RNA inverse folding and synthetic design

Author: Juan Antonio Garcia Martin

Boston College

Doctoral Thesis

RNA inverse folding and synthetic design

Juan Antonio Garcia Martin

A thesis 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

June 2016

Abstract

RNA inverse folding and synthetic design

by Juan Antonio Garcia Martin

Advisor: Dr. Peter Clote

Synthetic biology currently is a rapidly emerging discipline, where innovative and interdisciplinary work has led to promising results. Synthetic design of RNA requires novel methods to study and analyze known functional molecules, as well as to generate design candidates that have a high likelihood of being functional. This thesis is primarily focused on the development of novel algorithms for the design of synthetic RNAs. Previous strategies, such as `RNAinverse`, `NUPACK-DESIGN`, etc. use heuristic methods, such as adaptive walk, ensemble defect optimization (a form of simulated annealing), genetic algorithms, etc. to generate sequences that minimize specific measures (probability of the target structure, ensemble defect). In contrast, our approach is to generate a large number of sequences whose minimum free energy structure is identical to the target design structure, and subsequently filter with respect to different criteria in order to select the most promising candidates for biochemical validation. In addition, our software must be made accessible and user-friendly, thus allowing researchers from different backgrounds to use our software in their work. Therefore, the work presented in this thesis concerns three areas: Create a potent, versatile and user friendly RNA inverse folding algorithm suitable for the specific requirements of each project, implement tools to analyze the properties that differentiate known functional RNA structures, and use these methods for synthetic design of *de-novo* functional RNA molecules.

Contents

List of Figures

List of Tables

Acknowledgements

First and foremost, I would like express my gratitude to my advisor Prof. Peter Clote and Dr. Ivan Dotu for their guidance, advice and enthusiasm. I would like to thank them as well for their patience during the countless hours spent explaining the nuances of the computational RNA field. I have been fortunate to count on the guidance and experience of two exceptional mentors.

Besides, I would like to thank to other members of my thesis guidance committee, Prof. Welkin Johnson and Prof. Michelle Meyer, for insights and good advices during my graduate studies. And I am also grateful to Prof. David Mathews and Prof. Jacquin Niles for their interest in my work and for accepting to serve on my graduate committee.

I would like to thank as well current and former members of the CloteLab: Dr. Evan Senter and Amir Bayegan for stimulating discussions, to our collaborators at Prof. Encarnación Martinez-Salas' lab and the Meyer Lab for their meticulous work, and to anonymous reviewers for providing valuable suggestions. I am also grateful to my current colleagues at Boston College and the staff of the Biology Department for making of Higgins Hall a friendly and pleasant environment. I have a special gratitude towards my former colleagues at the Spanish National Centre for Biotechnology, who introduced me into academic research, in particular to Prof. Florencio Pazos.

I also want to acknowledge the National Science Foundation (NSF) for the funding provided for this research.

Finally, I want to thank my friends and family who encouraged me to pursue my endeavors and supported me during my whole life, and especially to my wife Bea for her love, encouragement and sacrifice along all these years.

Chapter 1

Introduction

Ribonucleic acid molecules are currently of great interest to the biological community, due to their primordial role in the presumed RNA world [1], anterior to DNA and proteins, and especially due to the many surprising, recently discovered regulatory roles played by RNA [2, 3, 4, 5].

RNA molecules are linear polymers of nucleotides: the purines adenine (A) and guanine (G); and the pyrimidines cytosine (C) and uracil (U). They are usually single stranded and tend to fold into an energetically favorable conformation, primarily determined by stacking interactions and hydrogen bonds between nucleotides. The most stable hydrogen bond interactions between non adjacent nucleotides occur between Watson-Crick base pairs (G-C,A-U) and G-U wobble base pairs. Although slightly weaker than Watson-Crick base pairs, G-U wobble base pairs are an especially interesting feature of RNA due to their unique chemical, structural, and ligand binding properties. In addition to increasing the structural possibilities, in contrast to DNA where uracil is replaced with thymine and G-U base pairs do not occur, they favor divalent metal-ion binding sites and stabilize backbone turns [6].

As in the case of proteins, the function of RNA is often determined by its structure; consider, for instance, the regulation of genes and alternative splicing by allostery (riboswitches) [4, 7] and the catalysis of enzymatic reactions (ribozymes) [8]. Since RNA structure is primarily determined by stacking base pair interactions, in contrast to protein folding which is predominantly driven by hydrophobic interactions, secondary structure is a good predictor for the function and is used for computational prediction instead of the more complex 3D 'tertiary' structure.

Given an arbitrary RNA sequence $\mathbf{a} = a_1,\ldots,a_n$, where $a_i \in \mathcal{N} = \{\mathrm{A},\mathrm{U},\mathrm{G},\mathrm{C}\}$, we can define a secondary structure s of $\mathbf{a}$ as a set of base pairs (i,j) satisfying the following conditions: (1) If $(i,j) \in s$ then a_i,a_j constitute a Watson-Crick or GU wobble pair, in other words, $ij \in \mathcal{B}$ which is the set {AU,UA,GC,CG,GU,UG}; (2) If (i,j),$(i,k) \in s$ then $j = k$, and if (i,j),$(k,j) \in s$ then $i = k$; (3) If $(i,j) \in s$ then $i+\theta < j$, where $\theta = 3$ (a minimum assumed for steric hindrance). In addition, unless stated otherwise, throughout this thesis we consider only secondary structures without pseudoknots – if $(i,j) \in s$ and $(k,\ell) \in s$, then either $i < k < \ell < j$ or $k < i < j < \ell$ or $i < j < k < \ell$ or $k < \ell < i < j$.

A secondary structure s without pseudoknots has some properties that simplify its computational study: a base pair (i,j) in s is defined as exterior to the base pair (k,ℓ) if $i < k < \ell < j$, and (k,ℓ) is respectively defined as interior to (i,j); and the secondary structure s can be decomposed into the following elements, depicted in Figure 1.1:

- Hairpin loop: A base pair (i,j) and all positions in $[i+1,\ldots,j-1]$, where positions $[i+1,\ldots,j-1]$ are unpaired.
- Base pair stack: Two consecutive base pairs (i,j),(k,ℓ) where $k = i+1$ and $\ell = j-1$.

- Bulge loop: Two base pairs (i,j),(k,ℓ), $i < k < \ell < j$, and all unpaired positions in $[i+1,\ldots,k-1,\ell+1,\ldots,j-1]$, where either: (1) $k = i+1$ and positions $[\ell+1,\ldots,j-1]$ are unpaired; or (2) $j = \ell+1$ and positions $[i+1,\ldots,k-1]$ are unpaired.

- Internal loop: Two base pairs (i,j),(k,ℓ), $i < k < \ell < j$, and all unpaired positions in $[i+1,\ldots,k-1,\ell+1,\ldots,j-1]$, where $[i+1,\ldots,k-1]$ and $[\ell+1,\ldots,j-1]$ are unpaired.

- Multiloop: A closing base pair (i,j) , k base pairs $(i_1,j_1),\ldots,(i_k,j_k)$, and all unpaired positions in the ranges $i\ldots i_1,j_n\ldots i_{n+1},j_k\ldots j$, where $k > 1$, $i < i_1 < j_1 < \ldots < i_k < j_k < j$, and there are no paired positions in the ranges $i\ldots i_1,j_n\ldots i_{n+1},j_k\ldots j$ for any $n \in [1\ldots k-1]$.

- External loop: The set of all unpaired positions u and base pairs (k,ℓ) in s for which there is no base pair $(i,j) : i < j$ where $i < u < j$ or $i < k < l < j$. Note an external loop always includes the first and last nucleotides of the RNA molecule.

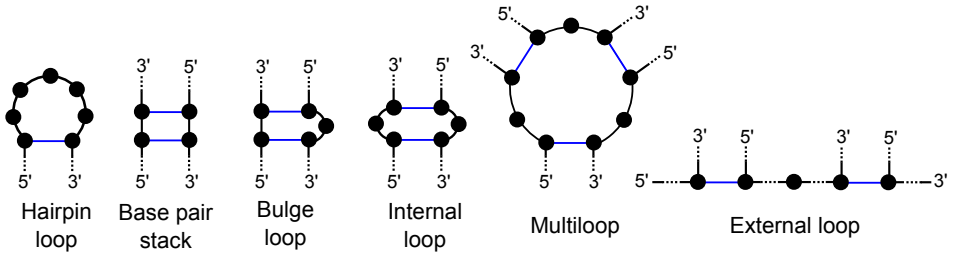


FIGURE 1.1: Elements of RNA secondary structure: hairpin loop, base pair stack, bulge loop, interior (internal) loop, multiloop and external loop. Lines illustrate base pair interactions (blue) and the ribose-phosphate backbone (black), where 5' and 3' ends are indicated. Dashed lines represent continuations of the backbone.

This decomposition allows the implementation of additive loop energy models based on estimations of the free energy contribution of each one of these elements. In some energy models, flanking positions of a closing stem (known as *dangling positions* or *dangles*) are also considered, since they also contribute to the total free energy of the structure.

The Turner nearest neighbor energy model is an additive loop model where a loop closed by external base pair (i,j) is designated as a k-loop, if it contains k base pairs interior to (i,j). If $k = 0$, then base pair (i,j) closes a hairpin loop. If base pair (i,j) stacks on the base pair $(i+1,j-1)$, or if (i,j) closes a bulge or internal loop, having one base pair internal to (i,j), then this is a 1-loop. If base pair (i,j) closes a multiloop with k components, or equivalently, containing k base pairs internal to (i,j), then this is a k-loop. Multiloops, or k-loops for $k \geq 2$, are also called $(k+1)$-way junctions, where the additional count is due to the outer component adjacent to (i,j)[9].

Due to the extensive study of RNA (secondary) structure, there is now software available for secondary structure prediction [10, 11, 12], motif discovery [13, 14], structure alignment [15, 16], riboswitch detection [17], precursor microRNA gene finders [18], non-coding RNA gene finders [19], etc. Due to the regulatory importance of RNA and the availability of such software, it is clear that some of the next important steps in synthetic biology will concern the computational design and experimental validation of RNA structures [20], as in the pioneering work of the lab of Niles Pierce [21].

Much of the work in synthetic biology concerns what might be called "synthetic genomics", pertaining to synthetic regulation of genes [22] and the development of genomic building blocks, from which "parts" of a novel genome can be constructed [23]. In contrast to such work, in this thesis, we instead consider RNA molecular design using computational methods from dynamic programming [24] and constraint programming [25, 26], with subsequent experimental validation.

In this thesis we present novel computational tools for RNA synthetic design, along with experimentally validated applications which show how computationally aided design can contribute

to the advance of the field of synthetic biology, a research area poised to make revolutionary contributions to the 21st century.

1.1 Thesis Content

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

- Garcia-Martin JA., Clote P. & Dotu I. (2013) RNAiFold: a constraint programming algorithm for RNA inverse folding and molecular design. J. Bioinform. Comput. Biol. 11(2), 1350001. http://doi.org/10.1142/S0219720013500017
- Garcia-Martin JA., Clote P. & Dotu I. (2013) RNAiFold: a web server for RNA inverse folding and molecular design. Nucleic Acids Res. 41(Web), W465–W470. http://doi.org/10.1093/nar/gkt280
- Dotu I., Garcia-Martin JA., Slinger BL., Mechery V., Meyer MM. & Clote P., (2014) Complete RNA inverse folding: computational design of functional hammerhead ribozymes. Nucleic Acids Res. 42(18), 11752–11762.http://doi.org/10.1093/nar/gku740
- Garcia-Martin JA., Dotu I. & Clote P. (2015) RNAiFold 2.0: a web server and software to design custom and Rfam-based RNA molecules. Nucleic Acids Res. 43 (W1): W513-W521. http://doi.org/10.1093/nar/gkv460
- Garcia-Martin JA. & Clote P. (2015) RNA thermodynamic structural entropy. PLoS One 10 (11), e0137859 http://doi.org/10.1371/journal.pone.0137859
- Garcia-Martin JA., Dotu I., Fernandez-Chamorro J., Lozano G., Ramajo J., Martinez-Salas E. & Clote P. (2016) RNAiFold2T: Constraint Programming design of thermo-IRES switches. Bioinformatics. (in press)
- Fernandez-Chamorro J., Lozano G., Garcia-Martin JA., Ramajo J., Dotu I., Clote P., & Martinez-Salas E. (2016) Designing synthetic RNAs to determine the relevance of structural motifs in IRES elements. Nature Scientific Reports 6, 24243 http://doi.org/10.1038/srep24243

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

1.2 Thesis Organization

The remainder of this thesis is organized in the following fashion. We begin in Chapter 2 with the presentation of `RNAiFold`, the first complete inverse folding algorithm which also includes a wide range of design constraints, along with several examples to illustrate how it can be used in different research fields. When we refer to completeness concerning inverse folding, we mean that `RNAiFold` can output all sequences that fold into the target structure (given sufficient time), or determine that there is no solution. We continue in Chapter 3 describing the synthetic design of functional cis-cleaving hammerhead ribozymes from `Rfam` alignments using `RNAiFold` in a computational pipeline. In Chapter 4 we extend `RNAiFold` for the design of meta-stable functional RNAs that adopt different stable conformations depending on the temperature (known as RNA thermometers, RNA thermosensors or thermoswitches) or the presence of a ligand (known as RNA switches or riboswitches). In addition, this chapter describes the design process and experimental validation of functional RNA thermometers and RNA switches: A thermosensor internal ribosome entry site (thermo-IRES) element, with increased cap-independent translation at higher temperatures; and theophylline *molecular scissors*, which combine into a single RNA chain both a theophylline riboswitch and a type III hammerhead ribozyme, resulting in a molecule capable of trans-cleavage of a second RNA chain only when activated by the presence of theophylline. The last chapters of this thesis are focused on algorithms to analyze the properties that differentiate known functional RNA structures. In Chapter 5 we present `RNAentropy`, two dynamic programming algorithms to

compute structural entropy for any user-specified temperature. Finally, in Chapter 6 we introduce the program `RNAdualPF`, which computes *the dual partition function* Z^* , defined as the sum of Boltzmann factors $exp(-E(a,s_o)/RT)$ of all sequences a with respect to the target structure s_o. Using `RNAdualPF`, we efficiently sample RNA sequences that (approximately) fold into s_o , where additionally the user can specify IUPAC sequence constraints at certain positions, GC-content, and whether to include dangles (energy terms for stacked, single-stranded nucleotides).

Chapter 2

RNAiFold

2.1 Introduction

In this chapter we present a *Constraint Programming* (*CP*) approach to solve the RNA inverse folding problem. Given a target RNA secondary structure, we determine an RNA sequence which folds into the target structure; i.e. whose minimum free energy structure is the target structure. Our approach represents a step forward in RNA design – we produce the first complete RNA inverse folding approach which allows for the specification of a wide range of design constraints. We also introduce a *Large Neighborhood Search* approach which allows us to tackle larger instances at the cost of losing completeness, yet while retaining the advantages of meeting design constraints (motif, GC-content, etc.). Results demonstrate that our software, `RNAiFold`, performs as well or better than all state-of-the-art approaches; nevertheless, our approach is unique in terms of completeness, flexibility and the support of various design constraints. Moreover, `RNAiFold` has applications beyond the pure RNA synthetic design, integrated as part of computational pipelines we show how it can be used for

the computational analysis of known RNAs and discovery of functional non coding RNAs. The algorithms presented in this chapter are publicly available via the interactive webserver `http://bioinformatics.bc.edu/clotelab/RNAiFold`; additionally, the source code can be downloaded from that site.

2.1.1 Organization

This chapter is organized in the following fashion. First, we provide background on the inverse RNA folding problem, as well as a brief overview of existing approaches, followed by a detailed description of the algorithm, where we describe in detail the implementation and features included in `RNAiFold`. Then we benchmark our algorithms against all currently available methods using benchmarking data used in other studies. We provide a table that compares various design features of each software, the number of solutions returned, etc. We continue with a brief description of the user interface via command line. Finally, we show how `RNAiFold` can be used in different research areas, such as the computational analysis of known RNAs, discovery of functional non coding RNAs, determining the relevance of structural motifs, and re-engineering of messenger RNAs to code the same or similar proteins and to contain desired RNA structural motifs.

2.2 Background

Given an RNA sequence, the *structure prediction* problem is to determine the *native structure* into which the sequence folds. Since the pioneering work of Anfinsen [27], it is widely accepted

that the native structure of a given macromolecule can be identified with its minimum free energy (MFE) structure. The 'RNA inverse folding' problem is the inverse; i.e. given a target structure, determine an RNA sequence whose MFE structure is the target structure. There are several widely-used thermodynamics-based software suites, which compute the MFE structure in time that is cubic in the RNA sequence length – for instance, `Vienna RNA Package` `RNAfold` [11, 28], `mfold` [29], `UNAFOLD` [10], and `RNAstructure` [12], all of which implement the Zuker algorithm [30], though with slightly different energy parameters [31, 32]. Since RNA MFE secondary structure can be efficiently computed, while determination of the MFE pseudoknotted (hence, *a fortiori*, tertiary) structure is an NP-complete problem [33], in this chapter we focus exclusively on the inverse folding problem for RNA secondary structures.

There is experimental evidence that RNA secondary structure forms independently of the tertiary structure [34]. From this data and newer NMR data [35], it is broadly believed that RNA folds in a hierarchical fashion [36], although there are exceptions [37, 38]. Since it appears that RNA secondary structure largely forms a scaffold for tertiary structure formation, any solution of the RNA secondary structure inverse folding problem is a major step towards functional RNA molecular design.

Several algorithms exist for the RNA inverse folding problem: `RNAinverse` [39], `RNA-SSD` [40], `INFO-RNA` [41], `MODENA` [42], `NUPACK-DESIGN` [21], `INV` [43], `FRNAkenstein` [44], `ERD` [45], `RNAfbinv` [46], `RNAdesign` [47], `EteRNABot` [48], `IncaRNAtion` [49]. With the exception of `IncaRNAtion` all of these algorithms can be classified as heuristic methods, which start with an initial sequence that is iteratively modified until it either folds into the target structure or some stopping criterion is reached.

The first approach found in the literature is `RNAinverse`, which forms part of the `Vienna RNA Package` [28, 39]. `RNAinverse` divides the given target structure S_0 into smaller subunits and attempts to find an RNA sequence by an *adaptive walk* algorithm. Sequence positions are randomly mutated; mutations are accepted if the objective function improves. In this case, the objective function is the Hamming distance between the MFE secondary structure of the current sequence and the target structure S_0. `RNAinverse` can return the correct solution, an approximate solution, or no solution at all.

`RNA-SSD` [40] is a different and very efficient algorithm, which nevertheless, shares the same overall approach of applying a divide-and-conquer strategy by hierarchically decomposing the target structure. In comparison with `RNAinverse`, `RNA-SSD` uses a more sophisticated initialization procedure to choose an initial RNA sequence, and applies *stochastic local search* in place of an adaptive walk. `RNA-SSD` is capable of finding a correct sequence for structures over one thousand nucleotides long.

The third approach is `INFO-RNA` [41]. Its main difference from previous approaches lies in the initialization step, which uses a dynamic programming algorithm to choose the sequence $s_1, \ldots, s_n$ that is compatible with the target structure S_0, having the lowest free energy. Although the free energy $E(s_1, \ldots, s_n; S_0)$ of target secondary structure S_0 on $s_1, \ldots, s_n$ is less than or equal to the free energy $E(s'_1, \ldots, s'_n; S_0)$ for all distinct sequences $s'_1, \ldots, s'_n$ that are compatible with S_0, this does *not* mean that the MFE structure of $s_1, \ldots, s_n$ is target structure S_0. `INFO-RNA` performs at least as well as `RNA-SSD`, and due to the initialization step, tends to yield RNA sequences, whose MFE structure has lower energy than sequences returned by other algorithms. Although this might seem to be a desirable feature, the solutions returned by `INFO-RNA` have

high GC-content and tend to have little resemblance with biologically active RNA, found in databases such as Rfam [50].

The fourth approach, MODENA [42], differs considerably from other inverse folding approaches, since it relies on a *multi-objective optimization* algorithm. MODENA uses the well-known NSGA2 [51] genetic algorithm to find solutions in the set of weak Pareto optimal solutions with respect to two optimization functions: structure stability (energy of the MFE structure of the proposed sequence) and structure similarity (distance between the MFE structure for the candidate sequence and the target structure). MODENA compares favorably to INFO-RNA and RNAinverse when benchmarked on a data set from Rfam [50].

NUPACK-DESIGN [21], is a remarkable, pioneering project of the Niles Pierce Lab, to design RNA molecules that have subsequently been synthesized and tested for folding properties, both *in vitro* and *in vivo*. NUPACK-DESIGN employs a similar approach to that of RNA-SSD, but, in this case, instead of finding sequences whose MFE structure is the given target structure, NUPACK-DESIGN attempts to find sequences having minimal *ensemble defect* [52]. *Ensemble defect* is the expected Hamming distance between the ensemble of secondary structures of an RNA sequence and a given user-specified target structure (See Appendix A for a formal definition of *ensemble defect*).

FRNAkenstein [44] is a recent Python program that calls Vienna RNA Package RNAfold and RNAeval within a genetic algorithm to evolve a collection of RNA sequences to have low energy structures with respect to one *or more* target structures (since solution sequences are compatible with than one target structure, structural compatibility constraints are supported). Source code can be downloaded from http://www.stats.ox.ac.uk/~anderson/Code/frnakenstein.html; however, there is no web server. FRNAkenstein [44] allows the

user to stipulate population size for its genetic algorithm, which thus determines the number of output sequences. A realistic upper bound on population size depends on run time, which is slow, since Python is an interpreted language.

`ERD` [45] is another genetic algorithm that performs a structural decomposition of the target structure into components. The main difference with other evolutionary approaches resides in the mutation and crossover steps, which operate at the component level.

Another algorithm developed in Java, `RNAfbinv` [46],includes a graphic user interface and also performs a decomposition of the structure into coarse-grained tree graphs. It uses a *simulated annealing* strategy to minimize free energy of the building blocks and base pair distance to the target structure.

`RNAdesign` [47] is a method specifically designed to solve the inverse folding problem for multiple target structures. It makes use of a combination of graph coloring and heuristic local optimization to find sequences whose energy landscapes are dominated by two or more given target structures.

The algorithm `INV` [43] uses a stochastic local search routine to determine a sequence whose minimum free energy *pseudoknotted* structure is a given target *3-noncrossing* RNA structure. Here, a 3-noncrossing structure is a (possibly pseudoknotted) structure, in which no three base pairs mutually cross each other. `INV` relies on the dynamic programming (exponential time) minimum free energy structure prediction algorithm for 3-noncrossing structures [53], and the fact that each 3-noncrossing RNA structure has a unique loop-decomposition. However, this software is no longer available.

`EteRNABot` [48], which also uses a local search optimization strategy, incorporates new design rules that are not currently included into any thermodynamic model. These new rules were extracted by machine learning from experimentally validated designs, which were created by the participants of the `EteRNA` game `http://www.eternagame.org/`. The novelty of the `EteRNA` project resides in the use of a collaborative science approach, where the joint effort of more than 37,000 non-expert participants is leading to the characterization of new features of RNA folding that are not accounted for in the current thermodynamic models.

Finally, `IncaRNAtion` [49] is a weighted sampling algorithm that, although not specifically intended to solve the inverse folding problem, is a very fast algorithm that returns sequences in the low free energy landscape of a given target structure. It performs a weighted sampling from the *partition function* of all sequences compatible with a given structure. However, it relies in a simplified energy model that only accounts for stacking energies.

In this chapter we present two algorithms to solve the inverse folding problem for RNA secondary structures. The first is a *Constraint Programming* (*CP*) implementation which performs surprisingly well, compared to the previously mentioned approaches. However, *CP* performs an exhaustive exploration of the search space which can lead, in some cases especially when the structures are large and complex, to a prohibitive inverse folding time. For this reason, we have also developed a *Large Neighborhood Search* (*LNS*) method which builds on the underlying *CP* framework, which achieves better results for larger structures. *LNS* can also be used when completeness is not required; i.e. when it is likely that a solution exists, and it is not necessary to prove that no solution exists.

2.3 Algorithm description

As previously mentioned, our algorithm is based on a *Constraint Programming* formulation of the RNA inverse folding problem. *Constraint Programming* (*CP*) has become one of the main methodologies for solving hard combinatorial optimization problems. Its salient features are its rich modeling language and its computational model based on *branch and prune*. At the modeling level, *CP* models a complex application in terms of decision variables, domains which specify the possible values for the variables, and constraints which capture its combinatorial substructures, giving the underlying solver significant information on the application structure. For instance, *CP* solvers feature global constraints such as *alldifferent*($x_1,\ldots,x_n$), which specifies that the variables $x_1,\ldots,x_n$ must be given different values. This contrasts with frameworks such as mixed-integer programs where all the constraints are linear.

Our algorithm was initially developed using the now defunct `COMET` framework[25], and subsequent versions were developed in C++ using the open source `OR-Tools` libraries[26] maintained by Google. Both `COMET` and `OR-Tools` libraries were selected by the efficiency of their *CP* engines along with several predefined global constraints that are key for the efficiency of our approach.

For direct folding of single or hybridized RNA sequences `RNAiFold` relies respectively on `RNAfold` and `RNAcofold` (from the `Vienna RNA Package` [28]) and `Fold` and `bifold` (from Mathews Lab `RNAstructure` [12]), adapted as plug-ins with `OR-Tools`.

The flexibility and modularity of our implementation made it possible to extend the capabilities of `RNAiFold` in subsequent versions, incorporating unique features such as: the use of different

thermodynamic energy models; amino acid constraints; structural *compatibility and incompatibility* constraints; *partial target* structures where specific positions may be either paired or not; and setting energy or *ensemble defect* limits for the solutions returned.

Before starting the formal definition the algorithm, it is important to understand some basic notions of *Constraint Programming*. The *domain* of a variable is the set of all the possible values that a can be assigned to that variable, which is not necessarily finite – i.e. in classic (imperative) programming paradigms the domain of a boolean variable is the finite set (TRUE,FALSE), while the domain of an integer variable is the infinite set of all integer numbers. *CP* variables are different from variables used in imperative programming. While imperative programming variables have unique values assigned, *CP* variables have associated *domains* which must be defined in the declaration. Moreover, the efficiency of a *Constraint Programming* algorithm depends on the domains selected in the modeling, and the choice of finite domains usually leads to more efficient algorithms. Therefore, it is key to define the appropriate domain for each *CP* variable according to the problem to be solved. The domain of a *CP* variable can change during the execution of the algorithm, and when the domain of a *CP* variable is reduced to a single value we say that this variable is *assigned*.

Besides variables and domains, the other two main components of *CP* are the *constraints* and the *search*. Constraints define relations between variables based in the properties of the problem to be solved, when the domain of a variable changes the *CP* engine is responsible for maintaining the consistency of such relations through *constraint propagation*. *CP* engines implement algorithms for efficiently propagate complex non-binary constraints such as the aforementioned *alldifferent* constraint.

In order to find solutions for a given problem, *CP* performs a search where the domain of some variables, called *search variables*, is sequentially explored and each possible value is assigned. The order in which the *search variables* are explored (variable heuristic) and the values are assigned (value heuristic) define the search tree, where each node is a possible assignment. Note that the search tree is not stored in memory at any time, since it is defined by the domain of the *search variables* and the value and variable heuristics. The search tree is traversed using a *depth-first search* (DFS) algorithm, where the *CP* engine tries to go deeper in the tree before exploring siblings. After each assignment, *constraint propagation* prunes the domain of the remaining variables to maintain consistency. If during the propagation the domain of a variable is reduced to size zero it means that the explored assignment is incompatible with the defined constraints. Therefore, the *CP* engine determines that there is no solution for the assignment given in the exploration, so the system prunes the search tree and performs a *backtracking*, restoring all the domains to the state previous to the last assignment and removing the value assigned from the domain of the variable. Figure 2.3 illustrates in a toy example of our *CP* implementation the process of constraint propagation from an initial assignment until the system determines that there is no solution for the given assignment. When the exploration reaches a leaf of the tree (a single value has been assigned for each variable) a solution is returned, and it continues until all values have been explored and therefore all possible solutions have been found.

Note the difference with local search algorithms, which explore only specific regions of the search space, with *CP* search, where the search finishes after the search tree has been completely explored. However, in *CP* specific stop conditions can be indicated such as a maximum

number of solutions or a running time limit. In addition, optimization constraints can be defined, indicating that the algorithm returns solutions that optimize the value of a given *CP* variable.

Therefore, when implementing a program using *CP*, we need to determine three different aspects: modeling (problem decomposition, variables, domains, and constraints), search (variable and value heuristics) and objective (stop condition). As mentioned in the introduction,we have developed two different algorithms, using *CP* and *LNS*. The modeling and objective parts are common to both and only the search part differs. We describe each of these in the following subsections. To simplify the algorithm description we will describe the implementation of the amino acid coding requirements in a separate section, since it involves specific variables, constraints and objective functions that will be easily understood once the general approach is assimilated.

2.3.1 Structural decomposition

The RNA inverse folding problem can be stated as follows: given a secondary structure S_0, presented as a dot-bracket expression of length n, find the RNA sequence (i.e. a word in the alphabet $\{A,G,C,U\}$) whose minimum free energy (MFE) structure is S_0. In our case, MFE structure is predicted by either `RNAfold`, a tool from the `Vienna RNA Package` [28], or `RNAstructure` depending on the option selected [1].

In order to minimize computational cost, we break down the target structure S_0 hierarchically, as previously done in most prior methods, `RNAinverse` [39], `RNA-SSD` [40]. First, we create

[1]The thermodynamics-based software suites `RNAfold` [11], `UNAFOLD` (replacing `mfold`) [10], and `RNAstructure` [12] all implement the same Zuker algorithm [30]; however, since free energy parameters may differ slightly between algorithms, the predicted MFE structure can depend slightly on the software used.

a tree-like decomposition T_1, where nodes correspond to substructures; from this, we next create a *reduced* tree-like decomposition T_2, obtained by repeatedly merging adjacent nodes of T_1 together. As explained below, adjacent nodes u,v of T_1 are merged when it happens that the substructure S_u corresponding to node u is energetically unstable (there is no sequence for which the free energy of S_u is negative), while the substructure S_{uv} is energetically stable (there is at least one sequence for which the free energy of S_{uv} is negative). Here if S_u [resp. S_v] represent the substructures corresponding to adjacent nodes u,v of T_1, then S_{uv} is the substructure corresponding to the concatenation of S_u with S_v. This operation is iterated, thus yielding a *reduced* tree T_2, with the property that the substructure corresponding to each node of T_2, which is designated as an *extended helix* (*EH*), has negative free energy. The structural decomposition into *EH* allows to associate specific structural constraints with each one of the nodes (*EH*) depicted in the right panel of Figure 2.1.

Formally, for the purpose of our decomposition, a *helix* is a set of *consecutive* base pairs, where consecutive is loosely defined so as to allow, within a helix, bulges of size at most 2, and internal loops of sizes at most (1×1), (2×1), (1×2), (2×2).[2]. The structure decomposition tree T_1 is defined as follows:

- The root of the tree is a node, corresponding to the (entire) target structure S_0.
- Recursively, create a node for each *helix* in the target structure. As shown in Figure 2.1 for the example target secondary structure S_0 given by

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

[2] An internal loop of size $(n \times m)$ is enclosed by base pairs (i,j) and $(i+n+1,j-m-1)$, where positions $i+1,\ldots,i+n$ and $j-m,j-m+1,\ldots,j-1$ are unpaired.

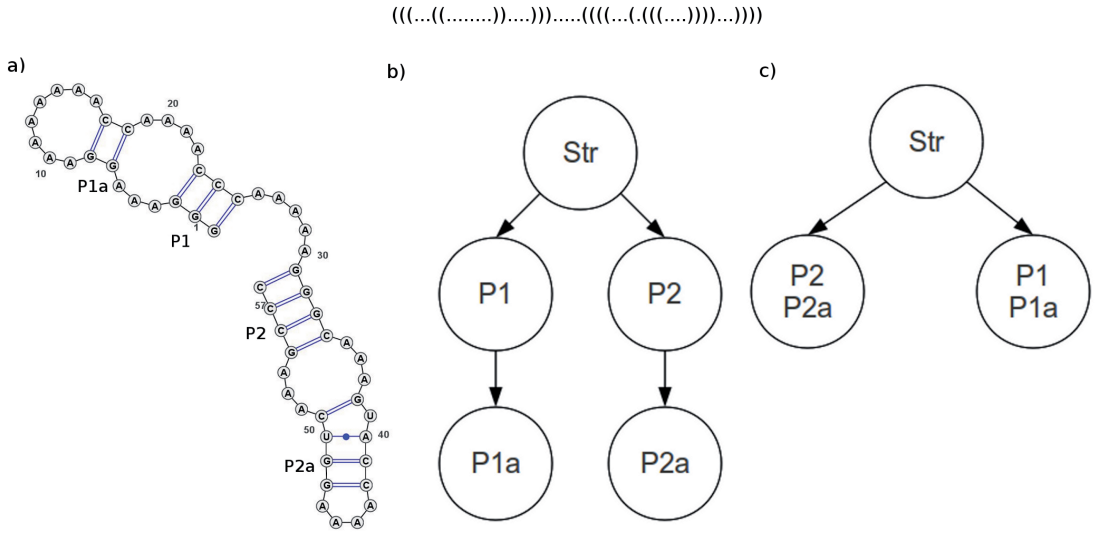


FIGURE 2.1: RNA structure tree-like decomposition and reduction. *(Left)* Target RNA secondary structure S_0. *(Middle)* Structure decomposition tree T_1. *(Right)* Reduced structure decomposition tree T_2.

the root of T_1 (corresponding to S_0) has two children, corresponding to *helices* $P1$,$P2$. For each node/substructure, recursively perform the same decomposition where a node is considered a parent node for the helices into which it can be decomposed. In our illustrative example, $P1$ [resp. $P2$] has child $P1a$ [resp. $P2a$]. If the currently considered node/helix $u \in T_1$ leads to a multiloop (also called multi-way junction), then u has children $v_1, \ldots, v_{k-1}$, corresponding to the $k-1$ remaining helices that are incident to the multiloop. If the currently considered node/helix $u \in T_1$ leads to an internal loop or bulge of size greater than 2, then u has a single child v, corresponding to the remainder of the stem after the internal loop or bulge.

- Leaves of the tree correspond to terminal helices, i. e. stem-loops, as depicted by $P1a$ and $P2a$ in Figure 2.1.

After computing the structure decomposition tree T_1, we subsequently perform a recursive merge operation, proceeding from leaves to the root. Initially T_2 is defined to be T_1. We recursively merge adjacent, non-bifurcating nodes of T_2 until no further merge operations are needed. This produces the final *reduced tree* T_2. Two adjacent nodes of T_2 are merged together, if either of the following holds.

- The stacking free energy of the stem (assuming that all base pairs in the stem are GC pairs) does not exceed, in absolute value, the free energy of the apical loop; i.e. the stem-loop structure is not energetically favorable, assuming base pairs are realized by GC pairs. This happens, for instance, in the stem-loop `((.......))`.
- The outermost or external base pair of the stem is separated from the rest of the stem by a bulge or internal loop of any size. One example is the stem-loop `(.(((....)))).` corresponding to *P2a* of the T_1 tree in Figure 2.1.

As mentioned, the merge operation is performed recursively from *leaves to root*. The reduction of tree T_1 to T_2 is very important, since certain nodes/substructures u of T_1 might be energetically unstable, meaning that no sequence would fold into the structure corresponding to u. Figure 2.1 depicts the reduction procedure, where, given the target structure S_0 (left panel)

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

we obtain the structure decomposition tree T_1 (middle panel), and after the merge procedure, the reduced *EH* tree decomposition T_2 (right panel). Finally, each node in the *EH* reduced tree T_2 corresponds to a structural constraint that is considered by our algorithm `RNAiFold`. Figure 2.2 depicts the structure decomposition tree T_1 for the Rhizobiaceae group bacterium NR64 RNA, with EMBL accession number Z83250.

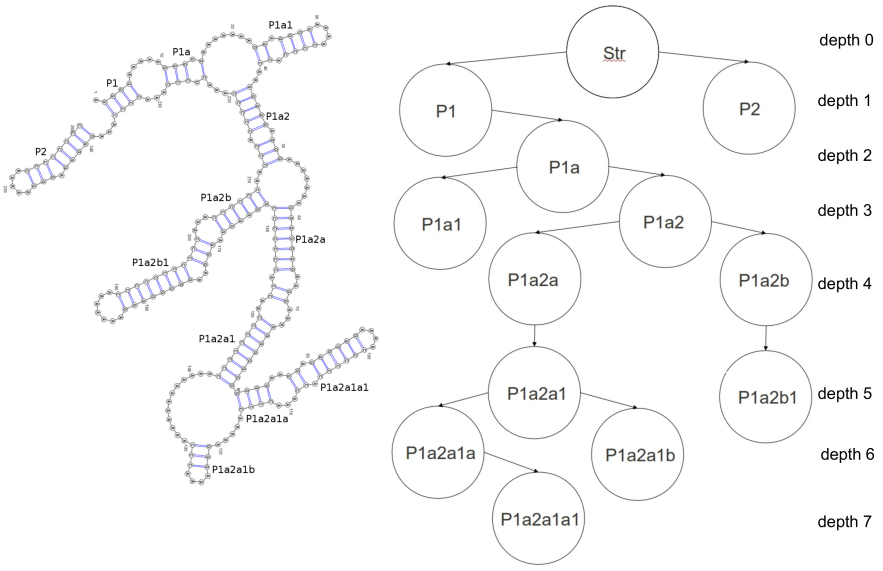


FIGURE 2.2: RNA structure and its tree-like decomposition by *depth*. *(Left)* RNA structure for Rhizobiaceae group bacterium NR64, with EMBL accession number Z83250. Image produced using `VARNA` [54]. *(Right)* Tree decomposition of helices for Z83250.

In some energy models, flanking positions of a closing stem (known as *dangling positions* or *dangles*) contribute to the free energy of the structure. Since *EHs* do not include dangling positions, it can happen that, although the MFE structure of given sequence a_{EH} for an *EH* is identical to the *EH* target structure S_{EH}, there is no assignment X,Y for the *EH* dangling positions i,j such as the MFE structure of the sequence $Xa_{EH}Y$ is identical to $iS_{EH}j$. To solve this problem, we extended the concept of *extended helices* to *extended helices with dangles* (*EHwD*), where dangling contribution is included. For this reason, *EH* decomposition trees of `RNAiFold1.0`[55] were replaced by *EHwD* decomposition trees in `RNAiFold 2.0` [56] and subsequent versions. A detailed description of *EHwD* tree decomposition is included in Appendix B.

2.3.2 Variables and Domains

The two basic components of the RNA inverse problem are the target structure S_0, given as a dot-bracket expression of length n, and the RNA sequence solution (i.e. a word of length in the alphabet $\{A,G,C,U\}$). The target secondary structure S_0 can be alternatively viewed as a set of canonical base pairs (GC,CG,AU,UA,GU,UG) and a set of unpaired positions.

The first modeling choice corresponds to the variables that define the problem and the values they can take (i.e. variable domains). In order to boost efficiency and create a framework that easily permits the addition of sequence constraints, we define several sets of *CP* variables.

- X: A set of variables corresponding to the nucleotides of the solution sequence $X = (x_1, x_2, ..., x_n)$. Where x_i denotes the nucleotide at position i of S_0.
- UP: A set of variables, $UP = (up_1, up_2, ..., up_k)$, corresponding only to those positions indicated as unpaired in the target structure S_0, where k is the number of unpaired positions in S_0 and up_i is the ith unpaired position in S_0.
- BP: A set of variables, $BP = (bp_1, bp_2, ..., bp_\ell)$, corresponding to every base pair in S_0, where ℓ is the number of base pairs in S_0 and bp_i is the base pair corresponding to the ith opening base pair position in S_0. Note that ℓ base pairs correspond to $2 \cdot \ell$ nucleotides in the sequence, the specific canonical base pairs found in RNA structures.
- BPT: A set of variables, $BPT = (bpt_1, bpt_2, ..., bpt_\ell)$, corresponding to every base pair in S_0, using the same order defined for BP, and indicating the type of the base pair (GC,AU,GU).

- *GC*: A set of boolean variables, $GC = (gc_1, gc_2, ..., gc_n)$, for each position in X representing whether it is assigned to a G or a C or not.

Note that X and the combination of UP and BP are two different modelings of the same problem, since each variable in UP and BP corresponds respectively to one or two positions of the sequence represented in X. As shown in Figure 2.3, where the *CP* variables are depicted in yellow, for the target structure `((...))` x_3 and up_1 are different representations of the same unpaired position and the combination x_1,x_7 and bp_1 represent the same base pair.

For this reason, it is important to distinguish between *search variables* and *auxiliary variables*. *Search* variables are the ones on which the search will focus, i.e., the ones that will be explicitly assigned a value. *Auxiliary* variables help simplify constraint declarations and/or heuristics, and they need to be unequivocally determined via *channeling constraints* [3]. In our approach, UP and BP are *search variables*, while X, BPT and GC are *auxiliary variables*.

The next modeling decision concerns the definition of the domains, a straightforward approach would be to choose letters among {*A*,*G*,*C*,*U*}, and pairs of letters among {*GC*,*CG*,*AU*,*UA*,*GU*,*UG*}, as domains for X and BP, respectively. However, this is not only more computationally costly, but also the implementation of correspondences between sequence variables X and base pairs and unpaired variables requires the use of several dictionaries[4] and/or *if-else* statements. For this reason we choose to use an integer representation for all the domain values.

Going a step further, we choose integers corresponding to the marks in an optimal *Golomb ruler* [57, 58] of size 5, for the domain of variables in X ({*A*,*G*,*C*,*U*}). A Golomb ruler is a ruler with marks placed at certain integer positions such that all the pairwise differences between marks

[3] In *Constraint Programming*, *channeling constraint* refers to a type of constraint that links two different modelings of the same problem and ensures that the solutions for both modelings are consistent with one another.

[4] A dictionary is a hash array of key:value pairs

are different. An optimal Golomb ruler, given a certain number of marks, is a Golomb ruler of minimum length. For 5 marks, the optimal Golomb ruler has marks in positions {0,1,3,7,12}. Excluding 0 which is always the first mark by definition. As depicted in Figure 2.3 in purple squares, the domain of the variable i in each one of the defined sets of variables is the following.

- $dom(X_i) = \{1, 3, 7, 12\}$ corresponding to $\{G,A,C,U\}$.
- $dom(UP_i) = \{1, 3, 7, 12\}$ corresponding to $\{G,A,C,U\}$.
- $dom(BP_i) = \{-11, -9, -6,6,9,11\}$ corresponding to $\{GU,AU,GC,CG,UA,UG\}$.
- $dom(BPT_i) = \{36,81,121\}$ corresponding to $\{GC,AU,GU\}$.
- $dom(GC_i) = \{0,1\}$ corresponding to $\{FALSE,TRUE\}$ for the statement $S_0[i] == (G \vee C)$.

Note that (as will be formally described below) the value of each possible base pair value in each variable of BP corresponds the difference of its sequence values in the corresponding variables of X, and each value representing base pair type in a variable of BPT is the squared difference of its sequence values in the corresponding variables of X. This allows for a direct implementation of certain constraints (see below) which, in turn, represents a great speed-up when checking their consistency and performing their propagation.

Additionally, we maintain the following dictionaries which provide the indexes to define the relations between the variables in UP and BP and the variables in X.

- $BPstart$. For each the base pair i in BP, $BPstart_i$ is the index of its corresponding opening position in X.
- $BPend$. For each the base pair i in BP, $BPend_i$ is the index of its corresponding closing position in X.

- $UPdict$. For each the base pair i in BP, $UPdict_i$ is the index of its corresponding unpaired position in X.

2.3.3 Constraints

There are four types of constraints in our approach: *channeling constraints*, *structural* constraints, compatibility and incompatibility constraints and *sequence* constraints. *Channeling constraints* are always used, while the three last types of constraints are optional. Note that structural constraints enforce that the sequence folds into the target structure so, although they are optional, they are activated by default since they are a basic requirement for inverse folding and should be only deactivated for very specific design purposes. On the other hand, sequence and compatibility constraints are used to specify biologically important motifs, GC-content, compatibility with user-designated pseudoknots, and other biologically relevant features desired for RNA molecular design.

2.3.3.1 Channelling Constraints

Channeling constraints allow us to unequivocally determine the value of all *auxiliary variables* from the *search variables*. They are the following.

- For each base pair i, $BP_i := x_{BPstart(i)} - x_{BPend(i)}$.
- For each base pair i, $BPT_i := (x_{BPstart(i)} - x_{BPend(i)})^2$.
- For each unpaired position i, $UP_i := x_{UPdict(i)}$.
- For each position i, $GC_i := (x_i == 1 \wedge x_i == 7)$.

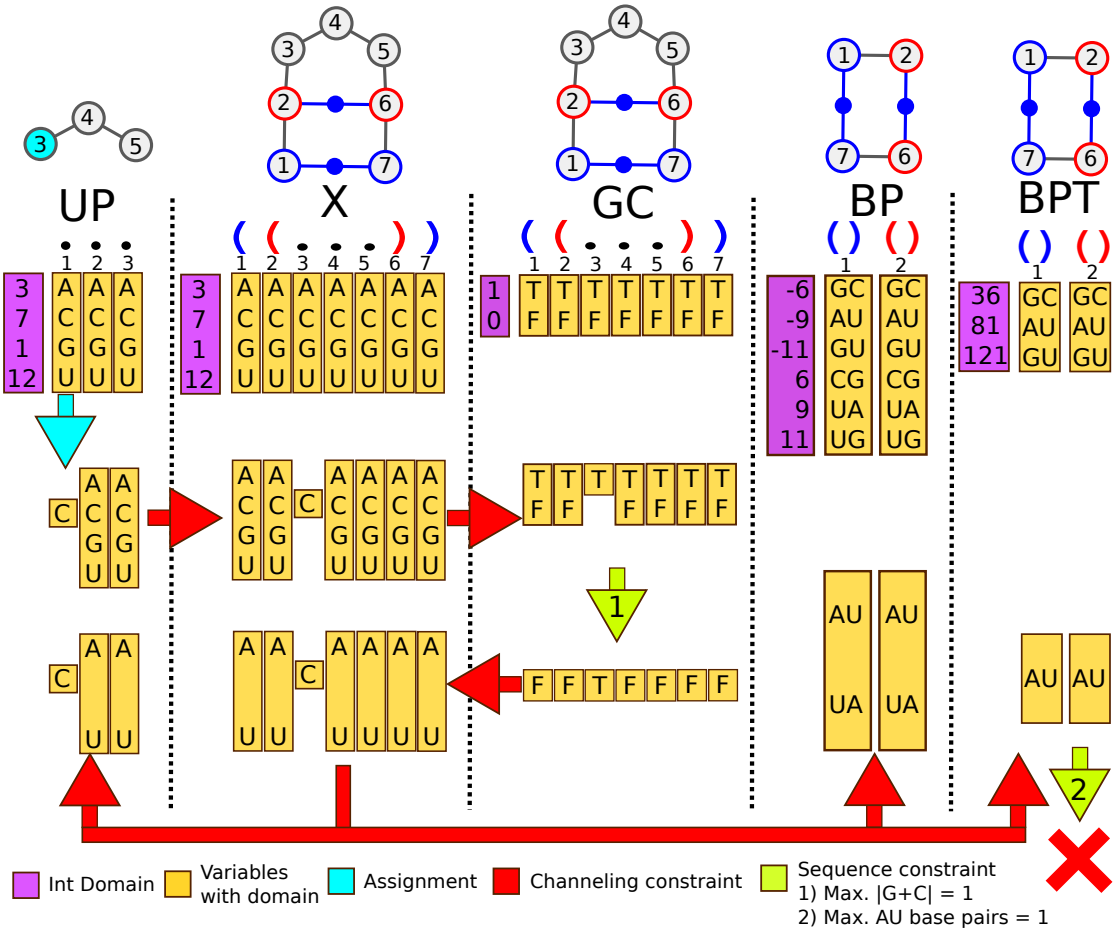


FIGURE 2.3: CP search propagation: Toy example of the *CP* modeling and constraint propagation for the target structure `((...))`. The figure illustrates the constraint propagation process from an initial assignment (in cyan) until the system determines that there is no solution for the given assignment (red cross). *CP* variables are organized in columns separated by dashed lines. The structures depicted on the top illustrate the positions of the target structure concerning each of the *CP* variables, where base pairs are colored in red and blue to distinguish the positions involved in each base pair. Purple squares indicate the integer representation of the initial domain in each one of the variables. Yellow squares represent the different sets of variables (indexes are shown in the initialized variable sets), the current domain is shown inside each variable (represented by the nucleotide values instead of the real integer representation for the sake of clarity). In this example, two sequence constraints are given 1) A maximum number of Gs and Cs of 1; 2) A maximum of 1 AU (or UA) base pair. The propagation starts after the assignment of a C at the first unpaired position UP_1 (cyan). Then *channeling constraints* propagate (represented by red arrows), reducing the domain of X_3 and subsequently the domain of GC_3. This triggers the propagation of the first sequence constraint (green arrow with number 1), since at most one G or C is allowed, the remaining variables of *GC* must be set to false. Again, the propagation of *channeling constraints* from *GC* to *X* reduces the domain of all variables in *X* except X_3, removing all Gs and Cs, and then constraints propagate from *X* to *UP*, *BP* and *BPT*. Finally, the second sequence constraint which indicates that there must be at most one AU (or UA) base pair is checked (green arrow with number 2). Since the domain of both base pairs is an AU base pair this constraint cannot be satisfied, therefore the *CP* engine determines that there is no solution that contains a C in UP_1 (indicated by a red cross). In this case the *CP* engine will perform a backtracking (the domain of all variables affected by the propagation will be restored to the initial state) and will continue with the exploration, removing *C* from the domain of UP_1 and assigning the next value available.

Figure 2.3 shows the propagation of the different *channeling constraints* (red arrows) in a toy example after a single position in UP has been assigned.

2.3.3.2 Sequence constraints

Sequence constraints are optional constraints that allow us to further specify desired features of a solution sequence. They are the following.

- Lower and upper bound on the number of base pairs of each type. Given a list of lower bounds *lbs* and a list of upper bounds *ubs* for the number of each type of base pair (*GC*,*AU*,*GU*), we can use the global constraint $cardinality(lbs,BPT,ubs)$ in `COMET` or $MakeBetweenCt(lbs,BPT,ubs)$ in `OR-Tools`.
- Maximum number of consecutive nucleotides of each type. Ensuring that the number of nucleotides of a particular type is bound by a specified maximum *maxcs*, can be realized by the following global constraint: $stretch(0,X,maxcs)$.
- Lower and upper bound on GC-content. This is handled in an analogous manner as in the base pair types.
- Limit number of bases. Minimum and maximum number of occurrences of each nucleotide in the MFE structure of any returned sequence. Minimum and maximum number of nucleotides can be constrained to specific regions by indicating the start and end positions.

Note the difference of *sequence constraints* with *channeling constraints*, which do not depend on any provided parameter. Figure 2.3, depicts the constraint propagation in a toy example for

the target structure `((...)))` where two sequence constraints have been defined: Number of Gs or Cs $\leq$ 1 and Number of AU base pairs $\leq$ 1. The assignment of a single variable in UP triggers the propagation of *channeling constraints* (red arrows) and sequence constraints (green arrows) until the *CP* engine determines that no solution exists that contains a C at position 3 of the sequence and satisfy the given sequence constraints.

2.3.3.3 Structural Constraints

Structural constraints are directly associated with the *EHs* defined in the structural decomposition. For each *EH* a structural constraint is defined, so it is triggered by the *CP* engine only when the corresponding subsequence is fully assigned (Figure 2.4), ensuring that a certain *EH* is the minimum free energy structure for the corresponding subsequence. The global structural constraint associated with the root node ensures that the whole sequence folds into the target structure S_0. However, note that this constraint will never be checked until all the other constraints are met, for a candidate sequence.

The user can specify which folding algorithm (`RNAfold` from `Vienna RNA` or `Fold` from `RNAstructure`) and energy model (Turner '99[31, 32], Turner '04[59] or Andronescu '07[60]) is used to compute the MFE structure and free energy of a sequence. In addition, `RNAiFold` structural constraints allow *partial target structures*, hybridization of two RNA chains and specifying free energy or *ensemble defect* limits.

Partial target structures Structural requirements can be relaxed by indicating *partial target structures* where specific positions may be paired or unpaired. Positions with indeterminate

pairing status are indicated in an expanded dot-bracket notation where all positions that contain a comma may be paired or unpaired; positions containing a dot must be unpaired; positions containing matching left and right parentheses must be paired together.

Cofolding Given a hybridization structure, together with optional constraints, `RNAiFold` returns two sequences, whose MFE hybridization is the input target structure. Note that a hybridization may involve inter-molecular base pairs, as well as intra-molecular base pairs, provided that there are no pseudoknots. `RNAiFold` uses `RNAcofold` from `Vienna RNA` or `bifold` from `RNAstructure` to compute the MFE structure of the hybridized complex.

The target hybridization structure is represented in dot bracket notation, following the convention of `RNAcofold` of the `Vienna RNA Package`. Namely, an ampersand ('&') separates the two single-molecule portions of the hybridization structure.

Free energy constraints `RNAiFold` also allows the user to stipulate the energy range $E(\mathbf{a},S_0)$ or *ensemble defect* $ED(\mathbf{a},S_0)$ for the sequences $\mathbf{a}$ returned, where S_0 denotes the target structure. Note that these constraints do not speed up the search since the energy or *ensemble defect* are computed once $\mathbf{a}$ is a solution.

On the other hand, due to the structure decomposition it is also possible to stipulate the energy range $E(\mathbf{a}_{i,j},S_{i,j})$ or *ensemble defect* $ED(\mathbf{a}_{i,j},S_{i,j})$ limits for the subsequence $S_{i,j}$ and the target substructure $S_{i,j}$ corresponding to the *EH* delimited by positions i and j. These constraints do reduce the search time because if a $S_{i,j}$ does not comply with the restrictions the subsequence, $\mathbf{a}_{i,j}$ is discarded as soon as the *EH* is assigned, pruning the search tree.

In `RNAiFold` the free energy $E(\mathbf{a},S_0)$ of a sequence a when is folded into a structure S_0 is calculated using the functions included in the `Vienna RNA Package` and `RNAstructure` libraries

for a given the energy model. The computed free energy value for the same structure and sequence can be slightly different depending on the library and energy model selected by the user.

2.3.3.4 Compatibility and incompatibility constraints

`RNAiFold` allows the user to specify those positions in the returned sequence(s) that *can* form a base pair, even if these base pairs are not part of the target structure. In this fashion, one could design an RNA whose MFE structure is the given target structure, but which is compatible with another structure.

For instance, the user could require all solutions to fold into the target structure .((((((.((((((...)))))).......((((........))))...)))))). and to be compatible with the additional structure ..............((((((.......))))))..... for which `RNAiFold` returns the following solution sequence

AGGCGGUAACCCGAUCCGGGUCUGAAGAGCUCGAGUUAAAGGGCGAAACCGCCC.

Solutions can be also required to be *incompatible* with base pair formation at those positions listed in a *prohibition list.* Base pair formation may be prohibited using 3 different formats. (i) If an *incompatible* secondary structure **s** is given, then positions (i,j) where a base pair occurs in **s** are not allowed to pair in every solution returned. (ii) A base pairing incompatibility stretch (i j k) may be indicated, which prevents position i from pairing with $j, j+1, j+2, \ldots, j+(k-1)$. (iii) A comma separated list of pairs $i_1 j_1, \ldots, i_n j_n$ can be specified, which prevents position i_1 from pairing with j_1, position i_2 from pairing with j_2, etc. The user may combine elements from (ii) and (iii) together.

To conclude this section, we include Table 2.1 which summarizes the capabilities of the different inverse folding methods available.

Software	⇓	WS	PK	H	MT	PT	T	EM	D	SeqC	StrC	AaC	O	Num
RNAiFold	✓	✓	–	✓	–	✓	✓	'99,'04	0,1,2,3	✓	✓	✓	various	MAX
RNAinverse	✓	✓	–	–	–	–	✓	'99,'04	0,1,2,3	IUPAC★	–	–	mfe, prob	100
RNA-SSD	–	✓	–	–	–	–	✓	'99	1	IUPAC★	–	–	mfe	10
Info-RNA	✓	✓	–	–	–	–	–	'04	1	IUPAC	–	–	mfe, prob	50
NUPACK	✓	✓	–	✓★	–	–	✓	'99,'04	0,1,2	✓	–	–	ens def	10
MODENA	✓	–	✓	–	–	–	–	I	def	–	–	–	mfe, prob	?
Frnakenstein	✓	–	–	–	✓	–	✓	I	def	–	–	–	various	?
IncaRNAtion	✓	–	–	–	–	–	✓	'04★	–	IUPAC	–	–	pf sampling	–
ERD	✓	✓	–	–	–	–	✓	I	def	IUPAC★	–	–	mfe	MAX★
RNAdesign	✓	–	–	–	✓	–	✓	'04	def	–	–	–	various	–
RNAfbinv	✓	–	–	–	–	✓	–	'99, I	def	local A,C,G,U	–	–	mfe	–

TABLE 2.1: Comparison table for RNA inverse folding software. Column headers: Software (method name), ⇓ (software can be downloaded), WS (web server), PK (pseudoknots), H (hybridization), MT (multiple targets), PT (partial targets), T (temperature),EM (energy model), D (dangles), SeqC (sequence constraints), StrC (structural constraints), AaC (amino acid constraints), O (objective), Num (maximum number of sequences returned). *Comments:* In column H, `RNAiFold` and `NUPACK-DESIGN` are the only programs that solve inverse folding for target *hybridizations*; moreover, `NUPACK-DESIGN` has '✓★', since it is the only algorithm that allows hybridization of more than 2 strands. In column EM, values are '99 (Turner'99), '04 (Turner'04), '04★ (Turner'04 base stacking parameters with no entropic free energies), I (installed, depending on the version of `Vienna RNA Package` installed on user's computer). In column D, dangle status is 0 (no dangle), 1 (max of 5′ and 3′-dangle), 2 (sum of 5′ and 3′-dangle), 3 (dangles and coaxial stacking), def (depending on default setting of user's version of `Vienna RNA Package`). In column SeqC, values are ✓(IUPAC plus additional constraints) IUPAC, IUPAC★ (limited subset of IUPAC symbols), and local A,C,G,U (oligonucleotide specified at a given position using only A,C,G,U). In column O, values are mfe (minimum free energy structure), prob (maximize Boltzmann probability), ens def (*ensemble defect*), pf sampling (partition function sampling with a restriction of Turner'04). In column Num, the number of solutions returned by the web server is given (—if no web server available); a question mark in this column appears for `MODENA` and `FRNAkenstein`, which are genetic algorithms, and have a population of evolving sequences, so the user cannot request a fixed number of solutions. `ERD` contains MAX★, since the web server allows the user to request an arbitrary number of *iterations* (distinct runs) of the program, where 10 minutes is the maximum computation time allowed per request. In contrast, `RNAiFold` contains MAX in this column, which indicates that as many solutions are returned as possible within the system-dependent run time bound.

2.3.4 CP Search

When implementing the search part of a *Constraint Programming* problem, we need to focus in the order in which variables will be assigned and on the order in which values will be assigned to the variables. Our *CP* algorithm is complete, meaning that it explores the search space exhaustively. This implies that, given sufficient time, our *CP* algorithm will either return a solution or prove that none exists. Moreover, we can as well return all the solutions, i. e., all the sequences that fold into the given target structure. Variable and value ordering heuristics give us the order in which we traverse the search space.

2.3.4.1 Variable ordering

Variable ordering determines the order in which the *search variables* are explored, recall that *search variables*, which are those included in the UP and BP sets, are the only *CP* variables that are instantiated in the search, and the propagation of *channeling constraints* determine the value of all *auxiliary variables*. Note that non structural constraints are not involved in variable ordering, so they are checked and propagated after any individual variable assignment.

Our variable ordering is specified on two levels, the first level depends in the *extended helices* (*EHwD*) to which the *search variables* belong. We start assigning a *depth* to each *EHwD*. Recursively define the *depth* of *EHwD* in decomposition tree T_2 as follows: the root has *depth* 0, while a non-root *EHwD* has *depth* one greater than its parent. Let $D(k)$ denote the number of *EHwDs* in T_2 at *depth* k. Define the node labels by applying breadth first search; i.e. the root has label 0; *EHwDs* at *depth* 1 have labels $1, \ldots, D(1)$; *EHwDs* at *depth* 2 have labels $D(1)+1, \ldots, D(1)+D(2)$, etc. An example of the *depth* assignment for *EHwD* decomposition is shown in the example

tree in the right panel of Figure 2.2. Then, *search variables* are grouped according to the *EHwD* to which they belong, and the order of exploration goes by *depth bottom-up*, where *search variables* in *EHwD* with higher *depth* are explored before. In the case of *EHwDs* with the same *depth*, variables from *EHwDs* in the left of the decomposition tree are explored first.

The second level of variable ordering heuristic deals with the exploration of nucleotide positions within a given *EHwD* structure, which depends on the pairing requirements of the *EHwD* structure. We define four types of elements in *EHwD* order assignment respectively:

1. Dangling position: Unpaired position at any side of a helix. Specific of *EHwDs*.
2. Unpaired position: Any other unpaired position.
3. Closing base pair: Outermost base pair of a helix.
4. Normal base pair: Any other base pair.

Then, this second level of variable ordering can be stated as follows:

1. First, variables in *BP* corresponding to non-outermost base-paired positions (x,y) of a given *EHwD* are instantiated from the innermost base pair to the outermost base pair.
2. Second, variables in UP corresponding to unpaired positions in a given *EHwD* are grouped together in consecutive runs, and these runs are ordered from largest to smallest and then instantiated from left to right.
3. Third, variables in *BP* corresponding to the outermost, closing base pair of a given *EHwD* is instantiated.

4. Finally, variables in UP corresponding to dangling positions of a given *EHwD* (if any) are instantiated (note that not all *EHwDs* contain dangling positions and *EHs* do not contain dangling positions).

To illustrate this ordering we have extracted the intermediate variable assignments for a toy example, which is depicted in Figure 2.4. Note the difference in variable ordering when using *EH* (left) and *EHwD* (left), since *EH* does not include dangling positions.

FIGURE 2.4: Trace of a toy example to illustrate variable ordering: In red full helix assignments corresponding to constraint check. *(Left)* Assignment using *extended helices*. *(Right)* Assignment using *extended helices with dangles*.

Additionally, we have added a slightly modified variable ordering heuristic, which we call *leaves to root*, where *EHwDs* are ordered by *height*. Define the *height* $ht(x)$ of each *EHwD* x of decomposition tree T_2 by induction: if x is a leaf, then $ht(x) = 0$; if x is the parent of *EHwDs* $x_1, \ldots, x_m$ and the *height* of $x_1, \ldots, x_m$ has been defined, then $ht(x) = 1+\max\{ht(x_1), \ldots, ht(x_m)\}$. Applying the *leaves to root* heuristic, (and opposed to *depth bottom-up* heuristic), in Figure 2.2, *EHwD* P1a2b1 with *depth* 5 will be assigned prior to *EHwD* P1a2a1a with *depth* 6.

2.3.4.2 Value ordering

`RNAiFold` determines the order in which values of base-paired positions in the target structure S are assigned. Note that the base paired positions are instantiated in the *search variable BP*. The base pair ordering is described as follows. If base pairs $(i,j) \in S$ and $(i+1,j-1) \in S$ and positions $i+1,j-1$ are currently instantiated, then base stacking free energies of are determined for each of the base pair choices G-C, C-G, A-U, U-A, G-U, U-G for positions i,j. A random number between 0 and 2 kcal/mol is added to each of the base stacking free energies, thus ensuring different value ordering depending on the random seed; subsequently, the base pair (i,j) is instantiated in order of increasing free energy of the resulting list.

UP values are assigned in the following order: $\{A,U,G,C\}$.

Note that randomizing the heuristic does not compromise completeness, it only entails that different runs of the algorithm will (potentially) yield different solutions, since the order in which the search space is visited would be different.

2.3.5 LNS

Large Neighborhood Search is a meta heuristic that attempts to find a high quality solution by iteratively changing a candidate (or tentative) solution. As opposed to other methods where differences between tentative solutions between two successive iterations is minimal, *LNS* fixes a small part of the tentative solution and explores (exhaustively if possible) the remaining, unfixed positions. This explains the origin of the name, '*Large Neighborhood Search*'.

`COMET` supports a straightforward implementation of *LNS*, where we reuse the program design and constraints from the *CP* implementation, while we add a 'restart' component. This restart component will fix some of the variables to their current values and will unassign the remaining variables. Thus, we only need to specify when to restart and what to do when we restart. `OR-Tools` includes predefined classes implementing *LNS* which are connected to the *CP* engine as *SearchMonitors*. Specific parameters such as the restart condition or the variable unassignment heuristic are defined during the *LNSSearchMonitor* initialization.

First of all, we choose to restart after an amount of time, which is proportional to the length of the target structure. Second, when a restart is triggered, a set of positions is selected as candidates to be fixed. The MFE structure for each EHwD of the current candidate solution is evaluated independently. If the MFE structure of an EHwD matches with the target structure, then all the EHwD positions are included in the set of candidates. When all the EHwDs have been evaluated, candidate positions are fixed with a probability of 0.9, and the set of candidate positions is stored.

Since the order of exploration is similar in each round, it could be possible that fixing similar parts of the sequence results in an exploration of almost the same region of the search space in subsequent searches, so two mechanisms are implemented to avoid this behavior: (1) In subsequent restarts, if the candidate positions to be fixed are the same as in the previous restart, then the probability of fixing positions decreases by 0.05, if not, then the initial probability of 0.9 is restored. (2) There is a hard restart (no nucleotide position is fixed) in the case that, after 10 restarts, the set of candidate positions remains unchanged, or if all possible solutions for the current subproblem have been explored.

In local search algorithms, there is always a trade-off between *exploitation* and *exploration*. Exploitation means focusing the search on promising regions, as reflected in our choice of probability 0.9 to remain close to currently instantiated portions of the sequence. Exploration means covering different, remote regions of the search space, as reflected in our choice to decrement the probability by 0.05 and our choice to perform a hard restart after 10 restarts.

2.3.6 Objective

The default behavior of `RNAiFold` using *CP* is to perform a complete exploration of the sequence space unless the specified search time limit is reached. However, in some cases the desired objective could not be finding all the sequences whose MFE structure is the target structure, but finding the sequence with the lowest free energy when folded into the target structure or with the lowest *ensemble defect* respect to the given target structure (see Appendix A).

It is important to remark that the MFE structure of a sequence **a** that minimizes the free energy for a target secondary structure S_0 ($E(\mathbf{a},S_0)$) is not necessarily S_0, so finding the sequence with the lowest free energy may not be the best strategy. However, objective conditions can be enabled or disabled independently and they are not mutually exclusive. Therefore, it is possible to use `RNAiFold` to find the sequence **a** whose MFE structure is the target structure S_0 and whose free energy $E(\mathbf{s},S_0)$ is the least among all sequences that fold into S_0.

`RNAiFold` can also be used to find the sequence **a**, which folds into the target structure S_0 and whose *ensemble defect* is the least among all sequences that fold into S_0; or to find the sequence

that minimizes *ensemble defect* independently of its MFE structure. In fact, *ensemble defect* minimization is the objective function used in `NUPACK-DESIGN` [21] local search.

2.3.7 Amino acid coding requirements

Amino acid constraints require solutions not only to fold into a target structure, but also to code one or more given proteins (or to code for the most similar proteins, as determined by the BLOSUM62 similarity matrix) in the given coding frames. There is no bound on the number of (possibly overlapping) coding regions for distinct peptides to `RNAiFold`.

The implementation of amino acid coding requirements requires some additions to the model. (1) a set of integer *CP* variables $AA = \{aa_1, aa_2, ..., aa_n\}$, representing each one of the codons with user specified coding restrictions. (2) a set of dictionaries defining the correspondence between amino acids and codons, so the initial domain of each codon variable is restricted by the given amino acids constraints.

The assignment of variables in *AA* is carried out by a *channeling constraint*, which propagates from the sequence domain of the *CP* variables in X, resulting in a unique value for each possible codon. For a codon k whose first nucleotide is at position i of the sequence $a_1, \ldots, a_n$ assigned in the *CP* variables of X, the value of AA_k is determined by the following *channeling constraint* $AA_k := (x_i - x_{i+1}) + ((x_i == x_{i+1} \cdot x_i) + 12) + (x_{i+2} \cdot 25)$.

This *channeling constraint*, which represents the relation between the domain of AA_k and the domain of x_i,x_{i+1} and x_{i+2} (recall that the domains of the variables in X is {1,3,7,12}, corresponding to the nucleotides {*G*,*A*,*C*,*U*}), is designed to produce a unique integer value for each

possible codon. We can distinguish three components in this constraint: (1) $x_i - x_{i+1}$ is a number between -11 and 11, unique for each combination of x_i, x_{i+1} unless $x_i == x_{i+1}$, when the value is 0; (2) $((x_i == x_{i+1} \cdot x_i) + 12)$ accounts for the case when $x_i == x_{i+1}$, adding $x_i + 12$ produces four values that do not overlap with the previous combinations of x_i and x_{i+1}: $AA = 15$, $CC = 19$, $GG = 13$, $UU = 24$; (3) Finally, $(x_{i+2} \cdot 25)$ adds the contribution of the assignment at x_{i+2} without introducing repetitions, since the maximum value of any combination of x_i and x_{i+1} is 24.

Therefore, the domain for an unrestricted codon variable is:

$dom(AA) = \{26,28,31,32,33,35,38,39,40,41,42,43,44,46,48,49,76,78,81,82,83,85,88,89,90,91,92,$

$93,94,96,98,99,176,178,181,182,183,185,188,189,190,191,192,193,194,196,198,199,301,303,306,307,$

$308,310,313,314,315,316,317,318,319,321,323,324\}$

corresponding to the 64 possible codons in the genetic code.

It is also possible to indicate constraints using symbols that represent a type of amino acids. To this end, we created a dictionary including the most common amino acid classifications such as polar, hydrophobic, negatively charged or aromatic.

A second way of specifying coding requirements is by auto-generating amino acid constraints. `RNAiFold` includes the option of selecting a BLOSUM62 similarity threshold t respect the input amino acid constraints, which indicates that each amino acid coded by a solution must have a BLOSUM62 similarity of at least t with the corresponding amino acid specified at each position of the input constraint.

Another novel feature of the amino acid constraints is the BLOSUM62 score maximization search, where `RNAiFold` determines a solution for which the sum of BLOSUM62 similarity

scores to the target peptide is an absolute maximum; i.e. no other solution of inverse folding codes a peptide having larger BLOSUM62 similarity to the specified target peptide. This is an objective function so, as explained before, it than can be used in combination with any other objective such as *ensemble defect* minimization.

2.4 Benchmarking

In this section we present a comparison of our approach against the approaches mentioned in the Introduction, excluding `INV`, which is not publicly available and which concerns 3-noncrossing structures. It should be mentioned that different sets of structures are used in benchmarking studies for different papers [40], [41], [42], [21]. Since we believe that the benchmarking set introduced by Taneda et al. [42] is the most unbiased and biologically relevant set of target structures, we believe the benchmarking results for this data set to be the most representative for the behavior of `RNAiFold` (see Tables 2.2 and 2.6). Nevertheless, in the remaining Tables 2.3 and 2.4, we benchmark `RNAiFold` against all the other data sets considered in the literature.

The benchmarking set of target secondary structures of Taneda et al. is built in the following manner.

- Download the seed alignment for various families from Rfam [50].
- Select the largest sequence in each seed alignment.
- Extract the annotated structure for the given sequence.
- Remove pseudoknotted pairs.

Since the Rfam database is modified and updated over time, to permit accurate benchmarking, we used the same set of Rfam structures used in the benchmarking from [42].

In order to compare with other approaches (mostly heuristic) we ran our algorithms for each instance a certain number of times (usually 50), and reported the number of times where the algorithm was able to return a solution, and the average time in which it did. For our *LNS* algorithm, which is heuristic, this is clearly understood. For our *CP* algorithm, even though it is complete, since we have added a random component to the variable (and value) ordering heuristic, different runs will explore the search space in a different order, and, thus, yield different results.

All benchmarking was carried out on an Intel Core i72630QM using 4 cores (2GH, 16GB memory, Linux Ubuntu 10.4), with a cutoff time of 10 minutes for all runs and for all algorithms. MODENA results are reported as in [42], where there is only 1 run with a population size of equal to the number of runs of the rest of the algorithms. Reported time is total time (in seconds) for MODENA to return the final population. All other times are reported also in seconds and are the average over all runs that returned a solution, where a *dash* ('–') corresponds to no solution found and thus no average time available. For all tables, best results are shown in bold face. Note that the algorithm that solves more runs might not be the fastest, since the average time is computed only over solved runs.

`INFO-RNA` 2.0 (newest version) was run, while allowing 0 mismatches in the final sequence (-n 0). `MODENA` was run with the maximum number of iterations allowed (9999) and a population equal to the number of runs. `RNA-SSD` code was modified to avoid premature termination due to the maximum number of tries and keep trying until a solution is found. `RNAinverse` was run with -R 1 (search until one solution is found).

We will discuss the results separately for *CP* and *LNS*.

2.4.1 CP results

Parameters		CP		INFO-RNA		MODENA		RNA-SSD		RNAinverse	
RF id	***n***	sol	time	sol	time	sol	time	sol	time	sol	time
RF00001.121	117	38	21.5	**50**	**0.0**	6	36.8	22	1.0	41	233.1
RF00002.2	151	**44**	29.5	4	62.6	20	39.4	6	**12.2**	0	-
RF00003.94	161	0	-	1	72.1	**29**	**70.2**	0	-	0	-
RF00004.126	193	**50**	1.5	**50**	**0.1**	34	52.9	**50**	2.0	**50**	48.3
RF00005.1	74	**50**	0.2	**50**	**0.0**	33	12.4	**50**	0.1	**50**	0.1
RF00006.1	89	**50**	0.3	**50**	**0.0**	37	15.1	**50**	0.6	**50**	4.3
RF00007.20	154	**50**	5.6	**50**	**0.0**	34	44.4	**50**	1.1	**50**	12.4
RF00008.11	54	**50**	0.1	**50**	**0.0**	26	8.7	**50**	**0.0**	**50**	**0.0**
RF00009.115	348	**48**	**20.8**	0	-	29	214.1	26	48.2	0	-
RF00010.253	357	**0**	-	**0**	-	**0**	-	**0**	-	**0**	-
RF00011.18	382	**0**	-	**0**	-	**0**	-	**0**	-	**0**	-
RF00012.15	215	**50**	**2.7**	15	25.0	27	64.5	28	28.8	1	139.4
RF00013.139	185	**50**	1.6	**50**	**0.8**	12	51.5	49	2.8	**50**	19.8
RF00014.2	87	**50**	0.3	**50**	**0.0**	33	17.5	49	0.1	**50**	**0.0**
RF00015.101	140	49	1.3	**50**	**0.2**	38	29.1	40	0.6	**50**	52.4
RF00016.15	129	**0**	-	**0**	-	**0**	-	**0**	-	**0**	-
RF00017.90	301	**50**	19.3	**50**	**0.0**	28	208.1	50	7.0	**50**	10.0
RF00018.2	360	**47**	**12.1**	1	697.0	28	331.5	0	-	0	-
RF00019.115	83	**50**	0.2	**50**	**0.0**	32	14.9	**50**	0.2	**50**	0.3
RF00020.107	119	**0**	-	**0**	-	**0**	-	**0**	-	**0**	-
RF00021.10	118	**50**	0.3	**50**	**0.0**	37	27.8	49	0.2	**50**	0.2
RF00022.1	148	**50**	0.7	**50**	**0.0**	38	32.6	24	0.9	35	225.5
RF00024.16	451	**0**	-	**0**	-	**0**	-	**0**	-	**0**	-
RF00025.12	210	**50**	**1.4**	9	47.9	33	54.2	29	2.9	0	-
RF00026.1	102	**50**	**0.4**	33	5.5	38	15.2	50	1.4	44	173.2
RF00027.7	79	**50**	0.1	**50**	**0.0**	32	17.4	**50**	0.1	**50**	0.4
RF00028.1	344	**39**	**6.2**	0	-	0	-	4	71.2	0	-
RF00029.107	73	**50**	0.3	**50**	**0.0**	37	10.4	**50**	0.2	**50**	0.3
RF00030.30	340	**46**	**6.8**	1	57.3	22	186.8	34	39.3	0	-
sum	-	**1111**	**133.2**	813	271.5	683	1555.5	860	220.9	771	919.7
avg	-	**38.3**	**5.7**	28.0	12.9	23.6	67.6	29.7	10.0	26.6	54.1

TABLE 2.2: Rfam *CP* Results. Summary of the experimental results. The first column is the Rfam identifier, the second column is the length of the structure. The rest of the columns are: (sol) number of runs where the algorithm returned a solution out of 50 executions (for MODENA is the number of sequences in the final population that fold into the target structure), and (time) the average time (in seconds) to find a solution (over the runs that did return a solution), for all the algorithms tested. The last two rows show sum and average values.

Tables 2.2,2.3,2.4 show the comparison results for our method against `MODENA`, `RNA-SSD`, `INFO-RNA` and `RNAinverse`. According to results from Table 2.2, we see that *CP* is far superior to other methods. There are more runs in which the algorithm returns a solution, and *CP* is only slightly slower than `INFO-RNA` on some of the easiest structures (those that are always solved in less than 1 second). Note that times are averaged over runs that returned a solution, and thus it is not entirely fair to compare speed for methods that return various numbers of solutions. In any case, our method is faster overall.

Parameters			CP		INFO-RNA		MODENA		RNA-SSD		RNAinverse	
RF id	*n*	runs	sol	time	sol	time	sol	time	sol	time	sol	time
Z83250	260	50	**50**	2.6	**50**	**0.0**	14	125.6	**50**	2.1	43	213.9
L11935	264	50	**50**	5.0	**50**	**0.0**	16	121.8	**50**	1.1	**50**	109.1
LIU92530	289	50	**50**	10.0	**50**	**0.0**	0	-	1	354.9	17	351.9
U84629	299	50	**50**	5.5	**50**	**0.0**	9	153.1	35	6.4	1	554.6
AF107506	337	50	**50**	9.5	**50**	**0.0**	28	218.2	49	6.6	7	347.6
AF106618	350	50	**50**	20.8	**50**	**0.0**	5	131.9	50	2.1	38	265
AJ011149	376	50	47	140.5	**49**	**0.0**	0	-	26	62.5	1	463.5
S70838	389	50	**50**	27.9	**50**	**0.0**	3	275.4	47	7.1	10	295.2
U63350	418	25	**25**	11.7	**25**	**1.2**	17	191.3	21	2.8	6	346.3
AF141485	473	25	17	51.4	**25**	**0.1**	13	266.6	22	65.1	0	-
U81771	491	25	**25**	28.8	**25**	**0.1**	10	221.6	23	26.2	0	-
AJ130779	506	25	22	70.1	**25**	**0.1**	12	227	23	11	2	507.2
AF096836	646	25	**25**	48.2	24	**0.3**	4	440.4	18	15.5	0	-
X61771	659	25	8	67.0	**18**	**0.3**	0	-	18	129.6	0	-
AJ236455	751	25	0	-	0	-	0	-	**19**	**39.2**	0	-
AJ132572	780	25	23	158.2	**24**	**0.3**	0	-	20	30	0	-
AB015827	856	10	4	245.2	**10**	**5.2**	0	-	9	49.7	0	-
D38777	858	10	1	173.3	**10**	**1.5**	0	-	**10**	17.3	0	-
AF029195	1053	10	7	321.0	**10**	**2.7**	0	-	**10**	42.2	0	-
X81949	1200	10	**6**	197.1	5	**15.7**	0	-	6	48.5	0	-
AJ133622	1296	10	0	-	**8**	**7.8**	0	-	4	128.6	0	-
AF056938	1398	10	5	477.9	**10**	**2.5**	4	319.7	7	58.5	0	-
X99676	1442	10	2	569.2	**8**	**9.8**	1	510.1	7	156.5	0	-
L77117	1475	10	0	-	**5**	**20.4**	0	-	**5**	90.4	0	-
sum	-	680	567	2640.9	**631**	**68**	136	3202.7	530	1353.9	175	3454.3
avg	-	28.3	23.6	125.8	**26.3**	**3.0**	5.7	246.4	22.1	56.4	7.3	345.4

TABLE 2.3: RNA-SSD set 1 *CP* Results. Summary of the experimental results. The first column is the Rfam identifier, the second column is the length of the structure and the third the number of runs executed for all the algorithms. The rest of the columns are: (sol) number of runs where the algorithm returned a solution out of *runs* (for MODENA is the number of correct individuals in the final population), and (time) the average time (in seconds) to find a solution (over the runs that did return a solution), for all the algorithms tested. The last two rows show sum and average values.

Tables 2.3 and 2.4 show a comparison over two sets of biologically relevant structures from [40]. In these cases, *CP* shows comparable performance, and it is only inferior for some of the larger structures, especially in the set from Table 2.3, where it is possible that, given a larger cutoff time, *CP* would find solutions as well. The newest version of `INFO-RNA` performs extremely well, especially in the benchmarks of Table 2.3. Our algorithm is slightly slower than both `RNA-SSD` and `INFO-RNA`.

Table 2.5 shows a summary of all the datasets. Our algorithm *CP* finds, overall, a greater total number of solutions and solves a similar number of structures when compared with `RNA-SSD` and `INFO-RNA`, while *CP* is only slightly slower than these two methods.

We do not claim our approach is faster than previous methods, but it solves more instances

Parameters		CP		INFO-RNA		MODENA		RNA-SSD		RNAinverse	
#	*n*	sol	time	sol	time	sol	time	sol	time	sol	time
1	100	**100**	0.1	**100**	**0.0**	77	19.3	**100**	0.1	**100**	0.1
2	100	**100**	**0.0**	**100**	**0.0**	73	26.2	**100**	0.1	**100**	0.1
3	100	**100**	2.7	**100**	**0.0**	75	69.4	98	1.5	**100**	4.1
4	100	**100**	0.7	**100**	**0.0**	82	104.5	**100**	0.9	**100**	4.1
5	100	**100**	**0.7**	2	165.7	53	245.7	0	-	2	407.9
6	100	**99**	6.2	93	0.8	62	192.2	**100**	**0.0**	3	362
7	100	**100**	9.8	84	**0.8**	68	405.9	64	12.8	4	254.6
8	100	**99**	**7.0**	22	19.5	57	421.1	76	48.4	0	-
9	100	**0**	-	**0**	-	**0**	-	**0**	-	**0**	-
10	100	92	32.9	**100**	**0.1**	57	397.2	99	6.9	13	287.6
sum	-	**890**	**60.0**	701	186.9	604	1881.5	737	70.7	422	1320.5
avg	-	**89**	**6.7**	70.1	20.8	60.4	209.1	73.7	8.8	42.2	165.1

Description	
1	Minimal catalytic domains of the hairpin ribozyme satellite RNA of the *tobacco ringspot virus* (Figure 1a) (Fedor, 2000)
2	U3 snoRNA 5'-domain from *Chlamydomonas reinhardtii*, in vivo probing (Figure 6B) (Antal et al., 2000)
3	*Haloarcula marismortui* 5S rRNA (Figure 2) (Szymanski et al., 2002)
4	VS Ribozyme from *Neurospora* mitochondria (Figure 1A) (Lafontaine et al., 2001)
5	R180 ribozyme (Figure 2B) (Sun et al., 2002)
6	XS1 ribozyme, *Bacillus subtilis* P RNA-based ribozyme (Figure 2A) (Mobley and Pan, 1999)
7	*Homo Sapiens* RNase P RNA (Figure 4) (Pitulle et al., 1998)
8	S20 mRNA from *Escherichia coli* (Figure 2) (Mackie, 1992)
9	*Halobacterium cutirubrum* RNAse P RNA (Figure 2) (Haas et al., 1990)
10	Group II intron ribozyme D135 from ai5γ (Figure 5) (Swisher et al., 2001)

TABLE 2.4: *CP* results for the Benchmarking set 2 used by `RNA-SSD`. Summary of the experimental results. The first column is the Rfam identifier, the second column is the length of the structure. The rest of the columns are: (sol) number of runs where the algorithm returned a solution out of 50 executions (for `MODENA` is the number of sequences in the final population that fold into the target structure), and (time) the average time (in seconds) to find a solution (over the runs that did return a solution), for all the algorithms tested. The last two rows show sum and average values.

	CP	INFO-RNA	MODENA	RNA-SSD	RNAinverse
Total solved	**2568**	2145	1423	2127	1368
Σ avg time	2834.1	526.4	6639.7	1645.5	5694.5
Str solved	53	53	45	**54**	35
avg time	53.5	**9.9**	147.5	30.5	162.7

TABLE 2.5: Summary of solved structures for benchmarking sets 1,2,3. Summary table showing: (1) Total number of successful runs, (2) sum of average times, i.e., the sum of all average times in previous tables, (3) number of structures solved, i.e., number of structures for which the algorithm returned at least one solution, and (4) average time per structure, obtained by dividing the sum of average times for all structures solved by the number of structures solved.

more often and it is at least comparable in speed, which can be counterintuitive given the exhaustive nature of our *CP* approach. We show that the addition of a large number of potentially relevant biological constraints does not jeopardize speed. However, times reported here correspond to finding one solution; finding all solutions or proving that none exists will, of course, require a greater amount of time.

Note that, given the stochastic nature of our algorithm (to prevent helices from being composed entirely of GC pairs), we ran `RNAiFold` several times and provide statistics on these multiple runs for comparison. Even though in the long run, each execution of `RNAiFold` will either return a solution or prove that none exists, the speed with which it can find a solution is influenced by the stochastic nature of our algorithm.

2.4.2 LNS results

Table 2.6 shows a comparison of our *LNS* algorithm over the Rfam set of structures. Recall that we added different variable and value heuristics with the goal of solving more inverse folding subproblems, and of increasing randomization to escape revisiting the same sequences again and again. We performed this comparison to sort out which combination of heuristics is best.

Boldface results signify the best result, i.e. which solves a higher percentage of runs and, in case of a tie, does so with a lower average time.

		Depth Bottom-Up				Leaves to root			
Parameters		A-U-C-G UP		variable UP		A-U-C-G UP		variable UP	
RF id	n	sol	time	sol	time	sol	time	sol	time
RF00001.121	117	50	8.86	50	14.11	**50**	**8.38**	50	13.83
RF00002.2	151	50	23.22	48	150.11	**50**	**22.53**	48	152.41
RF00003.94	161	0	-	13	241.69	0	-	10	253.70
RF00004.126	193	50	0.79	50	1.16	**50**	**0.41**	50	0.88
RF00005.1	74	**50**	**0.40**	50	0.86	50	0.46	50	0.51
RF00006.1	89	**50**	**0.39**	50	6.49	50	2.34	50	8.47
RF00007.20	154	50	5.20	50	6.85	**50**	**2.90**	50	6.43
RF00008.11	54	**50**	**0.01**	50	0.03	**50**	**0.01**	50	0.07
RF00009.115	348	**50**	**20.70**	50	185.07	50	25.46	49	181.30
RF00010.253	357	0	-	0	-	0	-	0	-
RF00011.18	382	0	-	0	-	0	-	0	-
RF00012.15	215	50	1.29	50	8.65	**50**	**1.25**	50	11.15
RF00013.139	185	50	0.23	50	2.00	**50**	**0.18**	50	3.13
RF00014.2	87	50	1.34	50	0.66	50	0.90	**50**	**0.10**
RF00015.101	140	**50**	**4.57**	50	7.80	50	4.94	50	10.10
RF00016.15	129	0	-	0	-	0	-	0	-
RF00017.90	301	50	15.94	50	18.11	**50**	**15.73**	50	21.79
RF00018.2	360	50	18.18	30	272.45	**50**	**15.67**	34	252.14
RF00019.115	83	**50**	**0.13**	50	0.70	50	0.19	50	0.61
RF00020.107	119	0	-	0	-	0	-	0	-
RF00021.10	118	50	0.07	50	0.92	**50**	**0.05**	50	0.65
RF00022.1	148	50	2.21	50	4.38	**50**	**1.10**	50	5.13
RF00024.16	451	0	-	0	-	0	-	0	-
RF00025.12	210	50	0.27	50	8.29	**50**	**0.21**	50	5.39
RF00026.1	102	**50**	**3.15**	50	10.92	50	4.47	50	4.89
RF00027.7	79	**50**	**0.03**	50	0.52	**50**	**0.03**	50	0.32
RF00028.1	344	49	56.48	50	101.35	**50**	**43.50**	50	93.38
RF00029.107	73	50	2.63	50	3.67	50	3.94	**50**	**2.34**
RF00030.30	340	48	9.76	49	49.76	**49**	**6.80**	45	34.10

TABLE 2.6: *LNS* Results for Rfam benchmarking set. Summary of the experimental results. Computational time (in seconds) was measured on an Intel Core i7-2630QM (2GH, 16GB memory, Linux Ubuntu 10.4. Time limit for was set to 10 minutes. The first column is the Rfam identifier, the second column is the length of the structure. The rest of the columns are number of runs where the algorithm returned a solution (over a total of 50 runs) and the average time to find a solution (over the runs that did return a solution), for all the algorithms tested. *Depth bottom-up* heuristic is explained in section 2.3.4.1 and it is the same variable ordering heuristic that the *CP* model uses; *leaves to root* heuristic is a variant which is introduced in section 2.3.5.

The results show that *LNS* with none of these added mechanisms is superior for a larger number of sequences. However, these tables also show that *LNS* (with added variable and value heuristics) is capable of solving more sequences, more quickly, for target structures that are larger and more complex.

2.4.3 Qualitative analysis

As explained before, not all the inverse folding methods use the same criteria to determine which sequences are solutions for a given target structure. In order to include a qualitative analysis of the methods we performed an additional benchmark including 10 different algorithms for all the sequences from datasets 1,2 and 3. Table 2.7 summarizes the properties of sequences returned by each software, including measures that quantify the extent to which the ensemble of low energy structures of a given sequence resembles a target structure (*ensemble defect*, *expected base pair distance*) or how diverse structures are from each other (*Morgan-Higgs structural diversity* and *Vienna structural diversity*), defined in Appendix A. For each of the 63 target structures, each software was run 10 min to generate a quantity of sequences using default settings. `ERD` returns an output 100% of the time, where 85% of the output sequences fold into the target structure. In contrast, `RNAiFold` returns an output 65% of the time, but 100% of its output is guaranteed to fold into the target structure. `IncaRNAtion` returns 41 535 sequences on average for each target, but less than 0.2% fold into the target structure, while `RNAiFold` returns 55 476 sequences on average and 100% fold into the target structure. `INFO-RNA` has over 72% GC-content, due to the initial choice of starting sequence, while `NUPACK-DESIGN` and `RNAiFold` have around 57% GC-content (and moreover, `RNAiFold` allows the user to set a desired GC-content range), while RNA-SSD has close to 36% GC-content.

Method	ERD	FRNA	IncaRNA	InfoRNA	MODENA	Nupack	R-SSD	Rfbinv	RNAiFold	RNAinv
Output (%)	100%	30%	60%	95%	60%	57%	90%	13%	65%	65%
Target (%)	85%	38%	0%	57%	45%	70%	82%	0%	100%	18%
Avg str len	397	122	352	393	234	256	400	74	363	208
Avg output	117	325	41,535	195	50	22	1	2	55,476	935
P(S)	3.32%	1.70%	0.06%	3.17%	11.30%	30.01%	2.24%	0.36%	23.21%	0.78%
Native cont. (%)	85 ± 9	61 ± 15	63 ± 13	76 ± 12	89 ± 9	98 ± 1	85	32 ± 6	93 ± 2	57 ± 12
Avg E	-0.41	-0.24	-0.46	-0.63	-0.46	-0.44	-0.30	-0.14	-0.56	-0.23
Pos entropy	0.33	0.71	0.41	0.44	0.15	0.07	0.36	0.88	0.12	0.80
MH diversity	0.16	0.35	0.21	0.22	0.07	0.03	0.18	0.45	0.06	0.38
Vienna diversity	0.11	0.23	0.15	0.16	0.05	0.02	0.11	0.30	0.05	0.26
Exp bp dist	0.09	0.21	0.27	0.16	0.06	0.01	0.08	0.38	0.03	0.24
Ens def	0.14	0.32	0.39	0.22	0.08	0.02	0.14	0.56	0.04	0.37
Exp num bp	0.28	0.29	0.34	0.30	0.26	0.29	0.28	0.28	0.27	0.28
GC-content (%)	55%	49%	71%	72%	50%	57%	36%	51%	57%	49%

TABLE 2.7: Comparison of 10 programs for RNA inverse folding, benchmarked on 63 target structures, as explained in the text. Averages are given, rounded either to two decimals or to the nearest integer as appropriate. Complete data, with averages and standard deviations, can be found on the `RNAiFold 2.0` web server (`http://bioinformatics.bc.edu/clotelab/RNAiFold2.0/`). FRNA stands for `FRNAkenstein`, R-SSD for `RNA-SSD`, IncaRNA for `IncaRNAtion`, Rfbinv for `RNAfbinv` and RNAinv for `RNAinverse`. Row labels are as follows, whereby measures appearing after the double line have been normalized by dividing by sequence length – for instance, *Avg E* denotes the *normalized* average free energy of the returned sequences, computed as the average, taken over all 63 individual target structures S_0, of average normalized free energies $E(\mathbf{a},S_0)/|\mathbf{a}|$, taken over all sequences $\mathbf{a}$ returned for target structure S_0, where $E(\mathbf{a},S_0)$ denotes the free energy of sequence $\mathbf{a}$ with respect to the structure S_0. The other normalized measures are defined in an analogous manner. *(Unnormalized measures)* Output (%): Fraction of the 63 target structures for which some output was produced. Target (%): Average fraction of output sequences whose MFE structure is the target. Avg str len: Average target structure length, taken over those target structures for which at least one output sequence was returned. Avg output: Total number of sequences returned for all 63 targets, divided by the number of targets for which at least one sequence was returned. P(S): average probability of target structure, defined as the average, taken over all 63 target structures S_0, of the average Boltzmann probability $P(s,S_0) = (\exp(-E(s,S_0)/RT)/Z$, taken over all sequences s returned for target structure S_0. *(Normalized measures)* Avg E: normalized average free energy with respect to target (previously defined). The remaining measures are length-normalized versions of *positional entropy*, *Morgan-Higgs structural diversity*, *Vienna structural diversity*, *expected base pair distance* from target structure, *ensemble defect* with respect to target structure, expected number of base pairs, proportion of native contacts, and GC-content. Measures are defined in Appendix A.

2.4.4 EteRNA results

Lastly, to show the use of introducing design constraints, we selected a set of 12 inverse folding problem instances from the `EteRNA` web site `http://www.eternagame.org/`. Results for both the *CP* and *LNS* programs are shown in Table 2.8. Note that no other approach in the literature can solve these inverse folding problems given their design constraints.

Parameters		Constraints			LNS		CP	
Description	n	MaxGC	MinGU	MaxG	sol	time	sol	time
Prion Pseudoknot	36	-	3	-	10	82.18	10	59.41
Human astrovirus	43	-	6	-	1	478.22	0	-
Homo Sapiens 1 Series	83	-	8	-	10	62.72	7	1.69
HIV Primer Binding Site	107	12	8	-	4	243.14	2	32.18
Homo Sapiens 3	109	10	20	-	1	482.54	0	-
Other Ribosomal RNA	112	12	6	2	10	122.03	10	1.05
Bacilus Subtilis sRNA	113	-	11	-	4	294.84	3	311.81
5s Ribosomal RNA	120	-	4	-	10	30.30	10	30.16
Tribolium Castaneum	123	18	13	-	7	224.71	4	83.77
Oryza sativa 4	176	40	20	-	10	215.83	0	-
Symbiotic plasmid	300	55	10	4	2	206.39	0	-
Telomerase RNA	546	-	15	-	6	297.43	0	-

TABLE 2.8: Results for *CP* and *LNS* on EteRNA data. Summary of the experimental results. Computational time (in seconds) were measured on an Intel Core i7-2630QM (2GH, 16GB memory, Linux Ubuntu 10.4). Time limit was set to 10 minutes. The first column is the description, the second column is the length of the structure, the third column is the maximum number of GC base pairs allowed, the fourth column is the minimum number of GU base pairs and the fifth column is the maximum number of consecutive Gs. The rest of the columns are number of runs where the algorithm returned a solution (over a total of 10 runs) and the average time to find a solution (over the runs that did return a solution), for all the algorithms tested.

The `EteRNA` structures were selected at random, from the vast set of structures available. `EteRNA` classifies its structures in 6 different levels of difficulty (from 0 to 5) and we selected two structures from each level. The constraints represented in this small data set correspond to:

- **MAX GC:** maximum number of GC base pairs allowed. GC stacked base pairs are the most stable base pairs, so limiting the maximum number of base pairs that can appear in the structure increases the difficulty of finding a sequence, at least, for someone trying to solve it "by hand".

- **MIN GU:** similarly, GU base pairs are less stable, and are penalized when they close a stem. Fixing a minimum number of GU base pairs increases difficulty as well.

- **MAX G:** maximum number allowed of consecutive Gs in the sequence. For similar reasons as MAX GC, this increases the difficulty of finding a sequence.

2.5 Interface

`RNAiFold` binaries for Linux and OS X and the source code are publicly available for download at `http://bioinformatics.bc.edu/clotelab/RNAiFold`. The usage is simple for users familiarized with command line tools, where each parameter of the design is preceded by the corresponding flag such as the target structure (e.g -RNAscdstr '(((...)))') or the sequence constraints (e.g. -RNAseqcon NNNAAANNN). Parameters can be also provided in an input file by using the appropriate label preceded by the 'pound' symbol ('#'), where the desired value appears in the next line (see online manual for more details).

For non-expert users `RNAiFold` can be used via webserver at `http://bioinformatics.bc.edu/clotelab/RNAiFold`, a user-friendly web interface that includes a fully automated pipeline to design synthetic RNAs, such as the synthetic hammerheads described in the next chapter.

Three possible types of results can be returned:

- No possible solution: If the target structure (with the specified constraints) has no solution.
- No solution found: If the search time limit is reached and no solution was found within this time limit.
- A list of solutions: For each solution `RNAiFold` shows additional information, unless specified otherwise, such as GC-content, the number of base pairs of each type (strong, weak and wobble), free energy of the structure in kcal/mol and several additional RNA structural measures (see Appendix A). This additional information could be very useful for further analysis and/or to filter or prioritize the solutions with respect to certain criteria.

2.6 Applications

Beyond the synthetic design of RNA molecules from `Rfam` alignments, which will be fully explained in the following chapter, `RNAiFold` has applications in other fields such as the computational analysis of known RNAs, discovery of functional non coding RNAs, determining the relevance of structural motifs and re-engineering messenger RNAs to code the same or similar proteins and to contain desired RNA structural motifs. In this section we describe examples of how `RNAiFold` can be used in each one of those areas.

2.6.1 Free energy analysis of natural RNAs

Given that our *CP* approach can return all sequences whose MFE structure is the given target structure, we can compute the minimum free energy of these structures, as well as their

structural diversity (see Appendix A) and analyze their distribution. Such analysis can provide insights into subtle differences between naturally occurring RNA and synthetic RNA whose minimum free energy structures are identical. Such insights may prove important in future work in synthetic biology and molecular evolution theory.

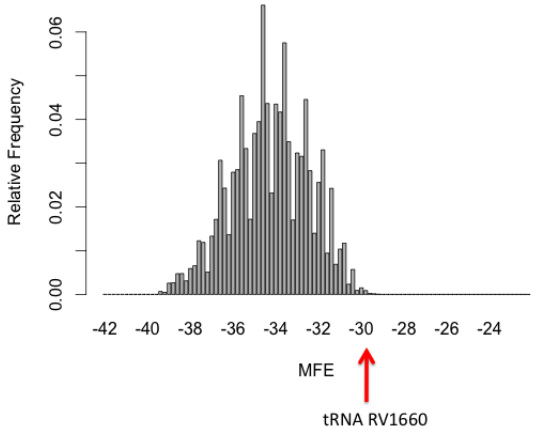


FIGURE 2.5: Distribution of the minimum free energies (MFE) of over 4 million sequences returned by `RNAiFold` *CP*, given as target structure the base paired region of the consensus structure from the `Rfam` RF00005 seed alignment. The only real sequence found in the 4 million sequences returned by *CP* was RV1660 from the `Sprinzl` tRNA database.

As proof of concept, we computed the free energy of all sequences that `RNAiFold` determined (until memory exhaustion), which fold into the following tRNA consensus secondary structure (consensus structure taken from the `Rfam` RF00005 seed alignment):

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

Figure 2.5 plots the minimum free energy distribution for over 4 million sequences generated by our program, which was run until memory exhaustion, where the arrow indicates the

free energy of the *Escherichia coli* val-tRNA (accession RV1600 from `Sprinzl` database [61], tdbR00000454 from `tRNAdb` [62]), a natural tRNA whose sequence was found in the search. The free energy of RV1600 is much higher than the average free energy of sequences that fold into the consensus secondary structure, showing that this natural sequence is not optimized for the free energy.

2.6.2 IRES-like domain discovery

The function of RNA is often determined by its structure. In fact, we usually observe higher secondary structure similarity than sequence similarity between functional non coding RNAs that share a common function [63]. Fast sequence alignment algorithms like `BLAST` are only useful to find functional RNAs with very high sequence homology. Therefore, successful strategies for finding functional RNAs in the growing number of sequenced genomes available should incorporate methods to detect secondary structure similarity. Covariance model-based approaches like `Infernal` [64] have produced good results, and for this reason `Infernal` is used to identify new putative members of the RNA families defined in the `Rfam` database [65]. However, `Infernal` produces low *E-scores* for RNA sequences that share not only structural similarity to secondary structures of RNAs in the training set, but also have sequences that are similar to those in the training set. For this reason, `Infernal` can fail to detect bona fide functional RNAs which have low sequence similarity to the sequences in the training set used to create the covariance model. A possible alternative strategy is to develop a moving-window algorithm, where current genomic window contents are folded using `Vienna RNA Package` `RNAfold`, in order to compare the minimum free energy structure of the window contents with the secondary structure of known functional RNAs of a certain family. However, the computational

time required for this genome-scanning strategy is prohibitive, since RNA folding algorithms require time that is cubic in input length.

Internal ribosomal entry site (IRES) elements are RNA chains that promote translation initiation of *mRNAs* independently of the five-prime m^7G cap, overriding part of the protein dependent translation initiation pathway. In [66] the authors discovered a previously unknown functional IRES-like subdomain in *D. melanogaster* using a novel approach that combines the exhaustive inverse folding capabilities of `RNAiFold` with the search speed of `BLAST` and the reasoned use of published knowledge about IRES elements. Here we briefly describe the computational pipeline used in this work, depicted in Figure 2.6 extracted from [66].

Using `RNAiFold` the authors generated more than a hundred thousand sequences whose MFE structure is the known secondary structure of the domain 3 of the type II IRES from foot and mouth disease virus (FMDV) and also contain three important sequence motifs (a GNRA, RAAA and C-rich region in specific hairpins) which are believed play a key role in the internal ribosomal translation.

Due to the exhaustive search strategy used in `RNAiFold` there is high sequence similarity between groups of the sequences returned, so sequences that differ in one nucleotide were clustered together and a representative sequence of each cluster was selected. Then, the representative sequences were used as input for the BLAST algorithm in order to find similar sequences in the transcriptomic databases.

Hits returned by `BLAST` were filtered by function and location; for example functional IRES must be located in the 5' untranslated region or at the beginning of the coding region due to its translation initiation function. A final assessment based on the codon usage bias and sequence

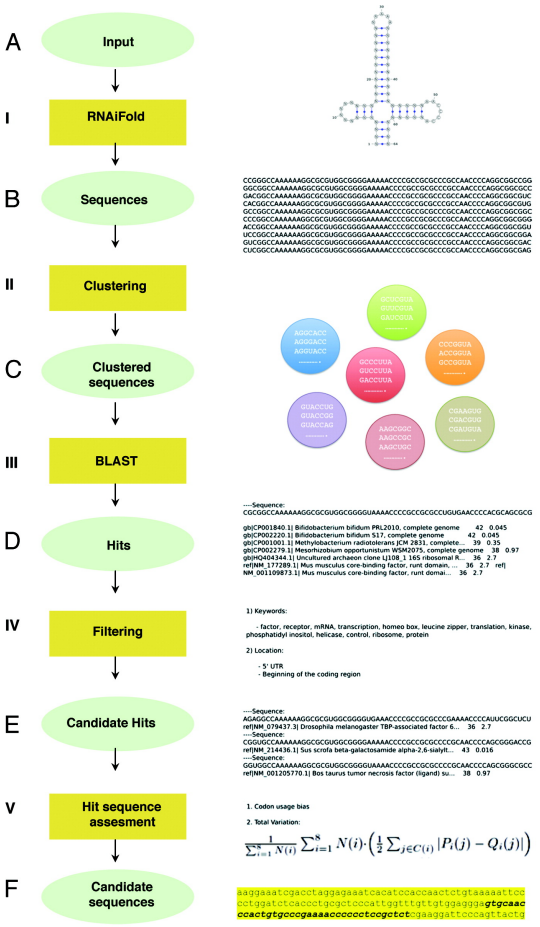


FIGURE 2.6: Computational pipeline for IRES-like domain discovery (figure extracted from [66]). (A) Target RNA structure and sequence constraints used as input for `RNAiFold`. (B) Sequences returned by `RNAiFold` were clustered (II). (C) One representative of each cluster was BLASTed (III) with match/mismatch weights of 1/-1 and gap existence/extension penalties of 1/2. (D) BLAST hits were then filtered (IV) using on the genomic locations and keywords indicated. (E) Candidate hits located in coding sequences were assessed (V) by computing total variation and codon usage bias. (F) Finally, the most promising candidate sequences were selected for experimental validation.

variation pointed to an IRES-like sequence located in the 5'UTR region of the TAF6 gene of *D. melanogaster*. Further biochemical validation by a luciferase reporter assay confirmed that this sequence in fact promotes 5' cap independent translation.

2.6.3 Determining the relevance of structural motifs

As we just showed, the function of IRES elements is intimately linked to their RNA structure. Specifically in picornavirus type II IRES elements, a conserved motif is the pyrimidine-tract (*Py tract*) at domain 5, located upstream of the functional initiation codon. By computationally designing synthetic RNAs to fold into a structure that sequesters the pyrimidine tract in a hairpin, in [67], we established a correlation between predicted inaccessibility of the pyrimidine tract and IRES activity, as determined in both *in vitro* and *in vivo* systems.

RNA viruses in general, and FMDV in particular, are characterized by a high genetic variability [68]. This feature, however, does not affect every position of the genome to the same extent. As it occurs in many RNA regulatory elements, evolutionary conserved motifs involved in IRES activity preserve RNA secondary structure in addition to short stretches of nucleotide sequence. The *Py tract* of picornavirus IRES elements belonging to type I and II tolerates some variations in the order of U/C residues CCC [69, 70]. In contrast, there is high sequence variability within the region that separates the *Py tract* from the first functional AUG codon, a feature that led to propose that this region was a spacer. However, both the length and the structure of the spacer region could contribute to ensure recognition of the authentic initiator codon by the translation machinery [71, 72] .

In spite of the mutational analysis carried out in the picornavirus IRES *Py tract*, it remained

elusive whether having the *Py tract* in a unique structural conformation is an absolute requirement for IRES activity. To answer this question we designed candidate RNA sequences adopting different conformations of domain 5, but harboring a pyrimidine tract of the same length as that of wild type IRES. Hence, the pyrimidine tract could be either unpaired or base-paired in stem-loops with different stability. For this, we made use of the FMDV IRES, a type II IRES element whose secondary structure is well characterized [73] to construct synthetic RNA domains capable of adopting different structures within domain 5, at the distal 3' end of the IRES element. Domain 5 consists of three structural motifs: a hairpin, a pyrimidine-rich tract and a variable sequence (Figure 2.7A). The hairpin has been described as the binding site of eIF4B [74, 75] , while the *Py tract* provides the binding site for PTB [76] . It should be noted that both, the hairpin and the pyrimidine tract are strongly conserved among field isolates, whereas the spacer region shows high sequence variability (Figure 2.7B). Taking advantage of this feature, novel subsets of IRES elements were generated by replacing the wild type sequence with the computationally designed RNA element fused to the luciferase open reading frame sequence. Functional and structural analysis of these elements provided information on the relation-ship between the accessibility of the *Py tract* and the structure of the hairpin of domain 5 with IRES activity.

We took advantage of the variability of the proximal spacer and the permissiveness of the distal spacer to design RNA candidates having different locations of the hairpin, while maintaining the pyrimidine tract sequence at the same position with respect to the wild type IRES sequence. RNA design strategy using `RNAiFold` involves the generation of hundreds of thousands or millions of sequences that fold into a given target structure, followed by the application of various computational filters to prioritize the best candidates for experimental validation. This is

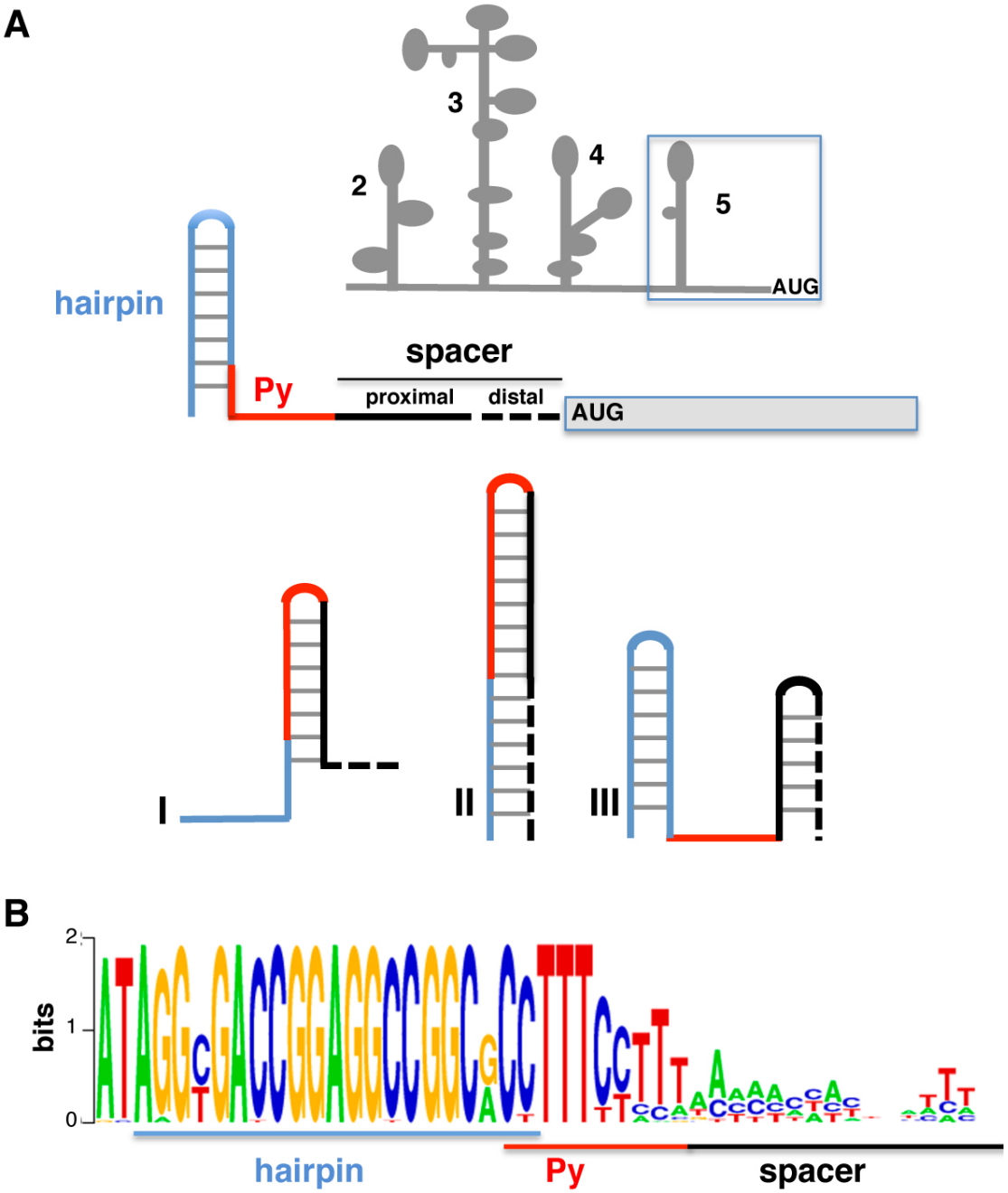


FIGURE 2.7: *(A)* Design strategy for synthetic RNAs harboring distinct conformations of the pyrimidine tract. *(Top)* Schematic representation of the FMDV IRES, organized in domains 2, 3 4 and 5. Diagram of the structural motifs of the wild type domain 5: a hairpin (blue line), pyrimidine tract (Py) (red line) and a spacer (black line), including a proximal and a distal sequence, upstream of the functional start codon (AUG) of the reporter gene. *(Bottom)* Design of RNA families I, II or III. *(B)* The pattern of nucleotide conservation (measured in bits) of domain 5 (nts 417–462) is represented by a sequence logo obtained from the alignment of the FMDV IRES sequences of field isolates deposited in data banks.

relatively straightforward by using the *in silico* design with `RNAiFold`, which allows exhaustive sequence generation and control over specific thermodynamic properties in the candidate selection process. In this work, we analyzed the effect in cap independent translation of sequestering the *Py tract* into short and long hairpins as depicted in Figure 2.7A I and II.

For short hairpins, we used the following input file for target structure used as input to `RNAiFold` [55].

```
> Py weak stem
#RNAscdstr
..............((((((..((((((....))))))..))))))...
#RNAseqcon
NUNNNNNNNNNNAGGNNNNNNCNUUYYYYYYNNNNNNNNNNNNNNNGAG
#temp
30
#MAXsol
0
#dangles
2
```

No sequence constraints were imposed in the inverse folding pipeline for the proximal spacer region, given the large sequence variability of this region. However, specific *Py tract* nucleotides that are conserved and might be relevant for IRES function independent of its structure were fixed in the design. `RNAiFold` generated a large number of sequences that fold into the target structure, subsequently filtered by Boltzmann probability of forming the target structure. Then, we computationally estimated the accessibility of the *Py tract* at 30 °C for each candidate using three different methods in order to prioritize candidates for experimental validation (Tables 2.9 and 2.10): First (`PLfold_PTB`), using `RNAplfold` [77] from `Vienna RNA Package` with the options `-L 84` (length of the sequence) and `-u 7`(length of *Py tract*) and extracting the value corresponding to the starting position of the *Py tract* we obtained an estimate of the probability of having all seven positions of the *Py tract* unpaired. Second (`Sample_PTB-5`), we

sampled 100,000 low energy structures from the thermodynamic ensemble using `RNAsubopt -d2 -p 100000` [78] from `Vienna RNA Package` and computed the proportion of structures in which at least five of the seven positions of the *Py tract* were unpaired. Third (`ProbUnpaired_PTB`), we computed the probability $1 - p(i)$ that position i of the py is unpaired, hence the probability that all the positions in the *Py tract* are unpaired is $prob(py) = \prod_{i \in py} 1 - p(i)$.

The *in silico* design and posterior filtering process produced sequences with high probability of having the *Py tract* base-paired. Among these sequences, sequence I-20 was selected (see Figure 2.8A) because it had the highest probability for the *Py tract* to be base-paired. Moreover, four additional sequences (I-2, I-3, I-4 and I-7)(see Figure 2.8A) with moderate and low probability among the filtered sequences were selected in order to later establish correlations between measures and IRES activity. Measures shown to have high correlation with IRES activity were then used as optimization criteria in the second design round. IRES activity of the selected candidates was determined using a cell-free system programmed with equal amounts of in vitro synthesized RNA. As shown in Figure 2.8B, the efficiency of protein synthesis measured as the ratio of ^{35}S-labeled luciferase (LUC) polypeptide to chloramphenicol acetyl transferase (CAT) polypeptide in rabbit reticulocyte lysates (RRL) was reduced in all selected candidates relative to the wild type RNA. Note that the activity of RNAs I-2, I-3, I-4, and I-7 was very similar. However, the activity of RNA I-20 was reduced to a higher extent than all other candidates. These results were confirmed using a different system, where we measured luciferase (Luc) activity expressed from bicistronic RNAs in transfected BHK-21 cells (Figure 2.8C).

Since the reduced activity of RNA I-20 could be related to the higher stability of the hairpin sequestering the *Py tract*, we attempted to generate a second round of candidates, selected on

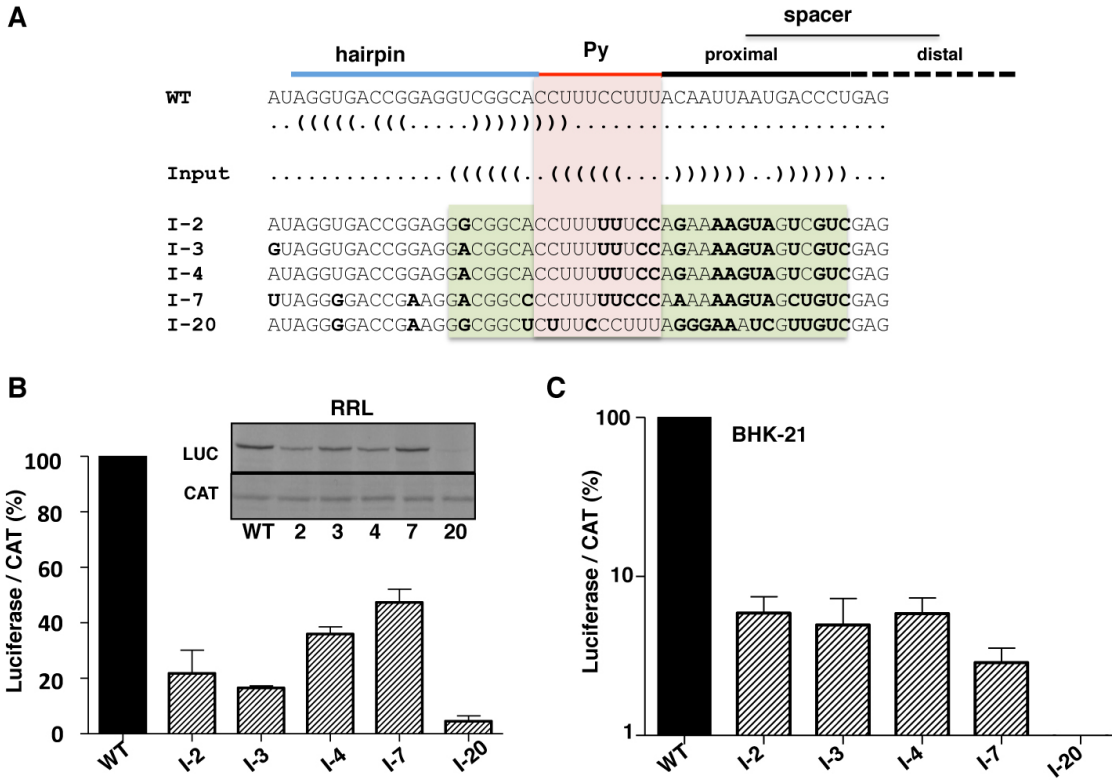


FIGURE 2.8: *(A)* Alignment of sequences belonging to family I candidates with the sequence of wild type (WT) domain 5. Blue, red and black lines depict the position of the hairpin, the pyrimidine tract and the spacer sequences, respectively. The RNA structure of domain 5 and the input for RNAiFold are shown in dot bracket notation. Bold letters denote the sequence changes relative to the wild type IRES sequence. A pink box denotes the location of the pyrimidine tract, while a green box depicts the residues predicted to form a hairpin in the selected candidates. *(B)* In vitro synthesized bicistronic RNAs (200 ng) bearing the WT or the candidate I-2, I-3, I-4, I-7, or I-20 sequences were used to program translation in rabbit reticulocyte lysates (RRL) during 60 min at 30 °C. ^{35}S-labeled proteins were resolved in 12% SDS-PAGE. An autoradiogram of a representative assay is shown in the insert. The intensity of ^{35}S-labeled luciferase and CAT polypeptides was measured in a densitometer; the ratio of luciferase/CAT was calculated and then normalized to the intensity observed in the WT RNA, which was set to 100%. Values correspond to the mean (± SD) of three assays. *(C)* Relative IRES activity was determined in transfected BHK-21 cells as the ratio of luciferase to chloramphenicol acetyl-transferase expressed from bicistronic constructs carrying the candidate sequences, normalized to the activity observed for the wild-type IRES (set to 100%). Each experiment was performed in triplicate and repeated at least three times.

Name	Luciferase intensity	Probability of structure[a]	Ensemble defect[a]	Expected base pair distance[a]	PLfold_PTB	SamplePTB_5	ProbUnpaired_PTB
I-2	0.38	0.3	9.72	6	0.0994	0.148	0.486
I-3	0.28	0.32	7.87	4.81	0.0734	0.114	0.502
I-4	0.36	0.33	8.01	4.9	0.0763	0.117	0.500
I-7	0.49	0.24	10.11	5.89	0.1506	0.194	0.451
I-20	0.06	0.22	14.52	9.06	0	0	0.569
Correlation[b]	1	0.1	0	-0.1	1	1	-1

TABLE 2.9: Measures of candidates in family I: [a] Calculated using the target structures, as described in Appendix A. [b] Spearman coefficient

the basis of adopting a stable hairpin that sequesters the *Py tract* within a long stem-loop. In order to generate sequences for this family, we again used RNAiFold with the following input.

```
> Py strong stem
#RNAscdstr
(((((((((((.((((.(((.((((((((((.....)).)))))))).)))).)))).)))))..)))).)))).........
#RNAseqcon
NUNNNNNANNNNAGGNNNNNRCYUUYYYYYYRNNNNNNNNNNNNNNNGAGCUCGAGCUUGGCAUUCCGGUACUGUUGGUAAAAUG
#temp
30
#MAXsol
0
#dangles
2
```

In this case, our target structure takes advantage of the distal region of the spacer to create a longer and more stable stem-loop. The sequence of the distal region was fixed, and all previous considerations for family I hold in this new design. Again, we filtered the thousands of sequences returned by `RNAiFold` using Boltzmann probability of target structure; in this case, we only considered sequences with a probability greater than 0.02. Among these, we selected two sequences (II-A and II-B)(see Figure 2.9A) with the highest `ProbUnpaired_PTB` since it was one of the measures that had the best correlations with IRES activity. As devised, the family II of candidates (Figure 2.9A) differed from family I in the capacity to adopt a stable hairpin including the entire spacer that separates the IRES from the functional AUG codon

for luciferase. Determination of IRES activity for family II members measured by in vitro assay indicated a strong decrease of luciferase synthesis (Figure 2.9B), which was akin to that found for construct I-20. Similar results were obtained using monocistronic constructs. Also, the relative IRES activity of the constructs II-A and II-B measured as the ratio of luciferase to CAT activities determined in the same extract in BHK-21 transfected cells showed a decreased activity relative to the wild type RNA(Figure 2.9C), which was similar to the results of the in vitro translation assays. These results suggest that sequestering the *Py tract* within a hairpin inactivates IRES activity; furthermore, the stronger the stability of the hairpin, the higher the inhibition of protein synthesis.

Name	Luciferase intensity	Probability of structure[a]	Ensemble defect[a]	Expected base pair distance[a]	PLfold_PTB	SamplePTB_5	ProbUnpaired_PTB
II-A	0.13	0.05	13.35	8.38	0.0159	0.021	0.976
II-B	0.09	0.05	13.34	8.38	0.0157	0.02	0.977

TABLE 2.10: Measures of candidates in family II: [a] Calculated using the target structures, as described in Appendix A.

In both design rounds one sequence was selected to analyze its SHAPE reactivity in solution in order to confirm the predicted secondary structure. In both cases, the RNA structure model obtained by imposing SHAPE reactivity on `RNAstructure` resembled the structure used as input for the inverse folding pipeline (Figure 2.9A), greatly differing from the wild type RNA.

To further test our hypothesis, we generated another construct by site-directed mutagenesis (Figure2.10A). This RNA was predicted to preserve the *Py tract* in an unpaired region within two hairpins; the first hairpin exactly matched the wild type, while the second hairpin occupied the spacer region (see Figure2.7A-III). Measurement of IRES activity by *in vitro* and *in vivo* assays indicated that this is almost as active as the wild type IRES element *in vitro* (Figure 2.10B), and at least 3 to 10-fold more active than any member of families I and II in BHK-21 cells. These

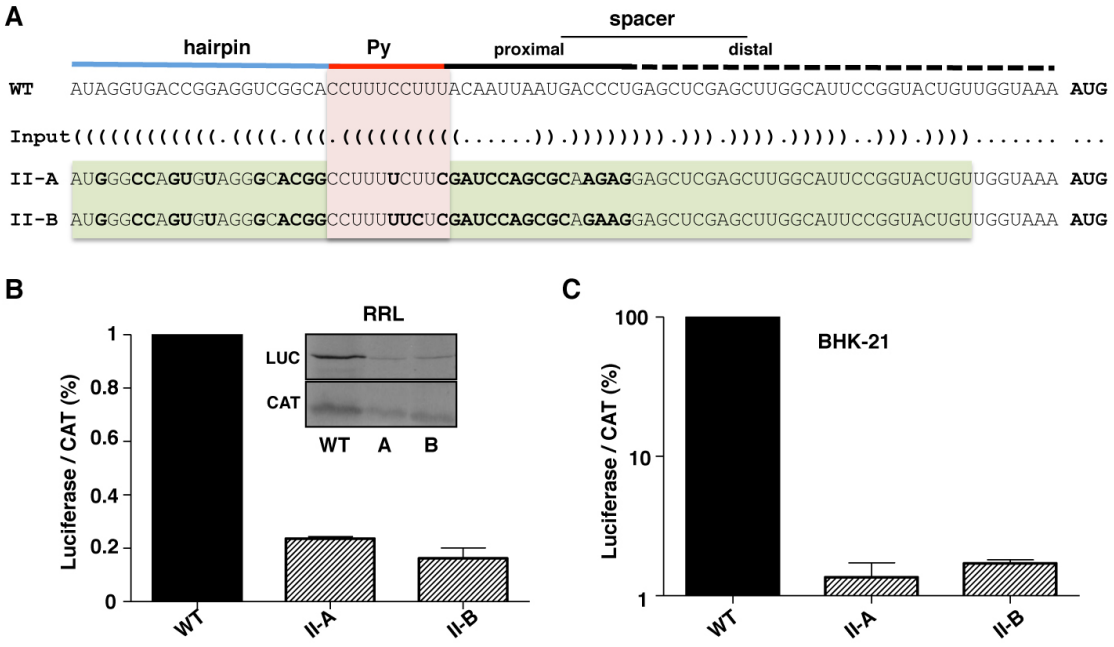


FIGURE 2.9: *(A)* Alignment of sequences belonging to family II candidates and the sequence of WT domain 5. The RNA structure of domain 5 and the input for RNAiFold are shown in dot bracket notation. All symbols are used as in Figure 2.8A. *(B)* In vitro synthesized bicistronic RNAs (200 ng) bearing the wild type RNA WT or the candidates II-A or II-B were used to program translation in RRL during 60 min at 30 °C. An autoradiogram of a representative assay is shown in the insert. The intensity of ^{35}S-labeled luciferase and CAT polypeptides was measured in a densitometer; the ratio of luciferase/CAT was calculated and then normalized to the intensity observed in the WT RNA, which was set to 100%. Values correspond to the mean (± SD) of three assays. *(C)* Relative IRES activity was determined in transfected BHK-21 cells as the ratio of luciferase to chloramphenicol acetyl-transferase expressed from bicistronic constructs carrying the candidate sequences, normalized to the activity observed for the wild- type IRES. Each experiment was performed in triplicate and repeated at least three times.

results suggest that maintaining the *Py tract* in an unpaired region is an important feature for the initiation of protein synthesis, whereas modification of the structural organization of the spacer region is tolerated.

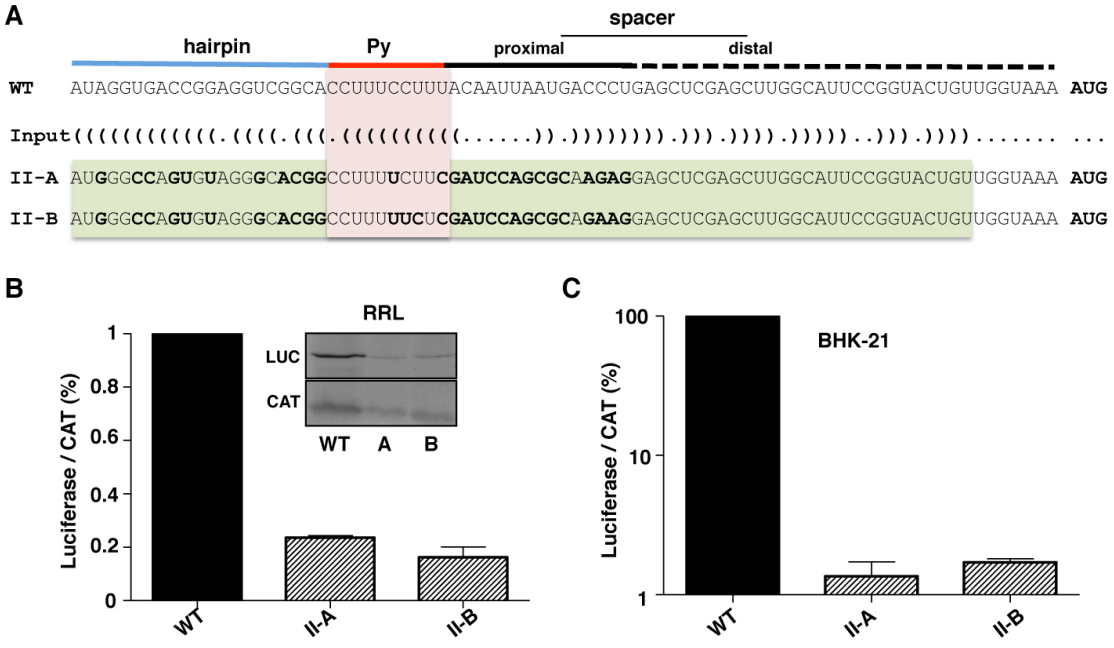


FIGURE 2.10: *(A)* Alignment of candidate III-1 with the sequence of domain 5. Symbols are used as in Figure 2.8A. A pink box denotes the location of the pyrimidine tract, while a blue box depicts the residues predicted to form a hairpin in the wild type RNA. *(B)* (Left panel) Invitrosynthesized bicistronic RNAs (200 ng) bearing the wt or the candidate III-1 domain 5 were used to program translation in RRL during 60 min at 30°C. An autoradiogram of a representative assay is shown in the insert. The intensity of 35S-labeled luciferase was measured in a densitometer, and normalized to the intensity observed in the wt RNA, which was set to 1. Values correspond to the mean (±SD) of three assays. (Right panel) Relative IRES activity was determined in transfected BHK-21 cells as the ratio of luciferase to chloramphenicol acetyl-transferase expressed from bicistronic constructs carrying the candidate sequence, normalized to the activity observed for the wild-type IRES (set to 100%). The experiment was performed in triplicate and repeated at least three times.

In addition, the test whether beyond the effect of sequestering the *Py tract*, the decrease in IRES function could be due to the disorganization of the hairpin of domain 5. We measured

the effect of mutations that destabilize the basal or the apical base pairs of the hairpin, but conserving the *Py tract*. The results indicated that the secondary structure of this hairpin is also important for IRES function.

In summary, we applied a combination of *in silico*, *in vitro* as well as *in vivo* approaches to design modified structural motifs of the type II picornavirus IRES element in order to determine the relevance of the presence of the stem-loop and the structural accessibility of the *Py tract* for initiation of protein synthesis, where use of `RNAiFold` made relatively straightforward the *in silico* design process. Our results showed that both, sequestering of the *Py tract* and disorganization of the hairpin could lead to a significant change in the binding of proteins that interact with this IRES region and appear to be necessary for functional picornavirus type II IRES elements.

2.6.4 SECIS design

Prokaryotes, archaea, and eukaryotes employ the UGA stop codon to code for selenocysteine, rather than terminating protein translation, provided that a *selenocysteine insertion* (SECIS) element occurs downstream of the UGA stop codon. The SECIS element is a $\sim$ 42 nt sequence having conserved nucleotides at certain positions, which folds into a stem-loop secondary structure [79] – see Figure 2.11. In prokaryotes, the SECIS element lies immediately after the UGA stop codon, while in eukaryotes and archaea it lies in the 3′ untranslated region [80]. In the formate dehydrogenase F (fdhF) gene of *Salmonella enterica* (GenBank: CDS70432.2), the 42-nt sequence UGACACGGCC CAUCGGUUGC AGGUCUGCAC CAAUCGGUCG GU consists of the UGA stop codon immediately followed by the SECIS element. This sequence folds

into the stem-loop structure shown in Figure 2.11 (left), and codes the 14 residue peptide UHGPSVAGLHQSVG ('U' denotes selenocysteine).

In contrast, the homologous 14 residue peptide of the fdhF protein of *Raoultella ornithinolytica* is given by CHGPSVAGLQQALG, where cysteine appears instead of selenocysteine. Unlike *S. enterica*, the $42 = 14 \cdot 3$ nt portion of the fdhF gene of *R. ornithinolytica* (Genbank AJF73661.1) begins with UGC, which codes for cysteine, rather than UGA, a stop codon that codes for selenocysteine in the presence of a SECIS element; moreover, the 42-nt sequence of *R. ornithinolytica* does not fold into a stem-loop SECIS structure.

Figure 2.11 illustrates how `RNAiFold` can be used to re-engineer selenoproteins from cysteine-bearing proteins. Using as target structure to be the MFE structure of the 42-nt RNA from *S. enterica*, we set as sequence constraints the bulged U18 and GGUC hairpin identity (known to be important for SECIS functionality [81, 82]), and as amino acid constraints the 14-mer of *R. ornithinolytica*, with 'C' replaced by 'U'. In order to avoid critical amino acid substitutions, we allowed a minimum BLOSUM62 similarity of -1 with respect to the target amino acid sequence. Using the following input:

```
> Selenocysteine insertion in AJF73661.1 140
.....(((((.((.(((.(((....)))))).)).)))))..
UGANNNNNNNNNNNNNNUNNNGGUCNNNNNNNNNNNNNNNNN
#MAXsol
0
#AAtarget
UHGPSVAGLQQALG
#AAsimilCstr
-1
#MaxBlosumScore
1
#dangles
2
```

In 0.24 seconds `RNAiFold` determined the optimal solution UGACACGGGC CCUCGCUUGC AGGUCUGCAG CAAGCGCUCG GA, which begins by the UGA stop codon, folds into the requisite SECIS stem-loop and translates the 14-mer UHGPSLAGLQQALG, which has an optimal BLOSUM62 similarity score of 68 out of 71 respect to the target amino acid sequence UHGPSVAGLQQALG. Note that `RNAiFold` did not only found a solution in less that one second, but also determined that there is no sequence that meets the given constraints and has a BLOSUM62 similarity score higher than 68. This construct has not been experimentally validated, however this example is provided to illustrate one of the possible applications of using amino acid constraints with BLOSUM62 similarity score optimization, a unique feature of `RNAiFold` among the inverse folding software.

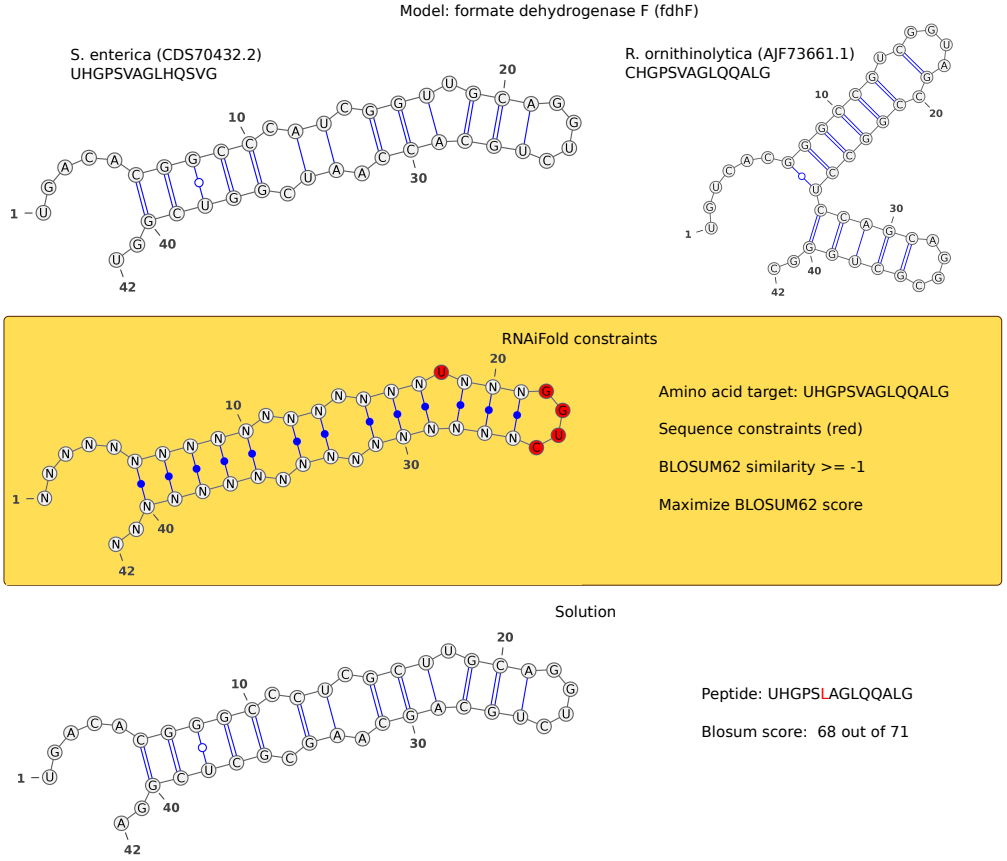


FIGURE 2.11: Using `RNAiFold` to re-engineer selenoproteins from cysteine-bearing proteins. *(Top)* Sequence model: *(Left)* The SECIS element from formate dehydrogenase F (fdhF) gene of *S. enterica* is used as secondary structure template. *(Left)* The peptide to be re-engineered is the homologous region of the fdhF protein of *R. ornithinolytica*, where cysteine appears instead of selenocysteine. *(Middle)* `RNAiFold` constraints: Target structure from the *S. enterica* SECIS element. Sequence constraints marked in red in the bulge and loop region. The target amino acid sequence is the original *R. ornithinolytica* peptide where the initial cysteine has been replaced by a selenocysteine. Minimum BLOSUM62 amino acid similarity is set to -1 and BLOSUM62 maximization option is enabled. *(Right)* Optimal solution returned by `RNAiFold`: There is no solution with the maximum score of 71. So the optimal solution contains a valine to leucine substitution, with a BLOSUM62 score of 68.

2.7 Conclusion

In this chapter we have presented the first algorithm for complete inverse folding. `RNAiFold` shows a performance that is at least comparable with the current available methods to solve the RNA inverse folding problem and incorporates more design constraints than any other software available, including the use of different energy models and folding algorithms, stipulation of a partial target structure, stipulation of prohibited (incompatible) base pairs and amino acid constraints. Given a target non-pseudoknotted hybridization complex of two structures, `RNAiFold` can output pairs of sequences, whose MFE hybridization complex is equal to the target.

The current implementation includes two search strategies, *CP* and *LNS*, appropriate for different design problems depending of whether completeness or search speed are required respectively. Its modular design makes `RNAiFold` easily scalable, as evidenced by the new capabilities included in the subsequent versions of the software.

All these features make `RNAiFold` a unique tool with applications in several fields such as the analysis of known RNAs, discovery of unknown functional RNAs and synthetic design.

In fact, to our knowledge `RNAiFold` and `NUPACK-DESIGN` are only RNA inverse folding methods that have been used for experimentally validated *de novo* synthetic design. Moreover, as we have shown, the capabilities of `RNAiFold` go beyond the scope of RNA synthetic design, with several applications in the field of RNA.

Chapter 3

Computational design of functional hammerhead ribozymes

3.1 Introduction

This chapter concerns the synthetic design of ribonucleic acid molecules, using `RNAiFold`, which can determine all RNA sequences whose minimum free energy secondary structure is a user-specified target structure. Using `RNAiFold`, we designed ten *cis*-cleaving hammerhead ribozymes, all of which were shown to be functional by a cleavage assay. We additionally used `RNAiFold` to design a functional *cis*-cleaving hammerhead as a modular unit of a synthetic larger RNA. Analysis of kinetics on this small set of hammerheads suggests that cleavage rate of computationally designed ribozymes may be correlated with *positional entropy*, *ensemble defect*, structural flexibility/rigidity and related measures.

Artificial ribozymes have been designed in the past either manually or by SELEX (Systematic Evolution of Ligands by Exponential Enrichment); however, this appears to be the first purely

computational design and experimental validation of novel functional ribozymes. `RNAiFold` is available at `http://bioinformatics.bc.edu/clotelab/RNAiFold/`.

3.1.1 Organization

This chapter is organized in the following fashion. First, we give an overview of previous approaches used to create functional ribozymes. Then we describe in detail the computational pipeline followed to design functional type III hammerhead ribozymes from `Rfam` alignments, as well as the modular design of type III hammerhead ribozymes within another structure. We continue showing the results of the experimental validation *in vitro*, where all ten candidates selected from the sequences produced the by our computational pipeline were shown to be functional. Our experimental validation includes kinetics of cleavage in order to determine the relation between the parameters used in the design and the cleavage efficiency. Finally, we investigate the structural implications of the conserved GUH motif in type III hammerheads using `RNAiFold`, where our results suggest that the conservation of the H (no G) at the position of cleavage among type III hammerhead ribozymes is driven by structural constraints.

3.2 Background

Ribonucleic acid enzymes (a.k.a. ribozymes) are catalytic RNAs with enzymatic capabilities that, similar to their protein counterparts, can catalyze and accelerate the rate of biochemical reactions while maintaining a great specificity with respect to the substrate they act upon. In general, ribozymes can catalyze the transesterification of phosphodiester bonds, acting in *cis* by self-cleavage, or in *trans* by cleaving other RNAs. There exist different types of ribozymes,

all with a very well defined tertiary structure: group I introns – self-splicing ribozymes, that were first observed for the intron of the nuclear 26S rRNA gene in *Tetrahymena thermophila* [83, 84]; group II introns – self-splicing ribozymes, which produce ligated exons and an excised intron-lariat as products of the splicing procedure [85]; ribonuclease P (RNase P) – a ubiquitous endoribonuclease that processes the 5′ end of precursor tRNA molecules, producing 5′ phosphoester and 3′ OH termini [86]; and small self-cleaving pathogenic RNAs, such as hammerhead ribozymes [87, 88], as well as the hairpin and the hepatitis delta virus ribozymes [89].

3.2.1 RNA Synthetic Biology

In response to the increased understanding and appreciation of the role RNA plays in biology, the last decade has seen a surge in the field of RNA synthetic biology. Several laboratories have successfully produced synthetic RNA sequences capable of self-cleaving, sensing small molecules *in vivo* or *in vitro*, as well as regulating gene expression [90, 91]. Many of these efforts have focused on the creation of allosteric ribozymes, or gene regulatory elements that can be used for further application.

Selection-based approaches (e.g. SELEX, or Systematic Evolution of Ligands by EXponential enrichment [92, 93]) have proved very powerful for generating a range of RNAs with a variety of capabilities. Allosteric ribozymes that are inhibited or activated by specific small molecules have been achieved by utilizing a pre-existing self-cleaving ribozyme sequence coupled to either an existing aptamer [94], or one derived through selection [95]. Additionally, SELEX has been coupled with *in vivo* screens to create RNAs with gene-regulatory activity in response to specific small molecule [96] or protein stimuli [97, 98].

Design-based approaches have also been successful at creating RNAs with engineered functions. By a series of manually determined pointwise mutations, where biological activity was repeatedly assayed for intermediate structures, a single RNA sequence was designed to simultaneously support the catalytic activities of both the self-cleaving hepatitis delta virus ribozyme, and the class III self-ligating ribozyme [99]. Several approaches to designing genetic regulators mimic the action of small regulatory RNAs by introducing engineered trans-acting RNAs to occlude a ribosome binding site or start codon to inhibit translation. Gene expression may be altered in such systems by inhibiting the original RNA with a second trans-acting RNA [100], or through utilization of a ligand binding domain (aptamer) to induce an alternative RNA structure that does not interact with the transcript of interest [101]. In addition, hammerhead ribozymes have been used to target the HIV virus [102, 103] by modifying sequences within base-pairing regions to target a specific sequence of viral RNA.

As the complexity of synthetic RNA devices increases, there is an increasing need to go beyond *ad hoc* manual approaches, and *in vitro* selection methods. RNA molecules have been rationally designed by the assembly of structural RNA tertiary fragments/motifs, extracted from X-ray and NMR structures of natural RNA molecules [104, 105]; see also [106]. Using computational methods with *reaction graphs*, with subsequent validation using atomic force microscopy, molecular programs have been executed for a variety of dynamic DNA constructs, ranging from hairpins, binary molecular *trees*, to bipedal walkers [107].

However, to the best of our knowledge, no group has previously designed a ribozyme by purely computational means, using RNA inverse folding, and subsequently validated the ribozyme functionality; this is our contribution in the present chapter.

3.3 Design process

3.3.1 Computational Methods

As explained in the previous chapter, `RNAiFold` returns sequences whose minimum free energy (MFE) structure is a given target structure, whereby the user may choose to use the free energy parameters from Turner 1999[31, 32], Turner 2004[59] or Andronescu 2007[60]. By default `Vienna RNA Package 1.8.5` and `Vienna RNA Package 2.0.7` use Turner 1999 parameters and Turner 2004 parameters respectively. By abuse of notation, let `RNAiFold'99` [resp. `'04`] denote the program `RNAiFold` with the corresponding energy parameters.

As target structure for our computationally designed type III hammerheads, we selected the secondary structure of a portion of the plus polarity strand of Peach Latent Mosaic Viroid (PLMVd) (isolate LS35, variant ls16b) from Rfam family RF00008 [50] having accession code AJ005312.1/282-335. The reason we chose PLMVd AJ005312.1/282-335 was that this is the only RNA sequence in the seed alignment of RF00008, whose MFE structure is identical to its Rfam consensus structure, when computed by `RNAiFold'99`– see Appendix C for a precise definition of Rfam consensus structure. Moreover, as shown in Figure C.1, the MFE structure computed by `RNAiFold'04` differs markedly from the Rfam consensus structure of PLMVd AJ005312.1/282-335, hence we used `RNAiFold` with the Turner '99 energy parameters from `Vienna RNA Package 1.8.5`. In summary, the target structure for `RNAiFold'99` was taken to be

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

which is both the RNAiFold'99 MFE structure as well as the Rfam consensus structure of PLMVd AJ005312.1/282-335.

Numerous biochemical and structural studies have pinpointed key nucleotides in the hammerhead ribozome that are required for catalysis [108, 109, 110]. However for an efficient, purely computational design of synthetic hammerheads, it is important to rely only on sequence conservation results from reliable multiple alignments. The Rfam web site image http://rfam.sanger.ac.uk/family/RF00008#tabview=tab3 clearly shows certain regions of the 56 nt consensus sequence have highly conserved sequence identity. Based on this observation, we computed the nucleotide frequency for the seed alignment of Rfam family RF00008 for those positions aligned to the nucleotides of the 54 nt PLMVd AJ005312.1/282-335. Figure 3.1 (left) shows the sequence logo of positions aligned to PLMVd AJ005312.1/282-335. Table C.1 shows that sequence identity exceeds 96% for the 15 positions 6-7, 22-25,27-29, 44-49 of PLMVd in the seed alignment for Rfam family RF00008 consisting of 84 sequences. For that reason, the nucleotides in PLMVd at these 15 positions were provided as a constraint for RNAiFold, thus fixing approximately 28% of the 54 nucleotides. Note that the cleavage site at C8, discussed below would have been included in the constraints, had we chosen to retain positions of at least 95%.

From the literature, it is well-known that hammerhead cleavage sites are of the form NUH (e.g. GUH and CUH); see, for instance, papers of Pan et al. [111] and Gonzalez-Carmona et al. [112], which provide experimental data on the efficiency of various target hammerhead cleavage sites. For PLMVd, cleavage occurs immediately after the cytidine at position 8. For this reason, IUPAC code H (i.e. not G) was given as an additional constraint at position 8 for RNAiFold.

Apart from nucleotide constraints at positions 6-7, 22-25,27-29, 44-49, and the constraint H8, all nucleotides at the remaining 38 positions were constrained to be *distinct* from those of PLMVd – this was done to prevent any unintentional use of other nucleotide identities in the computational design of a hammerhead. Summarizing, each sequence returned by `RNAiFold` was required to satisfy IUPAC sequence constraints given by `HBVHBGUHVH VHDVBBHDBD BCUGAVGAGV DVBVHBBBVH BHBCGAAACV DBVB` as shown in Figure 3.1 (right); moreover, the MFE structure of each returned sequence, determined by `RNAiFold'99`, is necessarily identical to the target consensus structure of PLMVd, as shown in Figure 3.2.

`RNAiFold` was run four times, each time additionally constraining GC-content to be within a specified range. Altogether, over one million solutions of RNA inverse folding were returned before memory exhaustion (using the 32 bit version of run-time system `COMET`): 200,072 with GC-content 30-39%, 352,924 with GC-content 40-49%, 349,325 with GC-content 50-59%, 366,323 with GC-content 60-69%, constituting a total of 1,268,644 sequences. Output sequences **s** were selected according to a number of criteria explained below.

Measures used in selecting promising hammerhead candidates from `RNAiFold` were of two basic types that addressed the following questions: *(1)* To what extent do low energy structures of **s** resemble the MFE structure? *(2)* To what extent are the same structural regions of PLMVd AJ005312.1/282-335 as *flexible/rigid* as those of **s**? In other words, the measures used for sequence selection concern either *structural diversity* or regional *structural flexibility/rigidity*; in particular, no sequence homology measures were used in selecting candidate hammerhead sequences for testing, including the program `Infernal` [114].

One measure of type 1 is the Boltzmann probability $P(S_0,\mathbf{s})$, where S_0 denotes the MFE structure of **s** (identical to the Rfam consensus structure of PLMVd AJ005312.1/282-335, since `RNAiFold`

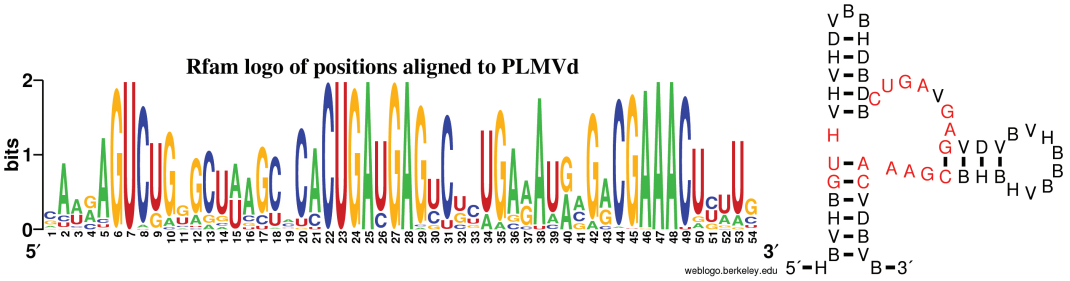


FIGURE 3.1: RF00008 sequence conservation and design sequence constraints. (*Left*) Sequence conservation for the 56 nt consensus sequence for type III hammerhead ribozymes from version 11.0 of the Rfam database [50]; image from `http://rfam.sanger.ac.uk/family/RF00008#tabview=tab3`. (*Left*) Sequence logo of conservation at positions aligned with the 54 nt Peach Latent Mosaic Viroid (PLMVd) AJ005312.1/282-335 from the type III hammerhead ribozyme seed alignment sequences from Rfam family RF00008. In-house program used to determine frequencies of positions aligned to those of PLMVd; sequence logo generated with WebLogo [113] (web server at `http://weblogo.berkeley.edu/`). The 15 positions 6-7, 22-25,27-29, 44-49 of PLMVd had sequence conservation in excess of 96%, while cleavage site C at position 8, adjacent to region 6-8, was conserved in 94.94% of RF00008 seed alignment sequences. `RNAiFold` was subsequently used to solve the inverse folding problem with consensus structure of PLMVd used as target, with sequence constraints at positions 6-8, 22-25,27-29, 44-49, as explained in text. Resulting from this analysis, the sequence constraints for `RNAiFold` were defined to be `HBVHBGUHVH VHDVBBHDBD BCUGAV-GAGV DVBVHBBBVH BHBCGAAACV DBVB`. (*Right*) Sequence constraints for `RNAiFold` with indicated target secondary structure. The 15 positions 6-7, 22-25,27-29, 44-49 having over 96% sequence conservation in the seed alignment of RF00008 were constrained to be those in Peach Latent Mosaic Viroid (PLMVd) AJ005312.1/282-335, and the cleavage site 8 was constrained to be H (not G). All 38 remaining positions were constrained to be distinct from the corresponding nucleotides in PLMVd.

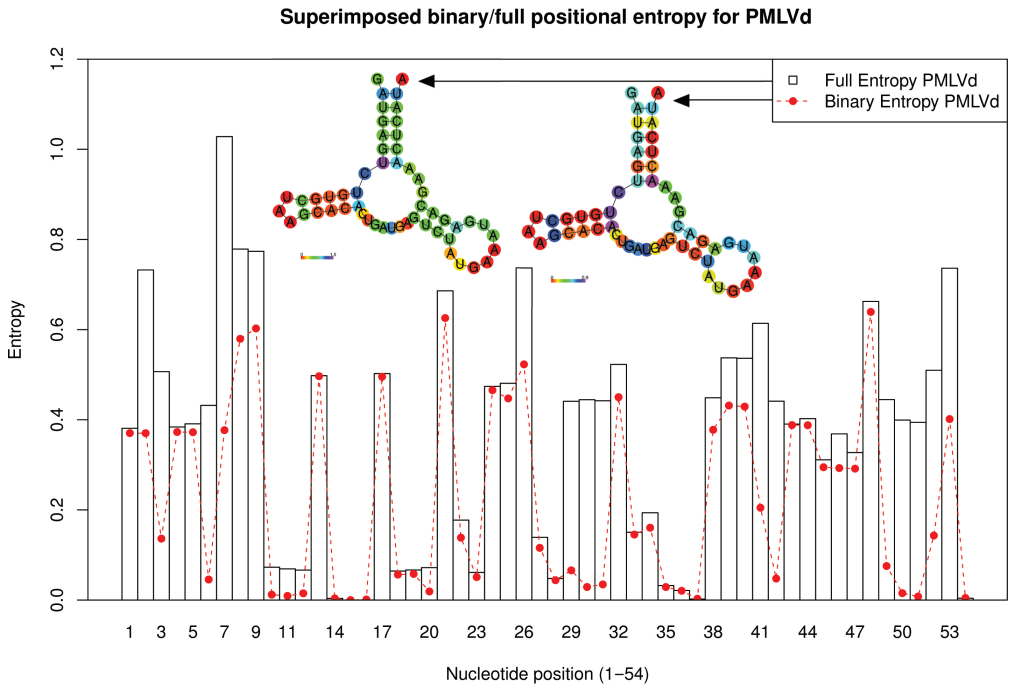


FIGURE 3.2: Binary and full structural *positional entropy* of hammerhead Peach Latent Mosaic Viroid (PLMVd) AJ005312.1/282-335. (*Left*) Full structural *positional entropy* H. (*Right*) *Binary positional entropy* H_b. Note that positions 50, 51 of have medium (full) entropy and high *binary positional entropy*, which indicates that these positions tend always to be base-paired in the low energy ensemble of structures, though with different base pairing partners. Note that the conserved region GUH in 6-8 has moderate to high entropy (G6: 0.62, U7: 1.48, H8: 1.12), GUC in 22-24 has low entropy (G22: 0.26, U23: 0.09, C24: 0.68), GAG in 27-29 has low entropy (*binary positional entropy* is very low) (G27: 0.12, A28: 0.04, G29: 0.07), while 44-49 has medium entropy. Left colored secondary structure figure created using `replot.pl` from `Vienna RNA Package` [28]; right upper colored secondary structure figure created by modifying code `replot.pl`.

solves inverse folding), and $P(S_0,\mathbf{s}) = \frac{\exp(-E(S_0,\mathbf{s})/RT}{Z}$, where $E(S_0,\mathbf{s})$ is the free energy of structure S_0 for sequence $\mathbf{s}$, as computed by Turner 1999 energies, and Z is the partition function. Other measures of type 1 are average structural *positional entropy* [115], *ensemble defect* [52], *expected base pair distance* [55], *Vienna structural diversity* [28], *Morgan-Higgs structural diversity* [116] (see Appendix A). Additionally, the restriction of these measures to the positions 6-8,

22-25, 27-29, 44-49, was computed. Throughout this chapter, we use the term *conserved site* to denote these 16 positions (we use the term *conserved site*, rather than *active site*, which has a different meaning in the biochemical literature). Thus we included measures such as *positional entropy* of conserved site, *ensemble defect* of conserved site, etc. Measures of type 2 concern the maximum discrepancy between values of type 1 for a candidate sequence **s** and wild type PLMVd AJ005312.1/282-335. These are briefly explained in the next section; see [55, 117] or (Apendix A)

Structural *positional entropy*: In selecting the most promising candidate hammerheads from the sequences returned by `RNAiFold`, we additionally considered *discrepancy* (deviation) from structural *positional entropy* of conserved positions in PLMVd. Unlike the notion of nucleotide positional entropy used in sequence logos [113], structural *positional entropy* is defined as follows. If n is the length of a given RNA sequence, then for $1 \leq i,j \leq n$, let $p^*_{i,j}$ denote the probability $p_{i,j}$ of base pair (i,j) if $i < j$, the probability $p_{j,i}$ of base pair (j,i) if $j < i$, and the probability that i is unpaired, $i = j$. With this notation, the (structural) entropy of position i is defined by $H(i) = -\sum_{j=1}^{n} p^*_{i,j} \cdot ln(p^*_{i,j})$. Base 2 logarithms are usually used, whereby entropy is given in bits, ranging from a minimum value of 0, where $p^*_{i,j_0} = 1$ for some j_0, to a maximum value of $\ln n / \ln 2$, in the case that $p^*_{i,j} = 1/n$ for each j.

An alternative to (full) structural *positional entropy* is *binary positional entropy* of position i, defined by $H_b(i) = -(q^*_i \cdot ln(q^*_i) + (1 - q^*_i) \cdot ln(1 - q^*_i))$ where q^*_i is probability that i is unpaired. *Binary positional entropy* values $H_b(i)$ range from a minimum value of 0 bits, where position i is either always base paired (though possibly to distinct partners) or always unpaired in the low energy ensemble of structures, to a maximum value of 1, where position i is paired (unpaired)

with exactly probability 1/2. Figure 3.2 displays full and *binary positional entropy* [115] for PLMVd AJ005312.1/282-335.

At the 16 conserved positions 6-8, 22-25, 27-29, 44-49 of PLMVd, there is a range of structural *positional entropy* values, suggesting that certain nucleotides may be located within a more flexible (high entropy) region of the structure, while other nucleotides may be located within a more rigid (low entropy) region. Figure 3.2 indicates the structural entropy of nucleotides within the consensus structure of PLMVd by appropriate colors, as well as a function of position.

Hypothesizing that low [resp. high] entropy regions of the hammerhead ribozyme could indicate structural *rigidity* [resp. *flexibility*] requirements necessary for hammerhead function, we scrutinized the sequences returned by `RNAiFold` by measures of *deviation (or discrepancy) from structural* positional entropy of PLMVd AJ005312.1/282-335. This led to a number of measures, formally defined in Appendix A, of which the most important are the following: *full/binary positional entropy* discrepancy for complete sequence defined in equations (A.2) and (A.4), *full/binary positional entropy* discrepancy for the conserved site defined in equations (A.22) and (A.23) (recall that 'conserved site' denotes the 16 positions 6-8, 22-25, 27-29, 44-49 constrained by `RNAiFold`). *Entropy discrepancy* for the complete sequence [resp. conserved site] is defined to be the maximum, taken over all 54 positions [resp. over positions 6-8, 22-25, 27-29, 44-49], of the absolute value of the difference between structural entropy of a candidate returned by `RNAiFold` and that of PLMVd.

Sequences selected: Table 3.1 shows the candidate hammerhead sequences finally selected for

cleavage assay, together with the selection criteria used for each sequence. Ten candidate hammerheads were selected: HH1-HH10. HH1-HH5 were chosen from sequences of specific GC-content ranges, to have the smallest *binary positional entropy* discrepancy for the 'conserved site'. HH1 was selected from sequences having GC-content 30-39%; HH2 from sequences having GC-content 40-49%; HH4 from sequences having GC-content 50-59%; HH5 from sequences having GC-content 60-69%. Since PLMVd AJ005312.1/282-335 has GC-content of 40.7%, HH3 was chosen to have second smallest *binary positional entropy* distance for the conserved site, selected from sequences having GC-content 40-49%.

ID	Sequence	Selection criteria
HH1	UUAAUGUAGAGCGAUUCGUUCCUGAAGAGCUAUAAUUUCUUAGCGAAACAUUAU	GC-content $30-39\%$, $P(S_0,\mathbf{s}) \geq 40\%$, smallest (binary) entropy distance for conserved site
HH2	UUAUUGUAGCGCGAUUCGCGCCUGAAGAGAUGCGUUUUAACAUCGAAACAGUAU	GC-content $40-49\%$, $P(S_0,\mathbf{s}) \geq 40\%$, smallest (binary) entropy distance for conserved site
HH3	CUAUUGUAGCGCGAUUCGCGCCUGAAGAGAUCUGUUUUAUGAUCGAAACAGUAU	GC-content $40-49\%$, $P(S_0,\mathbf{s}) \geq 40\%$, 2nd smallest (binary) entropy distance for conserved site
HH4	UGGAUGUAGCGCGAUUCGCGCCUGAAGAGCGGUCAUCCAUCCGCGAAACAUUCU	GC-content $50-59\%$, $P(S_0,\mathbf{s}) \geq 40\%$, smallest (binary) entropy distance for conserved site
HH5	CUCAGGUAGCGCGAUUCGCGCCUGAGGAGGGGUCUGGUAUCCCCGAAACCUGAU	GC-content $60-69\%$, $P(S_0,\mathbf{s}) \geq 40\%$, smallest (binary) entropy distance for conserved site
HH6	UGGCGGUAGCGCGAUUCGCGCCUGAAGAGGGGUAACGCGUCCCCGAAACCGUCU	GC-content $30-39\%$, $P(S_0,\mathbf{s}) \geq 40\%$, *largest* (binary) entropy distance for conserved site
HH7	UCAAUGUCGCGCGAUUCGCGCCUGAAGAGAUGGAAUUUAACAUCGAAACAUUGU	GUC in positions 6-8, smallest *ensemble defect*
HH8	UCAAUGUAGCGCGAUUCGCGCCUGAAGAGAUGGAAUUUAACAUCGAAACAUUGU	smallest *ensemble defect*
HH9	UUAAUGUCGCGCGAUUCGCGCCUGAAGAGAUCUGACUUCUGAUCGAAACAUUAU	$P(S_0,\mathbf{s}) \leq 20\%$, smallest (binary) entropy distance for conserved site
HH10	UUAAGGUCGCGCGAUUCGCGCCUGACGAGCUAUAUUUUAUUAGCGAAACCUUAU	smallest (binary) entropy distance for conserved site

TABLE 3.1: Hammerhead candidates selected and selection criteria used. Note that, subject to presence or absence of additional constraint C8, HH7 and HH8 had also the largest probability of structure, the smallest full structural *positional entropy*, the smallest (Morgan-Higgs and Vienna) structural diversity and smallest *expected base pair distance*.

Additional candidate hammerheads were chosen by different criteria, in order to determine their effect on functionality. HH6 was chosen to have the *largest binary positional entropy* discrepancy for the conserved site, selected from all sequences having C at cleavage position 8, provided that the Boltzmann probability of the MFE structure exceeded 40%. HH7 was chosen to have the *smallest ensemble defect* of all sequences having C at cleavage position 8. HH8 was

chosen to have the smallest *ensemble defect* of all sequences, regardless of nucleotide at position 8 (HH8 has A at cleavage site, instead of C). HH9 was chosen to have the smallest *binary positional entropy* discrepancy for the 'conserved site', selected from all sequences, for which the probability $P(S_0,\mathbf{s})$ of the target PLMVd structure was *at most* 0.2. Finally, HH10 was chosen to have the smallest *binary positional entropy* discrepancy for the conserved site, selected from all sequences, regardless of probability of target structure. Note that HH1-HH6 were selected with the requirement that $P(S_0,\mathbf{s}) \geq 0.4$, while HH7-HH10 were selected without this requirement. This was done in order to determine how important target structure probability might be in hammerhead functionality.

Computational Pipeline Summary: The following computational pipeline summarizes the generation and selection of candidate hammerhead sequences.

```
1. find Rfam sequence, whose MFE structure resembles family consensus structure
2. determine highly conserved positions in reliable multiple alignment
3. run RNAiFold to solve the constrained inverse folding problem
4. filter using Boltzmann probability, GC-content, entropy, ensemble defect, etc.
5. perform biochemical validation
```

A Python program can be downloaded from the `RNAiFold` web site, that automates steps 1,2. Of course, one can bypass step 1 without using Rfam, and instead use any reliable multiple sequence/structure alignment.

Design of modular hammerhead within another structure: It has many times been observed that aptamers, hammerheads and other functional RNAs constitute *modules*, capable of function even when engineered to form part of a larger RNA molecule. For instance, Wieland et al. [118] created artificial *aptazymes* by replacing a hammerhead helix by a theophylline

aptamer, and Saragliadis et al. [119] created artificial *thermozymes*, created by fusing a theophylline aptamer to a *Salmonella* RNA thermometer [119].

With the intent of designing a guanine-activated riboswitch with a modular hammerhead, we followed the following steps in rationally designing a synthetic 166 nt RNA, with putative type III hammerhead module. Target secondary structure S was taken to be the structure of the gene OFF xanthine phosphoribosyltransferase (XPT) riboswitch, depicted in Figure 1A of [120], whereby the terminator loop (expression platform) was replaced by the Rfam consensus structure for a type III hammerhead. Sequence constraints were chosen to be the highly conserved nucleotides of the Rfam consensus structures for the purine riboswitch (RF00167 seqcons view of consensus structure) and for type III hammerhead (RF00008 seqcons view of consensus structure). Figure 3.3 displays the target structure S for computational design of a modular hammerhead within the terminal stem-loop of a structure similar to the XPT riboswitch.

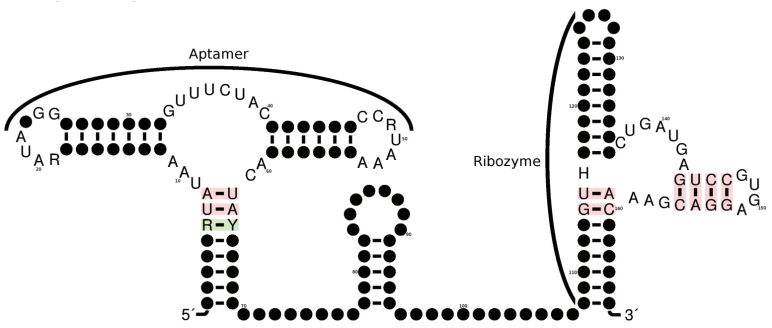


FIGURE 3.3: Target secondary structure S for modular placement of artificial hammerhead within larger RNA molecule. The structure and highly conserved nucleotides (sequence constraints) of the XPT-riboswitch appear on the left, while the structure and highly conserved nucleotides of the type III hammerhead ribozyme appear on the right.

We gave `RNAiFold` an additional compatibility constraint, whereby returned sequences were required to be compatible to a second structure S', in which the hammerhead cleavage site

(NUH) is fully sequestered within a base-paired region. Positions 60-118 of S' are given as follows:

```
5'-ACUAYNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNGUHNN-3'
5'-.....((((((((((((..(((((........)))))............))))))))))))-3'
```

while all positions in S' outside of 60-118 (i.e. from 1-59 and 119-166) are unpaired.

We filtered sequences output by `RNAiFold`, by applying `RNAbor` [121], and its faster sequel, `FFTbor` [122]. Given reference structure S, `RNAbor` and `FFTbor` return the *density of states* with respect to S, which depicts the Boltzmann probability $p(k) = \frac{Z_k}{Z}$ for secondary structures to have base pair distance k from S. Additionally, `RNAbor` computes, for each k, the MFE$_k$-structure; i.e. that structure having minimum free energy over all structures whose base pair distance from the reference structure S is exactly k.

From a partial output of 3,000 sequences from `RNAiFold`, only one sequence **s** satisfied the following two properties, when applying `RNAbor` with input **s** and reference MFE structure S: (1) The density of states figure has a pronounced peak at $k = 0$, corresponding to the location of the MFE structure S; (2) There was another pronounced peak for value $k \gg 0$, corresponding to a structure T containing the base pairs in S', which thus should sequester the ribozyme cleavage site NUH, located at position 114-116 (Figure 3.8).

The final, selected sequence 166 nt **s** is given as follows: GCCGC GUAUA AGGGC UGCGA UAAGG GCAGU CCGUU UCUAC GGGCG GCCGU AAACC GCCCA CUACG CGGCG UGGUU AAGCC GGAAA GGAGA CCGGC AGGAG GGUAA UGGGC CGCGU CGCGG CGCGG GAGCG CGCCG CCUGA UGAGU CCGUG AGGAC GAAAC GCGGCC.

3.3.2 Experimental Validation

Complementary DNA oligonucleotides, corresponding to the DNA sequence of the designed RNAs preceded by a T7 RNA polymerase promoter, were purchased from MWG Operon. The ten hammerhead candidate sequences HH1-HH10, extended 2 nt on the left by GG and 2 nt on the right by CC for transcriptional efficiency, and the 166 nt sequence, harboring a candidate hammerhead in the rightmost stem-loop of Figure 3.8 were constructed using primer extension and PCR amplified (5 U taq polymerase (New England Biolabs), 2.5 mM each NTP, 1x NEB Thermopol buffer). For each of the 10 designed hammerhead sequences, the H8G mutant was constructed in a similar manner, using alternative oligonucleotides containing the mutation. Similarly, C116G (analogous to H8G) and G142U mutations were constructed for the 166 nt designed ribozyme. The resulting PCR products were TOPO-cloned (Invitrogen), and the designed and mutant sequences were verified by sequencing plasmids containing full-length PCR products. These plasmids were subsequently used as templates for PCR reactions to generate template for *in vitro* transcription.

To generate the RNA, *in vitro* transcription was performed using T7 RNA polymerase (400 U T7 polymerase, 80 mM HEPES-KOH pH 7.5, 24 mM $MgCl_2$, 2 mM spermidine, 40 mM DTT, 2 mM each NTP) with the addition of 10 μCi of α-32P-GTP for transcriptions to generate body-labeled RNA when necessary. To prevent premature cleavage during transcription, 100 uM of oligonucleotides complementary to nucleotides 17-35 (numbering starts after the leading GG) were added to each reaction. Full-length RNAs were purified using denaturing PAGE (20% acrylamide).

To assess self-cleavage of designed hammerhead sequences, RNA was incubated for 1 hour in

cleavage buffer (5 mM $MgCl_2$, 50 mM tris pH 7.5) at 25°C. Subsequently, 1 volume of 2x gel-loading buffer (16 M urea (supersaturated), 10 mM EDTA, 20% sucrose, 0.1%SDS, 100 mM tris pH 8.0, 100 mM borate, 0.05% bromophenol blue) was added to quench the reaction with final urea and EDTA concentrations of 8 M and 5 mM respectively. The reaction was placed on ice until gel loading.

Samples lacking Mg^{++} were incubated in 50 mM tris pH 7.5 for 1 hour at 25°C. For the 166 nt RNA, cleavage experiments were conducted under similar conditions but reactions were incubated for a few seconds (0 h), 30 min, 5 h and 24 h, and samples lacking Mg^{++} were incubated in 50 mM tris pH 7.5 for 24 h at 25°C. Cleavage products were separated by denaturing PAGE (10% acrylamide), and the gels dried prior to exposure to phosphoimager plates (GE Healthcare) for 18 h. The gels were imaged using a STORM 820 phospoimager (GE Healthcare).

Kinetics: To determine the cleavage rates for designed hammerhead sequences, body-labeled RNA was incubated in cleavage assays as described above for varying amounts of time. Cleavage products were separated and gels imaged as described above. The cleavage products were quantified using ImageQuant software (GE Healthcare). To calculate the fraction cleaved at time t, $F(t)$, the sum of the quantified counts for 5′ and 3′ cleavage product bands was divided by the total quantified counts for the entire reaction (uncleaved, 5′ and 3′ cleavage products).

The observed cleavage rate K_{obs} was computed by using the `Matlab` function `nlinfit` with constant error model to fit cleavage time series data using the equation

$$F_{\max} - F(t) = (F_{\max} - F(0)) \cdot \exp(K_{obs} \cdot t) \tag{3.1}$$

where $F(t)$ denotes the amount of cleavage product measured at time t, and $F_{\max}$ the maximal fraction cleaved. The 95% confidence interval of this fit was calculated from the resulting residuals and variance-covariance matrix using the `Matlab` function `nlpredci`.

3.4 Results

Given the target Rfam consensus structure S of Peach Latent Mosaic Viroid (PLMVd) AJ005312.1/282-335, which is identical with the MFE secondary structure using `RNAiFold'99`, 16 highly conserved positions nucleotides were taken as constraints in the generation of over one million sequences solving the inverse folding problem, as determined by `RNAiFold'99`. Using distance measures of *dissimilarity* of low energy structures to the MFE structure (*positional entropy*, *ensemble defect*, structural diversity, etc.) together with measures of molecular structural flexibility/rigidity, ten putative hammerhead sequences were selected for *in vitro* validation using a cleavage assay. The selected sequences and selection criteria are given in Table 3.1. All ten hammerhead candidates, listed in this table, were shown to be functional, with cleavage rates listed in Table 3.2.

Cleavage assay gel images for the designed hammerheads HH1-HH10 are displayed in Figure 3.4, where each sequence shows Mg^{++}-dependent cleavage. In addition, the H8G mutant of each designed hammerhead shows no activity. These data strongly suggest that the designed sequences HH1-HH10 behave in a manner consistent with the expected mechanism for hammerhead ribozymes. Time series for cleavage fraction and kinetics curves for a typical designed hammerhead ribozyme (HH1) and the designed ribozymes are shown in Figure 3.5. Kinetics for the designed hammerheads should be compared with wild type hammerhead kinetics, where

ID	K_{obs}	F_{max}	MSE	Pos Ent	Ens Def	EBPD Dis Act
HH1	0.037	0.79	0.0029	0.270882	4.167687	0.0501207
HH2	0.0057	0.74	0.003	0.287235	4.552053	0.0386253
HH3	0.0027	0.65	0.0039	0.259577	4.121914	0.0410984
HH4	0.0127	0.55	0.0048	0.403846	6.755976	0.0354213
HH5	0.0085	0.52	0.0066	0.382235	6.240083	0.033132
HH6	0.102	0.73	0.0047	0.414872	8.138131	0.059864
HH7	0.25	0.74	0.0107	0.119159	2.383671	0.0406728
HH8	0.02	0.68	0.0124	0.078518	1.45179	0.0662421
HH9	0.025	0.76	0.0015	0.247886	4.525597	0.0328018
HH10	0.14	0.77	0.01	0.286425	4.975979	0.0269354

TABLE 3.2: Kinetics of cleavage for 10 computationally designed hammerheads, and correlation with several measures. Cleavage rate K_{obs} (min^{-1}), maximum percent cleavage F_{max}, mean squared error MSE, (full) structural *positional entropy* Pos Ent, *ensemble defect* Ens Def, and *expected base pair distance* discrepancy for the 'conserved (or active) site' EBPD Dis Act (see Appendix A). The Pearson correlation between cleavage rate and Pos Ent, Ens Def, EBPD Disc Active is respectively -0.461, -0.370, -0.438; i.e. cleavage rate is faster when these secondary structure deviation values are smaller. Other measures, such as structural diversity, had smaller correlation, while measures such as GC-content and MFE had almost no correlation with cleavage rate.

under standard conditions of 10 mM $MgCl_2$, pH 7.5, and 25° C, cleavage rates between 0.5 and 2 per minute have been observed for at least 20 different hammerheads [123]. It follows that kinetics of the computationally designed hammerheads described in this paper are slower than wild type hammerheads approximately by a factor of 10.

Pearson correlation coefficient was determined between cleavage rate K_{obs}, obtained by fitting equation (3.1) with data from three to five technical replicates, and 21 measures, including average *positional entropy*, GC-content, minimum free energy, etc. See Supplementary Information of [124] for all correlation values. The most pronounced correlations were observed between K_{obs} and (full) average structural *positional entropy*, *ensemble defect*, and *expected base pair distance* discrepancy for 'conserved site' with values respectively of -0.461, -0.370, -0.438; i.e. cleavage is faster when these measures are smaller. See equations (A.2),(A.9) and (A.24) in

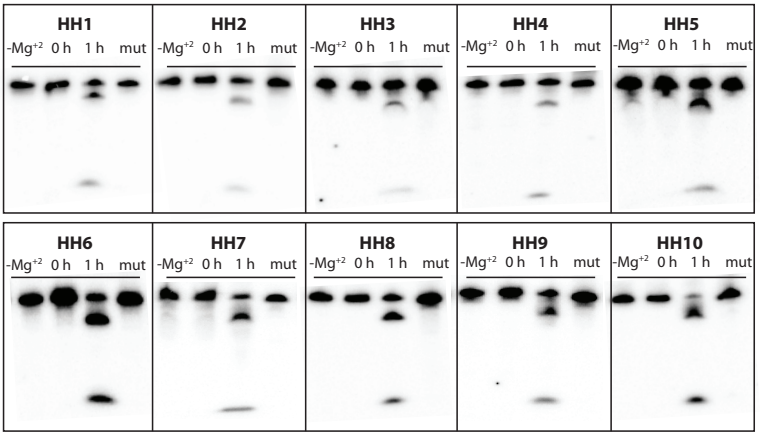


FIGURE 3.4: Summary of designed hammerhead cleavage. Each designed hammerhead RNA was incubated under mild conditions for 1 hour as described in the Methods to assess cleavage. As negative controls, a no magnesium, and a 0 hour reaction were also conducted for each RNA. Additionally, the 8G mutation, predicted to be incompatible with the hammerhead structure (see Section 3.3.1), was constructed for each designed sequence and examined under equivalent conditions to confirm that self-cleavage occurs using the expected hammerhead mechanism.

Appendix A for formal definitions of these notions.

It is known from literature [111, 112] that hammerhead cleavage sites are of the form NUH (e.g. GUH and CUH, but not GUG). Indeed, Carbonell et al. [125] suggest that G8 would pair with C22 (in our numbering) and impede its role in the catalytic pocket. Figure 3.4 shows that the H8G mutant of each designed sequence HH1-HH10 does not cleave under mild denaturing conditions that suffice for cleavage of HH1-HH10. In addition, `RNAiFold` determined that (provably) there is no RNA sequence, whose MFE structure is the Rfam consensus structure of Peach Latent Mosaic Viroid (PLMVd) AJ005312.1/282-335, having a guanine at cleavage site 8, as well as the 15 highly conserved nucleotides of PLMVd at positions 6-7, 22-25,27-29, 44-49 (left panel of Figure 3.6). This result holds for both the Turner 99 and Turner 2004 energy models.

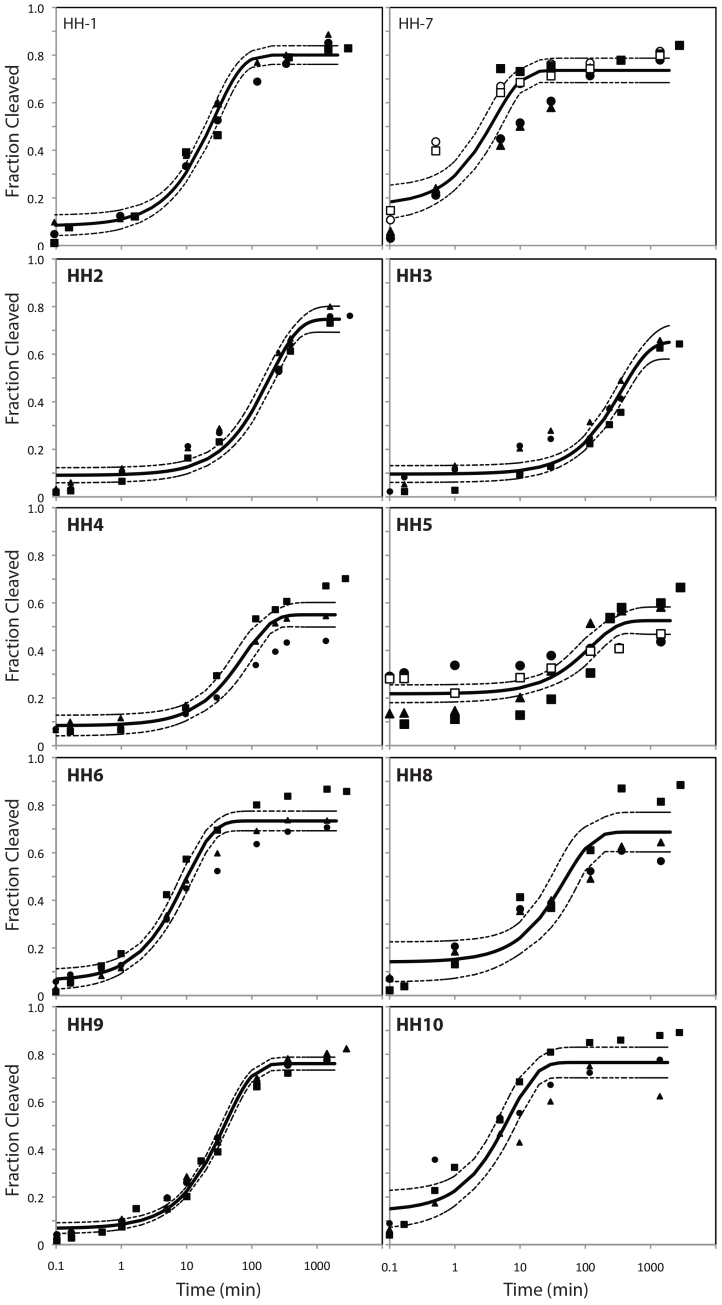


FIGURE 3.5: Best-fit kinetics curves for designed hammerhead sequences. *(Top Left)* HH1: typical cleavage time series curve with good error parameters (standard deviation <10% of mean, with mean squared error (MSE) = 0.0029). Solid line represents fitted line, and dotted lines indicate 95% confidence interval. Different data sets represented by filled and unfilled squares, triangles, etc. *(Top Right)* HH7: fastest hammerhead cleavage rate, though determined with considerable error (MSE=0.01). In data from the first experiments for HH7, indicated by filled squares, cleavage had been measured at times when maximum cleavage had nearly occurred (these points appear in the flat part of the fitted curve). Subsequent datasets have focused on shorter time periods. This curve was fitted using five data sets. *(Bottom)* Time series curves for cleavage data for the remaining 8 designed hammerheads HH2-HH6 and HH8-HH10.

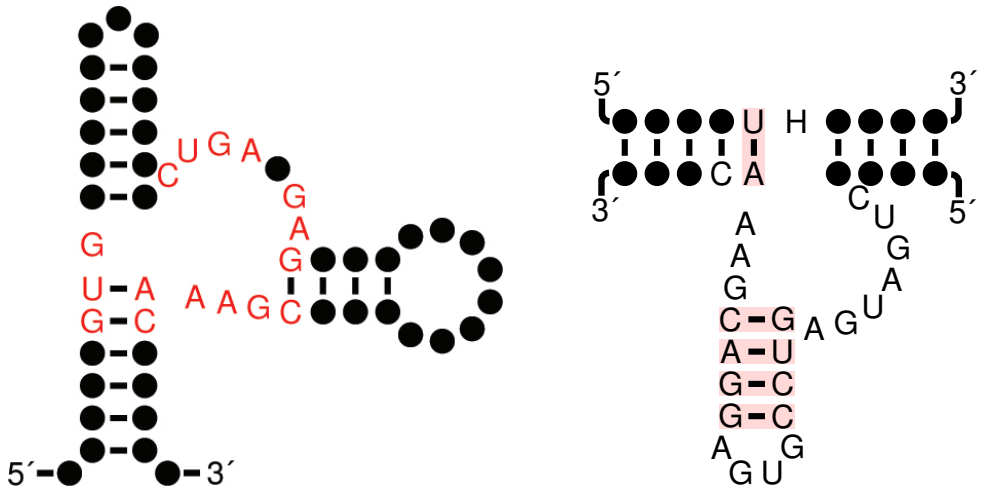


FIGURE 3.6: Target structures used in type III hammerhead ribozyme computational design. *(Left)* Target structure S used in computational experiment with `RNAiFold`, which determined that *no sequence exists*, having guanine at the cleavage site 8 along with those 15 nucleotides of Peach Latent Mosaic Viroid (PLMVd) AJ005312.1/282-335 having sequence conservation exceeding 96%, and which the Rfam consensus structure of PLMVd (i.e. whose `RNAiFold'99` MFE structure is the consensus secondary structure of PLMVd). *(Right)* hammerhead ribozyme (lower molecule) hybridized with *trans*-cleavage target RNA (upper molecule). Cleavage site NUH occurs at position 4-6 of the upper molecule, where 'H' denotes 'not G'. `RNAiFold` shows that no two sequences $\mathbf{s}_1$,$\mathbf{s}_2$ exist, where $\mathbf{s}_1$ contains 'GUG' at positions 4-6, both $\mathbf{s}_1$,$\mathbf{s}_2$ contain the other indicated nucleotides, for which the indicated structure is the MFE hybridization of $\mathbf{s}_1$,$\mathbf{s}_2$. The nonexistence, as determined by `RNAiFold`, of any sequence folding into target structure S, which has GUG at the cleavage site and satisfies certain additional minimal constraints, strongly suggests that GUG is not a hammerhead cleavage site is due to the inability of the molecule to fold into a structure necessary for nucleophilic attack. Image of right panel adapted from Figure 3A from [126], and both images produced by R2R [127].

Since `RNAiFold` also solves the inverse hybridization problem, we considered the NUH cleavage target of *trans*-cleaving hammerhead ribozymes, known from comparative sequence analysis [126]. Application of `RNAiFold` showed that there do not exist any two sequences, where the first contains GUG at the cleavage site location, for which the minimum free energy hybridization structure is the target structure appearing in the right panel of Figure 3.6. Taken together, these results provide a compelling computational explanation for the reason that GUG is not a hammerhead cleavage site.

To demonstrate the functionality of a computationally designed hammerhead, occurring within a larger rationally designed RNA, we synthesized the 166 nt sequence $\mathbf{s}$, designated as 'synthetic wild type', as well as two mutant sequences $\mathbf{s}_1$, $\mathbf{s}_2$, each containing a mutation that should inactivate hammerhead activity. Sequence $\mathbf{s}_1$ contains a C116G mutation at the GUC site of cleavage, while $\mathbf{s}_2$ contains a G142U mutation in a distal section of the ribozyme, known to be required for cleavage (the CUGAUGA sequence). Cleavage assays under mild conditions (5 mM $MgCl_2$, 50 mM tris pH 7.5, 25°C) showed that approximately 40% of our synthetic wild type sequence rapidly cleaves at the expected site, in the absence and presence of guanine.

The cleavage is Mg^{++}-dependent (Figure 3.7A), and the hammerhead appears to cleave rapidly within seconds. Neither of the mutant sequences displays any cleavage under the same conditions, even with significantly longer incubation times (Figure 3.7B,C). Kinetics for the 166 nt synthetic ribozyme are comparable with those of wild type hammerheads, with an observed cleavage rate K_{obs} of 1.3/min and Fmax of 0.47 (Figure 3.7D). Addition of 1 mM guanine has no significant affect on either the K_{obs} or the Fmax; i.e. the designed riboswitch was constitutively on.

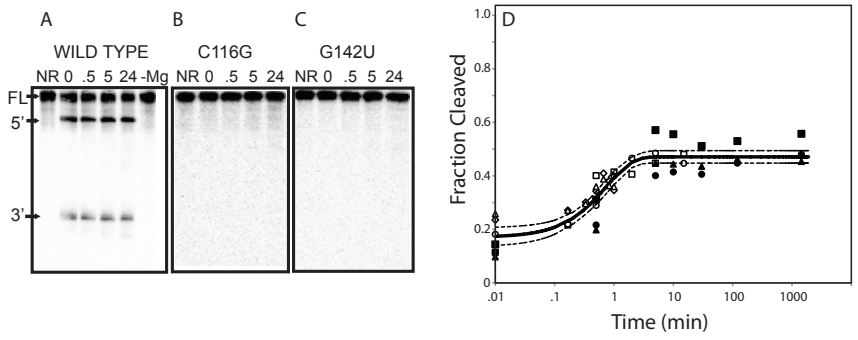


FIGURE 3.7: Cleavage assay reactions and time series curve for the 166 nt designed hammerhead. *(Left)* Cleavage assay reactions (A,B,C) of designed hammerhead (wild type), mutant C116G, and mutant G142U. For the wild type (A), mutant C116G (B), and mutant G142U (C) gel images, lane 1 is the undigested RNA (full-length, FT), lanes 2-5 are reactions in cleavage buffer (50 mM Tris pH 7.5, 5 mM $MgCl_2$) at the 0 s, 30 min, 5 h, and 24 h time points respectively (5′ and 3′ cleavage products indicated). For the wild type (A), lane 6 is a reaction lacking Mg (50 mM tris pH 7.5) incubated for 24 h. It is evident that cleavage only occurs for the wild type sequence, and when Mg is present. *(Right)* Cleavage time series curve (D) for the 166 nt designed hammerhead, with observed cleavage rate of 1.3/min with an Fmax of 0.47 and MSE of 0.0026. This construct displays kinetics comparable with that of wild type hammerheads, although the cleavage amount Fmax is much lower than that of wild type hammerheads.

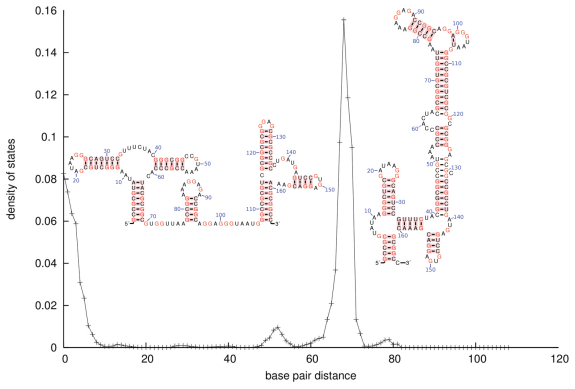


FIGURE 3.8: Density of states graph, computed by `RNAbor`, together with the MFE structure S and the MFE_{68} structure, found respectively at the left and right peaks. Metastable structure MFE_{68} contains all base pairs from compatibility constraint S', thus in theory sequestering the GUC hammerhead cleavage site at position 114-116. Density of states and secondary structures produced by `RNAbor`; superimposed secondary structure images produced by R2R [127].

3.5 Discussion

In this chapter, we have demonstrated the success of a purely computational approach for the rational design of artificial type III hammerhead ribozymes. Figure 3.4, clearly shows the Mg^{++}-dependent cleavage of each designed sequence HH1-HH10, as well as the non-cleavage of the 8G mutant of each sequence, strongly suggesting that cleavage is due to the usual hammerhead mechanism. Cleavage time series data for three to five technical replicates for each of the ten computationally designed hammerheads, displayed in Figure 3.5, lead to observed cleavage rates varying 100-fold from 0.0027 min^{-1} for HH3, to 0.25 min^{-1} for HH7. The relatively fast cleavage rate of HH7, selected from over one million sequences returned by `RNAiFold` solely on the criteria of minimizing *ensemble defect*, *with* the additional requirement of having GUC at the cleavage site, is slower only by a factor of 10 from wild type hammerhead cleavage rates (recall that wild type cleavage rates vary between 0.5 and 2 per minute [123]). In contrast, HH8 had an observed cleavage rate of 0.02 min^{-1}, although it was selected solely on the criteria of minimizing *ensemble defect– without* the additional requirement of having GUC at the cleavage site. This experimental result suggests that cleavage kinetics may be the underlying reason that cytidine is present at cleavage position 8 in 95% of the 84 sequences in the Rfam seed alignment of family RF00008.

Among more than 20 computational features, the features found to be most highly correlated with cleavage rate K_{obs} for HH1-HH10 were (full) average structural *positional entropy*, *ensemble defect*, and *expected base pair distance* discrepancy for 'conserved site' with values respectively of -0.461, -0.370, -0.438. However, this result is based on a tiny set of data and can only

be taken as a suggestive first step towards a more systematic determination of which measures of structural diversity/flexibility/rigidity might best predict ribozyme activity.

In the design phase, we selected HH1-HH5 to have a *positional entropy* profile similar to that of wild type PLMVd, i.e. to have small average structural *positional entropy* of conserved site, based on the intuition that certain positions in the wild type hammerhead may have high entropy to support cleavage. However, it is presently unclear whether discrepancy measures (absolute difference between wild type and synthetic) restricted to the conserved site are useful at all. Indeed, among all sequences returned by `RNAiFold`, HH6 had an observed cleavage rate of 0.102/min, a bit less than half that of HH7, yet HH6 was selected to have the *largest* entropy discrepancy from the conserved site among all sequences, such that the probability of the MFE structure exceeded 40%. Without additional experiments on a large collection of computationally designed hammerheads, and perhaps without extensive molecular dynamics modeling, it remains unclear to what extent hammerhead efficiency, as assayed by cleavage kinetics, is dependent on matching the positional stability and flexibility of the wild type PLMVd hammerhead.

It is interesting to note that HH1-HH6 are not recognized as hammerheads by the Rfam web server [50], which relies on the program `Infernal` [114], a sophisticated machine learning algorithm (stochastic context free grammar) that depends on recurring sequence and structural motifs. Rfam predicts only HH7-HH10 to be type III hammerheads, with the following confidence scores: HH7 41.3 bits (E-value 5.9e-09), HH8 38.1 bits (E-value 4.6e-08), HH9 37.5 bits (E-value 6.8e-08), HH10 38.9 bits (E-value 2.9e-08).

Currently, `NUPACK-DESIGN` [21] appears to be one of the most efficient tools to design RNAs by employing a heuristic computational search to minimize *ensemble defect*. Given the constraints

for synthetic hammerhead design described in this paper, the NUPACK-DESIGN server returned 10 sequences, nine of whose MFE structures were identical to that of PLMVd AJ005312.1/282-335. (The NUPACK-DESIGN philosophy is that minimizing *ensemble defect* is more important than guaranteeing that sequences be an exact solution of the inverse folding problem. The NUPACK-DESIGN web server has an upper limit of 10 sequences that can be returned. In contrast, after downloading and compiling the NUPACK-DESIGN source code, each run of NUPACK-DESIGN returns a single sequence; since the procedure is stochastic, repeated runs will usually return different sequences.) The first sequence returned by the NUPACK-DESIGN web server was CGC-CGGUAGC CUGACCCAGG CCUGAAGAGC UCUACCCCCC GAGCGAAACC GGCU, which has normalized *ensemble defect* of 2.5%, the same value as that of HH8 (1.45179/54 = 0.025030862). The cleavage rate of HH8, whose cleavage site is GUA (as in the NUPACK-DESIGN sequence) is 0.02/min, with five faster cleaving synthetic hammerheads. Despite the speed of NUPACK-DESIGN in designing RNAs with low *ensemble defect*, one advantage of RNAiFold is that prioritization of candidate sequences is performed in a postprocessing phase, thus allowing one to select solutions of inverse folding that are optimal with respect to various measures (not only *ensemble defect*), as we have done in this chapter.

We have additionally tested the programs RNAdesign [47] and IncaRNAtion [49], with the Rfam consensus structure of PLMVd hammerhead as target structure. Only 5.84% [resp. 2.57%] of the sequences returned by RNAdesign [resp. IncaRNAtion] actually folded into the target structure, thus requiring substantial additional computation time to select those sequences that fold into the target (in contrast, RNAiFold returns only sequences that correctly fold into the target structure). See Appendix C and http://bioinformatics.bc.edu/clotelab/SyntheticHammerheads/ for comparative results concerning entropy, *ensemble defect*, etc.

In addition to computationally designing the functional hammerheads HH1-HH10, we have designed the 166 nt sequence **s**, in which a synthetic hammerhead is embedded within the terminal stem-loop of the structure depicted in Figure 3.4. The sequence **s** is self-cleaving at the expected GUC cleavage site 114-116. Moreover, as shown in Figure 3.7, cleavage kinetics for this 166 nt artificial ribozyme (K_{obs}=1.3/min) are as fast as those of wild type hammerheads, although the cleavage amount (Fmax=0.47) is quite poor compared with our other designed ribozymes HH1-HH10. By utilizing two mutants, one at the cleavage site position 116, and one further downstream at position 142 in the CUGUAGA segment necessary for catalysis of cleavage, we show effectively that cleavage in the synthetic wild type, designed construct is due to the usual hammerhead mechanism. Additionally, we have demonstrated Mg^{++}-dependence, necessary for the cleavage mechanism, through the complete absence of 5′- and 3′-cleavage products when incubated for an extended period of time of 24 hours in buffer lacking Mg^{++}.

The software `RNAiFold` solves the inverse folding problem, not only for a target secondary structure, but as well when the target S is the hybridization of two secondary structures; i.e. when S contains both intra- and inter-molecular base pairs. Since `RNAiFold` uses constraint programming, it can perform a complete search of the space of compatible sequences, and thus return *all* sequences, whose MFE structure (resp. MFE hybridization) is a given target structure (resp. hybridization), or *can certify that no such solution exists.* The fact that `RNAiFold` determined that no solution of inverse folding exists for the GUH to GUG (resp. NUH to GUG) mutant of the target structure depicted in Figure 3.3 (resp. the right panel of Figure 3.6) provides very compelling computational evidence that there are structural reasons that prevent the occurrence of a GUG motif at the hammerhead cleavage site.

Chapter 4

RNAiFold2T

4.1 Introduction

RNA *thermometers* (RNATs) and RNA *switches* (riboswitches) are *cis*-regulatory elements that change secondary structure respectively upon temperature shift or binding to a ligand. Both elements constitute an interesting potential resource in synthetic biology, where engineered RNATs and riboswitches could prove to be useful tools in biosensors and conditional gene regulation.

RNATs are often involved in the regulation of heat shock, cold shock and virulence genes. Solving the 2-temperature inverse folding problem is critical for RNAT engineering. Here we introduce `RNAiFold2T`, the first *Constraint Programming* (*CP*) and *Large Neighborhood Search* (*LNS*) algorithms to solve this problem. In addition, `RNAiFold2T` incorporates new *local structural constraints* for the design of riboswitches. Benchmarking tests of `RNAiFold2T` against existent programs (adaptive walk and genetic algorithm) for 2-temperature inverse folding show

that our software generates two orders of magnitude more solutions, thus allowing ample exploration of the space of solutions. Subsequently, solutions can be prioritized by computing various measures, including probability of target structure in the ensemble, melting temperature, etc.

Using this strategy, we rationally designed two thermosensor internal ribosome entry site (*thermo*-IRES) elements, whose normalized cap-independent translation efficiency is approximately 50% greater at 42°C than 30°C, when tested in reticulocyte lysates. Translation efficiency is lower than that of the wild-type IRES element, which on the other hand is fully resistant to temperature shift-up. This appears to be the first purely computational design of functional RNA *thermometers*, and certainly the first purely computational design of functional *thermo*-IRES elements.

We applied a similar strategy for the computational design of a riboswitch-ribozyme capable of trans-cleavage of a second RNA molecule only when activated by the presence of theophylline. We refer to this riboswitch-ribozyme as *RNA molecular scissors*. The use of *local structural constraints* included in `RNAiFold2T` allowed us to generate thousands of solutions, which we filtered and prioritized by computing several measures such as the thermodynamic energy barrier between meta-stable conformations and the estimated concentration of structures in the presence or absence of theophylline. At the time of publication of this dissertation several RNA *molecular scissor* candidates are under experimental validation.

4.1.1 Organization

This chapter is organized in the following fashion. We start by providing some background on RNA *thermometers* and RNA *switches*, reviewing the methods available for the computational and experimental design of both types of RNA conformational switches. Then, we describe the modifications introduced in `RNAiFold` to create `RNAiFold2T`. These additions are specifically intended for the design of RNA *thermometers* and RNA *switches*. We then present benchmarking results against other software, showing that `RNAiFold2T` solves approximately the same number of 2-temperature inverse folding problems, yet provides a decided advantage by returning more solutions for each given 2-temperature inverse folding problem; moreover, `RNAiFold2T` incorporates many more design constraints than other software. Finally, we use `RNAiFold2T` to analyze naturally occurring RNA *thermometers*, and to rationally design *thermo*-IRES elements as well as theophylline-dependent RNA *molecular scissors*.

4.2 Background

RNA *thermometers* (RNATs), also known as *thermosensors* or RNA *thermoswitches*, are *cis*-regulatory elements that change secondary structure upon temperature shift. Examples include (1) repression of heat shock gene expression (ROSE) elements [128], that control the expression of small heat shock genes, such as hspA in *Bradyrhizobium japonicum* and ibpA in *Escherichia coli*, (2) FourU elements [129], such as the virulence factor LcrF in *Yersinia pestis*, (3) Hsp17 thermosensor [130, 131], which controls membrane integrity of the cyanobacterium *Synechocystis sp. PCC6803* under stress conditions, critical for photosynthetic activity. Additional

examples are described in [132]. ROSE elements and FourU elements operate as temperature-sensitive, reversible zippers, while the *Listeria monocytogenes* prfA thermosensor [133], phage λ cIII *thermoswitch* [134] and *E. coli* CspA cold shock thermometer [135] operate in a switch-like fashion. Here, the helix of a zipper melts gradually with increasing temperature, returning to the original structure when temperature is reduced, while a switch consists of two mutually exclusive structures determined by temperature.

Several bioinformatics search methods exist to identify and predict candidate RNA *thermometers*. In [136] the database RNA-SURIBA (Structures of Untranslated Regions In BActeria) was created; using regular expressions, particular structural motifs were detected in the minimum free energy (MFE) structure, as determined by `mfold` [29]. In contrast, the `RNAtips` web server [137] and the `RNAthermsw` [138] web server both rely on base pairing probabilities computed at different temperatures using `RNAfold` from the `Vienna RNA Package` [139].

For some time now, RNA thermosensors have been recognized as an attractive target for rational design [140, 141]. Indeed, within the broader context of synthetic biology, rationally designed RNA *thermometers* could be used as a *thermogenetic* tool to control expression by temperature regulation (i.e. *on-demand protein translation*), or even as a multifunctional nanoscale devices to measure temperature in the context of hyperthermic treatment of cancer cells, imaging, or drug delivery [142].

In [143], synthetic (zipper) thermosensors were *manually* designed to sequester the Shine-Dalgarno sequence AAGGAG within a single stem-loop structure containing 4-9 base pairs, several of which contained 1-2 bulges of size 1.

In [119], a *thermozyme* was created by fusing a *Salmonella* RNA *thermometer* (RNAT) to a hammerhead ribozyme, followed by *in vivo* screening – thus showing that naturally occurring hammerheads and RNATs appear to be modules that can be combined. In [144] small, heat-repressible RNA thermosensors (zippers) were manually designed in *E. coli*, which at low temperature sequester a cleavage site for RNaseE, and at high temperatures unfold to allow mRNA degradation.

`SwitchDesign` (`SD`) [145], the first approach in the literature to solve the 2-temperature inverse problem, achieves this by using an adaptive walk algorithm to optimize the following cost function for and input RNA sequence $\mathbf{a} = a_1, \ldots, a_n$:

$$(E_{T_1}(\mathbf{a},S_1) - G_{T_1}(\mathbf{a})) + (E_{T_2}(\mathbf{a},S_2) - G_{T_2}(\mathbf{a})) - \xi((E_{T_1}(\mathbf{a},S_1) - E_{T_1}(\mathbf{a},S_2)) + (E_{T_2}(\mathbf{a},S_2) - E_{T_2}(\mathbf{a},S_2)) \quad (4.1)$$

where $G_T(\mathbf{a})$ is the ensemble free energy sequence $\mathbf{a}$ at temperature T, $E_T(\mathbf{a},S)$ is the free energy of RNA sequence $\mathbf{a}$ with structure S at temperature T, and $0 < \xi < 1$ is a constant (see Appendix D for a complete definition).

RNA *thermoswitches* have been computationally designed and synthesized, that are as efficient as natural RNA *thermoswitches*, by applying the program `SwitchDesign`. In [141], synthetic (switch) thermosensors were *computationally* designed to switch between a single stem-loop structure that sequesters the Shine-Dalgarno sequence GGAGG, and two shorter stem-loop structures where the Shine-Dalgarno sequence is found in the apical loop of the second stem-loop. In particular, the 2-temperature `SwitchDesign` [145] was used to obtain 300 candidate sequences; only two candidate sequences survived after the application of several computational filters including the computation of melting curves with `RNAheat` [139]. Since neither

of these sequence displayed any temperature-dependent control of a reporter gene (bgaB) fusion, the top candidate sequence was used as a template in two rounds of error-prone PCR mutagenesis followed by selection, resulting in a successful thermosensor – see Figure 5 of [141].

A second approach capable of solving the 2-temperature inverse problem is the software `FRNAkenstein` (`FRNA`) [44], which implements a multi-objective genetic algorithm described in Chapter 2. However, `FRNAkenstein` has not been used in experimentally validated synthetic designs.

Despite these impressive results, [132] state that: "RNATs have little, if any sequence conservation and are difficult to predict from genome sequences. ...Therefore, the bioinformatic prediction and rational design of functional RNATs has remained a major challenge".

RNA *switches*, also known as riboswitches, are *cis*-regulatory elements that change secondary structure upon binding to small molecules. Functional RNA *switches* are composed of two parts, a small-molecule binding region or *aptamer*, and a regulatory region called the *expression platform*. The interaction between the aptamer and its ligand triggers a conformational change in the expression platform which, depending on the type of riboswitch, regulates gene expression using different mechanisms, such as: transcription termination by the formation of a rho-independent transcription termination hairpin[146]; translation inhibition or initiation by respectively sequestering[147] or releasing[148] the ribosomal binding site; alternative splicing[149]; and self-cleavage, where the expression platform acts as a ribozyme [150].

In [151], a non temperature-dependent riboswitch was manually designed, which promotes cap-independent translation in wheat germ cell lysate only upon binding of the ligand theophylline.

Recently, a synthetic *theophylline riboswitch* has been rationally designed to *transcriptionally* regulate the expression of a gene by fusing a theophylline aptamer with a computationally designed expression platform [152], which contains a rho-independent transcription termination hairpin. Moreover, this design was improved by introducing modifications that optimize the balance between the ON/OFF competing structures. Further analysis showed the importance of including cotranscriptional folding trajectory predictions in order to avoid thermodynamic folding traps [153]. In addition, the authors showed that tandem arrangement of several synthetic riboswitches linearly enhances the activation ratio, in contrast to natural riboswitches which show cooperative binding behavior [154].

One of the challenges for the synthetic design of riboswitches is understanding the biophysics in 3-dimensional aptamer-ligand interactions that trigger the conformational change. However, previous work shows that the thermodynamic computations of secondary structure free energy change between the gene ON and gene OFF conformations may well suffice for the computational design of synthetic riboswitch aptamers [155, 156].

In this chapter, we introduce the software `RNAiFold2T`, capable of solving the inverse folding problem for two or more temperatures, i.e. generating one or more RNA sequences whose minimum free energy (MFE) secondary structures at temperatures T_1 and T_2 [resp. $T_1, \ldots, T_m$] are user-specified target structures S_1 and S_2 [resp. $S_1, \ldots, S_m$], or which reports that no such solution exists. `RNAiFold2T` is unique in that it implements two different algorithms – *Constraint Programming* (*CP*) and *Large Neighborhood Search* (*LNS*). *CP* is an exact, non-heuristic method that uses an exhaustive yet efficient branch-and-prune process, and is the only currently available software capable of generating all solutions or determining that no solution exists (since there are possibly exponentially many solutions, a complete solution is feasible

only for structures of modest size). *CP* differs from what one might call a 'brute-force' approach only in that it relies on a highly efficient branch-and-prune search engine, that propagates the effects of currently instantiated variables held within a *constraint store*. *LNS* uses a local search heuristic, complemented with local calls of *Constraint Programming* to explore solutions of substructures of the target structures.

4.3 Algorithm description

`RNAiFold2T` uses *Constraint Programming* (*CP*) to determine those sequences, whose minimum free energy (MFE) structure at temperature T_1 [resp. T_2] is identical to a user-specified target structure S_1 [resp. S_2]. The target structures S_1,S_2 can also be hybridization complexes of two RNAs, rather than single secondary structures. *CP* performs a complete, exhaustive (branch and prune) exploration of the search space and therefore, it can return all possible solutions of the *thermoswitch* design problem or prove that no solution exists (given an unlimited amount of time). In addition to *CP*, `RNAiFold2T` also supports *Large Neighborhood Search* (*LNS*), a fast local (not complete) search metaheuristic that employs *CP* to exhaustively explore large neighborhoods of every candidate solution at each iteration step. Moreover, since it is written in C++ using the `OR-Tools` engine [26], together with plug-ins to `Vienna RNA Package` [139] and `RNAstructure` [157], the user can install and run `RNAiFold2T` locally, thus permitting much longer execution times than supported by our web server.

The overall methodology of `RNAiFold2T` is similar to its precursor, `RNAiFold 2.0` [55, 56], and `RNAiFold 3.0` now contains `RNAiFold2T`. Therefore, `RNAiFold2T` includes all the design

features described in Chapter 2. Moreover, the new additions do not compromise the performance of the algorithm when solving the inverse folding problem for one temperature. However, as explained below, there are a number of algorithmic details that are new and not present in `RNAiFold 2.0`– decomposition tree for 2 or more target structures, novel *local structural constraints* intended for the design of RNA *switches*, variable (helix) and value heuristics that are proper only to `RNAiFold2T`, a restart heuristic for multiple target structures in *LNS* to ensure a good trade-off between exploration of promising regions of the search space versus the exploration of remote portions of the search space. `RNAiFold2T` cleanly separates all constraints from the *CP* or *LNS* solver, thus permitting our software to be extended to support any future desirable constraints. Additionally, `RNAiFold2T` can determine a user-specified number of solutions, all solutions (given sufficient time), or whether no solution exists. Indeed, memory requirements for `RNAiFold2T` are minimal, and since there are no memory leaks, the software can be run for weeks.

In developing a *CP* solution to a given problem the main tasks are to define the problem (specify variables, domains and constraints) and to define the search procedure. The extension of the model from `RNAiFold 2.0` is trivial and consists of adding new constraints for the helices corresponding to the second structure. The search procedure, however, must be adapted to the new difficulties imposed by the 2-temperature problem. New variable and value ordering heuristics are needed in order to solve 2-temperature inverse folding efficiently. The algorithmic details related to the new search procedure are explained below.

4.3.1 Structure decomposition

As in other inverse folding methods, such as `RNAinverse` [139], `NUPACK-DESIGN` [21], etc., we rely on a decomposition tree of the structure into independent helices, called *extended helices* (*EHs*) and *extended helices with dangles* (*EHwDs*). See Section2.3.1 in Chapter 2 and Appendix B for definitions of *EH*, *EHwD* and decomposition tree, and see Figure 4.1 for an illustration of the *EHwD* decomposition tree for a FourU RNA *thermometer*. Decomposition trees play a special role in `RNAiFold2T` for the following reasons: (a) each node in the decomposition tree is a constraint, (b) the helix and variable heuristics (described later) cause the search tree to be searched in a specific order. To improve efficiency in solving multi-temperature inverse folding, we investigated various helix and value heuristics, which steer the search within a search space defined by a composite decomposition tree, comprising subtrees for each target structure at the corresponding temperature.

Consider the 65 nt FourU RNA *thermometer* (CP000647.1/1773227-1773291), whose MFE structures determined by `RNAfold` from `Vienna RNA Package2.1.9`[139] at 37°C and 53°C are given in dot-bracket notation by

```
>CP000647.1/1773227-1773291)
12345678901234567890123456789012345678901234567890123456789012345
GGACAAGCAAUGCUUGCCUUUAUGUUGAGCUUUUGAAUGAAUAUUCAGGAGGUUAAUUAUGGCAC
((.(((((..)))))))).....(((((.((((((((((....)))))))))).)))))........
.............((((.....(((((.((((((((((....)))))))))).)))))...)))).
```

Let S_1 [resp. S_2] denote the MFE structure of this FourU RNA *thermometer* at 37°C [resp. 53°C]. S_1 is identical to the consensus structure from `Rfam` 12.0 [65], as well as the structure displayed in Figure 1 of [129] for the FourU sequence taken from the 5′-UTR of the *Salmonella* agsA gene.

Giving labels as described to nodes in the *EHwD* decomposition trees $\mathcal{T}_1$,$\mathcal{T}_2$ respectively for S_1 and S_2, we find the *EHwD* decomposition of S_1 has *EHwD* 0 from positions 1-65, *EHwD* 1 from positions 1-19, and *EHwD* 2 from positions 23-58, while the *EHwD* decomposition of S_2 has *EHwD* 4 from positions 1-65, *EHwD* 5 from positions 14-65, and *EHwD* 5 from positions 23-58. Figure 4.1 depicts the *EHwD* decomposition trees for temperatures T_1,T_2, joined together with a (dummy) root that corresponds to the solution returned by `RNAiFold2T`.

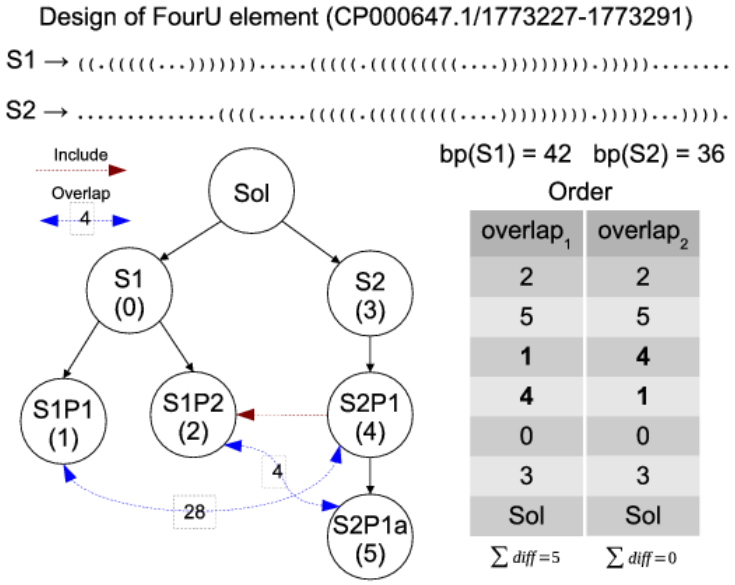


FIGURE 4.1: Example of *EHwD* tree decomposition *EHwD* heuristics for two target structures. Tree decomposition for target structure S_1 [resp. S_2] at temperature T_1 [resp. T_2], using target structures for the FourU element CP000647.1/1773227-1773291. The table on the right presents the order of *EHwD* exploration using $overlap_1$ and $overlap_2$ heuristics (these overlap heuristics are explained in Section 4.3.4.1 Helix ordering heuristics).

As explained in Chapter 2, it can happen that the MFE structure of an extended helix does not agree with the target substructure, while that of the extended helix with dangles does. This is the reason that the *EH* decomposition trees of `RNAiFold1.0` [55] were replaced by *EHwD* decomposition trees in `RNAiFold 2.0` [56], and why *EHwD* trees are used in `RNAiFold2T`.

In the *CP* and *LNS* search strategy, whenever a subsequence corresponding to a node of the decomposition tree has been instantiated, a check is made to determine whether the MFE structure of the subsequence is identical to the target substructure (see Figure4.2). In the case of structural disagreement, the instantiated subsequence is discarded and backtracking occurs. For any solution sequence returned by `RNAiFold2T`, it follows that at temperature T_1 [resp. T_2], each subsequence of the solution that corresponds to a node in decomposition tree $\mathcal{T}_1$ [resp. $\mathcal{T}_2$] folds into the corresponding substructure of target S_1 [resp. S_2].

4.3.2 Variables

The modular implementation of `RNAiFold` makes it easily scalable. Therefore, modeling the inverse folding problem for multiple structures in `RNAiFold2T` is trivial and it only requires the addition of a few *CP* variables to the model described in Section 2.3.2 of Chapter 2. In `RNAiFold2T`, the sets of *CP* variables UP, BP and BPT of `RNAiFold` were replaced by *arrays of sets of variables*, where each target structure s has three sets of variables $UP[s]$, $BP[s]$, $BPT[s]$ associated, corresponding to the unpaired positions, base pairs and base pair types in s respectively.

In addition, the dictionaries ($BPstart$,$BPend$,$UPdict$) were replaced by *arrays of dictionaries*, where $BPstart[s]$,$BPend[s]$ and $UPdict[s]$ respectively contains the indexes of the opening base pairing positions, closing base pairing positions and unpaired positions of the target structure s.

4.3.3 Constraints

4.3.3.1 Channeling constraints

Channeling constraints of `RNAiFold` were also modified in order to maintain the consistency of the new *CP* variables of `RNAiFold2T`. Therefore, for each target structure s `RNAiFold2T` defines the following *channeling constraints*:

- For each base pair i in s, $BP[s]_i := x_{BPstart[s](i)} - x_{BPend[s](i)}$.
- For each base pair i in s, $BPT[s]_i := (x_{BPstart[s](i)} - x_{BPend[s](i)})^2$.
- For each unpaired position i in s, $UP[s]_i := x_{UPdict[s](i)}$.

Rather than reducing the efficiency of the algorithm, the addition of new *channeling constraints* can increase the search speed, since the propagation of *channeling constraints* after each assignment prunes the search tree. The toy example depicted in Figure 4.2 shows (in blue) how after each assignment constraint propagation reduces the domain of other *CP* variables.

4.3.3.2 Local structural constraints

One of the most useful features included in `RNAiFold2T` is the implementation of *local structural constraints*, which are specifically intended for the design of RNA *switches*. *Local structural constraints* are similar to structural constraints defined in Section 2.3.3.3. However, *local structural constraints* are not directly associated with *EHwD*. Therefore, they allow the user to

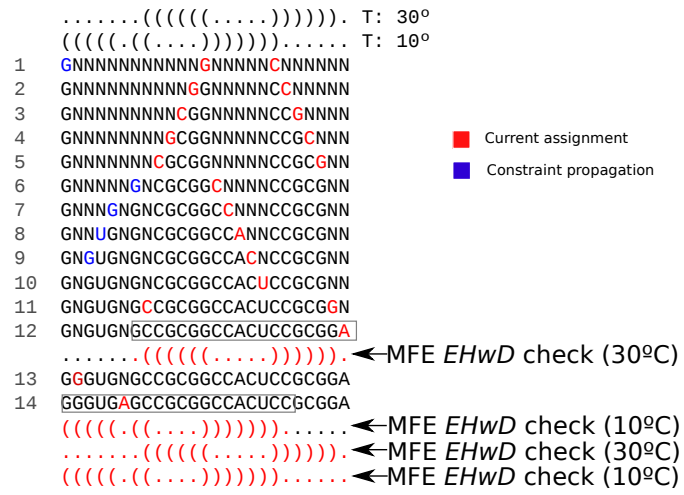


FIGURE 4.2: Example of *CP* search for target structures S_1 (top) at temperature 30°C, and S_2 (bottom) at temperature 10°C. Undetermined positions are assigned (red) in the following order: base pairs are instantiated in lines 1,2,3,4,5,13; unpaired positions in lines 6,7,8,9,10,14; a closing base pair in line 11; a dangling position in line 12. When the left nucleotide of a base pair is instantiated to C [resp. A], then propagation of the base pairing constraint reduces the domain of possible values of the right nucleotide of the base pair from $\{A,C,G,U\}$ to the set $\{G\}$ [resp. $\{U\}$] – this happens in lines 1,6-9. For simplification, this example assumes that correct value assignments always occur at each search step.

specify structural requirements for any fragment of the solutions returned, such as the minimum free energy structure of a fragment, and lower and upper bounds of free energy and *ensemble defect*. Recall that *ensemble defect* is the expected Hamming distance between the ensemble of secondary structures of an RNA sequence and a given user-specified target structure (See Appendix A for a formal definition of *ensemble defect*).

Let a be an RNA sequence and s its minimum free energy structure. If we select a fragment $a[i,j]$ between positions i and j the computed minimum free energy structure of the fragment $a[i,j]$ is not necessarily the same as $s[i,j]$. This observation is critical for the design of RNA *switches*. Consider a to be the sequence of an RNA *switch* of size n, composed by an aptamer of size i at positions $[1-i]$ and an expression platform of size $n-i$ at positions $[n-i+1,n]$. In the absence of a ligand, the MFE structure of a is s. When the aptamer changes its structure into

a more energetically favorable conformation upon binding to the corresponding ligand, the expression platform c folds independently into the MFE structure of the fragment $a_{[n-i+1,n]}$, which can be different from $s_{[n-i+1,n]}$.

Taking this into account, `RNAiFold2T` *local structural constraints* allow the user to define minimum free energy structure targets for fragments of the solutions. *Local structural constraints* are not only restricted to minimum free energy structure requirements; `RNAiFold2T` also allows to stipulate upper and lower bounds for free energy when a fragment is folded into a given structure, or *ensemble defect* limits of a fragment of a given structure.

4.3.4 Heuristics for variable and value order

In a *Constraint Programming* (*CP*) algorithm, one typically specifies the order in which variables are instantiated (assigned), known as the *variable ordering heuristic*, as well as the order in which the values belonging to the domain of each variable are to be assigned, known as the *value ordering heuristic*. The variable ordering heuristic is divided into two levels: first, the order in which extended helices with dangles (*EHwDs*) are to be assigned, and second, the order in which nucleotide positions within helices are to be assigned.

4.3.4.1 Helix ordering heuristics

In the search for thermosensors, there is often an overlap between *EHwDs* of structure S_1 and those of structure S_2 – this situation substantially complicates the task of finding an optimized order of exploration of the *CP* search space. In the *leaves to root* heuristic of `RNAiFold` (see Section 2.3.4.1 in Chapter 2), *EHwD* node H is explored before *EHwD* node H' if the *height*

$ht(H)$ of H is less than the *height* $ht(H')$ of H', or if $ht(H) = ht(H')$ and H appears to the left of H' in the decomposition tree for the single target structure S. In contrast to this heuristic, `RNAiFold2T` implements four different approaches in order to find an adequate exploration ordering for the extended helices with dangles for two target structures S_1 and S_2, whereby a high priority is given to solve those helices, whose sequence is determinant for other parts of the structure due to overlaps. Let N denote the number of nodes (*EHwDs*) in the decomposition tree for S_1, plus the number of nodes (*EHwDs*) in the decomposition tree for S_2. Suppose that H,H' are two distinct *EHwDs* belonging to S_1 or S_2, where the outermost base pair of H [resp. H'] is (i,j) [resp. (i',j')]. Define the following relations:

- $includes(H,H')$ is 1 if $i < i'$ and $j > j'$, i.e. interval $[i,j]$ properly contains $[i',j']$; otherwise $includes(H,H')$ is 0.
- $overlap_1(H,H')$ is 1 if $[i,j] \cap [i',j'] \neq \emptyset$, or equivalently $\max(i,i') \leq \min(j,j')$; otherwise $overlap_1(H,H')$ is 0.
- $overlap_2(H,H')$ is the number of positions k in H and H', for which k is base-paired in both H and H'.
- $overlap_3(H,H')$ is the total number of nucleotide positions k in H and H' (including possible bulges of size 1 or 2, as well as internal loops of sizes 1×1, 1×2, 2×1, and 2×2).
- $overlap_4(H,H')$ is a normalized version of $overlap_2(H,H')$ where the number of overlapping base paired positions in H and H' is divided by the number of base paired positions in H. (Note that $overlap_4$ is not necessarily symmetric.)
- $degree_\alpha(H) = \sum_{i=0}^{N} overlap_\alpha(H,H_i)$, for $\alpha = 1,2,3,4$.

- For $\alpha = 1,2,3,4$, $degreedist_{\alpha}(H,H')$ is equal to $degree_{\alpha}(H) - degree_{\alpha}(H')$, provided that $degree_{\alpha}(H) > degree_{\alpha}(H')$; otherwise, $degreedist_{\alpha}(H,H')$ is 0.

- For $\alpha = 1,2,3,4$, $diff_{\alpha}(\sigma,H,H')$ is equal to $degreedist_{\alpha}(H,H')$, provided that $\sigma(label(H)) < \sigma(label(H'))$; otherwise $diff_{\alpha}(\sigma,H,H')$ is 0.

The value $label(H)$ is defined in Section2.3.4.1 of Chapter 2, and corresponds to visitation order in breadth-first traversal of tree $\mathcal{T}$, and $\sigma : \{0,\ldots,N-1\} \rightarrow \{0,\ldots,N-1\}$ is a permutation that minimizes $\sum_{i=0}^{N} \sum_{j=0}^{N} diff_{\alpha}(\sigma,H_i,H_j)$, subject to the constraint that if H properly includes H', then $order(H) > order(H')$. Note that the partition σ orders the *EHwDs* of S_1 and S_2 in order to minimize the total *incremental overlap*. Before the search for thermosensors begins, `RNAiFold2T` executes a very fast *CP* search to determine the optimal ordering permutation σ. Finally, by setting the index α to 1, 2, 3 or 4 in the definition of $diff_{\alpha}(\sigma,H,H')$, we obtain the corresponding search heuristic. An example of two helix order heuristics (1,2) is shown in Figure 4.1 – note that for even small structures, there are several differences in the helix exploration order when α is 1 or 2.

4.3.4.2 Variable ordering at the nucleotide level

The second level of variable ordering heuristic deals with the exploration of nucleotide positions within a given *EHwD* structure. Variable ordering in `RNAiFold2T` is the same as in `RNAiFold`, and is described in Section 2.3.4.1 of Chapter 2.

4.3.4.3 Value ordering

Value ordering establishes the order in which values are assigned to variables – in our case, this means values GC,CG, AU, UA, GU, UG for base-paired positions and A,C,G,U for unpaired positions. The underlying idea for ordering domain values for variables for base-paired positions and unpaired positions is to allow the creation of thermodynamically stable helices and to take into account the nature of the overlap in overlapping positions. This requires specific value orderings for base-paired positions, depending on whether the position is a dangle, mismatch, normal or closing base pair of an *EHwD* for both targets S_1,S_2, or only one of the target structures.

For multiple target structures `RNAiFold2T` defines specific ordering heuristics for base-paired positions depending on the pairing status in each target structure and the respective target temperatures. These heuristics also incorporate a random component to ensure that parallel runs explore the search space in a different order. `RNAiFold2T` employs value ordering heuristics v_0, which is described in Section2.3.4.2 of Chapter 2, and v_1, where v_1 is summarized in Table 4.1 and the pseudocode in Appendix D. Let S denote the target structure at temperature T, S' denote the target structure at temperature T'. Value ordering heuristic v_1 employs base pair instantiation orderings that are specific to the environment in which the base pair is found, i.e. type 0-7 described as follows. Type 0: base pair $(i,j) \in S \cap S'$; Type 1: $(i,j) \in S$, i,j both unpaired in S', $T < T'$; Type 2: $(i,j) \in S$, i,j both unpaired in S', $T' < T$; Type 3: $(i,j) \in S$, either i or j paired differently in S'; Type 4: $(i,j) \in S$, i unpaired in S'; Type 5: $(i,j) \in S$, j unpaired in S'; Type 6: $(i,j) \in S$, $(i-1,j) \in S'$ or $(i+1,j) \in S'$; Type 6′: $(i,j) \in S$, $(i,j-1) \in S'$ or $(i,j+1) \in S'$; Type 7: $(i,j) \in S$, $i-1,i+1$ unpaired in S or $j-1,j+1$ unpaired in S. In the case of types 0,2,7, the same

procedure is employed as in v_0; i.e. the free energy for each base pair choice G-C, C-G, A-U, U-A, G-U, U-G for base pair (i,j) is tabulated, a random value between 0 and $2kcal/mol$ is added to the free energy, and the list is sorted in increasing order. Base pairs are then tried in that order. In case 1, the procedure is opposite that of 0,2,7; i.e. the list is sorted in decreasing order and base pairs are then tried in that order. In case 3, the following fictive *pseudo-energies* -2.90,-2.23,-1.90,-1.37,-1.03,-0.10 are assigned respectively to base pairs G-C, G-U, U-G, U-A, C-G, A-U. A random value between 0 and 2 kcal/mol is added to each pseudo-energy, and then base pairs are tried according to increasing pseudo-energies. In the remaining cases 4,5,6,6′, the same fictive *pseudo-energies* are taken, but are instead assigned to the base pairs indicated in Table 4.1. For instance, in case 4, pseudo-energies -2.90,-2.23,-1.90,-1.37,-1.03,-0.10 are assigned respectively to G-C, G-U, U-G, U-A, C-G, A-U. A random value between 0 and 2 kcal/mol is added to each pseudo-energy, and then base pairs are tried according to increasing pseudo-energies.

Type	Condition	Value order
$0^{\dagger}$	$(i,j) \in S \cap S'$	GC-CG-AU-UA-GU-UG
$1^{\ddagger}$	$(i,j) \in S$, i,j both unpaired in S', $T < T'$	UG-GU-UA-AU-CG-GC
$2^{\dagger}$	$(i,j) \in S$, i,j both unpaired in S', $T' < T$	GC-CG-AU-UA-GU-UG
3	$(i,j) \in S$, either i or j paired differently in S'	GC-CG-GU-UG-AU-UA
4	$(i,j) \in S$, i unpaired in S'	GC-GU-UG-UA-CG-AU
5	$(i,j) \in S$, j unpaired in S'	CG-UG-GU-AU-GC-UA
6	$(i,j) \in S$, $(i-1,j) \in S'$ or $(i+1,j) \in S'$	UG-GU-AU-UA-CG-GC
$6'$	$(i,j) \in S$, $(i,j-1) \in S'$ or $(i,j+1) \in S'$	UG-GU-AU-UA-CG-GC
$7^{\dagger}$	$(i,j) \in S$, $i-1,i+1$ unpaired in S or $j-1,j+1$ unpaired in S	GC-CG-AU-UA-GU-UG

TABLE 4.1: Value ordering for base pairs used in `RNAiFold2T`: Assume that S [resp. S'] is the target structure at temperature T [resp. T']. We consider cases where $(i,j) \in S \cap S'$, $(i,j) \in S - S'$, etc. Despite the order indicated in the table, the implementation in `RNAiFold2T` includes a random component, so that different parallel runs will explore the search space in a different fashion. This effects only the order of base pair value assignments, but not the completeness of *CP*– regardless of value order, *CP* involves a complete search of the search space using a branch-and-prune strategy. Types marked with a dagger † [resp. ‡] correspond to increasing [resp. decreasing] base stacking free energies.

4.3.5 LNS restart heuristics

Similarly as in the *LNS* variant of `RNAiFold`, the restart condition is a given amount of time, proportional to the length of the target structures, after which search is stopped and some variables are fixed in order to explore exhaustively a large neighborhood of the current solution. After the first restart, full exploration of the remaining space with no solution found is also a restart condition.

In `RNAiFold2T`, when a restart is triggered, a set of positions is selected as candidates to be fixed. The MFE structure for each *EHwD* of the current (attempted) solution is evaluated independently. If the MFE structure of an *EHwD* matches with the target structure at the desired temperature, and the MFE structure of each overlapping helix in the second target structure (at the corresponding temperature) also matches, then all the *EHwD* positions are included in the set of candidates. Then, candidate positions are fixed using the same strategy and probabilities described in Section 2.3.5 of Chapter 2 in order to adjust the trade-off between *exploitation* and *exploration*.

4.4 Benchmarking

In this section we compare the different helix ordering and value heuristics included in `RNAiFold2T`. In addition we benchmark the *Constraint Programming* (*CP*) and *Large Neighborhood Search* (*LNS*) programs of `RNAiFold2T` with the adaptive walk program `SwitchDesign` (`SD`) [145] and the genetic algorithm `FRNAkenstein` (`FRNA`) [44].

We created a benchmarking test set by retrieving the sequences of seven families of non-coding RNA *thermometers* (RNATs) from `Rfam`: RF00038, RF00433, RF00435, RF01766, RF01795, RF01804, RF01832. These families include both cold and heat shock RNA *thermometers*, taken from diverse organisms including phages, prokaryotes and eukaryotes, with sequence length ranging from 60 nt to 450 nt. The benchmarking was divided into two groups: sequences shorter than and longer than 130 nt.

For each sequence, we used `RNAfold` [139] with the Turner '99 energy parameters [59] to determine the MFE structures at temperatures T_1 and T_2, where temperatures were chosen (essentially) according to published experimental studies for each RNA *thermometer* family [134, 158, 159] – in particular, we increased the temperature difference $T_2 - T_1$ from the published values to ensure that `RNAiFold` produced distinct structures at T_1 and T_2 if possible. Turner '99 rather than Turner '04 energies were used, since it required less distortion from published temperatures T_1,T_2. All sequences, whose MFE structures at T_1 and T_2 were identical, were subsequently removed.

Appendix D contains a list of sequences, structures, and temperatures used. The resulting benchmarking set includes all 5 *Lambda* phage CIII thermoregulator elements (Lambda_thermo), all 3 FourU RNATs, 11 of 13 repression of heat shock (ROSE) elements, 8 of 14 sequences from a second family of repression of heat shock (ROSE_2) elements, 3 of 13 thermoregulators of PrfA virulence genes (PrfA), 4 of 6 HSP90 *cis* regulatory elements (HSP_CRE), and 14 of 15 cold shock protein regulator sequences (CspA).

Our first benchmark, summarized in Table 4.2 compares `RNAiFold2T` with respect to different helix ordering and value heuristics, using a cutoff time of 10 minutes. Since the data clearly

demonstrates the superiority of $overlap_2$ helix ordering, this is taken as the default for all other benchmarks and for the web server.

EMBL acc.	n	RF family	o-1-v0	o-1^t-v0	00-v0	01-v0	02-v0	o-1-v1	o-1^t-v1	00-v1	01-v1	02-v1
M13767.1/3-60	58	λthermo	0	**58**	17	17	23	0	4	10	8	6
CP000243.1/1246604-1246546	59	λthermo	0	**59**	16	17	14	1	0	13	4	8
CP000026.1/2520723-2520781	59	λthermo	0	0	1	0	3	0	14	4	**10**	7
CP001144.1/624595-624537	59	λthermo	0	9	22	19	11	0	9	**31**	24	30
AY736146.1/34404-34346	59	λthermo	0	**16**	5	0	0	0	0	1	4	2
CP000647.1/1773227-1773291	65	FourU	12	87	13	86	**94**	11	84	14	86	82
CP001127.1/1302123-1302187	65	FourU	0	71	19	73	**74**	7	62	8	62	60
CP001144.1/2031534-2031470	65	FourU	0	0	0	0	0	0	0	0	0	0
ACDJ01000026.1/381061-380989	73	ROSE_2	0	0	0	**32**	7	0	0	1	2	11
ABWL02000023.1/393416-393344	73	ROSE_2	0	1	1	10	11	4	0	2	3	**22**
CP000653.1/14627-14699	73	ROSE_2	0	97	91	92	**100**	0	98	**100**	99	**100**
CP000036.1/3699544-3699616	73	ROSE_2	0	0	0	0	0	0	0	1	0	**2**
CP000026.1/3798554-3798481	74	ROSE_2	0	16	0	0	0	0	11	2	1	**20**
AE017220.1/3951363-3951290	74	ROSE_2	2	0	11	86	**88**	5	0	14	86	83
BAAW01000185.1/6674-6747	74	ROSE_2	0	77	97	99	88	0	75	**100**	62	83
CP000647.1/4480191-4480116	76	ROSE_2	0	2	2	0	**24**	0	1	1	0	9
CP000009.1/1450710-1450627	84	ROSE	0	89	88	3	0	10	91	**94**	1	3
AP003017.1/94542-94451	92	ROSE	**73**	17	59	16	50	17	39	21	42	24
AE007872.2/441983-442075	93	ROSE	76	72	85	84	78	96	96	**98**	94	93
AE007872.2/51225-51317	93	ROSE	9	11	2	1	3	2	**76**	4	10	8
AL591985.1/872145-872052	94	ROSE	**92**	0	0	2	1	13	0	58	21	27
BA000012.4/1943819-1943723	97	ROSE	93	67	62	59	64	70	97	95	96	**100**
RU55047.1/3106-3215	110	ROSE	0	**2**	0	0	0	0	0	1	0	0
AJ003064.1/2697-2806	110	ROSE	2	0	0	5	1	1	3	0	**10**	2
U55047.1/5180-5291	112	ROSE	4	2	0	0	0	40	22	42	**57**	53
AJ010144.1/622-738	117	ROSE	0	90	84	75	91	58	96	**98**	95	97
AJ003064.1/2430-2312	119	ROSE	0	94	0	0	0	66	**98**	74	66	60
Total			363	937	675	776	825	401	976	887	943	**992**
Str. Solved			9	20	18	18	19	15	18	**25**	23	**25**

TABLE 4.2: `RNAiFold2T` heuristic combination test. Test of combinations of helix ordering heuristics ($o-1$, $o-1^t$, $o0$, $o1$, $o2$) and value ordering heuristics ($v0$, $v1$), using `RNAiFold2T CP`. Benchmarking statistics were obtained from 100 runs, with time limit set of 10 minutes per run, performed on a Core2Duo PC (2.8 GHz; 2 Gbyte memory; CentOS 5.5). $v0$ denotes the value ordering used in `RNAiFold`; $v1$ stands for the new value ordering from 4.1; $o-1$ denotes the helix ordering for a single structure, where the helix search order is from *leaves to root* in the *EHwD* decomposition tree of the first target structure alone (with intermediate checks whether any fully instantiated sequence for any fully instantiated sequence for an *EHwD* of the second structure folds into S_2 at T_2). $o-1^t$ denotes the helix ordering for a single structure, as in $o-1$, except that the *EHwD* decomposition tree is for the target structure at the *higher temperature*. $o0$ denotes the helix ordering from *leaves to root* in the combined decomposition tree for both target structures, as depicted in Figure 4.1, while $o1$ [resp. $o2$] denotes the helix ordering heuristic 1 ($overlap_1$) [resp. heuristic 2 ($overlap_2$)], described in Section 4.3.4.1.

The second benchmark compares the performance of `RNAiFold2T`, `SD` and `FRNA` for the task of finding a single solution of the 2-temperature inverse folding problem in a given amount of time. Table 4.3 [resp. Table 4.4] presents benchmarking data for *Large Neighborhood Search* (*LNS*) from `RNAiFold2T` and `SD` and `FRNA`, each with a cutoff time of 30 minutes, using `Rfam` thermosensor target structures of length less than 130 nt [resp. greater than 130 nt]. The

results show that RNAiFold2T has essentially the same performance as SD and FRNA for shorter sequences, while SD performs better than other methods for longer sequences.

Parameters			RNAiFold2T		Frnakenstein		SwitchDesign	
EMBL acc.	n	Rfam family	solved	Avg. cost	solved	Avg. cost	solved	Avg. cost
M13767.1/3-60	58	λ thermo	**30**	-1.12	**30**	-1.97	**30**	-3.66
CP000243.1/1246604-1246546	59	λ thermo	**30**	-1.22	**30**	-1.77	**30**	-3.70
CP000026.1/2520723-2520781	59	λ thermo	**29**	1.66	27	-1.09	27	-2.14
CP001144.1/624595-624537	59	λ thermo	**30**	0.54	27	-1.16	28	-2.37
AY736146.1/34404-34346	59	λ thermo	**30**	0.20	26	-1.30	24	-2.64
CP000647.1/1773227-1773291	65	FourU	**30**	2.50	**30**	2.82	16	0.95
CP001127.1/1302123-1302187	65	FourU	**30**	2.43	**30**	1.91	26	0.37
CP001144.1/2031534-2031470	65	FourU	3	2.31	**28**	1.48	25	0.93
ACDJ01000026.1/381061-380989	73	ROSE_2	11	3.75	**19**	2.19	8	1.10
ABWL02000023.1/393416-393344	73	ROSE_2	**2**	4.08	0	-	**2**	1.56
CP000653.1/14627-14699	73	ROSE_2	**26**	3.87	6	2.60	12	0.84
CP000036.1/3699544-3699616	73	ROSE_2	**30**	1.61	**30**	1.48	28	0.40
CP000026.1/3798554-3798481	74	ROSE_2	**30**	1.60	**30**	2.16	23	0.21
AE017220.1/3951363-3951290	74	ROSE_2	**30**	3.83	**30**	3.90	11	0.84
BAAW01000185.1/6674-6747	74	ROSE_2	**30**	3.09	**30**	4.29	28	0.99
CP000647.1/4480191-4480116	76	ROSE_2	**30**	2.82	**30**	2.35	25	0.22
CP000009.1/1450710-1450627	84	ROSE	**30**	2.63	**30**	2.07	26	0.31
AP003017.1/94542-94451	92	ROSE	**30**	2.25	**30**	2.05	**30**	0.14
AE007872.2/441983-442075	93	ROSE	**30**	4.67	**30**	4.09	16	1.10
AE007872.2/51225-51317	93	ROSE	**30**	6.88	27	5.62	12	2.08
AL591985.1/872145-872052	94	ROSE	**29**	6.13	20	3.70	19	1.44
BA000012.4/1943819-1943723	97	ROSE	**30**	5.08	**30**	5.16	23	1.61
RU55047.1/3106-3215	110	ROSE	**30**	6.43	7	3.47	14	1.03
AJ003064.1/2697-2806	110	ROSE	**30**	7.46	29	7.43	7	1.99
U55047.1/5180-5291	112	ROSE	6	8.08	1	7.66	**17**	1.58
AJ010144.1/622-738	117	ROSE	**30**	7.58	**30**	5.82	18	1.70
AJ003064.1/2430-2312	119	ROSE	4	8.91	0	-	**9**	3.04
Total/Avg. Cost			**680**	3.63	637	2.60	534	**0.37**
Str. Solved			**27**		25		**27**	

TABLE 4.3: Benchmark for sequences shorter than 130 nt: Summary of the computational results for Rfam structures shorter than 130 nucleotides comparing RNAiFold2T (LNS), SD and FRNA. Benchmarking was performed over 30 runs with time limit set to 30 minutes for each run, measured on a Core2Duo PC (2.8 GHz; 2 Gbyte memory; CentOS 5.5).

In addition, SwitchDesign (SD) [145], FRNAkenstein (FRNA) [44], and RNAiFold2T were benchmarked on Rfam family RF01804 of Lambda phage CIII thermoregulator elements to determine the maximum number of distinct solutions over 24 hours, restarting when no new solution is found within 1 hour. Since neither FRNA nor SD output more than one solution, we made the following modifications of each program. The genetic algorithm FRNAkenstein was run as many times as possible over 24 hours (each time with a time limit of 1 hour); we then

			RNAiFold2T		Frnakenstein		SwitchDesign	
EMBL acc.	n	Rfam family	solved	Avg. cost	solved	Avg. cost	solved	Avg. cost
M55160.1/297-426	130	PrfA	2	5.59	0	-	**4**	1.80
AJ002742.1/161-290	130	PrfA	**10**	2.69	**10**	2.13	9	0.30
X72685.1/1303-1435	133	PrfA	**10**	6.05	0	-	9	0.70
AAEU02000321/310682-310830	149	Hsp90_CRE	0	-	0	-	**3**	1.52
AAPQ01006501/448359-448507	149	Hsp90_CRE	**10**	6.17	6	5.50	1	1.45
AY122081/193-344	152	Hsp90_CRE	**10**	4.18	5	3.16	5	0.84
X03811/879-1030	152	Hsp90_CRE	**10**	4.31	9	3.05	4	1.41
L23115.1/459-834	376	CspA	0	-	0	-	0	-
AF017276.1/479-855	377	CspA	**10**	9.47	0	-	9	2.33
ABJD02000101.1/494629-495045	417	CspA	0	-	0	-	0	-
CP000647.1/4296745-4297166	422	CspA	0	-	0	-	0	-
CP000653.1/190938-191361	424	CspA	0	-	0	-	0	-
ABWM02000027.1/244578-245001	424	CspA	0	-	0	-	0	-
ABWL02000023.1/204545-204970	426	CspA	0	-	0	-	0	-
ABEH02000004.1/260631-260204	428	CspA	0	-	0	-	0	-
CP000946.1/174765-174338	428	CspA	0	-	0	-	0	-
CP000822.1/4602942-4603373	432	CspA	0	-	0	-	0	-
ACCI02000028.1/45583-45145	439	CspA	0	-	0	-	0	-
AAOS02000014.1/93711-93269	443	CspA	0	-	0	-	0	-
AALD02000025.1/10362-9916	447	CspA	0	-	0	-	0	-
ABXW01000053.1/281917-281471	447	CspA	0	-	0	-	0	-
Total/Avg. Cost			**62**	5.50	30	3.46	44	**1.29**
Str. Solved			7		4		**8**	

TABLE 4.4: Benchmark for sequences of length greater than 130 nt: Summary of the computational results for Rfam structures of length greater 130 nucleotides comparing `RNAiFold2T` (LNS), `SD` and `FRNA`. Benchmarking was performed over 10 runs with time limit set to 60 minutes for each run, measured on a Core2Duo PC (2.8 GHz; 2 Gbyte memory; CentOS 5.5).

output all sequences found in the most recent (internally stored) population which fold into the target structures S_1,S_2 at temperatures T_1,T_2. `SD` returns a single sequence which minimizes a cost function described in [145]; thus we modified `SD` source code in order to test whether any sequence explored in the search was a solution. Sequences were checked at two different points in `SD`: when a new sequence is generated by a single mutation (`SD` update), and when a sequence is selected by minimization of the cost function (`SD` selected). In all cases, `SwitchDesign` was restarted if no new sequence was found in one hour.

For each solution set obtained, additional solutions were generated by testing all single point mutations of any solution returned. Table 4.5 displays the number of solutions for λ phage CIII RNA *thermoswitches* from `Rfam` family RF01804. *Constraint Programming* (*CP*) from `RNAiFold2T`,

adaptive walk `SwitchDesign` [145] and genetic algorithm `FRNAkenstein` [44] were run on each thermosensor for 24 hours, forcing a restart if no new solution was found within 1 hour. Our results show that both versions of `SwitchDesign` and `FRNAkenstein`, return two orders of magnitude less solutions than `RNAiFold2T`.

EMBL	FRNA	SD-upd.	SD-sel	RNAiFold2T
CP000243.1/1246604-1246546	675/24,431 (36.2)	535/16,436 (30.7)	23/775 (33.7)	177,428/2,427,236 (13.7)
CP000026.1/2520723-2520781	296/11,529 (38.9)	598/18,519 (31.0)	16/787 (49.2)	68,800/674,593 (9.8)
CP001144.1/624595-624537	341/12,334 (36.2)	342/11,706 (34.2)	27/1,146 (42.4)	216,809/3,665,692 (16.9)
AY736146.1/34404-34346	321/11,976 (37.3)	290/10,163 (35.0)	20/641 (32.1)	64,853/1,027,058 (15.8)
M13767.1/3-60	811/27,008 (33.3)	520/14,733 (28.3)	18/682 (37.9)	49,598/533,629 (10.8)
Average	489/17,456 (35.7)	457/14,311 (31.3)	21/806 (38.8)	115,498/1,665,642 (14.4)

TABLE 4.5: Number of solutions for 2-temperature inverse folding with target structures for λ phage CIII thermoswitches from Rfam family RF01804. Column headers: EMBL accession code, `FRNA`, `SD` (with updates – see text), `SD` (selection – see text), `RNAiFold2T`. For each program, run time was 24 hours, where a restart was forced if no new solution was found within 1 hour. Results are presented as *A/B (C)*, where *A* is the number of distinct solutions returned, *B* the number of distinct solutions after additionally testing all single point mutations of sequences from A, and *C* is the ratio of B over A. Clearly, `RNAiFold2T` computes two orders of magnitude more solutions than the other methods.

4.5 Applications

4.5.1 Analysis of the cost function used in SwitchDesign

As with the synthetic hammerhead design in [124], our synthetic RNA design strategy consists of generating many solutions, which are prioritized for experimental validation by applying various computational filters. In our opinion, this strategy presents advantages over methods using `SD` or `NUPACK-DESIGN`, each of which returns a relatively small number of sequences that are optimized with respect to a single criterion – in the case of `SD`, this is the cost function defined in equation 4.1 [145], and in the case of `NUPACK-DESIGN`, this is *ensemble defect*

[21] (see Appendix D and A for a complete definition of the cost function and *ensemble defect* respectively).

In order to ascertain the viability of our approach of not committing to a particular cost function, we computed the cost of the sequences generated by each method in the last benchmark for target structures from `Rfam` family RF01804 (λ phage CIII thermoregulators). In addition, we generated a reference set of hundreds of thousands of solutions using the capabilities of `RNAiFold2T`.

Figure 4.3 and Figure D.1 in Appendix D show that that the cost function value of (real) `Rfam` sequences is not close to the minimum, but rather close to the average of the distribution. In particular, `SD` and `FRNA` return solutions having substantially lower cost values (i.e. more optimal) than those of natural thermosensors, whose cost value appears to be the mean value returned by `RNAiFold2T`.

Other figures can be found in Appendix D, where we investigated a variant of the cost function defined using *ensemble defect*. So, although `SD` benchmarking results (see Tables 4.3 and 4.4) indicate that cost function minimization is a good strategy to find sequences whose MFE structures at temperatures T_1 resp. T_2 are the target structures S_1 resp. S_2, it appears that naturally occurring RNA *thermoswitches* are not optimized for the `SD` cost function. This observation may be important for the future design of functional synthetic thermoregulators.

4.5.2 Design of thermo-IRES switches

As explained in Section 2.6.3 of Chapter 2, foot-and-mouth disease virus (FMDV) IRES element is composed of five domains, where the domain 5 stem-loop at positions 419-440 and *unpaired*

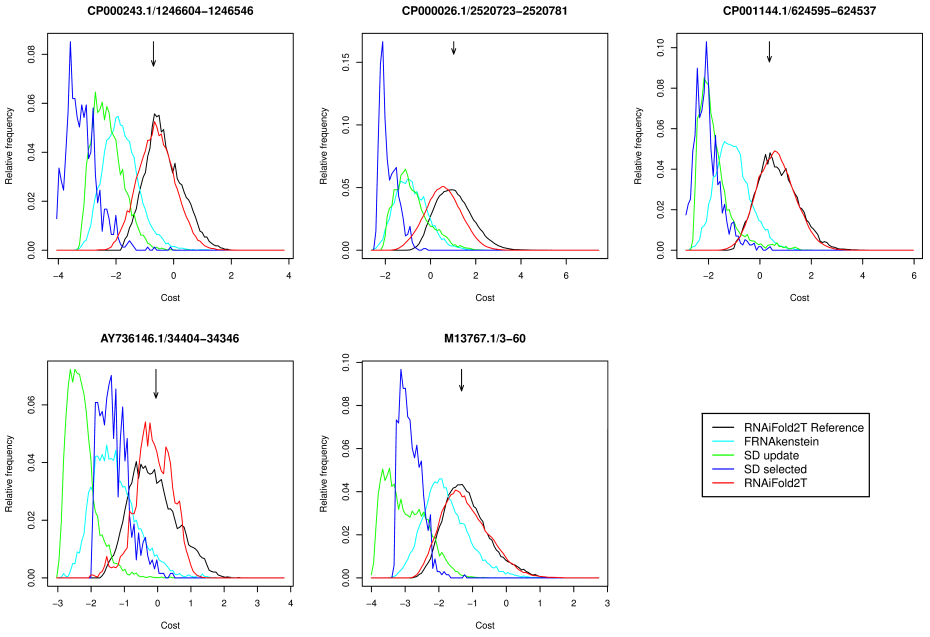


FIGURE 4.3: Relative frequency of the cost function optimized by `SwitchDesign`, for solutions returned by `RNAiFold2T`, `SwitchDesign` and `FRNAkenstein`, given target structure S_1 [resp. S_2] at temperature T_1 [resp. T_2] for λ phage CIII thermoregulators from `Rfam` family RF01804. This figure is a more generous representation of the data from `SwitchDesign` and `FRNAkenstein`, since all single point mutant solutions have been added to the raw output (Figure D.1 in Appendix D presents histograms for the raw output of these programs). The *reference* distribution for *RNAiFold2T Reference* (black curve), was produced by running `RNAiFold2T` for several days. Remaining curves are for `FRNAkenstein` (light green), `SwitchDesign` (dark green and purple) and `RNAiFold2T` (red). Arrows indicate cost values for the real λ phage CIII thermoregulators from `Rfam` RF01804. Distribution for `SD` and `FRNA` without additional single point mutants shown in SI. Figures D.1 and D.3 in Appendix D show clearly that cost function values for `Rfam` sequences approximately equal the reference distribution mean.

pyrimidine tract (*Py tract*) region at positions 441-447, are both known to be essential for IRES activity [160]. Domain 5 of wild-type FMDV IRES element contains 46 nucleotides at positions 417-462 with different sequence conservation, as shown in Figure 4.4, based on Figure 1 of [161].

We used `RNAiFold2T` to design a temperature-regulated internal ribosomal entry site (*thermo*-IRES) element by ensuring the presence at high temperatures of the domain 5 stem-loop and downstream single-stranded pyrimidine (*Py tract*) tract, both located upstream of the functional

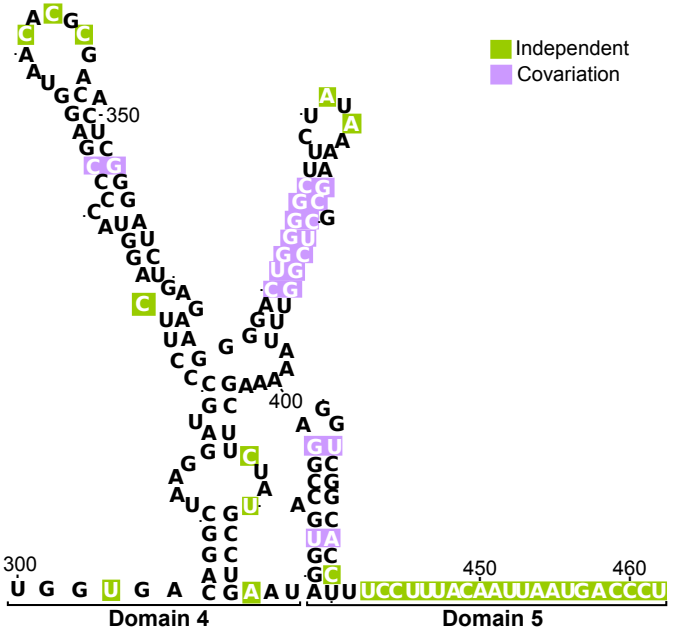


FIGURE 4.4: Sequence variability of domains 4 and 5 of FMDV IRES element: As described in [161], despite the structural conservation, there is some sequence variability among FMDV IRES elements. Invariant nucleotides are marked in bold, covariant nucleotide changes and substitutions are marked in purple and green respectively.

initiation codon and both known to be important for IRES functionality [162]. At low temperatures, our thermo-IRES element is designed to adopt a conformation that down-regulates protein product by disrupting both the domain 5 stem-loop and sequestering the *Py tract*.

Using the following pipeline, two candidate *thermo*-IRES elements were tested, along with a negative control and a positive control (wild-type IRES).

1. As shown in Figures 4.5a and 4.5b, the *inactive* target structure S_1 at $T_1 = 30°C$ was chosen to destroy domain 5 stem-loop and unpaired *Py tract* region, while target *active* structure S_2 at $T_2 = 42°C$ is the experimentally determined structure of wild-type domain 5 FMDV IRES element. Sequence constraints were chosen in accordance with conservation observed in a multiple alignment of 183 IRES elements [161], depicted in Figure 4.4,

where we added AUG start codon at positions 47-49 (corresponding to IRES positions 463-465).

```
Inactive S1: ..............((((((..((((((....))))))..))))))...
  Active S2: ..((((((.(((.....)))))))).........................
Constraints: NUAGGNGACCGNAGGNCGGCNCNUUYYYYYYRNNNNNNNNNNNNNNAUG
```

Using `RNAiFold2T`, 24,410 solutions were generated, although an additional 45,442 sequences were generated using variants of target S_1.

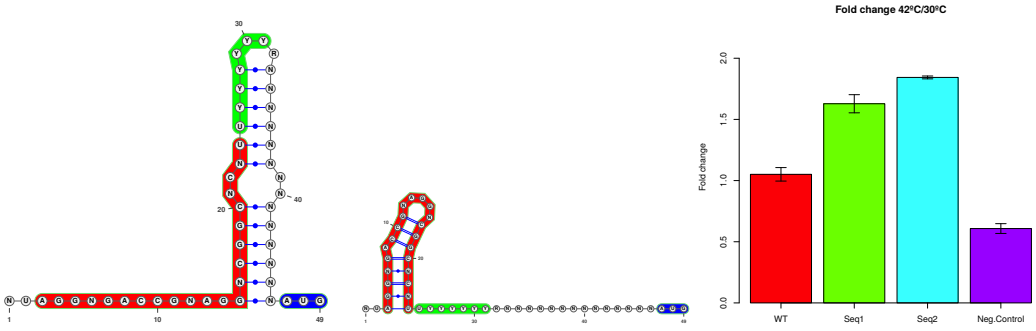
2. `RNAiFold2T` solutions were discarded if any of the following criteria were not met:

   - Wild-type structures for domain 4 and domain 5 appear as stable substructures using `Vienna RNA Package` `RNALfold-L 110 -T 42`.
   - Domain 4 appears as a stable substructure using `RNALfold-L 110 -T 30`
   - Probability $Pr(S_2,T_2)$ of active conformation at $T_2 = 42°C$ exceeds 0.2
   - Probability $Pr(S_1,T_1)$ of inactive conformation at $T_1 = 30°C$ exceeds 0.2
   - Probability of intended target structure at intended temperature is more than double that of unintended target, i.e. $Pr(S_1,42)$ / $Pr(S_2,42) < 0.5$ and $Pr(S_2,30)/Pr(S_1,30) < 0.5$.

3. Retained solutions were further filtered using various measures. For instance, candidate 1 (Seq1) had the highest value of A+B, where A is $a\cdot(b \cdot Pr(S_1,30) + c \cdot (Pr(S_1,30) - Pr(S_2,30)))$ and B is $d \cdot (1- Pr(S_1,30)) + e \cdot (b \cdot Pr(S_2,42) + c\cdot (Pr(S_2,42) - Pr(S_1,42)))$, and $a = 4, b = 0.5, c = 0.5, d = 2, e = 1$. This measure was designed to select sequences where the probability of the intended target at the intended temperature is high, while probability of the unintended target is low. The measure is weighted to increase the likelihood of

not having the inactive conformation at 42°C. In contrast, Candidate 2 (Seq2) is one of two sequences satisfying $Pr(S_2,42) > 0.3$ and $P(S_1,30) > 0.3$.

Seq1 and Seq2 consist of the following 46 nt: Seq1 is AUAGGUGACC GGAGGGCGGC ACCUUUUUUC CAGAAAAGUA GUCGUC (15/46 positions differ from wild-type) and Seq2 is GUAGGUGACC GGAGGACGGC ACCUUUUUUC CAGAAAAGUA GUCGUC (16/46 positions differ from wild-type).



(A) (B) (C)

FIGURE 4.5: *(a,b)* Target structures S_1 at temperature 30°C (a) and S_2 at temperature 42°C (b) for domain 5 thermo-IRES element with added AUG codon (blue). IUPAC sequence constraints are determined from an alignment of 183 IRES sequence[161]s as shown in Figure 4.4 . Domain 5 stem-loop (positions 3-24 in red) and *unpaired* pyrimidine tract (*Py tract* positions 25-32 in green) are known to be *essential* for IRES activity [160]. Target structure S_1 was designed to sequester the *Py tract* at low temperature, thus creating a thermo-IRES which should be functional only at high temperature. *(c)* Ratio of normalized IRES activity at 42°C over that at 30°C for wild-type FMDV IRES, a negative control, and two thermosensors designed using `RNAiFold2T`.

For the experimental validation, synthetic oligonucleotides containing the designed sequences (46 nts) in either positive or negative orientation were annealed in Tris 50 mM pH 7.5, NaCl 100 mM, $MgCl_2$ 10 mM, 15 min at 37°C and subsequently inserted into the HindIII and XhoI restriction sites of pBIC, which harbors the wild-type IRES, linearized with the same enzymes. Colonies that carried the correct insert were then selected, and prior to expression analysis, the

nucleotide sequence of the entire length of each region under study was determined (Macrogen).

In vitro transcription was performed for 1 h at 37°C using T7 RNA polymerase, as described in [163]. RNA was extracted with phenol-chloroform, ethanol precipitated and then resuspended in TE. Using gel electrophoresis, the transcripts were checked for integrity. Equal amounts of the RNAs synthesized in vitro were translated in 70% rabbit reticulocyte lysate (RRL) (Promega) supplemented with ^{35}S-methionine (10 μCi), as described in [164]. Each experiment was independently repeated in triplicate, using the wild type RNA as a control in all assays. Luciferase (LUC) and chloramphenicol acetyl transferase (CAT) activities were measured for the bicistronic plasmid, as previously described [165]. In particular, intensity of the LUC band, as well as the CAT band, produced by each transcript was determined in a densitometer, and normalized against the intensity of LUC and CAT bands produced by the wild type RNA, set at 100%. Values in Figure 4.5c represent the mean ± SD.

Luciferase activity reflects the efficiency of IRES-dependent translation, while CAT activity reflects the efficiency of 5′-dependent translation; thus the ratio LUC/CAT was determined at 30°C and 42°C for wild-type FMDV IRES, the negative control and two thermo-IRES constructs.

Figure 4.5c shows that Seq1 and Seq2 displayed an increase of approximately 50% normalized IRES-dependent translation efficiency in RRL at 42°C versus 30°C. Seq1 and Seq2 IRES elements displayed about 20% lower normalized activity than the wild type IRES. Nonetheless, the wild type IRES was equally active at all temperatures tested (30, 37 and 42 °C).

Our results indicate that our rationally designed thermo-IRES elements are functional, although they are not as efficient as the wild type FMDV IRES. However, since the focus of

this work is primarily using a purely computational design strategy, we have not taken steps to improve efficiency using error-prone mutagenesis and selection.

4.5.3 Design of theophylline *molecular scissors*

In the design of functional RNA *molecular scissors*, our objective is to combine into a single RNA molecule both a *theophylline riboswitch* and a type III hammerhead ribozyme, creating a molecule capable of trans-cleavage of a second RNA molecule only when activated by the presence of theophylline. Therefore, the design contains two RNAs: a molecule of 14nt which is the substrate to be cleaved, and a second RNA molecule of 81nt composed by a type III hammerhead ribozyme and a TCT8-4 theophylline aptamer [166]. Both molecules were designed together, therefore in this case the objective was solving a *2-molecule hybridization complex inverse folding problem* rather than a typical inverse folding problem. For simplicity in the notation we will refer to the full hybridized complex as *molecular scissors* and number it as a single sequence, where the substrate is at positions 1 to 14 and the riboswitch-ribozyme at positions 15 to 95. The hammerhead ribozyme is located at positions 15-53, and the theophylline aptamer at positions 54-95. The design process is modular, therefore in this section we will repeatedly refer to specific fragments of the complex: the substrate at positions 1-14 (`S`), the hammerhead ribozyme at positions 15-53 (`HH`), the substrate to be cleaved and the hammerhead ribozyme in complex at positions 1-53 (`HHC`) and the theophylline aptamer at positions 54-95 (`TA`).

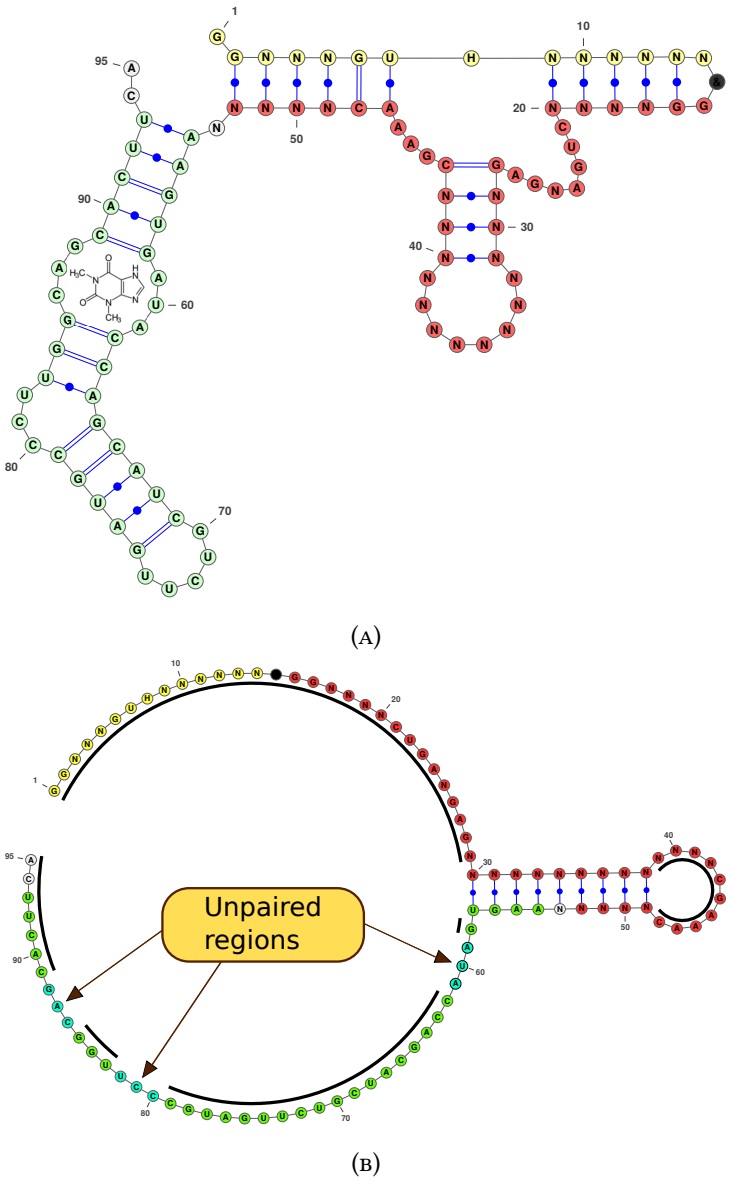
4.5.3.1 Sequence generation

For the rational design of *molecular scissor* candidates we used some of the unique features included in `RNAiFold`: *Partial target* structures, where specific positions may be either paired or not; target hybridization structures; *compatibility* structural constraints; and IUPAC nucleotide constraints (see Chapter 2). In addition, we used the novel features of `RNAiFold2T` for the design of RNA *switches*: *local structural constraints*; and optimized *CP* search for multiple target structures (see Section 4.3).

As explained in Chapter 2, `RNAiFold2T` uses an expanded dot-bracket notation that allows the user to indicate positions with undetermined pairing status and hybridization in the target structures, where a comma indicates that the position may be paired or not, and the ampersand symbol '&' indicates the separation between two RNA molecules that fold into a hybridized RNA complex. For the design of the *molecular scissors* we specified a target secondary structure where the active site of the hammerhead ribozyme is disrupted within a base-paired region (30-57) and the nucleotides involved in theophylline binding (59-61,80-82,86-88) remain unpaired. All other positions can be either paired or unpaired (Figure 4.6b).

The design forces sequences to be compatible with the active conformation in the presence of theophylline. In this conformation `HH` hybridizes with `S` leaving position 8 (where the cleavage occurs) and positions 21-27,44-46 (involved in the catalysis) unpaired (Figure 4.6a).

Sequence constraints for `HHC` were defined based on our previous work[124] described in Chapter 3, where we showed that fixing those positions of Peach Latent Mosaic Viroid (PLMVd) AJ005312.1/282-335 with more than 96% conservation within the Rfam RF00167 family is a good template for the design of an active type III hammerhead ribozyme. The sequence of



(A)

(B)

FIGURE 4.6: Global constraints: *(a)* Secondary structure of the active conformation. The active conformation for cleavage is a hybridized complex of two RNA molecules: the first is the RNA substrate to be cleaved (yellow); and the second includes the structure of the Peach Latent Mosaic Viroid (PLMVd) AJ005312.1/282-335 type III hammerhead ribozyme (red), using the sequence constraints defined in [124], and a TCT8-4 theophylline aptamer [166] (green). *(b)* Target minimum free energy structure for `RNAiFold2T`. The target MFE structure requirements are: a stem-loop pairing a portion of the hammerhead ribozyme with the theophylline aptamer; unpaired nucleotides at the theophylline binding sites (cyan) in order to facilitate theophylline binding; and regions with undefined pairing status (underlined with black curves).

the well-characterized TCT8-4 theophylline aptamer [166] was fixed in `TA`. Additionally, the first two nucleotides at the 5' end of each RNA molecule were set to Gs in order to improve transcriptional efficiency in the experimental validation (observation due to Prof. M.M. Meyer).

Several *local structural constraints* were included in the `RNAiFold2T` input file for the design of the *molecular scissors*: `S` must be completely unfolded in the absence of other RNAs (Figure 4.7a); the local minimum free energy structure of `HHC` must agree with the appropriate hybridized conformation for cleavage (Figure 4.7b); and the local MFE structure of `TA` must be the same that this region adopts when it is bound to theophylline (Figure 4.7c).

The following `RNAiFold2T` input file illustrates the full design. *Local structural constraints* are a single line, where continuation is indicated by a backslash (shown as displayed in order to fit in the page dimensions).

```
> Theophylline molecular scissors
#RNAscdstr
,,,,,,,,,,,,,&,,,,,,,,,,,,,,,(((((((((,,,,,,,,,)))))))))....................................,,,,,,
#RNAseqcon
GGNNNGUHNNNNNN&GGNNNNCUGANGAGNNNNNNNNNNNNNNCGAAACNNNNNAAGUGAUACCAGCAUCGUCUUGAUGCCCUUGGCAGCACUUCA
#RNAcompstr
,((((((.(((((,&,)))))).......((((........))))...)))))),(((((...(((((((.....)))))...)))...)))))..
#MFEstructure
1
#LocalCstrs
1 ............. MFE 1|27 .((((........)))). MFE 1|1 ,((((((.(((((,&,)))))).......((((........))))...)))))), MFE 1| \
53 .(((((...(((((((.....)))))...)))...))))). MFE 1
```

This design can be improved by taking advantage of the optimization strategies implemented in `RNAiFold2T`. As explained in Section 4.3.1, when two target structures (separated by the pipe symbol '|') are given as input, `RNAiFold2T` decomposes the target structures into *EHwDs* and optimizes the *EHwD* order of exploration for the *CP* search. `RNAiFold2T` uses both target structures as MFE constraints unless indicated otherwise. In our design, the global MFE

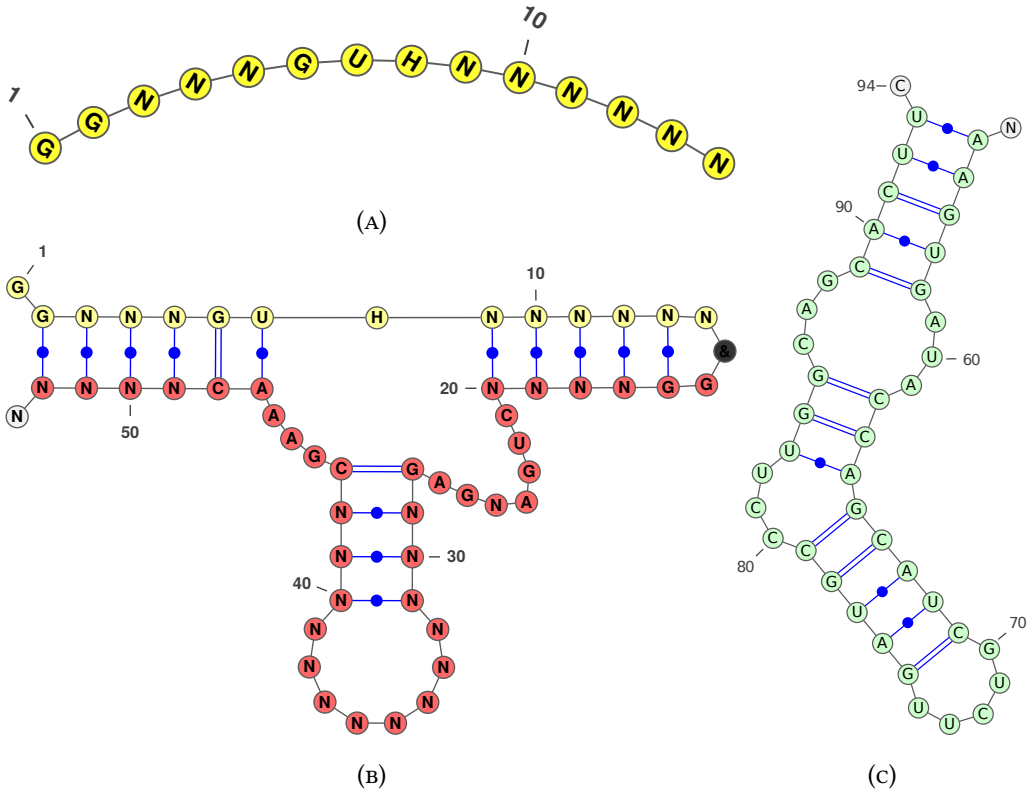


FIGURE 4.7: Local MFE structure constraints: *(a)* RNA substrate completely unfolded to facilitate hybridization *(b)* Hammerhead ribozyme and the substrate hybridized in the cleavage conformation *(c)* Theophylline aptamer adopts the same structure as when it is bound to the ligand. Recall that `S` denotes the substrate at positions 1-14, `HH` denotes the hammerhead ribozyme at positions 15-53, `HHC` denotes the substrate to be cleaved and the hammerhead ribozyme in complex at positions 1-53, and `TA` denotes the theophylline aptamer at positions 54-95.

structure constraint was deactivated by setting the `#MFEstructure` flag to zero, so the two given structures were used only for optimization and as compatibility constraints. MFE structure requirements were fulfilled by the addition of extra MFE *local structural constraints*, which ensure that the MFE structure of the solutions agrees with the inactive target structure.

The final input file of `RNAiFold2T` is as follows. Target structures and *local structural constraints* are single lines, where continuation is indicated by a backslash (shown as displayed in

order to fit in the page dimensions).

```
> Theophylline molecular scissors
#RNAscdstr
,((((((.((((((,&,))))).......((((........))))...))))))),(((((...(((((((((.....)))))...)))...)))))..| \
..............&,,,,,,,,,,,,,,,(((((((((,,,,,,,,,,)))))))))),...,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
#RNAseqcon
GGNNNGUHNNNNNN&GGNNNNCUGANGAGNNNNNNNNNNNNNNCGAAACNNNNNAAGUGAUACCAGCAUCGUCUUGAUGCCCUUGGCAGCACUUCA
#MFEstructure
0
#LocalCstrs
1 ,,,,,,,,,,,,,,&,,,,,,,,,,,,,,,(((((((((,,,,,,,,,,)))))))))),...,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,, MFE 1| \
15 ,,,,,,,,,,,,,,,(((((((((,,,,,,,,,,)))))))))),...,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,, MFE 1| \
29 ,(((((((((,,,,,,,,,,))))))))), MFE 1| \
1 .............. MFE 1| \
27 .((((........)))). MFE 1| \
1 ,((((((.((((((,&,))))).......((((........))))...))))))), MFE 1| \
53 .(((((...(((((((((.....)))))...)))...))))). MFE 1
```

After a few weeks of computation, when we decided to start the filtering and candidate selection process, `RNAiFold2T` had returned 207,585 sequences.

4.5.3.2 Filtering and selection of candidates

In order narrow down the number of sequences generated by `RNAiFold2T` to a list of 9 candidates for biochemical validation, a filtering and selection process was applied based on measures designed to estimate different features: likelihood that a conformational switch is triggered by the addition of theophylline; concentration of MFE energy structure and active conformation in the presence or absence of theophylline; probability of the cleavage reaction to occur in the absence of theophylline; and agreement in the predictions of the most used RNA folding methods (Figure 4.8).

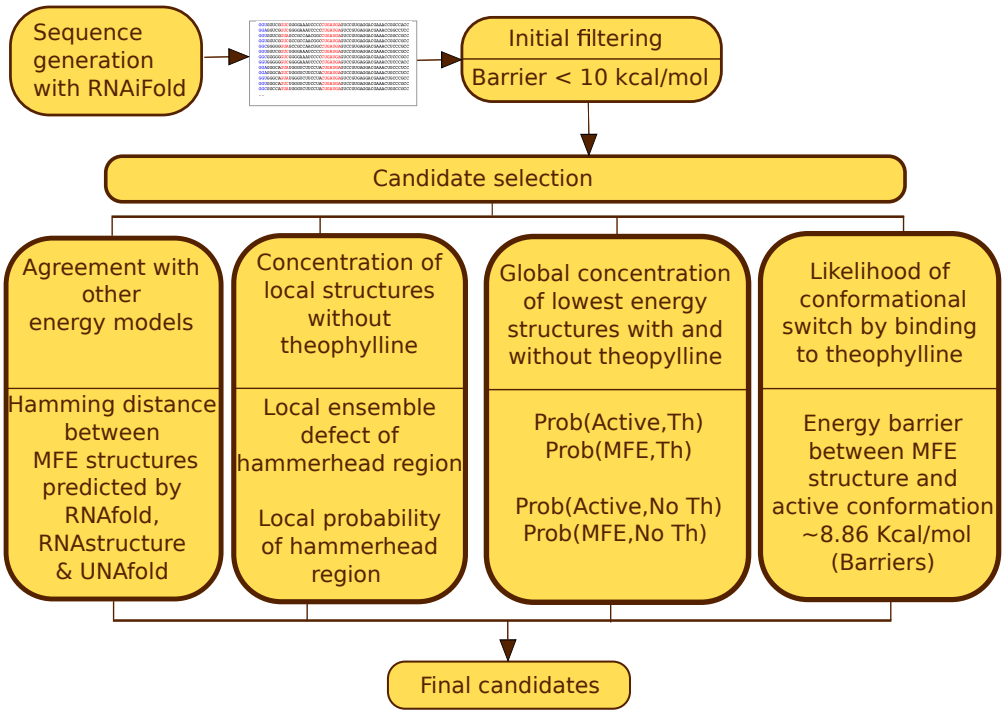


FIGURE 4.8: Computational design pipeline: This diagram depicts the full process of sequence generation, filtering and candidate selection. The order of importance of the selection criteria is from right to left.

Energy barrier: When an RNA molecule changes its conformation into a second metastable structure, the molecule has to adopt energetically unfavorable intermediate conformations that represent a thermodynamic energy barrier. The 'height' of this thermodynamic energy barrier is the difference between the free energy of the MFE structure and the free energy of the most energetically unfavorable structure adopted in the process. If this barrier is too high the conformational change is very unlikely to occur. Knowing that the experimentally estimated free energy contribution of theophylline bound to the aptamer used in the design is $\sim -8.86_{kcal/mol}$ [152], an initial filtering process was applied to discard those sequences whose energy barrier between the MFE structure and the active conformation is higher than $10_{kcal/mol}$.

`RNAsubopt` [78] program from `Vienna RNA Package` was used to compute all structures

whose energy lies within $10_{kcal/mol}$ of the minimum free energy. The list of suboptimal structures was used as input for `BARRIERS` 1.5.2 [167] to estimate the energy barrier between the MFE structure and any other suboptimal structure containing both the theophylline aptamer substructure and the hammerhead ribozyme hybridized with the substrate in a conformation where cleavage is likely to occur. Ideally, the energy barrier should be computed using two RNA molecules, however `BARRIERS` does not support hybridization. Therefore, the two sequences were concatenated using five 'A' nucleotides and the corresponding secondary structures were also concatenated using five unpaired positions. To ensure that the addition of the polyA fragment does not drastically affect the estimations, an additional test was performed comparing the difference between minimum free energies obtained from `RNAcofold` [168] for hybridized dimers and `RNAfold` [39] for sequences with the extra polyA fragment. This test showed that for all the sequences the difference is relatively low, with mean of $-0.96_{kcal/mol}$ and standard deviation of $1.04_{kcal/mol}$. The filtering process narrowed down the list of possible candidates to 2,106, each having an energy barrier below $10_{kcal/mol}$.

Energy barrier was also one of the four criteria in the final selection of candidates, whose energy barrier is close to $\sim 8.86_{kcal/mol}$.

Probability of a structure in the presence of theophylline: For a given sequence a, the *partition function* Z is the sum of the Boltzmann factors of the Turner energies of all the secondary structures compatibles with a; i.e. $Z = \sum_{s \cong a} exp(-E(s)/RT)$ where T is absolute temperature and R is the universal gas constant. The Boltzmann probability of a given secondary structure S is $P(S) = exp(-E(S)/RT)/Z$.

The partition function describes the statistical properties of the system in thermodynamic equilibrium. When theophylline is present, the thermodynamic equilibrium changes because the

thermodynamic ensemble also includes those molecules bound to theophylline, whose free energy is lower than other RNA molecules in the same conformation but not bound to the ligand. In order to calculate the probability of a structure we have to compute the partition function of this state, which is equal to $Z + Z_{theo}$, where Z_{theo} is the sum of the Boltzman factors of the Turner energies of those structures capable of binding to theophylline.

Given the estimated energy contribution of theophylline $E_{theo} = -8.86_{kcal/mol}$ [152], the free energy a sequence a adopting a specific secondary structure s and bound to theophylline is $E_{(a,s)} + E_{theo}$. In our computations we made the assumption that only those secondary structures where the theophylline aptamer adopts the conformation depicted in Figure 4.7c are subject to bind to the ligand, independently of the secondary structure into which the remaining part of the RNA molecule is folded.

For a sequence $\mathbf{a}$, let $\mathbf{a}(HHC)$ denote nucleotides 1-53 comprising the substrate (S) and hammerhead (HH), which together form the sequence fragment excluding the theophylline aptamer, let $\mathbf{a}(TA)$ denote nucleotides 54-95 comprising the fragment corresponding to the aptamer, and denote TA the structure of the aptamer depicted in Figure 4.7c. Then the partition function for the theophylline-bound complex Z_{theo} is the product of the partition function of $\mathbf{a}(HHC)$ ($Z_{\mathbf{a}(HHC)} = \sum_{s \in \mathbf{a}(HHC)} exp(-E_{(\mathbf{a}(HHC),s)}/RT)$), and the Boltzmann factor of $\mathbf{a}(TA)$ folded into the theophylline-bound aptamer structure TA ($exp(-(E_{(\mathbf{a}(TA),TA)})/RT)$) plus the Boltzmann factor of the theophylline energy contribution $exp(-(E_{theo}/RT)$ using the known binding free energy $E_{theo} = -8.6_{kcal/mol}$ of theophylline.

$$Z_{theo} = exp(-(E_{theo} + E_{(\mathbf{a}(TA),TA)}))/RT) \cdot Z_{\mathbf{a}(HHC)}.$$

Once $Z + Z_{theo}$ and Z for each sequence are computed, we can calculate the Boltzmann probabilities of the MFE structure ($P(MFE)$) and the active conformation ($P(Active)$) in the presence or absence of theophylline. Candidates selected for validation have low $P(MFE)$ and high $P(Active)$ when theophylline is present, and high $P(MFE)$ and low $P(Active)$ when theophylline is not present.

$$\begin{array}{lll} & Without\ Theophylline & With\ Theophylline \\ P(MFE) & \frac{e^{(\frac{-E_{MFE}}{RT})}}{Z} & \frac{e^{(\frac{-(E_{MFE}+E_{theo})}{RT}}}{Z+Z_{theo}} \\ P(Active) & \frac{e^{(\frac{-E_{active}}{RT})}}{Z} & \frac{e^{(\frac{-(E_{active}+E_{theo})}{RT}})}{Z+Z_{theo}} \end{array}$$

Agreement between energy models and RNA structure prediction algorithms: Different methods do indeed use different algorithms to compute base pairing probabilities; however the MFE structure is computed by all software using the same Zuker algorithm, though energy parameters, inclusion or not of coaxial stacking, treatment of dangles, etc. may be different. Therefore, it is important to have a measure of the agreement between the predictions from different methods and energy models. The average Hamming distance between the minimum free structures predicted by `UNAFOLD` [10], `RNAstructure` [157], `RNAfold` using Turner '99 and `RNAfold` using Turner '04 was used to score the agreement between different methods, where lower Hamming distance indicates higher similarity between the different predictions.

Estimation of cleavage activity in absence of theophylline: Another approach for estimating putative catalytic activity is to focus the analysis only on `HHC` and to measure the structural similarity between this region and the corresponding cleavage structure (Figure 4.7b). To

this end, we used a variation of *ensemble defect* [21], which is analogous to the average Hamming distance between a given target structure and the structures in the low energy Boltzmann ensemble of structures for a given RNA sequence (see Appendix A). Denote the base pairing probability of any two distinct positions $1 \leq k,l \leq n$ in a sequence a of length n be $p_{(k,l)}$, also denote $p^*_{(k,l)}$ the symmetrized base pairing probability, where $p^*_{(k,l)} = p_{(k,l)}$ if $k < l$, and $p^*_{(k,l)} = p_{(l,k)}$ if $l < k$, and denote q^*_k the probability that position k is unpaired, $q^*_k = 1 - \sum_{j>i} p_{i,j} - \sum_{j<i} p_{j,i}$. For a target substructure $S_{[i,j]}$, the *local ensemble defect* of the sequence a between positions i and j is defined as

$$localED(a,S_{[i,j]}) = (j - i + 1) - \sum_{i \leq k,l \leq j} p^*_{(k,l)} \cdot I[(k,l) \in S_{(i,j)}] - \sum_{i \leq k \leq j} q^*_k \cdot [k \; unpaired \; in \; S_{(i,j)}]$$

The value of $localED(a,S_{[i,j]})$ ranges between 0 and $j - i + 1$, where a high value indicates that there is a high average distance between the target structure $S_{[i,j]}$ and the region $[i,j]$ of the structures in the low energy Boltzmann ensemble of structures of the sequence a. In other words, the sequence a is unlikely to fold into structures whose region $[i,j]$ is similar to $S_{[i,j]}$.

For candidate selection, $localED(a,S_{[1,53]})$ was computed for each sequence a, where $S[1,53]$ is the secondary structure shown in Figure 4.7b. Then, only sequences with high *local ensemble defect* value were considered for selection, since this is an indicator that the `HHC` region a sequence is unlikely to fold into the structure depicted in Figure 4.7b, and therefore the sequence is expected to have low cleavage activity in absence of theophylline. All the selected candidates have a *localED* over 40, which corresponds to a length-normalized value of 0.75.

Sequences of the candidates selected for validation are included in Table 4.6 and the criteria for selection in Table 4.7.

ID	Sequence
RR1	GGUCGGUAAGAAUA
	GGUUCUCUGAUGAGCGCUUUGUCGUGUGCGAAACCGACAAAGUGAUACCAGCAUCGUCUUGAUGCCCUUGGCAGCACUUCA
RR2	GGCGGGUAAUAACG
	GGUUAUCUGAAGAGCACUUUGUGGCGUGCGAAACCCGCAAAGUGAUACCAGCAUCGUCUUGAUGCCCUUGGCAGCACUUCA
RR3	GGGGGGUAAGAGUA
	GGCUCUCUGAUGAGCACUUUGGAGUGUGCGAAACCUCCAAAGUGAUACCAGCAUCGUCUUGAUGCCCUUGGCAGCACUUCA
RR4	GGUCGGUCGAAACA
	GGUUUCCUGAAGAGCAUUUUGUCGCGUGCGAAACCGGCAAAGUGAUACCAGCAUCGUCUUGAUGCCCUUGGCAGCACUUCA
RR5	GGAUGGUAAUGAUG
	GGUCAUCUGAAGAGCACUUUGGCGUGUGCGAAACCGUUAAAGUGAUACCAGCAUCGUCUUGAUGCCCUUGGCAGCACUUCA
RR6	GGGAGGUAUAAGUA
	GGCUUACUGAUGAGUACUUUGGAGUGUGCGAAACCUCCAAAGUGAUACCAGCAUCGUCUUGAUGCCCUUGGCAGCACUUCA
RR7	GGUCGGUAAGUCUA
	GGGACUCUGAUGAGUACUUUGUCGUGUGCGAAACCGACAAAGUGAUACCAGCAUCGUCUUGAUGCCCUUGGCAGCACUUCA
RR8	GGUGGGUAUAAGUA
	GGCUUACUGAUGAGCACUUUAUGGUGUGCGAAACCCAUAAAGUGAUACCAGCAUCGUCUUGAUGCCCUUGGCAGCACUUCA
RR9	GGUCGGUAAGUUCG
	GGGACUCUGAAGAGCACUUUAUCGCGUGCGAAACCGAUAAAGUGAUACCAGCAUCGUCUUGAUGCCCUUGGCAGCACUUCA
NEG	GGGGUGUCUGCCCG
	GGGGCACUGACGAGUGCUUAGGGCGGCGCGAAACGCCCUAAGUGAUACCAGCAUCGUCUUGAUGCCCUUGGCAGCACUUCA

TABLE 4.6: Sequence of *molecular scissors* candidates. For each candidate and the negative control the first sequence corresponds to the substrate and the second to the putative riboswitch-ribozyme.

ID	Selection criteria
RR1	Low HD, High $P(MFE,\overline{T})$
RR2	High $P(MFE,\overline{T})$, High $P(active,T)$
RR3	Lowest HD, High $P(active,T)$
RR4	High $P(active,T)$, Low HD
RR5	Low B, High ED, Low HD
RR6	Lowest B
RR7	High $P(MFE,\overline{T})$, B close to $8.86_{kcal/mol}$, High $P(active,T)$, Low HD
RR8	Low B, Low HD, High $P(active,T)$
RR9	High $P(active,T)$, High $P(MFE,\overline{T})$
NEG	NEGATIVE CONTROL

TABLE 4.7: Selection criteria used for *molecular scissor* candidates: Criteria are ordered by their importance in the choice of each candidate. Features involved in the selection are: HD – Hamming distance between MFE structures predicted by different software (`RNAfold`, `UNAFOLD`, `RNAstructure`) and energy models; $P(MFE,\overline{T})$ – Probability of the MFE structure without theophylline; $P(active,T)$ – Probability of the cleavage conformation in the presence of theophylline; B – Energy barrier between the MFE structure and the active conformation; ED – Energy difference between the MFE structure and the active structure in the presence of theophylline.

4.5.3.3 Negative control

A negative control was designed to have the same properties as the molecular scissor candidates, except that it should not change conformation upon binding to theophylline– i.e. the negative control should have low or no cleavage activity under any conditions. To generate candidate sequences, we ran `RNAiFold2T` with the same input file we used for the design of the molecular scissor candidates, except that local structural constraints depicted in Figure 4.7 were removed. The resulting input file to `RNAiFold2T` is the following.

```
> Negative control
#RNAscdstr
,((((((.((((((,&,))))).......((((........))))...))))))..((((((...(((((((((.....)))))...)))...))))))..| \
..............&,,,,,,,,,,,,,,,(((((((((,,,,,,,,,,)))))))))...,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
#RNAseqcon
GGNNNGUHNNNNNNN&GGNNNNCUGANGAGNNNNNNNNNNNNNNNCGAAACNNNNNAAGUGAUACCAGCAUCGUCUUGAUGCCCUUGGCAGCACUUCA
#MFEstructure
0
#LocalCstrs
1 ,,,,,,,,,,,,,,,&,,,,,,,,,,,,,,,(((((((((,,,,,,,,,,)))))))))...,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,, MFE 1| \
15 ,,,,,,,,,,,,,,,(((((((((,,,,,,,,,,)))))))))...,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,, MFE 1
```

Therefore, the MFE of each solution returned must agree with the inactive conformation and, although solutions must be compatible with the active conformation, there are no specific structural requirements for the `HHC`, `TA` and `S` fragments. A total of 24,720,773 sequences were generated by `RNAiFold2T`. Then, different criteria were applied for filtering and selection in order to find a sequence predicted not to change its conformation in the presence of theophylline– i.e. (1) to have a large free energy difference between the MFE structure and the active conformation, and (2) to have low probability for the active conformation, even when the energy contribution of theophylline is included (probability approximately 0); and (3) to have a high probability of the inactive MFE structure.

For the selected negative control, the free energy difference between MFE and active structures is $17_{kcal/mol}$, approximately $8_{kcal/mol}$ higher than the estimated free energy contribution of theophylline. Moreover, for the selected negative control, probability of the active conformation is ~ 0, and probability of the inactive MFE structure is ~ 0.25.

At the time of publication of this dissertation, the cleavage activity of the nine RNA *molecular scissor* candidates and the negative control shown in Table 4.6 is being measured under different concentrations of theophylline and caffeine.

4.6 Conclusion

In this chapter, we present the software `RNAiFold2T` for the multi-temperature inverse folding problem, used to design functional thermoswitches. `RNAiFold2T` solves the k-temperature inverse folding problem for any $k \geq 2$. Most practical applications will concern 2-temperature inverse folding. There is less interest in k-temperature inverse folding for > 2, for the following reason. Although there always exists an RNA sequence compatible with any two given secondary structures, for three or more structures, it is possible that no sequence is compatible with all given structures[169].

`RNAiFold2T` is modular software, with a clear separation between search procedure and constraint descriptions, thus permitting the future addition of sequence and structural constraints. In its current form, `RNAiFold2T` includes constraints for *full* and/or *partial* target structures or hybridization complexes at two temperatures; a plug-in to use `RNAfold` or `RNAstructure` for MFE structure computation; IUPAC nucleotide constraints, IUPAC amino acid constraints

that require all returned RNA sequences to code specified peptides in one or more overlapping reading frames, structural compatibility and structural incompatibility constraints, *local structural constraints*, etc. These constraints support the design of temperature-sensitive selenocysteine insertion (SECIS) elements, precursor microRNAs, and mRNA domains that are targeted by microRNAs, etc. Since *Constraint Programming* (*CP*) is not a heuristic, unlike other methods such as adaptive walk, genetic algorithm, etc., `RNAiFold2T` can in principle return all 2-temperature inverse folding solutions, or prove that none exist.

The *Large Neighborhood Search* (*LNS*) algorithm of `RNAiFold2T` returns a single solution with approximately the same performance as state-of-the-art approaches `SD` and `FRNA`, while the *CP* variant of `RNAiFold2T` returns two orders of magnitude more solutions than other software. The software design of `RNAiFold2T` currently supports a much greater variety of user-defined structural and sequence constraints than other methods, and moreover can be extended to support future constraints.

`SwitchDesign` (`SD`), the first algorithm capable of designing RNA *thermoswitches*, achieves this by optimizing the cost function given in equation 4.1[145] – see Appendix D. In fact, the results of our benchmarks show that this is a good approach to solve the two temperature inverse folding problem. Surprisingly, our results shown in Figure 4.3 and Figure D.1 in Appendix D show that natural thermosensor sequences from `Rfam` appear *not* to be optimized for the cost function used in `SD`. In particular, `SD` and `FRNA` return solutions having substantially lower cost values (i.e. more optimal) than those of natural thermosensors, whose cost value appears to be the mean value returned by `RNAiFold2T`.

As with the synthetic hammerhead design in Chapter 3, our synthetic RNA design strategy

consists of generating many solutions, which are prioritized for experimental validation by applying various computational filters. Using this strategy we designed functional *thermo*-IRES, whose the cap-independent translational efficiency is approximately 50% higher at 42°C than at 30°C.

In addition, we generated promising RNA *molecular scissor* candidates by using the novel *local structural constraints* included in `RNAiFold2T` to generate hundreds of thousands of sequences. We subsequently filtered according to four different criteria: (1) likelihood that a conformational switch is triggered by the addition of theophylline; (2) probability of the MFE energy structure and the active conformation both in the presence and absence of theophylline; (3) probability of the cleavage reaction occurring in the absence of theophylline; and (4) agreement between the predictions of three RNA folding software packages.

In summary, the results presented in this chapter illustrate the excellent performance of `RNAiFold2T` in solving the 2-temperature inverse folding problem, and its versatility in designing RNA *thermoswitches* and RNA *switches*. This chapter also demonstrates the potential of our synthetic RNA design strategy, which is based on the generation of many solutions with subsequent prioritization for experimental validation by applying various computational filters.

Chapter 5

RNA Thermodynamic Structural Entropy

5.1 Introduction

Conformational entropy for atomic-level, three dimensional biomolecules is known experimentally to play an important role in protein-ligand discrimination, yet reliable computation of entropy remains a difficult problem. Here we describe the first two accurate and efficient algorithms to compute the conformational entropy for RNA secondary structures, with respect to the Turner energy model, where free energy parameters are determined from UV aborption experiments. An algorithm to compute the derivational entropy for RNA secondary structures had previously been introduced, using *stochastic context free grammars* (SCFGs). However, the numerical value of derivational entropy depends heavily on the chosen context free grammar and on the training set used to estimate rule probabilities. Using data from the `Rfam` database, we determine that both of our thermodynamic methods, which agree

in numerical value, are substantially faster than the SCFG method. Thermodynamic structural entropy is much smaller than derivational entropy, and the correlation between length-normalized thermodynamic entropy and derivational entropy is moderately weak to poor. In applications, we plot the structural entropy as a function of temperature for known RNA *thermoswitches*, such as the repression of heat shock gene expression (ROSE) element, we determine that the correlation between hammerhead ribozyme cleavage activity and total free energy is improved by including an additional free energy term arising from conformational entropy, and we plot the structural entropy of windows of the HIV-1 genome. Our software `RNAentropy` can compute structural entropy for any user-specified temperature, and supports both the Turner'99 and Turner'04 energy parameters. It follows that `RNAentropy` is state-of-the-art software to compute RNA secondary structure conformational entropy. Source code is available at `https://github.com/clotelab/RNAentropy/`; a full web server is available at `http://bioinformatics.bc.edu/clotelab/RNAentropy`, including source code and ancillary programs.

5.1.1 Organization

This chapter is organized in the following fashion. We start defining the different notions of entropy and their importance for the study of RNA molecules, along with the current methods available for computing them. Then we introduce two new dynamic programming algorithms to compute RNA structural entropy, followed by an analysis of sequences from `Rfam` comparing of structural entropy and derivational entropy. Finally, we show how incorporating structural entropy improves the correlation between hammerhead ribozyme cleavage activity

and free energy change, and how to use structural entropy for the detection of functional non coding RNAs in the HIV-I genome.

5.2 Background

Conformational (or configurational) entropy is defined by

$$S = -k_B \sum_s p(s) \ln p(s) \tag{5.1}$$

where k_B denotes the Boltzmann constant, and the sum is taken over all structures. As shown experimentally to be the case for calmodulin [170], conformational entropy plays an important role for the discrimination observed in protein-ligand binding. Since conformational entropy is well-known to be difficult to measure, this recent experimental advance involves using NMR relaxation as a proxy for entropy, a technique reviewed in [171].

It is currently not possible to reliably compute the conformational entropy for 3-dimensional molecular structures [171]; nevertheless, various methods have been developed, employing approaches from molecular, harmonic, and quasiharmonic dynamics [172, 173]. It appears likely that such computational methods will continue to improve, especially with the availability now of experimentally determined values by using NMR relaxation [171].

In contrast to the complex situation for 3-dimensional molecular structures, we show here that it is possible to accurately and efficiently compute the exact value of conformational entropy for RNA secondary structures, with respect to the Turner energy model [59], whose free energy parameters are experimentally determined from UV absorption experiments [174]. Our

resulting algorithm, `RNAentropy`, runs in cubic time with quadratic memory requirements, thus answering a question raised by M. Zuker (personal communication, 2009).

The *nearest neighbor* or *Turner* energy model is a coarse-grained RNA secondary structure model that includes free energy parameters for base stacking and various loops (hairpins, bulges, internal loops, multiloops) [59]. The exact definition of these loops can be found in the description of Zuker's algorithm [30] which computes the minimum free energy (MFE) secondary structure with respect to the Turner energy model. As explained in [174], values for base stacking enthalpy and entropy can be determined by plotting the experimentally measured UV absorption values of various double-stranded RNA oligonucleotide sequences at 280 nm (also 260 nm) as a function of RNA concentration. By least-squares fitting of the data, free energy parameters for base stacking, hairpins, bulges, etc. can be determined. Free energy and enthalpy parameters for an earlier model (Turner 1999) and a more recent model (Turner 2004) are described at the Nearest Neighbor Database (NNDB) [59]. For instance, the base stacking free energy for $\begin{matrix}5'\text{-GC-}3'\\3'\text{-CG-}5'\end{matrix}$ is −3.4 kcal/mol in the Turner 2004 parameter set. `mfold` [29], `UNAFOLD` [10] and the `Vienna RNA Package` [139] are software packages that implement the Zuker dynamic programming algorithm [30] to compute the MFE structure as well as the McCaskill algorithm [175] to compute the partition function over all secondary structures. Applications of such software are far-reaching, ranging from the prediction of microRNA target sites [176] to the design of synthetic RNA [124, 177].

Throughout this chapter, for a given RNA sequence $\mathbf{a} = a_1, \ldots, a_n$, *structural entropy*, denoted by $H(\mathbf{a})$, is defined to be (Shannon) entropy

$$H(\mathbf{a}) = -\sum_{s \in \mathbb{SS}(\mathbf{a})} p(s) \ln p(s) \tag{5.2}$$

where the sum is taken over all secondary structures s of $\mathbf{a}$, denoted by $\mathbb{SS}(\mathbf{a})$; $p(s)$ denotes the Boltzmann probability $\exp(-E(\mathbf{a},s)/RT)/Z(\mathbf{a})$, R denotes the universal gas constant (Boltzmann constant times Avagadro's number), $E(\mathbf{a},s)$ is the free energy of the secondary structure s of $\mathbf{a}$ with respect to the Turner energy model [59], and $Z(\mathbf{a})$ denotes the partition function, defined as the sum of all Boltzmann factors $\exp(-E(\mathbf{a},s)/RT)$ over all secondary structures s in $\mathbb{SS}(\mathbf{a})$. When the RNA sequence $\mathbf{a}$ is clear from the context, we generally write $E(s)$, H, $\mathbb{SS}$ and Z, rather than $E(\mathbf{a},s)$, $H(\mathbf{a})$, $\mathbb{SS}(\mathbf{a})$ and $Z(\mathbf{a})$. It follows that the conformational entropy is equal to the Boltzmann constant times the structural entropy: $S = k_B H$.

Before presenting our results and describing our algorithms, we first survey several distinct notions of entropy that have appeared in the literature of RNA secondary structures – each quite different from the notion of thermodynamic structural entropy described in this chapter.

5.2.1 Pointwise entropy in multiple alignments

Shannon entropy is used to quantify the variability of positions in a multiple sequence alignment. This application is particularly widespread due to the ubiquitous use of sequence logos [113, 178] to present motifs in proteins, DNA and RNA. Letting $\mathcal{N}$ denote the 4-letter alphabet $\{A,C,G,U\}$, the pointwise entropy $H_1(k)$ at position k in the alignment is defined by $H_1(k) = -\sum_{a\in\mathcal{N}} p_a \ln p_a$, where p_a is the proportion of nucleotide a at position k. Entropy values range from 0 to $\log 4$, where high entropy entails uncertainty or disagreement of the nucleotides at position k. Average pointwise sequence entropy is often expressed in bits, where logarithm base 2 is used instead of the natural logarithm. The concept of sequence logo has

many generalizations; indeed, logos for DNA major groove binding are described in [113], logos for tertiary structure alignment of proteins are described in [179], logos for RNA alignments including mutual information on base pair covariation are described in [180], and logos with secondary structure context of RNAs that bind to specific riboproteins are described in [181, 182].

5.2.2 Positional entropy

For a given RNA sequence $\mathbf{a} = a_1, \ldots, a_n$, and for $1 \leq i < j \leq n$, define the base pairing probability $p_{i,j}$ to be the sum of Boltzmann factors of all secondary structures that contain base pair (i,j), divided by the partition function, i.e.

$$p_{i,j} = \sum_{\{s \in \mathbb{SS} : (i,j) \in s\}} p(s) = \frac{\sum_{\{s \in \mathbb{SS} : (i,j) \in s\}} \exp(-E(s)/RT)}{Z} \tag{5.3}$$

Here $p(s)$ is the Boltzmann probability of structure s of $\mathbf{a}$, $E(s)$ is the Turner free energy of secondary structure s [59], $R \approx 0.001987_{kcal/(mol \cdot K)}$ is the universal gas constant, T is absolute temperature, and the *partition function* $Z = \sum_{s \in \mathbb{SS}} \exp(-E(s)/RT)$, where the sum is taken over all secondary structures s of $\mathbf{a}$. Base pairing probabilities can be computed in cubic time by McCaskill's algorithm [175], as implemented in various software, including the `Vienna RNA Package` `RNAfold -p` [139].

Define the positional base pairing probability distribution at fixed position $1 \leq i \leq n$ by

$$p^*_{i,j} = \begin{cases} p_{i,j} & \text{if } i < j \\ p_{j,i} & \text{if } i > j \\ 1 - \sum_{j \neq i} p^*_{i,j} & \text{if } i = j \end{cases} \tag{5.4}$$

For each fixed value of i, $p^*_{i,j}$ is a probability distribution, where j ranges over $1,\ldots,n$, the structural *positional entropy* $H_2(i)$ at position i is defined by

$$H_2(i) = -\sum_{j=1}^{n} p^*_{i,j} \ln p^*_{i,j}. \tag{5.5}$$

Low values of *positional entropy* at position i indicate that there is a strong agreement among low energy structures in the Boltzmann ensemble that either i is unpaired, or that i is paired with the same position j. The *average* positional entropy $\langle H_2 \rangle$ is the average $\sum_{i=1}^{n} \frac{H_2(i)}{n}$ taken over all positions of the sequence. Structural *positional entropy* was first defined by Huynen et al. [115], who used the term S-value for average *positional entropy*, and showed that RNA nucleotide positions having low entropy correspond to positions where the minimum free energy (MFE) structure tends to agree with that determined by comparative sequence analysis. In [183], Mathews made a similar analysis, where in place of S-value, a normalized pseudo-entropy value was used, defined by $-\sum_{1 \le i < j \le n} p_{i,j} \ln p_{i,j}/n$. *Positional entropy* of RNA secondary structures can be presented by color-coding each nucleotide, where the color of the kth nucleotide reflects the *positional entropy* $H_2(k)$ as defined in equation (5.5). The Rfam 12.0 database [65] uses such color-coded secondary structures, since the base-pairing of positions having low entropy is likely to be correct [115, 183].

5.2.3 Derivational entropy using stochastic context free grammars

Manzourolajdad et al. [184], Sukosd et al. [185] and Anderson et al. [186] describe the computation of structural entropy for stochastic context free grammars (SCFGs), defined by $-\sum_{s \in \mathbb{SS}} p(s) \ln p(s)$, where the sum is taken over all secondary structures s of a given RNA sequence, and $p(s)$ is the probability of deriving the structure s in a particular grammar G, defined as follows.

Suppose that $S = S_0$ is the starting nonterminal for the grammar G, $s = S_m$ is the secondary structure s consisting only of terminal symbols belonging to the alphabet $\{$ `(`, `)`,•$\}$, and that $S_1, \ldots, S_{m-1}$ are expressions consisting of a mix of nonterminal and terminal symbols. If $S_0 \rightarrow_G S_1 \rightarrow_G S_2 \rightarrow_G \cdots \rightarrow_G S_m$ is a leftmost derivation using production rules from grammar G and for each $i = 0, \ldots, m-1$, we let p_i denote the probability of applying the rule $S_i \rightarrow S_{i+1}$, then $p(s)$ is defined to be the product $\prod_{i=0}^{m-1} p_i$. It should be noted that the derivational probability $p(s)$ heavily depends on the choice of grammar G as well as on the rule application probabilities p_i, obtained by applying expectation maximization to a chosen training set of secondary structures.

Anderson et al. [186] are motivated to compute derivational entropy of a multiple alignment of RNAs, in order to provide a numerical quantification for the quality of the alignment – specifically, their paper shows that accurate alignment quality corresponds to low derivational entropy. In [187], Sukosd et al. describe the software `PPfold`, a multithreaded version of the `PFold` RNA secondary structure prediction algorithm. Subsequently, Sukosd et al. [185] describe how to compute the derivational entropy for the grammar used in the `PFold` algorithm (grammar G6 as defined in [188]), and show that derivational entropy is correlated with the accuracy of `PPfold` structure predictions, as measured by F-scores. In contrast, Manzourolajdad et al. [184] computed the derivational entropy of various families of noncoding RNAs, using the trained stochastic context free grammars G4,G5,G6 [188], which they denote respectively as RUN (G4), IVO (G5) and BJK (G6). The Linux executable and trained models can be downloaded from `http://rna-informatics.uga.edu/malmberg/` for three RNA stochastic context free grammars, each with three trained models using the training sets 'Rfam5', 'Mixed80', and 'Benchmark' – see [184] for description.

5.3 Algorithm description

In this section, we describe the two novel algorithms to compute RNA thermodynamic structural entropy using the Turner energy model [59]. Section 5.3.1 describes the relation between entropy and expected energy, and provides two variants of a simple sampling method to approximate the value of structural entropy. The approximation does not yield accurate entropy values, so two accurate methods are described: (1) *formal temperature derivative* (FTD) method, (2) dynamic programming (DP) method. An overview of both algorithms is provided in this section. Full details of each algorithm are then provided in Sections 5.3.4 and 5.3.5.

5.3.1 Statistical mechanics

Shannon entropy for the Boltzmann ensemble of secondary structures of a given RNA sequence $\mathbf{a} = a_1, \ldots, a_n$ is defined by

$$\begin{aligned} H(\mathbf{a}) &= -\sum_{s \in \mathbb{SS}} p(s) \ln p(s) = -\sum_{s \in \mathbb{SS}} \frac{\exp(-E(s)/RT)}{Z} \ln\left(\frac{\exp(-E(s)/RT)}{Z}\right) \\ &= -\sum_{s \in \mathbb{SS}} \frac{\exp(-E(s)/RT)}{Z} \cdot \left[-\frac{E(s)}{RT} - \ln Z\right] \\ &= \frac{1}{RT} \sum_{s \in \mathbb{SS}} p(s) E(s) + \frac{\ln Z}{Z} \cdot \sum_{s \in \mathbb{SS}} \exp(-E(s)/RT) \\ &= \frac{\langle E \rangle}{RT} + \ln Z = \frac{\langle E \rangle - G}{RT} \end{aligned} \tag{5.6}$$

where G denotes the ensemble free energy $-RT \ln Z$. It follows that if the energy $E(s)$ of every structure s is zero, or if the temperature T is infinite, then entropy is equal to the logarithm of the number of structures. Note as well that in the Nussinov energy model [189], where each base pair has an energy of -1, it follows that the expected energy is equal to -1 times the

expected number of base pairs, i.e. $\langle E \rangle = -\sum_{i<j} p_{i,j}$, where $p_{i,j}$ is the probability of base pair (i,j) in the Nussinov model.

By sampling RNA structures with the `RNAsubopt` program from `Vienna RNA Package` [139], we can approximate the value of expected energy, and hence obtain an approximation of the thermodynamic entropy by using equation (5.6). This can be done in two distinct manners.

In the first approach, a user-specified number `N` of low energy structures from the thermodynamic ensemble can be sampled by using the algorithm of Ding and Lawrence [190], as implemented in `RNAsubopt -p N`. A sampling approximation for the expected energy is then defined to be the arithmetic average of the free energy of the `N` sampled structures. In the second approach, all structures can be generated, whose free energy lies within a user-specified range `E` of the minimum free energy, by using the algorithm of Wuchty [78], as implemented in `RNAsubopt -e E`. Let Z_0 be an approximation of the partition function, defined by summing the Boltzmann factors $\exp(-E(s)/RT)$ for all generated structures. Define the (approximate) Boltzmann probability of a generated structure s to be $p(s) = \exp(-E(s)/RT)/Z_0$. An approximation for the expected energy is in this case taken to be $\sum_{s \in \mathbb{SS}} p(s) \cdot E(s)$, where the sum is taken over all structures s, whose free energy is within `E` kcal/mol of the minimum free energy. In either case, the resulting entropy approximation is not particularly good. For instance, the thermodynamic entropy of the 78 nt arginyl-tRNA from *Aeropyrum pernix* (accession code tdbR00000589 in the *Transfer RNA database* tRNAdb [62]) is 5.44, as computed by the algorithm `RNAentropy` described in this chapter, while the entropy approximation by the first sampling approach with $N = 10{,}000$ is 4.71 and that of the second sampling approach with $E = 10$ is 4.68. Since the estimate from each sampling approach has greater than 13% relative error, sampling cannot be used to provide accurate entropy values. For that reason, we now briefly describe

two novel, cubic time algorithms to compute the exact value of structural entropy– details of the algorithms are further described in Sections 5.3.4 and 5.3.5.

5.3.2 Algorithm 1: Formal temperature derivative (FTD)

It is well-known from statistical physics that the average energy $\langle E \rangle$ of N independent and distinguishable particles is given by the following formula (cf equation (10.36) of [191]):

$$\langle E \rangle \quad = \quad RT^2 \cdot \frac{\partial}{\partial T} \ln Z(T). \tag{5.7}$$

This equation does not hold in the case of RNA secondary structures with the Turner energy model; however, equation (5.7) is close to being correct. The idea of Algorithm 1 is to use finite differences $\frac{\ln Z(T+\Delta T)-\ln Z(T)}{\Delta T}$ to approximate the derivative $\frac{\partial}{\partial T} \ln Z(T)$, thus obtaining the expected energy $\langle E \rangle$, from which we obtain the structural entropy by applying equation (5.6). As shown later, certain technically subtle issues arise in this approach; in particular, the derivative $\frac{\partial}{\partial T} \ln Z(T)$ must be taken with respect to the *formal temperature*, which represents only those occurrences of the temperature variable within the expression RT. Formal temperature is distinct from *table temperature*, which latter designates all occurrences of the temperature variable in the Turner energy parameters. This will be fully explained in Section 5.3.4. For this reason, Algorithm 1 is named FTD, for *formal temperature derivative*.

5.3.3 Algorithm 2: Dynamic Programming (DP)

Recall that the partition function for a given RNA sequence **a** is defined by $Z = \sum_{s \in \mathbb{SS}} \exp(-E(s)/RT)$, where the sum is taken over all secondary structures of **a**. Letting $BF(s) = \exp(-E(s)/RT)$ denote the Boltzmann factor of s, it follows that the Boltzmann probability of secondary structure s satisfies $p(s) = BF(s)/Z$, and hence

$$\langle E \rangle = \sum_{s \in \mathbb{SS}} p(s) \cdot E(s) = \sum_{s \in \mathbb{SS}} \frac{BF(s) \cdot E(s)}{Z} = \frac{Q}{Z} \tag{5.8}$$

where $Q = \sum_{s \in \mathbb{SS}} BF(s) \cdot E(s)$. The partition function Z can be computed by McCaskill's algorithm [175], while in Section 5.3.5, we describe a dynamic programming algorithm to compute $Q(a)$. Since this method uses dynamic programming, Algorithm 2 is named DP.

Both FTD and DP support the Turner'99 and Turner'04 energy models [59], and all references to FTD and DP mean FTD'04 and DP'04, unless otherwise stated (there are small numerical differences in the entropy, depending on the choice of Turner parameters). Moreover, both algorithms allow the user to specify an arbitrary temperature T for the computation of structural entropy. This latter feature could prove useful in the investigation of thermoswitches, also called RNA thermometers, discussed later. The software `RNAentropy` implements both algorithms, and is available at `http://bioinformatics.bc.edu/clotelab/RNAentropy`.

5.3.4 Entropy by statistical physics

Here we show that for the Turner energy model of RNA secondary structures, expected energy satisfies

$$\langle E \rangle \approx RT^2 \cdot \frac{\partial}{\partial T} \ln Z(T) \tag{5.9}$$

although equality does not strictly hold. Indeed,

$$\begin{aligned} RT^2 \cdot \frac{\partial}{\partial T} \ln Z(T) &= \frac{RT^2}{Z(T)} \cdot \frac{\partial}{\partial T} Z(T) = \frac{RT^2}{Z(T)} \sum_{s \in \mathbb{SS}(\mathbf{a})} \frac{\partial}{\partial T} \exp(-E(s)/RT) \\ &= \frac{RT^2}{Z(T)} \sum_{s \in \mathbb{SS}(\mathbf{a})} \left\{ \frac{E(s)}{RT^2} - \frac{1}{RT} \cdot \frac{\partial}{\partial T} E(s) \right\} \cdot \exp(-E(s)/RT) \\ &= \sum_{s \in \mathbb{SS}(\mathbf{a})} E(s) \cdot \frac{\exp(-E(s)/RT)}{Z(T)} - \\ &\quad T \sum_{s \in \mathbb{SS}(\mathbf{a})} \frac{\exp(-E(s)/RT)}{Z(T)} \cdot \frac{\partial}{\partial T} E(s) \\ &= \langle E \rangle - T \cdot \langle \frac{\partial}{\partial T} E \rangle \end{aligned} \tag{5.10}$$

Let *formal temperature* denote each occurrence of the temperature variable T within the expression RT, while *table temperature* denotes all other occurrences (i.e. table temperature refers to the temperature-dependent Turner free energy parameters [59]). This will shortly be explained in greater detail. From equation (5.10), it follows that expected energy $\langle E \rangle$ is equal to RT^2 times the derivative of $\ln Z(T)$ with respect to *formal temperature*, which later we define to be the *formal temperature derivative* of $\ln Z(T)$.

If we treat the energy $E(s)$ of structure s as a constant (computed at either the default temperature of 37° C, or at a user-specified temperature T), then the second term of equation (5.12) disappears, and we can approximate $RT^2 \cdot \frac{\partial}{\partial T} \ln Z(T)$ by the finite difference $RT^2 \cdot \frac{\ln Z(T+\Delta T) - \ln Z(T)}{\Delta T}$, where for instance $\Delta T = 10^{-7}$. This requires a modification of McCaskill's algorithm [175] for the partition function $Z(T)$, where we distinguish between *formal temperature* and *table temperature*. Our software `RNAentropy` implements such a modification, and thus supports the *formal temperature derivative* (FTD) method of computing thermodynamic structural entropy.

Note that the function $\ln Z(T)$ is decreasing and concave down, so barring numerical precision errors, the finite difference $\frac{\ln Z(T+\Delta T) - \ln Z(T)}{\Delta T}$ is negative and slightly larger in absolute value

than the *formal temperature derivative* $\frac{\partial}{\partial T} \ln Z(T)$. From equation (5.6), structural entropy H is equal to $\langle E \rangle / RT + \ln Z$ and so there will be a small numerical deviation between the value of H, computed by the FTD (*formal temperature derivative*) method currently described, and the exact value of H computed by the DP (dynamic programming) method, described in Section 5.3.5. In particular, entropy values computed by FTD should be slightly smaller than those computed by DP, where the discrepancy will be visible only for large sequence length. This is indeed observed in Figure 5.1B and in data not shown.

We now show that the expression, $\langle \frac{\partial}{\partial T} E(s) \rangle$, occurring as the second term in the last line of equation (5.10), is equal to $-T \cdot \langle S_t \rangle$ where $\langle S_t \rangle$ denotes the expected change in entropy using the Turner parameters [59], determined as follows. From statistical physics, the free energy $E(s)$ of a secondary structure s satisfies

$$E(s) \quad = \quad H_t(s) - T \cdot S_t(s) \tag{5.11}$$

where $H_t(s)$ [resp. $S_t(s)$] denotes change in enthalpy [resp. entropy] from the empty structure to structure s using the Turner parameters. The term S_t measures the entropic loss due to stacked base pairs, hairpins, bulges, internal loops and multiloops using parameters obtained from least-squares fitting of UV absorption data. In the Turner energy model, entropy S_t and enthalpy H_t are assumed to be independent of temperature, so it follows from equation (5.11) that $\frac{\partial}{\partial T} E(s) = -S_t$, and hence

$$\langle E \rangle \quad = \quad RT^2 \frac{\partial}{\partial T} \ln Z(T) + T \cdot \langle S_t \rangle \tag{5.12}$$

To compute $S_t(s)$ for a given secondary structure s of an RNA sequence $\mathbf{a}$, determine the free energy $E(s,37)$ [resp. $E(s,38)$] of structure s at 37° C [resp. 38° C] by using `Vienna RNA Package` `RNAeval` [139]; it then follows from equation (5.11) that $S_t(s) = E(s,37) - E(s,38)$. Throughout

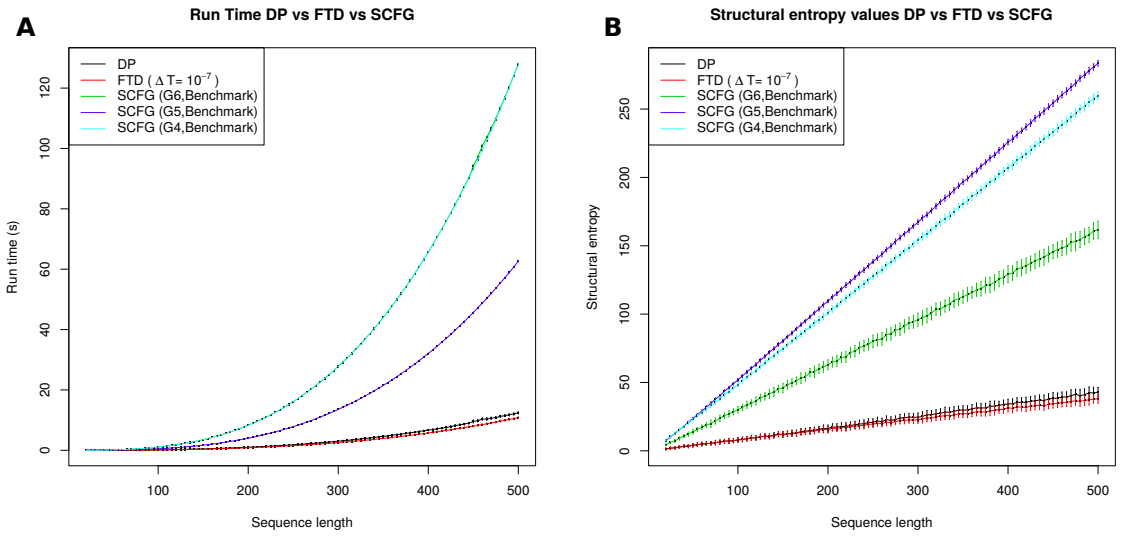


FIGURE 5.1: *(A)* Average run times, with (tiny) error-bars of ±1 standard deviation, for each of the five methods DP, FTD ($\Delta T = 10^{-7}$), SCFG (G6,Benchmark), SCFG (G4,Benchmark), and SCFG (G5,Benchmark). Averages were determined for 100 random RNA sequences of length n, each having expected compositional frequency of 0.25 for A,C,G,U, where n ranges from 20 to 500 with increments of 5. Methods tested are as follows: (1) DP: dynamic programming computation of expected energy $\langle E \rangle$ and partition function to yield $H = \langle E \rangle / RT + \ln Z$, with Turner 2004 energy parameters. (2) FTD: *formal temperature derivative* method which computes $\langle E \rangle \approx RT^2 \cdot \frac{\ln Z(T+\Delta T) - \ln Z(T)}{\Delta T}$, where the temperature increment $T + \Delta T$ is applied only to occurrences of T within the expression RT – i.e. *formal temperature*, as explained in the text. Increment ΔT is 10^{-7}, and Turner 2004 energy parameters are used. (3) SCFG: computation of derivational entropy using the method of [184], for the grammars G_4, G_5, G6 with grammar rule probabilities from 'Benchmark' data (see [184, 188]). SCFG executables and models downloaded from `http://rna-informatics.uga.edu/malmberg/`. The methods, ordered from fastest to slowest, are as follows: FTD, DP, G_5, G6, G_4, where FTD and DP are approximately equally fast, while the slowest methods, G6 and G_4, have almost identical run times. DP and FTD are an order of magnitude faster than G6. *(B)* Average entropy values, with error bars of ±1 standard deviation, computed by the methods DP, FTD ($\Delta T = 10^{-7}$), SCFG (G_4,Benchmark), SCFG (G_5,Benchmark), and SCFG (G6,Benchmark) for the same data set as in the left panel. The methods, ordered from those returning smallest entropy values to largest, are as follows: FTD, DP, G6, G_4, G_5. FTD and DP return essentially identical values, with a small deviation for larger sequences due to the finite approximation of the *formal temperature derivative*.

this paragraph, the reader should not confuse the notion of *conformational entropy* from equation(5.1), which is always non-zero and is computed by the novel algorithms described in this chapter, with the notion of *Turner change of entropy* $S_t(s)$ of secondary structure s, which is always negative due to entropic loss in going from the empty structure to a fixed structure s. Nor should the reader confuse the notion of *structural entropy*, denoted by H and defined in equation (5.2), with *Turner change of enthalpy* $H_t(s)$ of secondary structure s.

5.3.5 Entropy by dynamic programming

Throughout this section, $\mathbf{a} = a_1, \cdots, a_n$ denotes an arbitrary but fixed RNA sequence. Below, we give recursions for $Q(a)$, defined by $Q(a) = \sum_{s \in \mathbb{SS}} BF(s) \cdot E(s)$, where the sum is taken over all secondary structures s of RNA sequence $\mathbf{a}$, $E(s)$ is the free energy of s, using the Turner 2004 parameters, $BF(s) = \exp(-E(s)/RT)$ is the Boltzmann factor of structure s, where R is the universal gas constant and T the temperature in Kelvin.

Recursions are also given for the partition function $Z(a) = \sum_{s \in \mathbb{SS}} \exp(-E(s)/RT)$, where the sum is taken over all secondary structures of $\mathbf{a}$. It follows that the expected energy

$$\langle E \rangle = \sum_{s \in \mathbb{SS}} \frac{BF(s) \cdot E(s)}{Z} = \frac{Q(a)}{Z(a)} \tag{5.13}$$

For $1 \leq i \leq j \leq n$, the collection of all secondary structures of $\mathbf{a}[i,j] = a_i, \ldots, a_j$ is denoted $\mathbb{SS}[i,j]$. In contrast, if s is a secondary structure of $a_1, \ldots, a_n$, then $s[i,j]$ is the *restriction* of s to the interval $[i,j]$, defined by $s[i,j] = \{(x,y) : i \leq x \leq y \leq j, (x,y) \in s$.

5.3.6 Initial steps

For notational convenience, we define $Q_{i,i-1} = 0$ and $Z_{i,i-1} = 1$. If $i \leq j < i + 4$, then for any secondary structure s, the restriction $s[i,j]$ is the empty structure, denoted by $j - i + 1$ dots with zero energy, and so $Q_{i,j} = 0$. As well, the only secondary structure on $[i,j]$ is the empty structure, so $Z_{i,j} = 1$.

Now assume that $i + 4 \leq j$. Since

$$Q_{i,j} = \sum_{\substack{s \in \mathbb{SS}[i,j] \\ j \text{ unpaired in } s}} BF(s)E(s) + \sum_{k=i}^{j-4} \sum_{\substack{s \in \mathbb{SS}[i,j] \\ (k,j) \in s}} BF(s)E(s). \tag{5.14}$$

we treat each sum in a separate case. Let $bp(k,j)$ be a boolean valued function with the value 1 if k can base-pair with j; i.e. $a_k a_j \in \{AU,UA,CG,GC,GU,UG\}$. For secondary structure $s \in \mathbb{SS}[i,j]$, let $bp(k,j,s)$ be a boolean function with value 1 if it is possible to add the base pair (k,j) to s and obtain a valid secondary structure; i.e. without creating a base triple or pseudoknot.

Case 1: j is unpaired in $[i,j]$. For $s \in \mathbb{SS}[i,j]$ in which j is unpaired, $s = s[i,j-1]$, $BF(s) = BF(s[i,j-1])$, and $E(s) = E(s[i,j-1])$. The contribution to $Q_{i,j}$ in this case is given by $Q_{i,j-1}$.

Case 2: j is paired in $[i,j]$. The contribution to $Q_{i,j}$ in this case is given by

$$\begin{aligned}
Q_{i,j} \quad += \quad & \sum_{k=i}^{j-4} \sum_{\substack{s \in \mathbb{SS}[i,j] \\ (k,j) \in s}} BF(s)E(s) = \sum_{k=i}^{j-4} \sum_{\substack{s \in \mathbb{SS}[i,j] \\ (k,j) \in s}} BF(s)\left[E(s[i,k-1]) + E(s[k,j])\right] \\
= \quad & \sum_{k=i}^{j-4} bp(k,j) \cdot \left\{ \sum_{s_1 \in \mathbb{SS}[i,k-1]} \sum_{\substack{s_2 \in \mathbb{SS}[k,j] \\ (k,j) \in s_2}} BF(s_1) \cdot BF(s_2)\left[E(s_1) + E(s_2)\right] \right\} \\
= \quad & \sum_{k=i}^{j-4} bp(k,j) \cdot \left\{ \sum_{s_1 \in \mathbb{SS}[i,k-1]} BF(s_1)E(s_1) \sum_{\substack{s_2 \in \mathbb{SS}[k,j] \\ (k,j) \in s_2}} BF(s_2) + \right. \\
& \left. \sum_{s_1 \in \mathbb{SS}[i,k-1]} BF(s_1) \sum_{\substack{s_2 \in \mathbb{SS}[k,j] \\ (k,j) \in s_2}} BF(s_2)E(s_2) \right\} \\
= \quad & \sum_{k=i}^{j-4} bp(k,j) \cdot \left\{ Q_{i,k-1} \cdot ZB_{k,j} + Z_{i,k-1} \cdot QB_{k,j} \right\}. \qquad (5.15)
\end{aligned}$$

Putting together the contributions from both cases, we have

$$Q_{i,j} \quad = \quad Q_{i,j-1} + \sum_{k=i}^{j-4} bp(k,j)\left[Q_{i,k-1}ZB_{k,j} + Z_{i,k-1}QB_{k,j}\right]. \qquad (5.16)$$

5.3.7 Recursions for the Turner nearest neighbor energy model

In the nearest neighbor energy model [9, 59], free energies are defined not for base pairs, but rather for *loops* in the loop decomposition of a secondary structure. In particular, there are stabilizing, negative free energies for stacked base pairs and destabilizing, positive free energies for hairpins, bulges, internal loops, and multiloops.

In this section, free energy parameters for base stacking and loops are from the Turner 2004 energy model [59]. As in the previous subsection, Q,Z are defined, but now with respect to the

Turner model.

$$\begin{aligned} Q_{i,j} &= \sum_{s \in \mathbb{SS}[i,j]} E(s) \cdot \exp(-E(s)/RT) \\ Z_{i,j} &= \sum_{s \in \mathbb{SS}[i,j]} \exp(-E(s)/RT). \end{aligned} \tag{5.17}$$

It follows that $Z = Z_{1,n}$ is the partition function for secondary structures (the Boltzmann weighted counting of all structures of **a**) and

$$\langle E(s) \rangle = \frac{Q_{1,n}}{Z_{1,n}} = \sum_{s \in \mathbb{SS}[1,n]} p(s) \cdot E(s) = \sum_{s \in \mathbb{SS}[1,n]} E(s) \cdot \frac{\exp(-E(s)/RT)}{Z}. \tag{5.18}$$

To complete the derivation of recursions, we must define $QB_{i,j}$ and $ZB_{i,j}$ for the Turner model.

To provide a self-contained treatment, we recall McCaskill's algorithm [175], 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 5.1 (Partition function Z and related function Q)**.**

- $Z_{i,j} = \sum_s \exp(-E(s)/RT)$ where the sum is taken over all structures $s \in \mathbb{SS}[i,j]$.
- $ZB_{i,j} = \sum_s \exp(-E(s)/RT)$ where the sum is taken over all structures $s \in \mathbb{SS}[i,j]$ which contain the base pair (i,j).
- $ZM_{i,j} = \sum_s \exp(-E(s)/RT)$ where the sum is taken over all structures $s \in \mathbb{SS}[i,j]$ 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 $s \in Q[i,j]$ 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 $i < r \leq j$.

- $Q_{i,j} = \sum_s E(s) \cdot \exp(-E(s)/RT)$ where the sum is taken over all structures $s \in \mathbb{SS}[i,j]$.

- $QB_{i,j} = \sum_s E(s) \cdot \exp(-E(s)/RT)$ where the sum is taken over all structures $s \in \mathbb{SS}[i,j]$ which contain the base pair (i,j).

- $QM_{i,j} = \sum_s E(s) \cdot \exp(-E(s)/RT)$ where the sum is taken over all structures $s \in \mathbb{SS}[i,j]$ which are contained within an enclosing multiloop having *at least* one component.

- $QM1_{i,j} = \sum_s E(s) \cdot \exp(-E(s)/RT)$ where the sum is taken over all structures $s \in \mathbb{SS}[i,j]$ 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 $i < r \leq j$.

For $j - i \in \{0,1,2,3\}$, $Z(i,j) = 1$, since the empty structure is the only possible secondary structure. For $j - i > \theta = 3$, we have

$$Z_{i,j} = Z_{i,j-1} + ZB_{i,j} + \sum_{r=i+1}^{j-4} Z_{i,r-1} \cdot ZB_{r,j} \tag{5.19}$$

$$\begin{aligned} ZB_{i,j} = {} & \exp(-HP(i,j)/RT) + \sum_{i \leq \ell \leq r \leq j} \exp(-IL(i,j,\ell,r)/RT) \cdot ZB_{\ell,r} + \\ & \exp(-(a+b)/RT) \cdot \left(\sum_{r=i+1}^{j-\theta-2} ZM_{i+1,r-1} \cdot ZM1_{r,j-1} \right) \end{aligned} \tag{5.20}$$

$$ZM1_{i,j} = \sum_{r=i+\theta+1}^{j} ZB_{i,r} \cdot \exp(-c(j-r)/RT) \tag{5.21}$$

$$\begin{aligned} ZM_{i,j} = {} & \sum_{r=i}^{j-\theta-1} ZM1_{r,j} \cdot \exp(-(b + c(r-i))/RT) + \\ & \sum_{r=i+\theta+2}^{j-\theta-1} ZM_{i,r-1} \cdot ZM1_{r,j} \cdot \exp(-b/RT). \end{aligned} \tag{5.22}$$

Base Case: For $j-i \in \{-1,0,1,2,3\}$, $Q_{i,j} = QB_{i,j} = 0$, $Z_{i,j} = 1$, $ZB_{i,j} = ZM_{i,j} = ZM1_{i,j} = 0$.

Inductive Case: Assume that $j-i > 3$.

Case A: (i,j) closes a hairpin.

In this case, the contribution to $QB_{i,j}$ is given by

$$A_{i,j} \quad = \quad \exp\left(-\frac{H(i,j)}{RT}\right) \cdot H(i,j) \tag{5.23}$$

Case B: (i,j) closes a stacked base pair, bulge or internal loop, whose other closing base pair is (ℓ,r), where $i < \ell < r < j$.

In this case, the contribution to $QB_{i,j}$ is given by the following

$$\begin{aligned} B_{i,j} \quad &= \quad \sum_{\ell=i+1}^{\min(i+30,j-5)} \sum_{r=j-1}^{\max(\ell+4,j-(30-(\ell-i)))} \sum_{\substack{s\in\mathbb{SS}[\ell,r] \\ (\ell,r)\in s}} \exp\left(-\frac{IL(i,j,\ell,r)}{RT}\right) \\ &\quad \cdot BF(s) \cdot [IL(i,j,\ell,r) + E(s)] \end{aligned} \tag{5.24}$$

$$\begin{aligned} &= \quad \sum_{\ell=i+1}^{\min(i+31,j-5)} \sum_{r=j-1}^{\max(\ell+4,j-(30-(\ell-i)))} \exp\left(-\frac{IL(i,j,\ell,r)}{RT}\right) \cdot IL(i,j,\ell,r) \\ &\quad \cdot ZB(\ell,r) + \exp\left(-\frac{IL(i,j,\ell,r)}{RT}\right) \cdot QB(\ell,r). \end{aligned} \tag{5.25}$$

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

$$C_{i,j} = \sum_{\substack{s\in\mathbb{SS}[i,j],(i,j)\in s\\ (i,j)\text{ closes a multiloop}}} BF(s)E(s) \tag{5.26}$$

$$= \sum_{r=i+6}^{j-5} \exp\left(-\frac{a+b}{RT}\right)\cdot \sum_{\substack{s_1\in\mathbb{SS}[i+1,r-1],s_2\in\mathbb{SS}[r,j-1]\\ r\text{ base-paired in }s_2}} BF(s_1)\cdot BF(s_2)\cdot \tag{5.27}$$

$$[a+b+E(s_1)+E(s_2)]$$

$$= \sum_{r=i+6}^{j-5} \exp\left(-\frac{a+b}{RT}\right)\cdot \sum_{\substack{s_1\in\mathbb{SS}[i+1,r-1],s_2\in\mathbb{SS}[r,j-1]\\ r\text{ base-paired in }s_2}} BF(s_1)\cdot BF(s_2)\cdot [a+b] +$$

$$\sum_{r=i+6}^{j-5} \exp\left(-\frac{a+b}{RT}\right)\cdot \sum_{s_1\in\mathbb{SS}[i+1,r-1]} BF(s_1)\cdot E(s_1) \sum_{\substack{s_2\in\mathbb{SS}[r,j-1]\\ r\text{ base-paired in }s_2}} BF(s_2) +$$

$$\sum_{r=i+6}^{j-5} \exp\left(-\frac{a+b}{RT}\right)\cdot \sum_{s_1\in\mathbb{SS}[i+1,r-1]} BF(s_1) \sum_{\substack{s_2\in\mathbb{SS}[r,j-1]\\ r\text{ base-paired in }s_2}} BF(s_2)\cdot E(s_2)$$

$$= \sum_{r=i+6}^{j-5} \exp\left(-\frac{a+b}{RT}\right)\cdot [(a+b)\cdot ZM(i+1,r-1)\cdot ZM1(r,j-1)+$$

$$QM(i+1,r-1)\cdot ZM1(r,j-1) + ZM(i+1,r-1)\cdot QM1(r,j-1)]\,. \tag{5.28}$$

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.

$$QM1_{i,j} = \sum_{k=i+4}^{j} \sum_{\substack{s\in\mathbb{SS}[i,k]\\ (i,k)\in s}} \exp\left(-\frac{c(j-k)}{RT}\right)\cdot BF(s)\cdot [c(j-i)+E(s)]$$

$$= \sum_{k=i+4}^{j} \exp\left(-\frac{c(j-k)}{RT}\right)\cdot \left[c(j-i)\cdot ZB(i,k) + QB_{i,k}\right]. \tag{5.29}$$

$$
\begin{aligned}
QM_{i,j} &= QMA_{i,j} + QMB_{i,j} \qquad (5.30)\\
QMA_{i,j} &= \sum_{r=i}^{j-\theta-1} \sum_{\substack{s\in\mathbb{SS}[r,j]\\ r \text{ pairs in } [r,j]}} \exp\left(-\frac{b+c(r-i)}{RT}\right)\cdot\exp\left(-\frac{E(s)}{RT}\right)\cdot[b+c(r-i)+E(s)]\\
&= \sum_{r=i}^{j-\theta-1} \exp\left(-\frac{b+c(r-i)}{RT}\right)\cdot\{ZM1(r,j)\cdot(b+c(r-i))+QM1(r,j)\}\\
QMB_{i,j} &= \sum_{r=i+5}^{j-\theta-1} \sum_{s_1\in\mathbb{SS}[i,r-1]} \sum_{\substack{s_2\in\mathbb{SS}[r,j]\\ r \text{ pairs in } [r,j]}} \exp\left(-\frac{b}{RT}\right)\cdot\exp\left(-\frac{E(s_1)}{RT}\right)\cdot\\
&\quad \exp\left(-\frac{E(s_2)}{RT}\right)\cdot[b+E(s_1)+E(s_2)]\\
&= \exp\left(-\frac{b}{RT}\right)\cdot\sum_{r=i+5}^{j-4}\{b\cdot ZM(i,r-1)\cdot ZM1(r,j)+\\
&\quad QM(i,r-1)\cdot ZM1(r,j)+ZM(i,r-1)\cdot QM1(r,j)\}\,.
\end{aligned}
$$

(5.31)

This completes the derivation of the recursions for expected energy.

5.4 Results

In this section, we describe a detailed comparison of our thermodynamic entropy algorithms FTD and DP, both implemented in the publicly available program `RNAentropy`, with the algorithm of Manzourolajdad et al. [184] which computes the derivational entropy for trained RNA stochastic context free grammars. Subsequently, we show that by accounting for structural entropy, there is an improvement in the correlation between hammerhead ribozyme cleavage activity and total free energy, extending a result of Shao et al. [192].

5.4.1 Comparison of structural entropy and derivational entropy

Using random RNA, 960 seed alignment sequences from Rfam family RF00005, and a collection of 2450 sequences obtained by selecting the first RNA from the seed alignment of each family from the Rfam 11.0 database [193], we show the following.

1. The thermodynamic structural entropy algorithms DP, FTD compute the same structural entropy values with the same efficiency, although as sequence length increases, FTD runs somewhat faster and returns slightly smaller values than does DP, since FTD uses a finite difference to approximate the derivative of the logarithm of the partition function.

2. DP and FTD appear to be an order of magnitude faster than the SCFG method of [184], which latter requires two minutes for RNA sequences of length 500 that require only a few seconds for DP and FTD. Run times and derivational entropy values returned by the program of [184] heavily depend on the grammar chosen and the training set used for production rule probabilities (Table 5.1).

3. Derivational entropy values computed by the method of [184] are much larger than thermodynamic structural entropy values of DP and FTD, ranging from about 4-8 times larger,depending on the SCFG chosen (Table 5.1).

4. The length-normalized correlation between thermodynamic structural entropy values and derivational entropy values is poor to moderately weak.

Unless otherwise specified, throughout this chapter, FTD, DP and SCFG refer to the *formal temperature derivative* method (Algorithm 1, with Turner'04 parameters), the dynamic programming method (Algorithm 2, with Turner'04 parameters), and the stochastic context free

grammar method of [184]. SCFG (G4), SCFG (G5), SCFG (G6) respectively refer to the SCFG method of [184] using the stochastic context free grammars G4, G5, and G6. Additionally, there are three different training sets for each grammar: Rfam5, Mixed80 and Benchmark – see [184] for explanations of the training sets. Thus SCFG (G6,Benchmark) refers to derivational entropy, computed by the algorithm of [184], using grammar G6 with training set Benchmark, etc.

Table 5.1 lists the average values, plus or minus one standard deviation, for the entropy values and run time (in seconds) for 960 transfer RNAs from the seed alignment of family RF00005 from Rfam 11.0 [193]. Results for five methods are presented: (1) the dynamic programming method of this chapter, using the Turner 2004 free energy parameters (DP), (2) approximating the *formal temperature derivative* $\frac{\partial}{\partial T} \ln Z(T)$ by finite differences, and subsequently applying equations (5.10, 5.6), using Turner 2004 free energy parameters (FTD); (3,4,5) using the program of [184] respectively with the stochastic context free grammars G4, G5, and G6 trained on the dataset 'Rfam5'.

Table 5.2 presents the Pearson correlation for entropy values of 960 transfer RNAs from the seed alignment of family RF00005 from the database Rfam 11.0 [193]. The upper-triangular [resp. lower-triangular] entries are correlations for *unnormalized* [resp. *length-normalized*] entropy values. Entropy values were computed for the same methods as in Table 5.1. Since there is little variation in sequence length for the transfer RNAs in the seed alignment of RF00005 (average length is 73.41 ± 5.13), any correlation due to sequence length is eliminated. The table shows the poor correlation between SCFG structural entropy, as computed by each grammar, with thermodynamic structural entropy.

Method	Entropy ($\mu \pm \sigma$)	Run Time ($\mu \pm \sigma$)
DP	5.953 ± 1.381	0.074 ± 0.017
FTD ($\Delta T = 10^{-7}$)	5.532 ± 1.342	0.058 ± 0.014
SCFG(G4,Rfam5)	39.917 ± 2.885	0.437 ± 0.096
SCFG(G5,Rfam5)	40.682 ± 3.053	0.204±0.046
SCFG(G6,Rfam5)	21.207 ± 2.412	0.433±0.096

TABLE 5.1: Average values for structural entropy and run time (in seconds) for the 960 transfer RNA sequences from the seed alignment of Rfam family RF00005. Methods include: DP: dynamic programming algorithm from our program `RNAentropy`, using the Turner 2004 energy parameters; FTD ($\Delta T = 10^{-7}$): finite difference computation of $\langle E \rangle = RT^2 \cdot \frac{\ln Z(T+\Delta T) - \ln Z(T)}{\Delta T}$, where formal and table temperature are *uncoupled*, and *formal temperature* increment is 10^{-7}; SCFG (G4,Rfam5): SCFG method [184] using grammar G4 with training dataset 'Rfam5'; SCFG (G5,Rfam5): SCFG method using grammar G5 with training dataset 'Rfam5'; SCFG (G6,Rfam5): SCFG method using grammar G6 with training dataset 'Rfam5'. FTD returns very similar values for temperature increments $10^{-7} \leq \Delta T \leq 10^{-11}$; however, for smaller temperature increments, there is a slight deviation due to numerical precision issues – for example, average entropy of FTD with $\Delta T = 10^{-12}$ is 5.238878 ± 1.504748, with similar run times as other FTD runs.

Norm \ Unnorm	DP	FTD ($\Delta T = 10^{-7}$)	G4	G5	G6
DP	1	0.905	0.294	0.256	0.451
FTD ($\Delta T = 10^{-7}$)	0.919	1	0.142	0.116	0.398
SCFG(G4,Rfam5)	0.314	0.301	1	0.969	0.666
SCFG(G5,Rfam5)	0.247	0.263	0.720	1	0.619
SCFG(G6,Rfam5)	0.428	0.458	0.541	0.462	1

TABLE 5.2: Pearson correlation for entropy values of 960 transfer RNAs from the seed alignment of family RF00005 from the database Rfam 11.0 [193]. Upper-triangular entries are for *unnormalized* entropy values, while lower-triangular entries are for *length-normalized* entropy values. Entropy values were computed for the same methods described in Figure 5.1; in particular, all SCFGs were trained with RF00005, as described in [184].

Table 5.3 presents the average *positional entropy*, length-normalized structural entropy, and corresponding Z-scores for a small collection of experimentally confirmed conformational switches, collected by Giegerich et al. [194], and available on the `RNAentropy` web server. There appears to be no clear entropic signal for conformational switches, at least with respect to this small collection of sequences.

RNA	Seq len	Pos Ent	Norm str ent	Z-score, pos ent	Z-score, str ent
Spliced-Leader	56	0.802	0.075	0.755	-0.697
Attenuator	73	0.326	0.054	-0.871	-0.983
MS2	73	0.076	0.061	-1.660	-1.366
S15	74	0.191	0.079	-2.242	-0.734
E coli dsrA	85	0.331	0.096	-0.557	1.444
HDV ribozyme	107	0.326	0.034	-2.037	-2.424
Tetrahymena Group I intron	108	0.515	0.076	-1.062	0.434
E. coli alpha operon mRNA	130	0.251	0.059	-1.448	-1.865
hok	142	0.340	0.087	0.700	0.608
3'-UTR of AMV RNA	145	0.336	0.077	-0.517	-0.316
T4 td gene intron	163	0.542	0.042	-1.129	-2.365
thiM-Leader	165	0.515	0.085	-1.660	-0.474
btuB	202	0.830	0.092	-0.691	0.362
Sbox-metE	247	0.237	0.097	-1.350	0.727
HIV-1 leader	280	0.324	0.086	-1.425	0.109
B. subtilis ribD leader	304	0.471	0.067	-1.835	-1.460
B. subtilis ypaA leader	342	0.428	0.076	-1.659	-0.184

TABLE 5.3: Thermodynamic structural entropy, *positional entropy*, and corresponding Z-scores for a small collection of experimentally confirmed conformational switches, collected in [194] – sequences available at the `RNAentropy` web site. For each sequence, the positional (resp. structural) entropy x was computed, along with the mean μ and standard deviation σ of 1000 dinucleotide shuffles of the sequence. The Z-score is then $\frac{x-\mu}{\sigma}$. Dinucleotide shuffles were computed, using the Altschul-Erikson algorithm [195] as implemented in [196]. Pearson correlation between Z-scores for positional and structural entropy is 0.4103.

Table 5.4 presents the number of sequences, average *length-normalized* thermodynamic entropy, average entropy Z-score, average *length-normalized ensemble defect*, and average Z-score for for sequences in the seed alignment of several RNA familes from the Rfam 11.0 [193], as well as the precursor microRNAs from the repository MIRBASE [197]. Average values are given,

plus or minus one standard deviation. The Z-score is defined as $\frac{x-\mu}{\sigma}$, where x is the entropy (resp. *ensemble defect*) of a given sequence, and μ (resp. σ denotes the mean (resp. standard deviation) of corresponding values for 100 random sequences having the same dinucleotides, obtained by using the Altschul-Erikson dinucleotide shuffling algorithm [195]. As shown by this table, Rfam family members appear to have lower structural entropy as well as *ensemble defect* than random RNA having the same dinucleotides, although the family RF00005 of transfer RNAs shows an exception to this rule for structural entropy. The most pronounced Z-scores for structural entropy and *ensemble defect* are for precursor microRNAs, which have very stable stem-loop structures. These results are generally comparable, with the exception of entropy Z-scores for RF00005, with results concerning minimum free energy (MFE) Z-scores from [196, 198]. Indeed, the particularly low MFE Z-scores of precursor miRNAs is used as a feature in the support vector machine `miPred` to detect microRNAs [199].

RNA family	seq	H	Z-score, H	ens def	Z-score, ens def
RF00001	712	0.071 ± 0.016	-0.354 ± 1.056	0.198 ± 0.123	-0.423 ± 0.965
RF00004	208	0.068 ± 0.014	-1.425 ± 1.018	0.177 ± 0.103	-0.901 ± 0.863
RF00005	960	0.081 ± 0.019	-0.049 ± 0.949	0.189 ± 0.105	-0.405 ± 0.820
RF00167	133	0.077 ± 0.020	-0.606 ± 1.111	0.164 ± 0.105	-0.782 ± 0.858
MIRBASE	28645	0.056 ± 0.018	-1.791 ± 1.491	0.101 ± 0.076	-1.324 ± 0.791

TABLE 5.4: For several large families from the Rfam 11.0 database [193], and for MIRBASE precursor microRNA [197], the table presents the number of sequences (seq), length-normalized values of thermodynamic structural entropy (H) and *ensemble defect* (ens def), and the corresponding Z-scores for entropy and *ensemble defect*. For each sequence from a given RNA family, 100 random sequences were generated with the same dinucleotides, using the Altschul-Erikson dinucleotide shuffling algorithm [195] as implemented in [196] – in the case of MIRBASE, only 10 random sequences were generated for each sequence. Subsequently, Z-scores were computed as $\frac{x-\mu}{\sigma}$, where x is the entropy (resp. *ensemble defect*) of a given sequence, and μ (resp. σ) is the mean (standard deviation) of 100 random sequences having the same dinucleotides.

We now turn to the figures that support each of the four assertions made at the beginning of Section 5.4.1. Figure 5.1 shows the average run times and entropy values for for DP, FTD ($\Delta T = 10^{-7}$), and the SCFG method of [184] using each of the grammars G4, G5 and G6 with training data from the set 'Benchmark'. According to benchmarking work of [184] and [200], the grammar G6 seems somewhat better than G4 and G5. It is for this reason that we focus principally on the grammar G6, which was first introduced in the SCFG algorithm `PFold` for RNA secondary structure prediction – see [187]. Figure 5.1A depicts average run times for DP, FTD, and SCFG methods, for 100 random RNA sequences of length n, where n ranges from 20 to 500 with an increment of 5. This figure shows that FTD and DP run faster by an order of magnitude than the SCFG methods – indeed, for length 500 RNAs, derivational entropy is computed in two minutes, while thermodynamic structural entropy is computed in a few seconds. The Figure 5.1B depicts the entropy values computed by DP, FTD ($\Delta T = 10^{-7}$), and SCFG methods. Note that for large RNA sequence length, entropy values returned by FTD are slightly smaller than those returned by DP, in agreement with the discussion in Section 5.3.4. Entropy values for the grammar G5 are considerably larger than those of FTD and DP, while entropy values for G4 and G6 are almost identical and approximately twice the size of those from G5.

Figure 5.2A presents graphs of length-normalized entropy values, computed by DP and SCFG. Using methods from algebraic combinatorics [201, 202], it is possible to prove that the length-normalized asymptotic structural entropy is constant, as observed in this figure. By numerical fitting, we find that the slope of the DP line is 0.087, while that of G6 is 0.329; i.e. SCFG entropy values using the G6 grammar are 3.78 times those of DP entropy. This is supported by Table 5.1, which suggests that G6 entropy values are 3.56 times larger than DP, while G4 and G5 entropy

values are 6.71 resp. 6.85 times larger than DP entropy values. Figure 5.2B depicts the relative frequency of structural entropy values for DP, FTD, and SCFG methods for 960 transfer RNA sequences from the seed alignment of the Rfam 11.0 database [193].

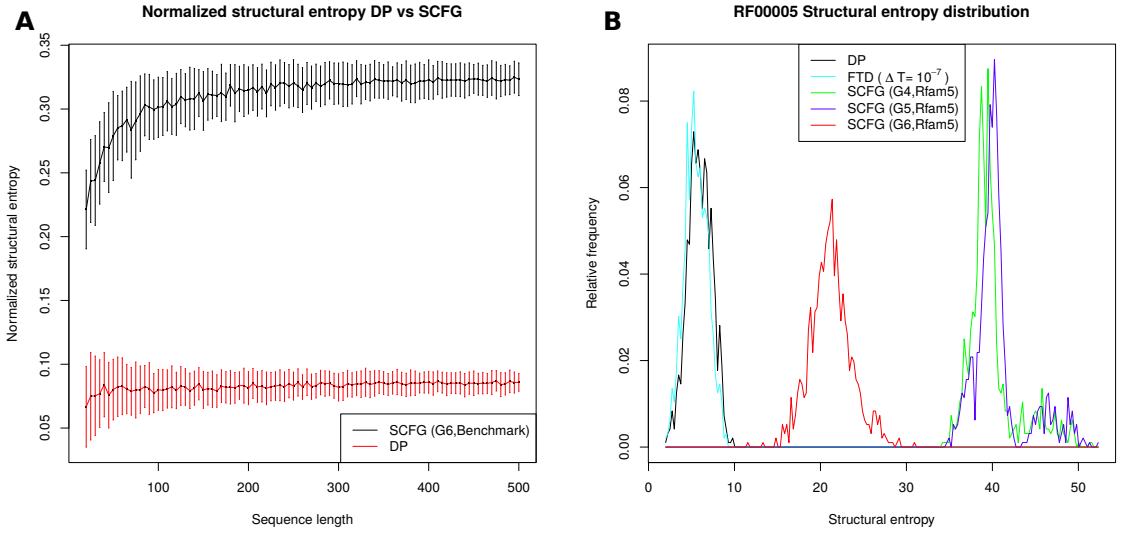


FIGURE 5.2: *(A)* The average of length-normalized entropy values, as computed by DP and SCFG (G6,Benchmark), using the same data as described in the caption of Figure 5.1. Using methods from algebraic combinatorics, it can be proven that the length-normalized entropy for a homopolymer is asymptotically constant. By numerical fitting, we find that SCFG values are roughly four times as large as DP values (approximate fitted value 3.78). *(B)* Relative frequency of entropy values for the 960 transfer RNA sequences in the seed alignment of RF00005 family from Rfam 11.0 [193], as computed for each of the five methods DP, FTD ($\Delta T = 10^{-7}$), SCFG (G4,Rfam5), SCFG (G5,Rfam5) and SCFG (G6,Rfam5). See the caption from Figure 5.1 for explanation of each method, where in contrast to previous figures, the training set 'Rfam5' was used in place of 'Benchmark'. Average entropy values for RF00005 are given as follows. FTD ($\Delta T = 10^{-7}$): 5.53±1.34. DP: 5.95±1.38; G4: 39.92±2.88; G5: 40.68±3.05; G6: 21.21 ± 2.41. Note the bimodal distribution of entropy values computed with the SCFGs G4 and G5. Relative frequency plot for 712 5S ribosomal RNAs from RF00001 is very similar (data not shown).

Figure 5.3 presents scatter plots and Pearson correlation of length-normalized entropy values and several notions of structural diversity that have been used for RNA design [21, 124]. Values were computed in this figure for a set of 2450 RNAs of various lengths, by selecting the

first sequence from the seed alignment of each family from the Rfam 11.0 database [193], after discarding a few families having too few sequences. Figure 5.3A depicts the Pearson correlation between length-normalized structural entropy values, as computed by DP, FTD, and the SCFG method using grammars G4, G5, G6. Length-normalized derivational entropy values remain highly correlated, regardless of training set, but the correlation of all SCFG methods is poor with DP. The Pearson correlation of 0.79 for length-normalized entropy values obtained by G4 and G5 is high; however the correlation with G6 drops to 0.56 (G4-G6) and 0.34 (G5-G6). Figure 5.3B depicts scatter plots and Pearson correlation for 960 transfer RNAs from family RF00005 of Rfam 11.0, for length-normalized structural entropy, as computed by DP, and various notions of structural diversity used in synthetic RNA design. (By minimizing values such as the *positional entropy*, structural entropy, *ensemble defect*, *expected base pair distance*, it is more likely that computationally designed RNAs will fold into their predicted structures when experimentally validated.) Brief definitions of the notions of structural diversity that are compared in Figure 5.3B are given as follows. *Native Contacts*: proportion of base pairs in the Rfam consensus structure that appear in the low energy Boltzmann ensemble, defined by $\sum_{s \in \mathbb{SS}} p(s) \cdot \frac{|s \cap s_0|}{|s_0|}$, where s_0 is the Rfam consensus structure. Positional entropy: average *positional entropy* $\sum_{i=1}^{n} H_2(i)/n$, where $H_2(i)$ is defined by equation (5.5). Expected base pair distance: length-normalized value determined from $\sum_{s \in \mathbb{SS}} p(s) \cdot d_{\mathrm{BP}}(s, s_0)$, where s_0 is the Rfam consensus structure, computed by $\sum_{1 \le i < j \le n} I[(i,j) \notin s_0] \cdot p_{i,j} + I[(i,j) \in s_0] \cdot (1 - p_{i,j})$ where I denotes the indicator function – see [124]. *Ensemble defect*: length-normalized value determined from $n - \sum_{i \neq j} p^*_{i,j} \cdot I[(i,j) \in s_0] - \sum_{1 \le i \le n} p^*_{i,i} \cdot I[i \text{ unpaired in } s_0]$, where s_0 is the Rfam consensus structure, I denotes the indicator function, and $p^*_{i,j}$ is defined in equation (5.4). Vienna structural diversity: Boltzmann average base pair distance between each pair of structures in the ensemble, called *ensemble diversity* in the output of `RNAfold -p` [139], formally defined by

$\sum_{i<j} p_{i,j}(1 - p_{i,j}) + (1 - p_{i,j})p_{i,j}$, where $p_{i,j}$ and output as *ensemble diversity* by `RNAfold -p`. Morgan-Higgs structural diversity: Boltzmann average Hamming distance between each pair of structures in the ensemble, where a structure s is represented by an array where $s[i] = j$ if (i,j) or (j,i) is a base pair, and otherwise $s[i] = i$, formally defined by $n - \sum_{i,j} p^*_{i,j} \cdot p^*_{i,j}$ (see Appendix A).

Length-normalized DP entropy values are moderately highly correlated with *positional entropy*, but not with the other measures. In synthetic design of RNAs, it is our opinion that one should prioritize for experimental validation those synthetically designed RNAs by consideration of *ensemble defect*, structural entropy, etc., where the measures selected are not highly correlated. From this standpoint, one might use *ensemble defect*, structural entropy and proportion of native contacts as suitable measures for synthetic RNA design – see [52].

Figure 5.4 displays the heat capacity and structural entropy for a thermoswitch (also called RNA thermometer) from the ROSE 3 family RF02523 from the Rfam 11.0 database [193], with EMBL accession code AEAZ 01000032.1/24229-24162. The heat capacity, computed by `Vienna RNA Package` `RNAheat`, presents two peaks, corresponding to two critical temperatures T_1,T_2, where one of the two conformations of this thermoswitch is stable in the temperature range between T_1 and T_2. The entropy plot also suggests the presence of a stable structure in the temperature range between T_1 and T_2, since small entropy values entail small diversity in the Boltzmann ensemble of structures.

As shown in the tables and figures, the DP and FTD methods return almost identical values and have very similar (fast) run times, contrasted with the SCFG method, which is slow and whose values are much larger than those of DP and FTD. For a sequence of length 500, SCFG (G6,Benchmark) takes 2 minutes, compared with a few seconds for DP and FTD. Since FTD

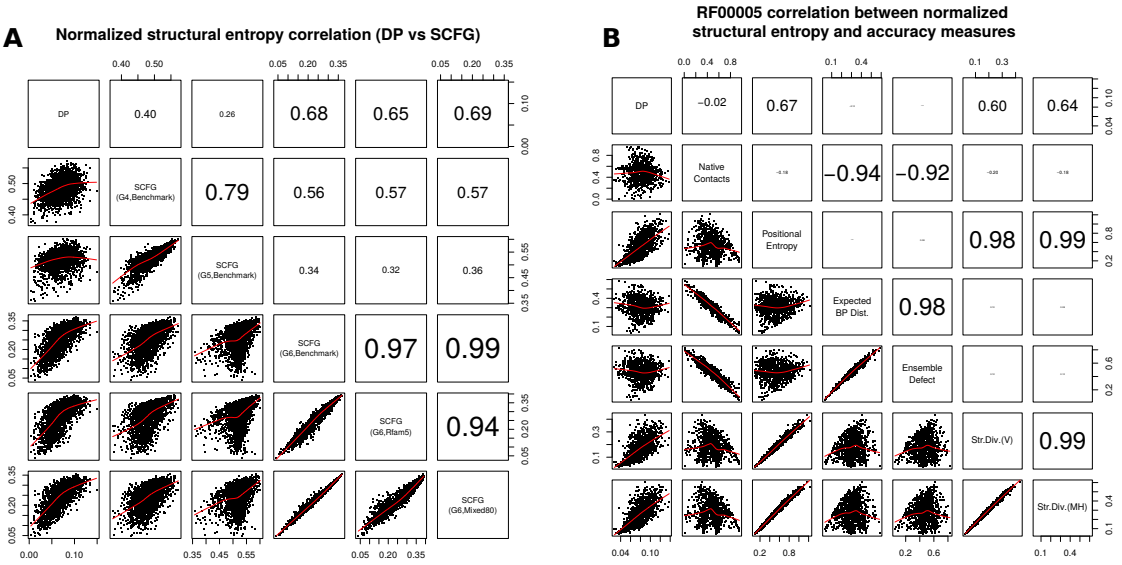


FIGURE 5.3: *(A)* Correlation between length-normalized structural entropy values, as computed by DP and five stochastic context free grammars: grammars G4, G5 and G6 for the 'Benchmark' training set, and G6 for 'Rfam5' and 'Mixed80' training sets (see [184]). Low correlation is shown between length-normalized thermodynamic structural and derivational entropies. For the fixed grammar G6, very high correlation is displayed between length-normalized entropy values for each of the training sets 'Benchmark', 'Rfam5', 'Mixed80' (similar results for fixed grammars G4,G5 – data not shown). Although grammars G4 and G5 display a moderately high correlation together, there is low correlation with length-normalized entropy values determined by the grammar G6. Benchmarking set consists of the first sequence in the seed alignment from each family in the database Rfam 11.0 [193]. *(B)* Scatter plots and correlation between thermodynamic structural entropy and several measures of *structural diversity*, computed from 960 tRNA sequences in the seed alignment of family RF00005 from the Rfam 11.0 database [193]. Correlation is computed between the following normalized values: (1) DP: length-normalized thermodynamic structural entropy computed by DP algorithm. (2) Native Contacts, (3) *Positional entropy*: average *positional entropy*, (4) *Expected base pair distance*: length-normalized *expected base pair distance*, (5) *Ensemble defect*: length-normalized *ensemble defect* (6) Str. Div. (V): *Vienna structural diversity*, output as *ensemble diversity* by `RNAfold -p` [139], (7) Str. Div. (MH): *Morgan-Higgs structural diversity*. See Appendix A for the formal definition of these measures. *Positional entropy* is moderately correlated with DP; *ensemble defect* and *expected base pair distance* are highly correlated, and each is moderately correlated with the proportion of native contacts. *Vienna structural diversity* and *Morgan-Higgs structural diversity* are highly correlated with *positional entropy*, but only (surprisingly) only moderately correlated with conformational entropy DP, in spite of the fact that all these measures concern properties of the ensemble of structures. *Ensemble defect*, *expected base pair distance* and expected number of native contacts are all highly correlated; this is unsurprising, since all measures concern the deviation of structures in the ensemble from the minimum free energy structure. Note that *positional entropy* is poorly correlated with the proportion of native contacts, although Huynen et al. [115] show that base pairs in the MFE structure of 16S rRNA tend to belong to the structure determined by comparative sequence analysis when the nucleotides have low *positional entropy*.

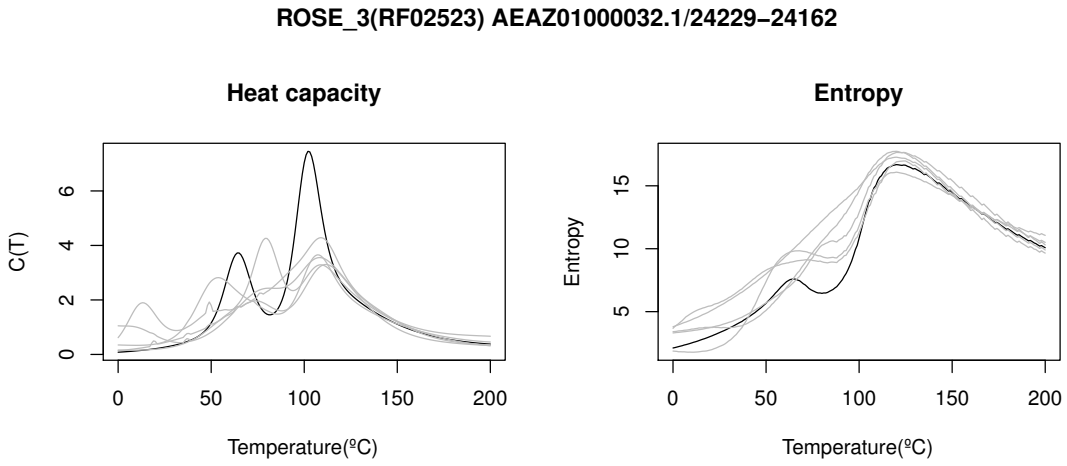


FIGURE 5.4: Heat capacity (left) and thermodynamic structural entropy (right) for a thermoswitch, or RNA thermometer, from the ROSE 3 family RF02523 from the Rfam 11.0 database [193], with EMBL accession code AEAZ01000032.1/24229-24162. Lighter curves in the background correspond to the heat capacity (left) and thermodynamic structural entropy (right) of random RNAs having the same dinucleotides, obtained by the implementation in [196] of the Altschul-Erikson dinucleotide shuffle algorithm [195]. Since structural entropy $H = \langle E(T)\rangle/RT + \ln Z(T)$ and heat capacity $C(T) = \frac{\partial}{\partial T}\langle E(T)\rangle$, the derivative of entropy H with respect to temperature closely follows the curve of the heat capacity (data not shown). Heat capacity computed using `Vienna RNA Package` `RNAheat` [139], and entropy computed by method DP.

approximates a derivative by a finite difference, one expects a small discrepancy in the values of DP and FTD for thermodynamic structural entropy. According to [184], the sensitivity and specificity of G4 and G6 grammars are "significantly" higher than that of the G5 grammar. Since G6 is the underlying grammar of the `PFold` software, for many of our comparisons, we compute derivational entropy using grammar G6 with the 'Benchmark' training set. (In data not shown, we benchmarked all nine combinations of grammars and training sets.)

5.4.2 Using RNAfold to compute conformational entropy

We have recently learned that newer versions of `Vienna RNA Package` [139] allow the user to modify the value RT by using the flag `-betaScale` (kindly pointed out by Ivo Hofacker). It follows that `RNAfold` can easily be used to compute conformational entropy by using the FTD method. Let $T = 310.15$ be the absolute temperature corresponding to 37° C, let $\Delta T = 0.01$, let $T_2 = T + \Delta T = 310.16$ and $T_1 = T - \Delta T = 310.14$. Define the scaling factors $\beta_2 = \frac{T+\Delta T}{T} = 1.0000322424633241$, and $\beta_1 = \frac{T-\Delta T}{T} = 0.9999677575366759$. Run `RNAsubopt-p -betaScale` β_2 to compute the ensemble free energy $-R(T+\Delta T)\ln Z(T+\Delta T)$, and `RNAsubopt-p -betaScale` β_1 to compute the ensemble free energy $-R(T-\Delta T)\ln Z(T-\Delta T)$, where $Z(T+\Delta T)$ [resp. $Z(T-\Delta T)$] temporarily denotes the value of the partition function where table temperature is 37° C (as usual), and *formal temperature* is $T+\Delta T$ [resp. $T-\Delta T$] in Kelvin. It follows that the *uncentered finite difference* equation (5.32)

$$RT^2 \cdot \frac{\ln Z(T+\Delta T) - \ln Z(T)}{\Delta T} \tag{5.32}$$

as well as the *centered finite difference*

$$RT^2 \cdot \frac{\ln Z(T+\Delta T) - \ln Z(T-\Delta T)}{2\Delta T} \tag{5.33}$$

both provide good approximations for the expected energy $\langle E \rangle$. Now run `RNAsubopt-p` to compute the ensemble free energy $G = -RT\ln Z$ where table and *formal temperature* are (as usual) 310.15 in Kelvin, and so compute the entropy

$$H = \frac{\langle E \rangle - G}{RT}. \tag{5.34}$$

Let `Vienna RNA` [resp. `Vienna RNA` *] denote the entropy computation just described, where expected energy is approximated by the uncentered equation (5.32) [resp. centered equation (5.33)]. Similarly, we let FTD [resp. FTD *] denote the uncentered [resp. centered] version of our code from Algorithm 1 in Section "Statistical Mechanics" in Methods. In computing entropy for Rfam family RF00005, both `Vienna RNA` and `Vienna RNA` * sometimes return entropy values that are *larger* than the correct values computed by DP, while entropy values of FTD [resp. FTD *] are always smaller than [essentially always smaller] than those of DP, as expected when using finite differences to approximate the derivative of the strictly decreasing, concave-down function $\ln Z(T)$.

Figure 5.5A shows that `Vienna RNA` is somewhat faster than FTD, and for each method, the *uncentered* version is faster than the *centered* version, which is clear since the former [resp. latter] computes the partition function twice [resp. three times]. Figure 5.5B shows that the standard deviation of entropy values for 100 random RNA is larger for `Vienna RNA` than FTD, and the uncentered form of `Vienna RNA` displays the largest standard deviation when $\Delta T = 10^{-4}$ (for $\Delta T = 0.01$, all four finite derivative methods are comparable). These results are unsurprising due to numerical precision issues; e.g. for the 98 nt purine riboswitch with EMBL accession code AE005176.1/1159509-1159606, the algorithm DP determines a value of conformational entropy 9.975439, whereas by using (centered) `Vienna RNA` * with version 2.1.8 of `RNAfold` with $\Delta T = 10^{-2}$, we obtain 9.93425742505. For $\Delta T = 10^{-4}$, 10^{-5} and 10^{-6}, `Vienna RNA` * computes entropies of 9.59831636855, $6.94285165005 \cdot 10^{-8}$, $-5.9169597422 \cdot 10^{-7}$. Such numerical instability issues are of much less concern to our method FTD and FTD *, as Figure 5.1 demonstrates for the uncentered method FTD with $\Delta T = 10^{-7}$.

Figure 5.6 shows the distribution of entropy differences (DP-FTD, DP-FTD *, DP-ViennaRNA,

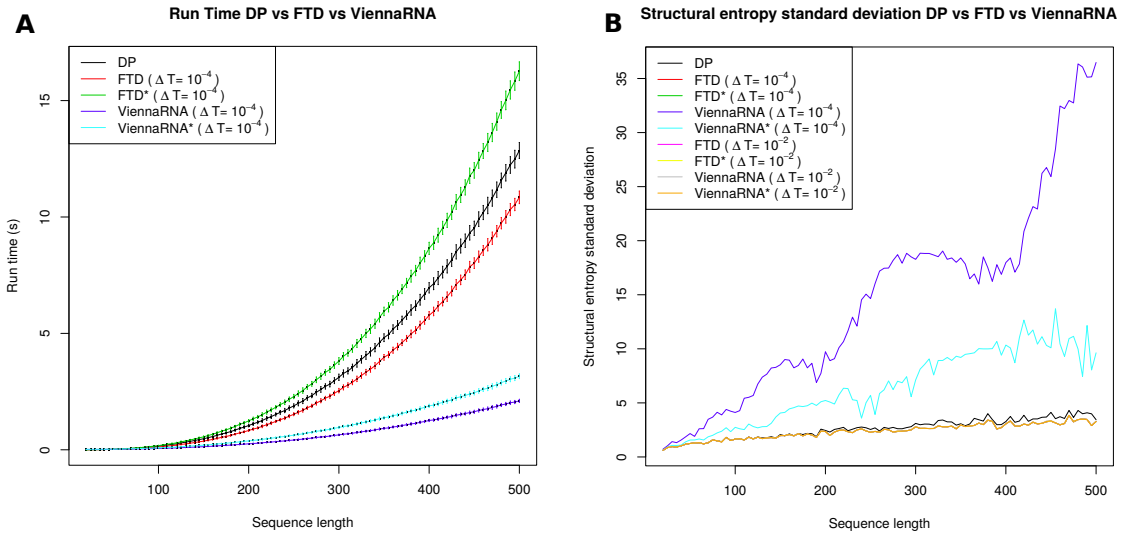


FIGURE 5.5: Average values for the run time and the entropy values for 100 random RNA sequences of length n, each having expected compositional frequency of 0.25 for A,C,G,U, where n ranges from 20 to 500 with increments of 5 for conformational entropy. *(A)* Average run times as a function of sequence length, where error bars represent ± 1 standard deviation. Methods used: DP, FTD, FTD *, `Vienna RNA`, `Vienna RNA` *. For random RNAs of length 500 nt, `Vienna RNA Package` is about three times faster than our code. *(B)* Standard deviation of the entropy values computed for 100 random RNA, displayed as a function of sequence length. From top to bottom, the first three curves represent uncentered `Vienna RNA` with $\Delta T = 10^{-4}$, centered `Vienna RNA` * with $\Delta T = 10^{-4}$, and DP. The bottom curve represents centered FTD with $\Delta T = 10^{-4}$, centered FTD * with $\Delta T = 10^{-2}$, uncentered `Vienna RNA` with $\Delta T = 10^{-2}$, centered `Vienna RNA` * with $\Delta T = 10^{-2}$. The average entropy values computed by FTD, FTD *, `Vienna RNA`, and `Vienna RNA` * are indistinguishable and since FTD values are shown in the right panel of Figure 5.1, they are not shown here.

DP-ViennaRNA*) for 960 transfer RNAs from family RF00005 from the Rfam 11.0 database [193]. Reasons for the behavior of `Vienna RNA` and `Vienna RNA` * are presumably due to numerical precision issues. These differences are small, so when plotted as a function of sequence length in a manner analogous to Figure 5.1 (not shown), average entropy values computed by FTD, FTD *, `Vienna RNA`, and `Vienna RNA` * for $\Delta T = 10^{-2}$ and 10^{-4} are visually indistinguishable.

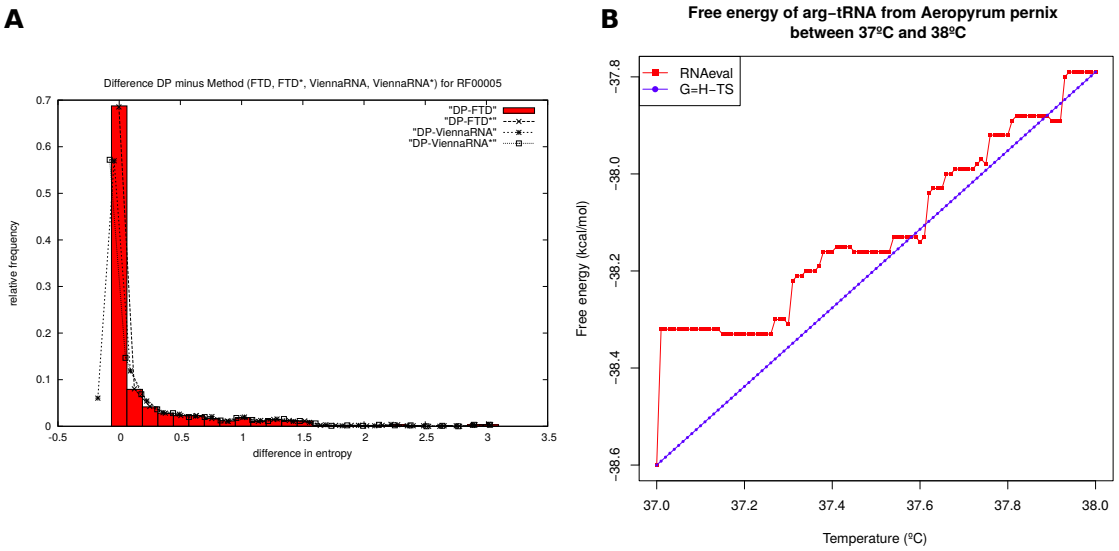


FIGURE 5.6: *(A)* Relative frequency of the difference in entropy values for 960 transfer RNAs from the RF00005 family of the Rfam 11.0 database. (1) DP-FTD with average entropy difference 0.2512 ± 0.4935 with maximum of 3.1622 and minimum of 0. (2) DP-FTD * with average entropy difference 0.2502 ± 0.4934 with maximum of 3.1602 and minimum of -0.0020. (3) DP-ViennaRNA with average entropy difference 0.2475±0.4975 with maximum of 3.1520 and minimum of -0.1743. (4) DP-ViennaRNA* with average entropy difference 0.2494 ± 0.4946 with maximum of 3.1572 and minimum of -0.0777. It is noteworthy that FTD is *always* less than DP, FTD * exceeds DP by a tiny margin only rarely, while ViennaRNA and ViennaRNA* more often exceed DP. Recall that the average deviation DP-FTD increases with increasing sequence length, as shown in the right panel of Figure 5.1. The same is true for DP-FTD *, DP-ViennaRNA, DP-ViennaRNA* (data not shown). *(B)* Free energy of arginyl-transfer RNA from *Aeropyrum pernix* with tRNAdb accession code tdbR00000589 [62] for temperatures ranging from 37° C to 38° C in increments of 0.01. The blue piecewise linear curve was created using `RNAeval-T` from the `Vienna RNA Package` [139]. The red linear curve was created by (1) calculating the entropy $S_t = G(37) - G(38)$ of the tRNA cloverleaf structure by subtracting the free energy at 38° C from the free energy at 37° C, as determined using `RNAeval-T`, (2) computing the enthalpy $H_t = G(37) + (273.15 + 37) \cdot S_t$, and then (3) computing the free energy at temperature T by $G(T) = H_t - T \cdot S_t$. The jagged free energy curve is due to the fact that `Vienna RNA Package` represents energies as integers (multiples of 0.01 kcal/mol), so that loop energies jump at particular temperatures.

Due to numerical stability issues, `Vienna RNA` and `Vienna RNA` * perform optimally with $\Delta T = 10^{-2}$. Note that when using `RNAfold`, it is essential to use `-betaScale`; indeed, if one attempts to compute the entropy using equation (5.34) where expected energy is computed from equation (5.32) [resp. equation (5.33)] by running `RNAfold-p -T 37.01` and `RNAfold-p -T 37` [resp. `RNAfold-p -T 37.01` and `RNAfold-p -T 36.99`], then the resulting entropy for the 98 nt purine riboswitch with EMBL accession code AE005176.1/1159509-1159606 is the impossible, *negative* value of -208.13 [resp. -210.61]. The large negative entropy values in this case are not only due to the lack of distinction between formal and table temperature, but as well to the fact that `Vienna RNA Package` represents energies as integers (multiples of 0.01 kcal/mol), so that loop energies jump at particular temperatures, as shown in the right panel of Figure 5.6. These issues should not be construed as shortcomings of the `Vienna RNA Package`, designed for great speed and high performance, but rather as a use of the program outside its intended parameters. As shown by Figure 5.5, the methods `Vienna RNA` and `Vienna RNA` * can rapidly compute accurate approximations of the conformational entropy.

5.4.3 Correlation with hammerhead cleavage activity

In [192], Shao et al. considered a 2-state thermodynamic model to describe the hybridization of hammerhead ribozymes to messenger RNA with subsequent cleavage at the mRNA GUC-cleavage site. In that paper, they define the total free energy

$$\Delta G_{\text{total}} = \Delta G_{\text{hybrid}} - \Delta G_{\text{switch}} - \Delta G_{\text{disrupt}} \tag{5.35}$$

where each of these energies is defined on p. 10 of [192], and obtained by averaging over 1000 low energy structures sampled by `Sfold` [203]. The authors show a (negative) high correlation between ΔG_{total} and the cleavage activity of 13 hammerhead ribozymes for GUC cleavage sites in ABCG2 messenger RNA (GenBank NM_004827.2) of *H. sapiens*; i.e. the lower the total change in free energy, the more active is the ribozyme. (Shao et al. originally considered 15 hammerheads; however two outlier hammerheads were removed from consideration.) Here, we show that the correlation with cleavage activity can be improved slightly by taking secondary structure conformational entropy into consideration.

To fix ideas, we consider the first GUC cleavage site considered by Shao et al. The minimum free energy (MFE) hybridization complex, as predicted by `RNAcofold` from the `Vienna RNA Package` [139] is shown in Figure 5.7A. The MFE structure of the 21 nt portion of mRNA, followed by a linker region of five adenines, followed by the hammerhead ribozyme, as computed by `RNAfold` from the `Vienna RNA Package` yields the same structure (where the linker region appears in a hairpin). It follows that to a first approximation, MFE hybridization structures can be predicted from MFE structure predictions of a chimeric sequence that includes a linker region. (Before the introduction of hybridization MFE software [139, 204], this approach was used to predict hybridization structures.)

In this case, enzyme activity is 0.843, $\Delta G_{\text{total}} = -5.423$ kcal/mol, structural entropy of the hammerhead is 2.830, structural entropy of the 21 nt portion of mRNA is 2.146, and structural entropy of the 21 nt portion of mRNA portion with linker and hammerhead is 2.328. Assuming that the entropy of a rigid structure is zero, the change in structural entropy ΔH(hammerhead) is $0-2.830 = -2.830$, and similarly ΔH(21 nt mRNA + linker) is -2.146, ΔH(21 nt mRNA+linker+hammerhead) is -2.328. The net change in structural entropy ΔH is ΔH(21 nt mRNA+linker+

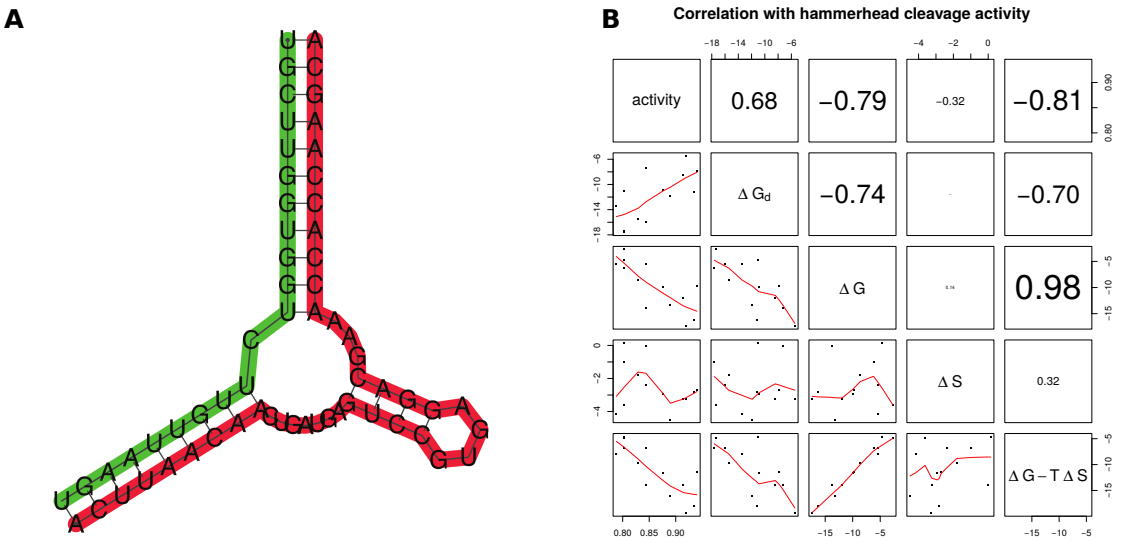


FIGURE 5.7: *(A)* Hybridization structure predicted by `RNAcofold` [168] of a 21 nt portion of messenger RNA for *H. sapiens* ABC transporter ABCG2 messenger RNA (GenBank . NM_004827.2) hybridized with a hammerhead ribozyme (data from the first line of Table 1 of [192]). The 21 nt portion of mRNA is 5′-UGCUUGGUGG UCUUGUUAAG U-3′ and the 42 nt hammerhead ribozyme is 5′-ACUUAACAAC UGAUGAGUCC GUGAGGACGA AACCACCAAG CA-3′. Messenger RNA is shown in green, while the hammerhead appears in red. In data not shown, we determined the secondary structure of the 21 nt mRNA portion, followed by a linker region of 5 adenines, followed by the 42 nt hammerhead ribozyme, by using `RNAfold` [139]. The base pairs in the hybridization complex are identical to the base pairs in the chimeric single-stranded sequence (not shown) – i.e. except for the unpaired adenines from the added linker region, the structures are identical. This fact permits us to approximate the structural entropy for the hybridization of two RNAs by using `RNAentropy` to compute the entropy of the concatenation of the sequences, separated by a linker region. *(B)* Correlation between hammerhead ribozyme cleavage activity, as assayed by Shao et al. [192], with ΔG_d (change in free energy due to disruption of mRNA, denoted $\Delta G_{\text{disrupt}}$ in text), ΔG (change in total free energy, denoted ΔG_{total} in text), both taken from [192], with ΔS (change in conformational entropy $k_B \cdot \Delta H$), and ΔG(total) - T ΔS. Cleavage activity was measured by Shao et al. for the cleavage of GUC sites in ABC transporter ABCG2 messenger RNA (GenBank NM_004827.2). Values of ΔG_d, ΔG were taken from Table 1 of [192], while the change in conformational entropy ΔS was computed by `RNAentropy`. Note modest increase in the correlation of cleavage activity with ΔG, when adding the free energy contribution $-T\Delta S$, due to conformational entropy.

hammerhead) minus ΔH(21 nt mRNA + linker) minus ΔH(hammerhead), so $\Delta H = -2.328 - (-2.146 - 2.830) = 2.648$. The net change in conformational entropy $\Delta S = k_B \cdot \Delta H$ is then 0.00526, hence the free energy contribution $-T\Delta S = -RT\Delta H = -1.632$. The correlation between ΔG_{total} and $-T\Delta S$ is the value of 0.108, while the correlation value of −0.788 between hammerhead activity and ΔG_{total} is increase in absolute value to −0.806 (p-value 0.000878) when also taking into account $-T\Delta S$. See Figure 5.7 for a scatter plot and correlations between enzyme activity and ΔG [resp. $\Delta G - T\Delta S$], which correspond to the total free energy change without [resp. with] a contribution from conformational entropy.

Figure 5.7A depicts the minimum free energy *hybridization* structure of a 21 nt portion of the ABC transporter ABCG2 messenger RNA from *H. sapiens* (GenBank NM_004827.2), hybridized with a hammerhead ribozyme (data from the first line of Table 1 of [192]). The MFE hybridization structure was computed by `Vienna RNA Package` `RNAcofold` [139]. We obtain the same structure by applying `RNAfold` to the chimeric sequence obtained by concatenating the 21 nt portion of mRNA, given by 5′-UGCUUGGUGG UCUUGUUAAG U-3′, with a 5 nt linker region consisting of adenines, with the 42 nt hammerhead ribozyme, given by 5′-ACUUAACAAC UGAUGAGUCC GUGAGGACGA AACCACCAAG CA-3′ (data not shown). By such concatenations with a separating 5 nt linker region, we can compute the structural entropy of hybridizations of the 21 nt mRNA with the hammerhead ribozyme. (In future work, we may extend `RNAentropy` to compute the entropy of hybridization complexes without using such linker regions.)

5.4.4 Structural entropy of HIV-1 genomic regions

Figure 5.8A depicts the structural entropy, computed as a moving average of 100 nt portions of the HIV-1 complete genome (GenBank AF033819.3). Using `RNAentropy`, the structural entropy was computed for each 100 nt portion of the HIV-1 genome, by increments of 10 nt; i.e. entropy was computed at genomic positions 1, 11, 21, etc. for 100 nt windows. To smooth the data, moving averages were computed over five successive windows. The figure displays the moving average entropy values, as a function of genome position (top dotted curve), entropy Z-scores, defined by $\frac{x-\mu}{\sigma}$, where x is the (moving window average) entropy at a genomic position, and μ [resp. σ] is the mean [resp. standard deviation] of the entropy for all computed 100 nt windows. Figure 5.8B is a portion of the NCBI graphics format presentation of GenBank file AF033819.3. Regions of low Z-score are position 4060 (Z-score of -2.69), position 8700 (Z-score of -2.46) and position 4040 (Z-score of -1.95). Since positions do not appear to correspond to the start/stop position of annotated genes, we ran `cmscan` from `Infernal 1.1` software [64] on the HIV-1 genome (GenBank AF033819.3). We obtained 11 predicted noncoding elements as listed in Table 5.5, including the trans-activation response (TAR) element. Many of the predicted noncoding RNAs are much shorter than the 100 nt window used in the `RNAentropy` genome-scanning approach just described – it follows that low entropy Z-scores cannot be expected for such elements. Nevertheless, certain elements have quite low entropy Z-scores, such as the 5′-UTR and TAR element, both of which are known to be involved in the packaging of two copies of the HIV-1 genome in the viral capsid [205].

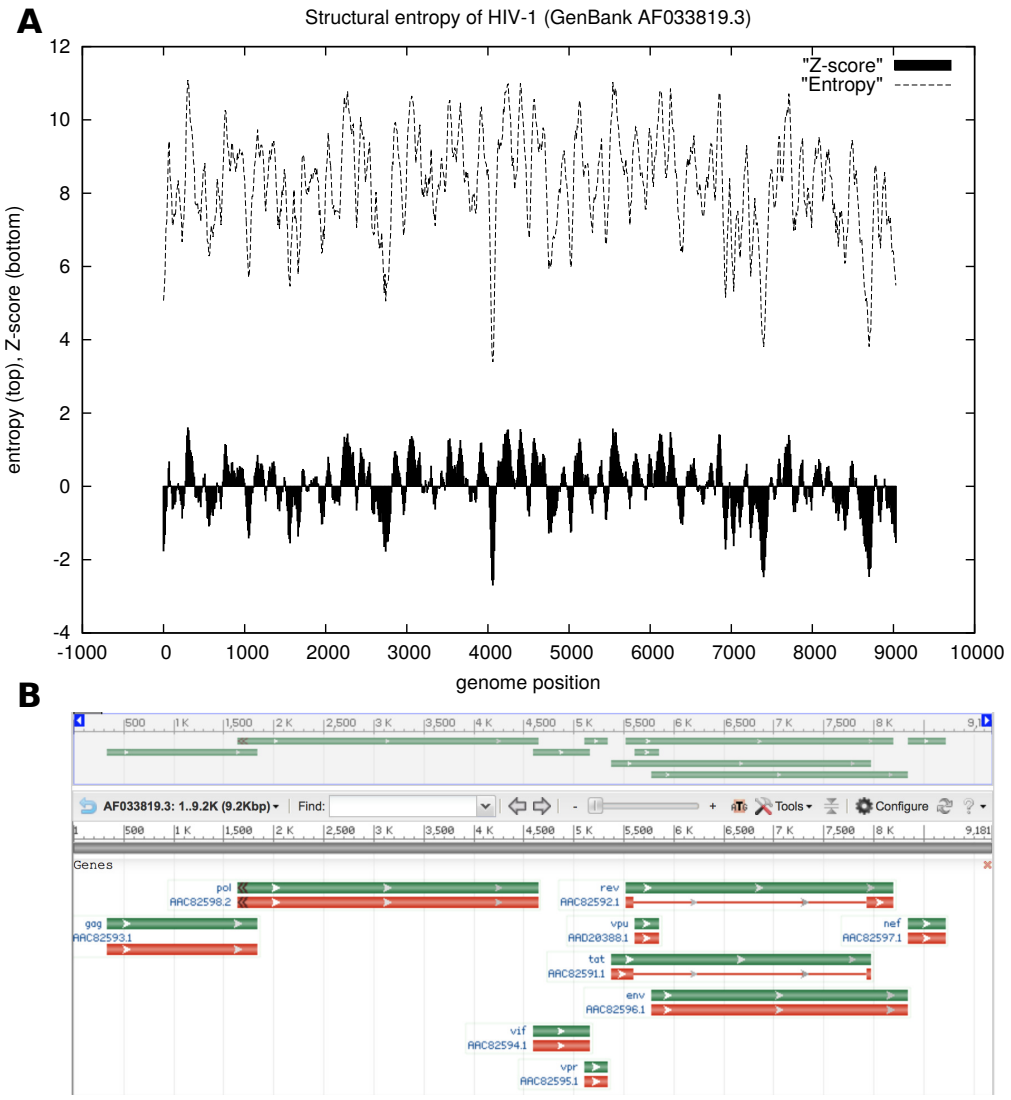


FIGURE 5.8: Structural entropy plot for the HIV-1 genome (GenBank AF033819.3). Using `RNAentropy`, the structural entropy was computed for each 100 nt portion of the HIV-1 genome, by increments of 10 nt; i.e. for 100 nt windows starting at genome position 1, 11, 21, etc. To smooth the curve, moving averages were computed over five successive windows. *(A)* Dotted-line displays moving average values of structural entropy; solid curve displays entropy Z-scores, defined by $\frac{x-\mu}{\sigma}$, where x represents the (moving window average) entropy at a genomic position, and μ [resp. σ] represents the mean [resp. standard deviation] of the entropy for 100 nt windows. Some of the lowest entropy Z-scores are -2.69 at position 4060, -2.46 at position 8700, -1.95 at position 4040. *(B)* NCBI graphics display of the HIV-1 genome, for comparison purposes. Low entropy (negative Z-score) regions do not appear to correspond with the start/stop location for annotated genes. In data not shown, we also computed positional entropy values [115] for the same windows, and determined a Pearson correlation of 0.7025 [resp. Spearman correlation of 0.6829] between (moving window average) values of entropy and *positional entropy*.

Name	Start	Stop	Len	E-score	entropy Z-score
RRE	7265	7601	66	7.6e-125	-1.389
HIV PBL	125	223	99	1.6e-30	-0.589
HIV POL-1 SL	2012	2124	113	3.1e-29	+0.066
HIV GSL3	400	483	84	1.2e-23	-0.299
mir-TAR	9085	9145	61	7e-21	-1.528
mir-TAR	1	60	60	1.1e-18	-1.759
HIV FE	1631	1682	52	3.6e-11	-0.506
HIV-1 DIS	240	279	40	3.7e-11	-0.205
HIV-1 SL3	309	331	23	7.1e-09	+0.907
HIV-1 SL4	337	356	20	1.9e-05	+0.907
HIV-1 SD	282	300	19	3.7e-05	-0.529

TABLE 5.5: Computationally annotated RNA noncoding elements from the HIV-1 genome with corresponding entropy Z-scores. Running `cmscan` from `Infernal 1.1` [64] on the HIV-1 genome (GenBank AF033819.3), we obtain 11 noncoding elements as listed in the table, along with the nucleotide beginning and ending positions, length of noncoding element, E-score, and entropy Z-score. Entropy Z-scores were computed using `RNAentropy` as explained in the text. Many of the annotated noncoding elements are much shorter than 100 nt, the length of the window size used; however, sporadic checking of entropy Z-scores computed for a moving window of size 50 does not seem to radically change the entropy Z-scores. Nevertheless, certain elements have low entropies and corresponding entropy Z-scores, such as the 5′-UTR and TAR (trans-activation response) element, both of which are known to be involved in the packaging of the HIV-1 genome in the viral capsid [205].

Chapter 6

RNAdualPF

6.1 Introduction

In this chapter, we consider a less stringent definition of RNA inverse folding, which is the problem of finding one or more sequences that (approximately) fold into a user-specified target structure s_0, i.e. whose minimum free energy structure with respect to the Turner energy model is (approximately) s_0. Despite the availability of inverse folding software, there is no unbiased representation of the (astronomically large) collection of sequences that fold into a given target structure s_0. Here, we introduce the program `RNAdualPF`, which computes the *dual partition function* Z^*, defined as the sum of Boltzmann factors $\exp(-E(\mathbf{a},s_0)/RT)$ of *all RNA nucleotide sequences* $\mathbf{a} \in \mathbb{AA}(s_0)$ with respect to the target structure s_0, where $\mathbb{AA}(s_0)$ denotes all sequences of the same length as s_0. Using `RNAdualPF`, we efficiently sample RNA sequences that (approximately) fold into s_0, where additionally the user can specify IUPAC sequence constraints at certain positions, and whether to include dangles (energy terms for stacked, single-stranded

nucleotides). Moreover, the user can require that all sampled sequences have a precise, specified GC-content, since, optionally, we compute the *dual partition function* $Z^*(k)$ simultaneously for all values $k = G+C$. Using Z^*, we can compute the *dual expected energy* $\langle E^* \rangle$, *dual ensemble free energy* G^*, *dual conformational entropy* S^* and *dual heat capacity* C_p^* for the collection of *sequences* that (approximately) fold into target structure s_0. Using `RNAdualPF`, we show that natural RNAs from the `Rfam` 12.0 database have *higher* minimum free energy than expected, thus suggesting that functional RNAs are under evolutionary pressure to be only marginally thermodynamically stable.

Using `RNAdualPF`, we corroborate previous studies by confirming that *C. elegans* microRNA is significantly mutationally robust; however, in contrast to previous work, *C. elegans* microRNA appears *not* to be significantly robust when GC-content is controlled. In addition, when GC-content is controlled, bacterial small noncoding RNAs are significantly non-robust. The thermodynamic parameters Z^*,C_p^*,G^*,H^*,S^* of the *ensemble of sequences* that approximately fold into a target structure s_0, together with sampled sequences from this ensemble, either with or without strict control over GC-content, provide a novel description of the universe of possible sequences that fold into a given structure. These aspects make `RNAdualPF` a unique tool for the field of molecular evolution. Source code for the C++ software `RNAdualPF` is available at `http://bioinformatics.bc.edu/clotelab/RNAdualPF`.

6.1.1 Organization

This chapter is organized in the following fashion. First, we provide some background about robustness and plasticity analysis of non coding RNAs, including the different formal definitions of robustness used in previous studies. Then, we introduce the notion of *dual partition*

function and describe the algorithmic details of the implementation of `RNAdualPF`, where we precisely define the computations required to perform a correct weighted sampling from sequences in the low energy ensemble of a given target structure, including IUPAC sequence constraints and exact GC-content control. Next, we benchmark our algorithm against the software `IncaRNAtion` [49], showing that `RNAdualPF` is not only faster, but also samples sequences with higher probability of folding into the given target structure. Then, we use `RNAdualPF` to analyze different properties of non coding RNAs, where we corroborate the findings of a previous study based on sequences generated by `RNAinverse`, in which *C. elegans* microRNAs were shown to be significantly mutationally robust [206], provided that GC-content is not controlled. On the other hand, our results contrasts the findings of [206] when GC-content is controlled. Since `RNAinverse`, used in [206] does not control GC-content, Borenstein and Ruppin had filtered out inverse folding solutions to have the same GC-content as that of given *C. elegans* microRNAs. Since the number of solutions having a desired GC-content was very small, it seems likely that the conclusions of [206] could be based on inadequate sampling due to their use of non-optimal third-party software. Finally, we describe how to use `RNAdualPF` to compute other thermodynamic parameters of the *ensemble of sequences* that approximately fold into a target structure s_0, and we provide further evidence that natural RNAs have *higher* free energy than expected.

6.2 Background

In [206], Borenstein and Ruppin define *neutrality* of an RNA sequence $\mathbf{a} = a_1, \ldots, a_n$ by $\eta(\mathbf{a}) = 1 - \frac{\langle d \rangle}{n}$, where in this section $\langle d \rangle$ denotes the average, taken over all $3n$ single-point mutants of $\mathbf{a}$, of the base pair distance d_{BP} between the minimum free energy (MFE) structure s_0 of $\mathbf{a}$

and the MFE structures of single-point mutants of **a**. An RNA sequence **a** is then defined to be *robust* if $\eta(\mathbf{a})$ is greater than the average neutrality of 1000 control sequences generated by the program `RNAinverse` [139], which fold into the same target structure s_0. The main finding of [206] is that precursor microRNAs exhibit a significantly higher level of mutational robustness when compared with random RNA sequences having the same structure. To control for sequence composition bias in their computational study, the authors filtered the output of `RNAinverse`, because GC-content is not controlled by this program. Since the filtering step required enormous run time and computational resources, the authors restricted their attention to a small set of 211 microRNAs, generating only 100 control sequences per microRNA, for which the GC-content approximately agreed with that of the given microRNA. Borenstein and Ruppin conclude that robustness of precursor microRNAs is not the byproduct of a base composition bias or of thermodynamic stability.

A similar analysis, also using the program `RNAinverse`, was undertaken by Rodrigo et al. [207] for bacterial small noncoding RNAs (sncRNAs), albeit using somewhat different definitions – precise definitions are given in Section 6.2.1. The main finding of [207] is that bacterial sncRNAs are not significantly robust when compared with 1000 sequences having the same structure, as computed by `RNAinverse`; however, bacterial sncRNAs tend to be significantly *plastic*, in the sense that the ensemble of low energy structures are structurally diverse. Unlike the case of precursor microRNAs [206], Rodrigo et al. did not control for sequence compositional bias.

This raises the question of whether the control sequences analyzed in [206, 207] are *representative* or to what extent features shared by sequences output by the program `RNAinverse` are artefacts of the program. Indeed, the number of RNA sequences that fold into a given target structure can be astronomically large. Over a few weeks, before we elected to terminate the

execution, our state-of-the-art inverse folding software `RNAiFold` [56] generated 273,926,421 many 52-nt sequences that fold exactly into the MFE secondary structure s_0 of HIV-1 ribosomal frameshift stimulating signal from the Gag-Pol overlap region AF033819.3/1631-1682, and which additionally code 17-mer peptides in the Gag and Pol reading frames having amino acids that appear in Gag/Pol peptides found in the Los Alamos HIV-1 database[208]. The number of 52 nt RNA sequences that fold into target s_0 without additionally imposing the constraint of coding particular peptides in overlapping Gag/Pol reading frames is certain to dwarf the previous number. Moreover, the number of sequences that fold into the MFE structure of an animal precursor microRNA (length 68 to 91 nt [209]) or into the MFE structure of bacterial sncRNA (length 53-436 nt [207]) is certain to be even more daunting.

Motivated by such considerations, we developed `RNAdualPF` to efficiently sample a *representative* set of sequences that approximately fold into a given target structure, and additionally control GC-content and support IUPAC sequence constraints. Sampling is performed in a manner distinct but somewhat analogous to that by which `Sfold` [203] and `RNAsubopt -p` [139] sample representative secondary structures from the Boltzmann ensemble of all structures of a given sequence. Using `RNAdualPF`, we perform a pilot study that is similar, though not identical, to that of [206, 207] for two classes of RNA: *C. elegans* precursor microRNA from Rfam 12.0 [65] and bacterial small noncoding RNAs [207].

6.2.1 Formal definitions of robustness

Let $\mathbf{a} = a_1, \ldots, a_n$ denote an arbitrary RNA sequence and s a secondary structure (see Chapter 1). The collection of all secondary structures of the RNA sequence $\mathbf{a}$ is denoted $\mathbb{SS}(\mathbf{a})$, and the free energy [59] of s is denoted by $E(\mathbf{a},s)$, or simply by $E(s)$ provided that the sequence $\mathbf{a}$ is

clear from context. The *Boltzmann probability* $p(s) = p_{\mathbf{a}}(s)$ for structure s of $\mathbf{a}$ is defined by $\exp(-E(\mathbf{a},s)/RT)/Z$, where the partition function $Z = Z(\mathbf{a}) = \sum_{s \in \mathbb{SS}(\mathbf{a})} \exp(-E(\mathbf{a},s)/RT)$. Given two secondary structures s,t of $\mathbf{a}$, the *base pair distance* $d_{\mathrm{BP}}(s,t)$ between s and t is defined to be the size of the symmetric difference of s,t, i.e. $|s - t| + |t - s|$.

In [207], Rodrigo et al. define *intrinsic distance*

$$d_0(\mathbf{a}) \quad = \quad \sum_{s,t} p(s) \cdot p(t) \cdot d_{\mathrm{BP}}(s,t) \tag{6.1}$$

i.e. intrinsic distance is another name for *ensemble diversity* earlier defined in [210], and computed by `Vienna RNA Package` [139]. *Plasticity* is defined in [207] to be *normalized ensemble diversity*; i.e.

$$P(\mathbf{a}) \quad = \quad \frac{d_0(\mathbf{a})}{n/2} \tag{6.2}$$

obtained by dividing ensemble diversity by (essentially) the maximum possible number $n/2$ of base pairs in a structure of $\mathbf{a}$. Given two RNA sequences $\mathbf{a} = a_1,\ldots,a_n$ and $\mathbf{b} = b_1,\ldots,b_n$ of the same length n, Rodrigo et al. define $d_1(\mathbf{a},\mathbf{b})$ to be the expected base pair distance between structures of $\mathbf{a}$ and structures of $\mathbf{b}$ minus the ensemble diversity of $\mathbf{a}$, i.e.

$$d_1(\mathbf{a},\mathbf{b}) \quad = \quad \sum_{s \in \mathbb{SS}(\mathbf{a})} \sum_{t \in \mathbb{SS}(\mathbf{b})} p_{\mathbf{a}}(s) \cdot p_{\mathbf{b}}(t) \cdot d_{\mathrm{BP}}(s,t) - d_0(\mathbf{a}). \tag{6.3}$$

Since d_1 is not symmetric, this measure is not a metric. In contrast, *ensemble distance* as defined in [210] is a valid metric, defined by the following:

$$D_V(\mathbf{a},\mathbf{b}) = \sqrt{\sum_{s\in\mathbb{SS}(\mathbf{a})}\sum_{t\in\mathbb{SS}(\mathbf{b})} p_{\mathbf{a}}(s)\cdot p_{\mathbf{b}}(t)\cdot d_{BP}(s,t) - \frac{d_0(\mathbf{a})+d_0(\mathbf{b})}{2}} \tag{6.4}$$

$$= \sqrt{\sum_{i<j}(p_{i,j}(a)-p_{i,j}(b))^2} \tag{6.5}$$

In [207], Rodrigo et al. define the *mutational robustness*

$$R_m(\mathbf{a}) = 1 - \frac{\langle d_1(\mathbf{a},\mathbf{a}')\rangle}{n/2} \tag{6.6}$$

where $\langle d_1(\mathbf{a},\mathbf{a}')\rangle$ denotes the average value of $d_1(\mathbf{a},\mathbf{a}')$ taken over all single point mutants $\mathbf{a}'$ of $\mathbf{a}$. Since $d_1(\mathbf{a},\mathbf{a}')$ is not a true metric, we replace it by the metric $D_V(\mathbf{a},\mathbf{b})$ in our computation of mutational robustness. Clearly both notions are closely related.

6.3 Algorithm description

In [175], McCaskill described a cubic time algorithm to compute the *partition function*

$$Z = Z(\mathbf{a}) = \sum_{s\in\mathbb{SS}(\mathbf{a})} \exp(-E(\mathbf{a},s)/RT) \tag{6.7}$$

for an RNA sequence $\mathbf{a} = a_1,\ldots,a_n$, where the sum is taken over all secondary structures $\mathbb{SS}(\mathbf{a})$ of $\mathbf{a}$, $E(\mathbf{a},s)$ denotes the free energy for the structure s of $\mathbf{a}$ with respect to the Turner energy parameters [59], R denotes the universal gas constant and T is absolute temperature. Subsequently Ding and Lawrence [190] described how to use the partition function together with a simple backtracking strategy to *sample* secondary structures of $\mathbf{a}$ from the Boltzmann ensemble of low energy structures.

If s_0 is a given secondary structure of length n, we define the *dual partition function*

$$Z^* = Z^*(s_0) = \sum_{\mathbf{a} \in \mathbb{AA}(s_0)} \exp(-E(\mathbf{a},s_0)/RT) \tag{6.8}$$

where the sum is taken over all RNA sequences $\mathbf{a} = a_1,\ldots,a_n$ of length n, denoted by $\mathbb{AA}(s_0)$. Note that if a sequence $\mathbf{a}$ is not compatible with the target structure s_0, then the energy $E(\mathbf{a},s_0)$ is infinite, so the corresponding Boltzmann factor $\exp(-E(\mathbf{a},s_0)/RT)$ is zero. Here we describe the efficient software `RNAdualPF` to compute the *dual partition function* Z^* and to sample from the low energy ensemble of *sequences* that are compatible with a given secondary structure s_0.

6.3.1 Dual partition function

If s is a secondary structure on sequence $\mathbf{a} = a_1,\ldots,a_n$, then the *length* of s, denoted by $\mu(s)$ or sometimes simply by μ, is equal to n, while the *size* of s, denoted by $|s|$, is the number of base pairs belonging to s. Similarly, if secondary structure s is restricted to the interval $[i,j]$, where $1 \le i \le j \le n$, then the length of the restriction of s to $[i,j]$, denoted by $\mu(s[i,j])$, is equal to $j - i + 1$, while the size of the restriction of s to $[i,j]$, denoted by $|s[i,j]|$, is the number of base pairs (x,y) of s that satisfy $i \le x < y \le j$.

Given an RNA sequence $\mathbf{a} = a_1,\ldots,a_n$, the McCaskill algorithm [175] computes the partition function $Z(\mathbf{a})$ defined in equation 6.7. When $\mathbf{a}$ is clear from context, $Z(\mathbf{a})$ is usually denoted by Z.

Given a target secondary structure s_0, we describe below an algorithm to compute the *dual partition function* $Z^*(s_0)$, defined as the sum of all Boltzmann factors $\exp(-E(\mathbf{a},s_0))$, where the sum is taken over all RNA sequences $\mathbf{a} \in \mathbb{AA}(s_0)$. Unlike the McCaskill algorithm, which

requires time that is cubic in the length of $\mathbf{a}$, the algorithm presented below requires time that is (essentially) linear[1] in the length of s_0. Our algorithm is motivated by the initialization step of the algorithm INFO-RNA [41], in which a sequence is determined, for which the free energy with respect to target structure s_0 is a minimum – i.e. `INFO-RNA` determines $\text{argmin}_{\mathbf{a}} E(\mathbf{a},s_0)$.

The algorithm specification requires the notation $Z^*(i,j;x,y)$, which denotes the sum

$$Z^*(i,j;x,y) = \sum_{\mathbf{a}[i,j],a_i=x,a_j=y} \exp(-E(\mathbf{a}[i,j],s_0[i,j])/RT) \tag{6.9}$$

of Boltzmann factors for sequences $\mathbf{a}[i,j] = a_i,\ldots,a_j$ for which $a_i = x, a_j = y$, and for the restriction $s_0[i,j]$, defined by

$$s_0[i,j] = \{(x,y) \in s_0 : i \le x < y \le j\}. \tag{6.10}$$

The function $Z^*(i,j;x,y)$ will be defined for all base pairs $(i,j) \in s_0$; these values will be stored in an array, whose rows index base pairs of s_0, and whose columns are indexed by the six canonical base pairs GC, CG, AU, UA, GU, UG (see example in Table 6.1). Once $Z^*(i,j;x,y)$ has been computed for all base pairs that are *visible*, for which there is no base pair (x,y) for which $x < i < j < y$, we can compute the full partition function $Z^*(s_0)$.

Following [41], we define a total ordering on base pairs (i,j) belonging to the target structure s_0 that satisfy following precedence rule for any two base pairs $(i,j),(x,y)$.

$$(i,j) \prec (x,y) \Leftrightarrow x < i < j < y \text{ or } i < j < x < y \tag{6.11}$$

[1] When dangling positions are not included in the computation (-do), the algorithm clearly requires linear time. When dangling positions are included (-d2), run time is exponential in the number of components of the largest multilooop; however, it is possible to modify the algorithm so that even this case takes linear time.

From this ordering, we assign a *base pair index* to each base pair (i,j), which is defined to be the rank of (i,j) in the total ordering.

The following definitions correspond to the Turner nearest neighbor energy model, which is an additive loop model described in Chapter 1. Recall that in this energy model, a loop closed by external base pair (i,j) is designated as a k-loop, if it contains k base pairs interior to (i,j). Therefore, hairpin loops are 0-loops; base pair stacks, bulge loops and internal loops are 1-loops; and multiloops are k-loops for $k \geq 2$ (also called $(k+1)$-way junctions), where the additional count is due to the outer component adjacent to (i,j)[9]. We reintroduce the following notation, where $\mathcal{N}$ denotes the set of nucleotides {A, U, G, C} which can be assigned to a position i in the sequence $\mathbf{a}$; and $\mathcal{B}$ denotes the set {AU,UA,GC,CG,GU,UG} corresponding to the possible combinations of nucleotides that constitute a Watson-Crick or GU wobble pair in a base pair (i,j).

Since AU-base pairs that close a loop are energetically unfavorable, in the Turner energy model, there is an AU-penalty we now define.

$$e_{AU}(i,j,X,Y) = \begin{cases} 0.5 & \text{if } (i,j) \text{ is the outermost pair in a stem of } s_0\text{, having AU,UA,GU,UG} \\ 0 & \text{otherwise.} \end{cases}$$

This AU-penalty is applied only if (i,j) is a base pair adjacent to a triloop, a bulge, an internal loop or a multiloop, or if it is the outermost base pair of an external loop in target structure s_0; and if the pair (i,j) has one of the nucleotides AU, UA, GU, UG. When base-paired positions i,j are clear from the context, we write $e_{AU}(X,Y)$.

Here, we assume that in parsing the input target structure, a list *BPcloseELorML* has been created of those base pairs (i,j), which close either an external loop or a multiloop. Let I

be the indicator function, it follows that if (i,j) closes an external loop or multiloop, then $\exp\left(-\frac{I[(i,j)\in BPcloseELorML]\cdot e_{AU}(X,Y)}{RT}\right)$ is the Boltzmann factor for a special AU-penalty, otherwise this factor equals 1. For clarity in the notation, this factor is denoted by $e^{(-\frac{e^I_{AU}(X,Y)}{RT})}$. Note that this term is different from the factor $\exp(-\frac{e_{AU}(X,Y)}{RT})$ applied to base pairs adjacent to a triloop, a bulge or an internal loop, which does not depend on the indicator function.

6.3.1.1 Hairpins

Let (i,j) close a hairpin in s_0. The hairpin free energy term $H(j-i-1)$, arising solely from entropic considerations, is defined by

$$H(j-i-1) = \begin{cases} hairpinE(j-i-1) & \text{if } j-i-1 \leq 30 \\ hairpinE(30) + 1.75RT\ln\left(\frac{j-i-1}{30}\right) & \text{otherwise} \end{cases}$$

where $hairpinE(j-i-1)$ designates the hairpin free energy obtained from table look-up, when $j-i-1 \leq 30$.

Triloop Let $TriLoop_{x,y}$ denote the collection of special triloops, $xabcy$, having an energy bonus $triloopE(xabcy)$.

$$\begin{aligned} Z^*(i,j;x,y) &= e^{(-\frac{e^I_{AU}(x,y)}{RT})} \cdot \exp(-\frac{H(j-i-1)+e_{AU}(xy)}{RT}) \cdot \\ &\left((4^3 - |TriLoop_{x,y}|) + \sum_{abc\in TriLoop_{x,y}} \exp(-\frac{triloopE(xabcy)}{RT})\right) \end{aligned}$$

Tetraloop Let $TetraLoop_{x,y}$ denote the collection of special tetraloops, $xabcdy$, having an energy bonus $tetraloopE(xabcdy)$. Similarly, given nucleotides $n_1,n_2 \in \mathcal{N}$, $TetraLoop_{x,y}(n_1,n_2)$ denotes the collection of special tetraloops of the form xn_1abn_2y.

$$Z^*(i,j;x,y) = e^{(-\frac{e^I_{AU}(x,y)}{RT})} \cdot \exp(-\frac{H(j-i-1)}{RT}) \cdot \sum_{n_1,n_2 \in \mathcal{N}} \left(\exp(-\frac{mismatch(x,y,n_1,n_2)}{RT}) \cdot \left((4^2 - |TetraLoop_{x,y}(n_1,n_2)|) + \sum_{ab \in TetraLoop_{x,y}(n_1,n_2)} \exp(-\frac{tetraloopE(xn_1abn_2y)}{RT}) \right) \right)$$

Hexaloop Let $HexaLoop_{x,y}$ denote the collection of special hexaloops, $xabcdefy$, having an energy bonus $hexaloopE(xabcdefy)$. Similarly, given nucleotides n_1,n_2, $HexaLoop_{x,y}(n_1,n_2)$ denotes the collection of special hexaloops of the form xn_1abcdn_2y.

$$Z^*(i,j;x,y) = e^{(-\frac{e^I_{AU}(x,y)}{RT})} \cdot \exp(-\frac{H(j-i-1)}{RT}) \cdot \sum_{n_1,n_2 \in \mathcal{N}} \left(\exp(-\frac{mismatch(x,y,n_1,n_2)}{RT}) \cdot \left((4^4 - |HexaLoop_{x,y}(n_1,n_2)|) + \sum_{ab \in HexaLoop_{x,y}(n_1,n_2)} \exp(-\frac{HexaloopE(xn_1abcdn_2y)}{RT}) \right) \right)$$

Hairpin size exceeds four and is different than six

$$Z^*(i,j;x,y) = e^{(-\frac{e^I_{AU}(x,y)}{RT})} \cdot \exp(-\frac{H(j-i-1)}{RT}) \cdot \left(\sum_{n_1,n_2 \in \mathcal{N}} \exp(-\frac{mismatch(x,y,n_1,n_2)}{RT}) \cdot 4^{j-i-3} \right)$$

6.3.1.2 Stacked base pairs, bulges and internal loops

Here, we consider the case of a 1-loop, which comprises the case of stacked base pairs, bulges and internal loops. The following cases correspond to each possibility.

Stacked base pair In this case, (i,j) stacks on the base pair $(i+1,j-1)$, and the partition function $Z^*(i+1,j-1;U,V)$ has been computed. Let $stack(X,Y,U,V)$ denote the free energy of base stack $\begin{matrix} 5'\text{-}\mathtt{XU}\text{-}3' \\ 3'\text{-}\mathtt{YV}\text{-}5' \end{matrix}$ obtained by table look-up.

$$Z^*(i,j;X,Y) \quad = \quad e^{(-\frac{e^I_{AU}(X,Y)}{RT})} \cdot \sum_{UV \in \mathcal{B}} \exp(-\frac{stack(X,Y,U,V)}{RT}) \cdot Z^*(i+1,j-1,U,V)$$

Bulge loop In this case, (i,j) closes a bulge in s_0. Since bulge size may exceed the values in table look-up, we define the free energy for a bulge of size r by

$$bulge(r) = \begin{cases} bulgeE(r) & \text{if } r \leq 30 \\ bulgeE(30) + 1.75RT \ln\left(\frac{r}{30}\right) & \text{otherwise.} \end{cases}$$

If (i,j) closes a left bulge of size r in s_0, then the bulge is closed by base pair $(i+r+1,j-1)$ involving nucleotide pair U,V, and

$$Z^*(i,j;X,Y) = e^{(-\frac{e^I_{AU}(X,Y)}{RT})} \cdot \sum_{UV \in \mathcal{B}} \exp(-\frac{e_{AU}(i,j,X,Y)}{RT}) \cdot \exp(-\frac{bulge(r)}{RT}) \cdot 4^r \cdot Z^*(i+r+1,j-1,U,V)$$

while if (i,j) closes a right bulge in s_0, then the bulge is closed by base pair $(i+1,j-r-1)$ involving nucleotide pair U,V, and

$$Z^*(i,j;X,Y) = e^{(-\frac{e^I_{AU}(X,Y)}{RT})} \cdot \sum_{UV \in \mathcal{B}} \exp(-\frac{e_{AU}(i,j,X,Y)}{RT}) \cdot \exp(-\frac{bulge(r)}{RT}) \cdot 4^r \cdot Z^*(i+1,j-r-1,U,V)$$

Internal loop In this case, (i,j) closes an internal loop in s_0, whose left [resp. right] portion is of size r_1 [resp. r_2]. Since internal loop size $r = r_1 + r_2$ may exceed the values in table look-up, we define the free energy for an internal loop of size r by

$$internal(r) = \begin{cases} internalE(r) & \text{if } r \leq 30 \\ internalE(30) + 1.75RT \ln\left(\frac{r}{30}\right) & \text{otherwise.} \end{cases}$$

The closing base pair $(i+r_1+1,j-r_2-1)$ of the internal loop of size $r = r_1 + r_2$ may involve the nucleotides $UV \in \mathcal{B}$, while the unpaired (mismatch) nucleotides in positions $i+1,j-1,i+r_1,j-r_2$

may involve $A,B,C,D \in \mathcal{N}$. In addition, there is an energy penalty for non symmetric internal loops, $min(asym \cdot |r_1 - r_2|, maxAsym)$, where the value of the constants $asym$ and $maxAsym$ are given in the Turner energy model. Thus

$$\begin{aligned} Z^*(i,j;X,Y) \quad = \quad & e^{(-\frac{e^I_{AU}(X,Y)}{RT})} \cdot \exp(-\frac{min(asym \cdot |r_1 - r_2|, maxAsym)}{RT}) \cdot \\ & \cdot \sum_{UV \in \mathcal{B}} \sum_{A,B,C,D \in \mathcal{N}} \exp(-\frac{e_{AU}(i,j,X,Y)}{RT}) \cdot \exp(-\frac{internal(r_1 + r_2)}{RT}) \cdot 4^{r_1+r_2-4} \cdot \\ & \exp(-\frac{mismatch(X,Y,A,B) + mismatch(V,U,D,C)}{RT}) \cdot Z^*(i + r_1 + 1, j - r_2 - 1, U, V) \end{aligned}$$

6.3.1.3 External loop

Despite the fact that, by following the total order on base pairs defined in equation 6.11, the *dual partition function* of multiloops is always computed before the *dual partition function* of the external loop, the computation of the *dual partition function* of multiloops will be easier to understand if the *dual partition function* of the external loop is defined in advance.

In order to improve speed, some implementations of RNA thermodynamics-based algorithms ignore the contribution of dangling positions, which corresponds to `Vienna RNA Package -d0` flag. `RNAdualPF` also includes this option, which dramatically increases the speed of the algorithm (see benchmarking in Section 6.4 below). The reason behind this difference of performance is clear from the following definitions.

Suppose that $H = [(i_1,j_1),\ldots,(i_k,j_k)]$ constitutes the list of k external base pairs of s_0, where $i_1 < j_1 < i_2 < j_2 < \cdots < i_k < j_k$. For each (i_r,j_r), with $1 \leq r \leq k$, and for each choice of base pair GC, CG, AU, UA, GU, UG, the value $Z^*(i_r,j_r;X_r,Y_r)$ has been previously computed and stored by dynamic programming, as well as the sum $Z^*(i_r,j_r)$. When the contribution of

dangles is ignored, the *dual partition function* of an external loop with ℓ nucleotide positions external to every base pair is defined by

$$Z^*(s_0) \quad = \quad 4^{\ell} \cdot \prod_{r=1}^{k} Z^*(i_r,j_r) \tag{6.12}$$

where $\ell = n - \sum_{r=1,\ldots,k} (j_r - i_r + 1)$ and n is the length of the target structure s_0.

The default treatment of dangles in `RNAdualPF` described below corresponds to `Vienna RNA Package -d2` flag, where both flanking positions of each external base pair contribute to the free energy. Let $D = [a_1,b_1,\ldots,a_k,b_k] \subseteq [i_1 - 1,j_1 + 1,\cdots,i_k,j_k]$ be a list of those nucleotide positions that are adjacent to the k external base pairs $(i_1,j_1),\ldots,(i_k,j_k)$. The ordered multiset $[a_1,b_1,\ldots,a_k,b_k]$ can be considered as a collection of constraints, so that (for instance) if $a_2 = i_2 - 1$, and $a_2 = j_1 + 1$, then $a_2 = b_1$ and any nucleotide value that is assigned to b_1 must simultaneously be assigned to a_2. Moreover, there can also be an overlap between the list of base paired positions in H $[i_1,j_1,\ldots,i_k,j_k]$ and the multiset $D = [a_1,b_1,\ldots,a_k,b_k]$. If (for instance) $j_1 = i_2 - 1$, then $b_1 = i_2$ and $a_2 = j_1$. Therefore, in the computation we have to account for these constraints. Let m denote the number of unpaired positions in D, without repetitions, and define A_r,B_r as the nucleotides instantiated respectively at a_r,b_r. The energy term for a 5′-dangle [resp. 3′-dangle] on base pair (x,y) with nucleotides U,V is denoted by $E_{d_5}(x,y,x-1;U,V,W)$ [resp $E_{d_3}(x,y,y+1;U,V,W)$] where the dangle position $x-1$ [resp. $y+1$] is assigned nucleotide W. With the notation just described, we have

$$Z^*(s_0) = \sum_{\langle (U_1,V_1),\ldots,(U_k,V_k)\rangle \in \mathcal{B}^k} \sum_{\{A_1,B_1,\ldots,A_k,B_k \in \mathcal{N}^{2k}\}} 4^{\ell-m} \cdot \prod_{r=1}^{k} \left(Z^*(i_r,j_r;U_r,V_r) \cdot \exp(-\frac{E_{d_5}(i_r,j_r,a_r;U_r,V_r,A_r) + E_{d_3}(i_r,j_r,b_r;U_r,V_r,B_r)}{RT} \right) \tag{6.13}$$

Depending on the target structure s_0, it can happen that the second sum of equation 6.13 must be restricted to range over strictly less than 4^{2k} many RNA sequences. This is explained as follows. If $i_1 = 1$ [resp. $j_r = n$] then there is no position for a 5′ [resp. 3′] dangle, and hence the nucleotide sequences considered in the second summation would have length strictly less than $2k$. Moreover, certain 5′ dangled positions could be identical to 3′ dangle positions, which arises for instance when $j_k + 2 = i_{k+1}$; alternatively, certain dangled positions could be identical with base-paired positions, which arises for instance when $j_k + 1 = i_{k+1}$. In such situations, instantiations of the 3′-dangle on (i_k,j_k) and the 5′-dangle on (i_{k+1},j_{k+1}) are not independent, thus leading to a restriction of the range of the second summation in equation 6.13. A similar restriction is implicitly assumed in the treatment of external loops in this section and of multiloops in the next section.

The algorithm performance can be improved by dividing the external loop into groups of components having interdependently constrained dangling positions, as just explained. Define two base pairs (x,y),(x',y') as adjacent if $x < y < x' < y'$ and $x' - y \leq 2$ – i.e. dangling positions of the base pairs (x,y),(x',y') are constrained. Let G denote a *maximal* collection of *adjacent* base pairs belonging to $H = [(i_1,j_1),\ldots,(i_k,j_k)]$, together with their associated dangle positions in $D = [i_1 - 1, j_1 + 1, \ldots, i_k - 1, j_k + 1]$. It is important to note that $H \cup D$ is thus partitioned into a collection of g disjoint groups $\mathcal{G} = [G_1,\ldots,G_g]$. Therefore, we can divide an external loop of k helices into a collection groups $\mathcal{G}$ of size $g \leq k$, and p unpaired positions that are external to

every base pair of s_0 and not adjacent to any base pair.

For a group G with h base pairs, let $H(G) = [(\kappa_1,\lambda_1),\ldots,(\kappa_k,\lambda_k)]$ denote the list of base pairs in G, and let $D(G) = [\alpha_1,\beta_1,\ldots,\alpha_h,\beta_h] \subseteq [\kappa_1 - 1,\lambda_1 + 1,\cdots,\kappa_h - 1,\lambda_h + 1]$ denote their associated dangle positions. If U_r,V_r,A_r,B_r denote the nucleotides instantiated at the base pair $r = (\kappa_r,\lambda_r)$ and its respective dangling positions α_r,β_r respectively, then the *dual partition function* of G is the following.

$$Z^*(G) = \sum_{\langle (U_1,V_1),\ldots,(U_h,V_h)\rangle \in \mathcal{B}^h} \sum_{\{A_1,B_1,\ldots,A_h,B_h \in \mathcal{N}^{2h}\}} \prod_{r=1}^{h} \left(Z^*(\kappa_r,\lambda_r;U_r,V_r) \cdot \exp(-\frac{E_{d_5}(\kappa_r,\lambda_r,\alpha_r;U_r,V_r,A_r) + E_{d_3}(\kappa_r,\lambda_r,\beta_r;U_r,V_rB_r)}{RT} \right) \tag{6.14}$$

where the range of the second summation can be constrained by the overlap among positions in $D(G)$ and between positions in $D(G)$ and $H(G)$, as explained for equation 6.13.

Finally, since there are no shared dangling positions between groups, the *dual partition function* of an external loop is defined by

$$Z^*(s_0) = 4^p \cdot \prod_{r=1}^{g} Z^*(G_r). \tag{6.15}$$

6.3.1.4 Multiloop

Suppose that (i,j) closes a multiloop in s_0, which is a k-loop, or $(k+1)$-way junction, for $k > 1$, where there are ℓ unpaired bases in the multiloop. Suppose that the k components of the multiloop are closed by the base pairs $(i_1,j_1),\ldots,(i_k,j_k)$ with the property that $i < i_1 < j_1 < i_2 <$

$j_2 < \cdots < i_k < j_k < j$. For all nucleotide choices in $\mathcal{B}$ for each (i_r,j_r), for $1 \leq r \leq k$, the value $Z^*(i_r,j_r;X_r,Y_r)$ has been previously computed and stored by dynamic programming, as well as the sum $Z^*(i_r,j_r)$. The computation of the *dual partition function* is similar to that of the external loop. However, in this case we have to add the contribution of the base pair closing the multiloop (i,j), the AU-penalties applied to this base pair, and the energetic penalty of a multiloop $a + b \cdot (k+1) + c \cdot \ell$, where the values of the constants a, b and c are given in the Turner energy model. Then, the *dual partition function* of a multiloop without accounting for dangling positions is

$$\begin{aligned} Z^*(i,j;X,Y) \quad = \quad & e^{(-\frac{e^I_{AU}(X,Y)}{RT})} \cdot \exp(-\frac{a + b \cdot (k+1) + c\ell}{RT}) \cdot 4^{\ell} \cdot \\ & \exp(-\frac{e_{AU}(i,j,X,Y)}{RT}) \cdot \sum_{\langle (U_1,V_1),\ldots,(U_k,V_k)\rangle \in \mathcal{B}^k} \prod_{r=1}^{k} Z^*(i_r,j_r;U_r,V_r) \end{aligned} \tag{6.16}$$

The terminology to define the *dual partition function* of multiloops with dangling positions is similar to that described for external loops. However, it requires some modifications in the previous definitions, since we have to take into account the flanking positions of the base pair (i,j) closing the multiloop. Let $H = [(i_1,j_1),\ldots,(i_k,j_k),(i,j)]$ be the collection of k base pairs closing one of the k components of the multiloop, and the base pair (i,j) closing the multiloop, and define the multiset $D = [a_1,b_1,\ldots,a_{k+1},b_{k+1}] \subseteq [i_1 - 1, j_1 + 1, \cdots, i_k - 1, j_k + 1, i + 1, j - 1]$ of nucleotide positions adjacent to the base pairs in H. Due to the possible overlap with the base pair closing the multiloop and its flanking positions, there are additional constraints in the ordered multiset $[a_1,b_1,\ldots,a_{k+1},b_{k+1}]$, so that (for instance) if $a_1 = i_1 - 1$, and $i_1 = i + 1$, then $a_1 = a_{k+1}$ and any nucleotide value that is assigned to a_1 must simultaneously be assigned to a_{k+1}. Moreover, there can also be an overlap between the list of base paired positions $[i_1,j_1,\ldots,i_k,j_k,i,j]$ and the multiset $[a_1,b_1,\ldots,a_{k+1},b_{k+1}]$. If (for instance) $i = i_1 - 1$,

then $a_{k+1} = i_1$ and $a_1 = i$.

Let m denote the number of unpaired positions in D, without repetitions. Then, the *dual partition function* of a multiloop with dangling positions is defined as follows.

$$
\begin{aligned}
Z^*(i,j;X,Y) \quad = \quad & e^{(-\frac{e^I_{AU}(X,Y)}{RT})} \cdot \sum_{\langle (U_1,V_1),\ldots,(U_k,V_k)\rangle \in \mathcal{B}^k} \sum_{\{A_1,B_1,\ldots,A_{k+1},B_{k+1} \in \mathcal{N}^{2(k+1)}\}} \\
& \exp(-\frac{a + b \cdot (k+1) + c\ell}{RT}) \cdot 4^{\ell - m} \cdot \exp(-\frac{e_{AU}(i,j,X,Y)}{RT}) \cdot \\
& \prod_{r=1}^{k} \left(Z^*(i_r,j_r;U_r,V_r) \cdot \exp(-\frac{E_{d_5}(i_r,j_r,a_r;U_r,V_r,A_r) + E_{d_3}(i_r,j_r,b_r;U_r,V_r,B_r)}{RT} \right) \cdot \\
& \exp(-\frac{E_{d_3}(j,i,a_{k+1};Y,X,A_{k+1}) + E_{d_5}(j,i,b_{k+1};Y,X,B_{k+1})}{RT})
\end{aligned}
\tag{6.17}
$$

As explained for equation 6.13, it can happen that the second summation must be restricted to range over strictly less than 4^{2k} many RNA sequences.

A decomposition similar to the one described for external loops can be performed to improve the performance in the computation of the *dual partition function* of a multiloop. In a multiloop, in addition to the adjacency definition given for external loops, we consider the base pair (i,j) that closes the multiloop as adjacent to a base pair (x,y) that closes a component of the multiloop, where $i < x < y < j$, if either $x \leq i + 2$ or $y \geq j - 2$. Then, let G denote a *maximal* collection of *adjacent* base pairs belonging to $H = [(i_1,j_1),\ldots,(i_k,j_k),(i,j)]$, together with their associated dangle positions in $D = [i_1 - 1, j_1 + 1, \ldots, i_k - 1, j_k + 1, i + 1, j - 1]$. This decomposition produces a collection $\mathcal{G}$ of g disjoint groups $G_1,\ldots,G_g$, one of which, designated the *closing group* G_c contains the closing base pair (i,j) of the multiloop, and $g - 1$ of which, designated as *non-closing groups* G_{nc}, do not contain the base pair (i,j).

Non-closing groups have the same composition as those defined for external loops – i.e. a collection of h base pairs $H(G_{nc}) = [(\kappa_1,\lambda_1),\ldots,(\kappa_h,\lambda_h)]$ and a set of dangling positions $D(G_{nc}) = [\alpha_1,\beta_1,\ldots,\alpha_h,\beta_h] \subseteq [\kappa_1 - 1,\lambda_1 + 1,\cdots,\kappa_h - 1,\lambda_h + 1]$. Therefore, we can compute the *dual partition function* $Z(G_{gc})$ of a *non-closing group* as described in equation 6.14. In addition, the collection of *non-closing groups* of size $g-1$ of a multiloop of k components is denoted by $\mathcal{G}_{nc}$, where $0 \leq (g-1) \leq k$.

Therefore, a multiloop of k components and ℓ unpaired positions can be decomposed into one closing group G_c, a collection of non-closing groups $\mathcal{G}_{nc}$, and p unpaired positions that are not adjacent to any base pair, with $0 \leq p \leq \ell$.

In a *non-closing group*, the collection of base pairs of size $h+1$ is denoted by $H(G_c) = [(\kappa_1,\lambda_1),\ldots,(\kappa_h,\lambda_h),(i,j)]$, where the base pair (i,j) closing the multiloop is at the last position. The ordered multiset of adjacent positions is denoted by $D(G_c) = [\alpha_1,\beta_1,\ldots,\alpha_{h+1},\beta_{h+1}] \subseteq [\kappa_1 - 1,\lambda_1 + 1,\cdots,\kappa_h - 1,\lambda_h + 1,i+1,j-1]$, where the positions adjacent to i and j are at the last positions are respectively denoted by α_{h+1},β_{h+1}. A graphical example of a *closing group* and a *non-closing group* is shown in Figure 6.2e, where the positions of a *non-closing group* with 1 base pair are highlighted in green and the positions of the *closing group* are highlighted in red and blue, and where the base pair (i,j) that closes the multiloop is depicted in red.

For a *closing group* G_c with $h+1$ base pairs in $H(G_c) = [(\kappa_1,\lambda_1),\ldots,(\kappa_h,\lambda_h),(i,j)]$ and their flanking positions $D(G_c) = [\alpha_1,\beta_1,\ldots,\alpha_{h+1},\beta_{h+1}] \subseteq [\kappa_1 - 1,\lambda_1 + 1,\cdots,\kappa_h - 1,\lambda_h + 1,i+1,j-1]$, let X,Y denote the nucleotides assigned to the closing base pair of the multiloop (i,j), and let U_r,V_r,A_r,B_r denote the nucleotides assigned respectively to the base pair $r = (\kappa_r,\lambda_r,)$ and its flanking positions α_r,β_r. Then, the the *dual partition function* of the *closing group* is defined by

$$
\begin{aligned}
Z^*(G_c;X,Y) \quad = \quad & e^{(-\frac{e^I_{AU}(X,Y)}{RT})} \cdot \sum_{\langle (U_1,V_1),\ldots,(U_k,V_h)\rangle \in \mathcal{B}^h} \sum_{\{A_1,B_1,\ldots,A_{h+1},B_{h+1} \in \mathcal{N}^{2(h+1)}\}} \exp(-\frac{e_{AU}(i,j,X,Y)}{RT}) \cdot \\
& \prod_{r=1}^{h} \left(Z^*(\kappa_r,\lambda_r;U_r,V_r) \cdot \exp(-\frac{E_{d5}(\kappa_r,\lambda_r,\alpha_r;U_r,V_r,A_r) + E_{d3}(\kappa_r,\lambda_r,\beta_r;U_r,V_r,B_r)}{RT} \right) \cdot \\
& \exp(-\frac{E_{d3}(j,i,\alpha_{h+1};Y,X,A_{h+1}) + E_{d5}(j,i,\beta_{h+1};Y,X,B_{h+1})}{RT}) \qquad (6.18)
\end{aligned}
$$

In the same way as in equation 6.13, the values of the second summation are constrained to the possible choices among overlapping positions.

Then, the *dual partition function* $Z^*(i,j;X,Y)$ of the multiloop with k components and ℓ unpaired positions, where p of which are not adjacent to any base pair, is defined by

$$
Z^*(i,j;X,Y) \quad = \quad \exp(-\frac{a + b \cdot (k+1) + c\ell}{RT}) \cdot 4^p \cdot Z^*(G_c;X,Y) \cdot \prod_{G_{nc} \in \mathcal{G}_{nc}} Z^*(G_{nc}) \qquad (6.19)
$$

6.3.2 Sampling

Once the *dual partition function* $Z^*(i,j)$ and its subcases $Z^*(i,j;X,Y)$ for each base pair (i,j) have been computed, it possible to perform a Boltzmann weighted sampling of positions i and j. For example, given the target structure with sequence constraints depicted in Figure 6.1, `RNAdualPF` computes the *dual partition function* table shown in Table 6.1. The *dual partition function* of the substructure enclosed by the base pair (i,j) is $Z^*(i,j)$, and the *dual partition function* of the substructure enclosed by the base pair (i,j) given the nucleotides G,C at positions i,j respectively is $Z^*(i,j;G,C)$. Therefore, the Boltzmann probability of G,C at positions i,j in

the substructure enclosed by the base pair (i,j) is $Z^*(i,j;G,C)/Z^*(i,j)$ and can be sampled using the roulette wheel method.

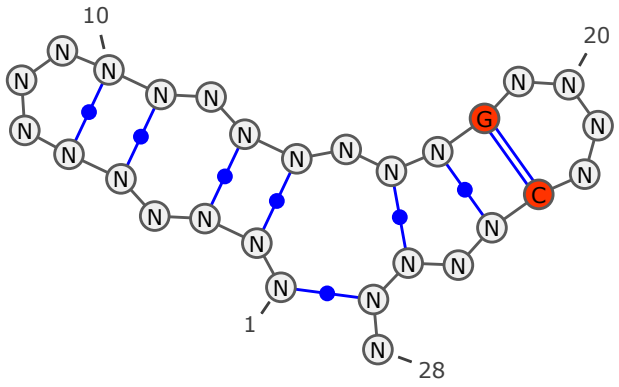


FIGURE 6.1: Target structure with sequence constraints used as input of `RNAdualPF` to compute the *dual partition function* values shown in Table 6.1. Sequence constraints are highlighted in red.

Index	i	j	Type	$Z^*(i,j;A,U)$	$Z^*(i,j;C,G)$	$Z^*(i,j;G,C)$	$Z^*(i,j;U,A)$	$Z^*(i,j;G,U)$	$Z^*(i,j;U,G)$	$Z^*(i,j)$
1	18	23	Tetraloop	0.000	0.000	0.364	0.000	0.000	0.000	0.364
2	17	24	Stack	10.977	17.859	76.923	10.977	10.977	3.525	131.238
3	16	26	R. bulge	11.690	70.834	184.603	12.771	13.347	3.915	297.160
4	6	10	Triloop	0.004	0.010	0.010	0.004	0.004	0.004	0.038
5	5	11	Stack	0.750	3.022	5.234	0.899	0.960	0.256	11.120
6	3	13	Int. loop	109.842	256.875	424.976	108.653	117.851	108.132	1126.330
7	2	14	Stack	10853.104	86208.448	170643.321	12575.544	13285.398	3647.077	297212.891
8	1	27	Multiloop	1558.575	7895.583	7895.583	1558.575	1558.575	1558.575	22025.464
9	1	28	S_0	–	–	–	–	–	–	88101.856

TABLE 6.1: Base pair dual partition function table. Given the target structure with sequence constraints depicted in Figure 6.1, `RNAdualPF` computes and stores all the partial *dual partition function* values for the substructures enclosed by each base pair. The first column indicates the *base pair index* which dictates the order in which the dual partition function is computed for different loops closed by the base pair (i,j), where we the *index* of base pair (i,j) is defined to be the rank of (i,j) in the total ordering defined in equation 6.11. Columns i and j indicate the opening and closing positions of each base pair. Type indicates the type of element in the secondary structure closed by each base pair, where R. bulge stands for right bulge, Stack for stacking base pair, and Int. loop for interior loop. The *dual partition function* $Z^*(i,j)$ of the substructure closed by base pair (i,j) appears in the rightmost column, while the partition function $Z^*(i,j,X,Y)$ for each of the six canonical base pairs is given in columns 5-10. Note that for base pair 1, sequence constraints depicted in Figure 6.1 force i and j to be instantiated respectively to G and C, hence the dual partition function $Z^*(i,j;X,Y)$ is zero for any base pair different than GC. The last column of the last row of the table shows the total dual partition function $Z^*(s_0)$ for the target structure s_0.

Due to the Turner energy model, it is necessary to determine nucleotide positions whose instantiation influences the energy (hence Boltzmann probability) of other positions, and subsequently those positions of mutual dependency must be instantiated together. Figure 6.2 illustrates the mutual dependencies that must be considered when sampling different types of elements, where the base pair (i,j) to be sampled is highlighted in red, positions whose sampling probability is dependent on the instantiation of (i,j) are highlighted in blue, and positions that are mutually dependent, but independent of the instantiation of (i,j), are highlighted in green.

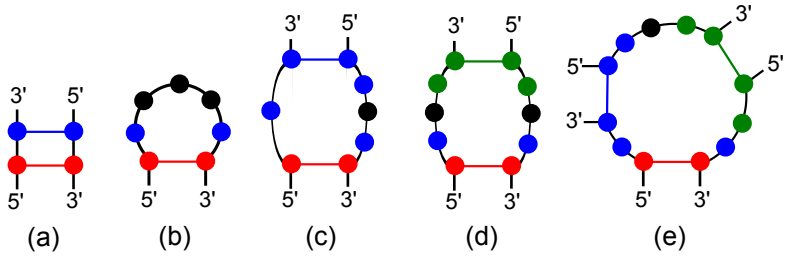


FIGURE 6.2: Sampling dependency examples in `RNAdualPF`. Base pair (i,j) to be sampled is highlighted in red, positions whose energy contribution is dependent on the instantiation of (i,j) are highlighted in blue, and positions that are mutually dependent, but independent of the instantiation of (i,j), are highlighted in green. Unpaired positions where the nucleotide choice has no effect in the free energy of the structure are indicated in black.

Since the dynamic programming algorithm for the *dual partition function* proceeds from inner to outer base pairs, using the total ordering $\prec$ in equation 6.11, the sampling order of base pairs proceeds from outer to inner positions, i.e. from largest *base pair index* to smallest. In order to account for mutual dependencies in the sampling step, we define the function $sample(k,T,i,j,X,Y)$ for each base pair (i,j) in S_0, where k indicates the *base pair index* defined from equation 6.11, T indicates the type of structural element closed by base pair (i,j) in the target RNA secondary structure, as shown in Table 6.1, and X,Y are the nucleotides instantiated

at positions (i,j). Due to the mutual dependencies, sampling a base pair with *base pair index* k closing a n-loop, with $n > 0$, forces the instantiation of all innermost closing base pairs of the n-loop, which corresponds to base pairs with index *base pair index* $< k$. For this reason, except in the case of external loops, the outermost base pair (i,j) has been always instantiated before $sample(k,T,i,j,X,Y)$ is called, and therefore the instantiation X,Y is given as a parameter of the sampling function.

The Boltzmann probability of each possible instantiation of mutually dependent positions can be computed on the fly in the backward step. However, in order to improve the speed of the algorithm, in the forward step `RNAdualPF` stores (for each base pair) the conditional *dual partition function* values of instantiations of interdependent positions. These tables are used by the sampling function, since each value corresponds to the *dual partition function* conditional on a specific instantiation of the positions to be sampled by $sample(k,T,i,j,X,Y)$. As mentioned before, the procedure of sampling depends on the type of element (T). Therefore, we define the flow of the function $sample(k,T,i,j,X,Y)$ for each one of the examples depicted in Figure 6.2, and we formally define the values stored in the conditional *dual partition function* table associated with the base pair (i,j) in each of these cases.

6.3.2.1 Hairpins

When hairpin size exceeds three (Figure 6.2a), since the base pair (i,j) has been previously instantiated, flanking positions $i+1,j-1$ are sampled first. Given the current assignment X,Y, the Boltzmann probability of sampling respectively the nucleotides U,V at the flanking positions $i+1,j-1$ is

$$P(i+1=U,j-1=V|i=X,j=Y) \quad = \quad \frac{Z^*(i,j,i+1,j-1;X,Y,U,V)}{Z^*(i,j;X,Y)}$$

Therefore, in the forward step `RNAdualPF` stores in a table the conditional *dual partition function* of each possible instantiation $\{X,Y,U,V\}$ of the base pair (i,j) and its flanking positions $i+1,j-1$ respectively, defined by

$$Z^*(i,j,i+1,j-1;X,Y,U,V) \quad = \quad e^{(-\frac{e^I_{AU}(X,Y)}{RT})} \cdot \exp(-\frac{H(j-i-1)}{RT}) \cdot \exp(-\frac{mismatch(X,Y,U,V)}{RT}) \cdot 4^{j-i-3}$$

Then, remaining unpaired positions are uniformly sampled, since the nucleotide choice does not change the final free energy. Triloops, tetraloops and hexaloops are exceptions to this rule, since there are special loops that contribute or penalize to the free energy. In those cases, we have to account for the special loops, as defined in Section 6.3.1.1.

Although storing a different conditional *dual partition function* table for each base pair (i,j), even for two different hairpins of the same size in the target structure, could seem a waste of space, note that `RNAdualPF` allows sequence constraints, and therefore $Z^*(i,j)$ could possibly differ from $Z^*(i',j')$ among hairpins of the same size, closed respectively by (i,j) and (i',j').

6.3.2.2 Stacking base pairs

As depicted in Figure 6.2b, sampling probability of a base pair with *base pair index* $k-1$ is dependent on the value sampled at the adjacent stacking base pair with *base pair index* k. Therefore, $sample(k,Stack,i,j,X,Y)$ samples the base pair $(i+1,j-1)$ using the probability conditional on the given assignment X,Y for (i,j), defined by

$$P(i+1=U,j-1=V|i=X,j=Y) \quad = \quad \frac{Z^*(i,j,i+1,j-1;X,Y,U,V)}{Z^*(i,j;X,Y)}$$

Then, the conditional *dual partition function* values stored in the forward step correspond to each instantiation $\{X,Y,U,V\}$ of the base pairs $(i,j),(i+1,j-1)$, denoted by

$$Z^*(i,j,i+1,j-1;X,Y,U,V) \quad = \quad e^{(-\frac{e^I_{AU}(X,Y)}{RT})} \cdot \exp(-\frac{stack(X,Y,U,V)}{RT}) \cdot Z^*(i+1,j-1,U,V)$$

6.3.2.3 Internal loops

The energy contribution of internal loops in the Turner energy model is always dependent on the flanking unpaired positions of both closing base pairs, so the sampling probability of the innermost base pair cannot be separated from the adjacent unpaired positions. Moreover, on specific sizes of internal loop (1×1, 1×2, 2×1, $1\times N$ and $N\times1$), base pairs share flanking positions. In these cases, all the unpaired positions and the outermost base pair must be sampled at the same time, since the energy contribution of each combination of base pairs and flanking positions is different. In the 1x3 internal loop depicted in Figure 6.2c, given the instantiation X,Y at the outermost base pair (i,j), the probability of sampling the nucleotides U,V,A,B,C respectively at positions k,l,n_1,n_2,n_3, where (k,l) is the innermost closing base pair, n_1 is the flanking position at $i+1$ shared by the base paired positions i and k, and n_2 and n_3 are the adjacent positions to j and l respectively, is given by

$$P(k=U,l=V,n_1=A,n_2=B,n_3=C|i=X,j=Y) \quad = \quad \frac{Z^*(i,j,k,l,n_1,n_2,n_3;X,Y,U,V,A,B,C)}{Z^*(i,j;X,Y)}$$

`RNAdualPF` computes and stores the conditional *dual partition function* of each possible instantiation $\{X,Y,U,V,A,B,C\}$ respectively at positions i,j,k,l,n_1,n_2,n_3 , defined as

$$\begin{aligned} Z^*(i,j,k,l,n_1,n_2,n_3;X,Y,U,V,A,B,C) &= e^{(-\frac{e^I_{AU}(X,Y)}{RT})} \cdot \exp(-\frac{min(asym \cdot |(k-i)-(j-l)|, maxAsym)}{RT}) \cdot \\ & 4^{j-l-3} \cdot \exp(-\frac{e_{AU}(i,j,X,Y)}{RT}) \cdot \exp(-\frac{internal(k-i+j-l-2)}{RT}) \cdot \\ & \exp(-\frac{mismatch(X,Y,A,B) + mismatch(V,U,C,A)}{RT}) \cdot Z^*(k,l,U,V) \end{aligned}$$

For internal loops of sizes (1×1, 1×2, 2×1, $1\times N$ and $N\times1$) similar conditional *dual partition function* tables are computed following the definitions in Section 6.3.1.2.

Other internal loops: When there are no shared flanking positions between the two base pairs that close an internal loop, as depicted in Figure 6.2d, the energy contribution of innermost base pair and its respective flanking positions is independent of those of the outermost base pair.

In this case, `RNAdualPF` samples first the flanking positions $i+1,j-1$ of the outermost base pair (i,j), whose sampling probability is solely dependent on the instantiated nucleotides X,Y at positions i,j. Is not necessary to store any conditional *dual partition function* for sampling these positions, since the probability of sampling the values A,B at the flanking positions $i+1,j-1$, given the assignment X,Y is defined by

$$P(i+1=A,j-1=B|i=X,j=Y) = \frac{\exp(-\frac{mismatch(X,Y,A,B)}{RT})}{\sum_{C,D\in\mathcal{N}} \exp(-\frac{mismatch(X,Y,C,D)}{RT})}$$

where mismatch penalties are obtained from table look-up.

Finally, the innermost base pair (k,l) and its flanking positions $k-1,l+1$ are sampled together. In this case, we need to store an additional value $Z^*(k-1,l+1)$, which is given by

$$Z^*(k-1,l+1) = \sum_{UV \in \mathcal{B}} \sum_{C,D \in \mathcal{N}} \exp(-\frac{mismatch(V,U,D,C)}{RT}) \cdot Z^*(k,l,U,V)$$

Then, following the same notation, the probability of sampling the nucleotides V,U,D,C respectively at positions $k,l,k-1,l+1$ is

$$P(k=V,l=U,k-1=D,l+1=C) = \frac{Z^*(k,l,k-1,l+1;V,U,D,C)}{Z^*(k-1,l+1)}$$

Therefore, the conditional *dual partition function* of each possible instantiation $\{V,U,D,C\}$ stored in the corresponding table is defined as

$$Z^*(k,l,k-1,l+1;V,U,D,C) = \exp(-\frac{mismatch(V,U,D,C)}{RT}) \cdot Z^*(k,l,U,V)$$

Finally, since the remaining unpaired position does not contribute to the free energy, it is uniformly sampled.

6.3.2.4 Multiloops and external loops

As explained in Section 6.3.1.3, if dangling positions are not included in the computation, sampling an external base pair or the closing base pair (i,j) of a multiloop from $Z^*(i,j)$ is trivial. On the other hand, by including dangling positions in the sampling, there is a dramatic increase in the space complexity of `RNAdualPF`, albeit the space used is only a constant factor

larger. However, the decompositions into groups described in Sections 6.3.1.3 and 6.3.1.3 allow to sample the positions of each group independently.

The example shown in Figure 6.2e depicts a multiloop with two groups: a *non-closing group* G_{nc} highlighted in green, and a *closing group* G_c highlighted in red and blue, where the closing base pair of the multiloop (i,j) is marked in red.

In a *non-closing group* G_{nc} all base pairs in $H(G_{nc})$ and dangling positions in $D(G_{nc})$ must be sampled together. Therefore, the conditional *dual partition function* of each possible instantiation of nucleotides at the h closing pairs in $H(G_{nc})$ and their adjacent positions in $D(G_{nc})$ is stored. Let $\mathcal{U} = \{U_1,V_1,\ldots,U_h,V_h\}$ denote an instantiation of the h base pairs in $H(G_{nc}) = [\kappa_1,\lambda_1,\ldots,\kappa_h,\lambda_h]$, and let $\mathcal{W} = \{A_1,B_2,\ldots,A_h,B_h\}$ denote an instantiation of the h flanking positions in $D(G_{nc}) = [\alpha_1,\beta_1,\ldots,\alpha_h,\beta_h]$ in the *non-closing group* G_{nc}. Then, the probability of sampling $\mathcal{U},\mathcal{W}$ is

$$P(H(G_{nc}) = \mathcal{U},D(G_{nc}) = \mathcal{W}) \quad = \quad \frac{Z^*(G,H(G_{nc}),D(G_{nc});\mathcal{U},\mathcal{W})}{Z^*(G)}$$

Therefore, the conditional *dual partition function* of each instantiation $\mathcal{U},\mathcal{W}$ at $H(G_{nc}),D(G_{nc})$, stored in the table of the group, is defined by

$$\begin{aligned} Z^*(G,H(G_{nc}),D(G_{nc});\mathcal{U},\mathcal{W}) \quad = \quad & \prod_{r=1}^{h}\Big(Z^*(\kappa_r,\lambda_r;U_r,V_r)\cdot \\ & \exp(-\frac{E_{d_5}(\kappa_r,\lambda_r,\alpha_r;U_r,V_r,A_r) + E_{d_3}(\kappa_r,\lambda_r,\beta_r;U_r,V_r,B_r)}{RT})\Big) \end{aligned}$$

Recall that the base pairs in $H(G_{nc})$ are adjacent. Therefore, due the constraints given by the overlapping positions within $D(G_{nc})$, and between $D(G_{nc})$ and $H(G_{nc})$, explained in Section 6.3.1.3, the number of possible instantiations $\mathcal{U},\mathcal{W}$ of $H(G_{nc}),D(G_{nc})$ is $\leq (6^h \cdot 4^{h+1})$.

In a similar way, sampling from the *closing group* G_c closed by the base pair (i,j), with $h+1$ base pairs in $H(G_c)$ and their corresponding flanking positions in $D(G_c)$ requires us to store the conditional *dual partition function* of each instantiation of nucleotides $\{X,Y,\mathcal{U},\mathcal{W}\}$ respectively at $i,j,H(G_c),D(G_c)$, where $\mathcal{U} = \{U_1,V_1,\ldots,U_h,V_h\}$ denotes an instantiation of the h first base pairs $[(\kappa_1,\lambda_1),\ldots,(\kappa_h,\lambda_h)]$ in $H(G_c)$, $\mathcal{W} = \{A_1,B_2,\ldots,A_{h+1},B_{h+1}\}$ denotes an instantiation of the $2 \cdot (h+1)$ flanking positions in $D(G_c) = [\alpha_1,\beta_1,\ldots,\alpha_{h+1},\beta_{h+1}]$, and X,Y denotes an instantiation of (i,j). The probability of the instantiation $\mathcal{U},\mathcal{W}$, given the nucleotides X,Y is

$$P(H(G_c) = \mathcal{U},D(G_c) = \mathcal{W}|i = X,j = Y) \quad = \quad \frac{Z^*(G_c,i,j,H(G_c),D(G_c);X,Y,\mathcal{U},\mathcal{W})}{Z^*(G_c;X,Y)}$$

Then, the values stored in the table of the closing group correspond to the conditional *dual partition function* of each instantiation $\{X,Y,\mathcal{U},\mathcal{W}\}$, defined by

$$\begin{aligned} Z^*(G_c,i,j,H(G_c),D(G_c);X,Y,\mathcal{U},\mathcal{W}) \quad = \quad & e^{(-\frac{e^I_{AU}(X,Y)}{RT})} \cdot \exp(-\frac{e_{AU}(i,j,X,Y)}{RT}) \cdot \prod_{r=1}^{h}\Big((Z^*(\kappa_r,\lambda_r;U_r,V_r) \cdot \\ & \exp(-\frac{E_{d_5}(\kappa_r,\lambda_r,\alpha_r;U_r,V_r,A_r) + E_{d_3}(\kappa_r,\lambda_r,\beta_r;U_r,V_r,B_r)}{RT})\Big) \cdot \\ & \exp(-\frac{E_{d_3}(j,i,\alpha_{h+1};Y,X,A_{h+1}) + E_{d_5}(j,i,\beta_{h+1};Y,X,B_{h+1})}{RT}) \end{aligned}$$

As a final remark, we would like to recall that all the conditional *dual partition function* values

are computed and stored in the forward step at the same time as the *dual partition function*. Therefore, despite the consequent increase of space complexity in the algorithm, the computation of the values required for correct sampling does not involve an greater time complexity.

6.3.3 Scaling

The sequence partition function $Z^*(s_0)$ grows much faster than the usual structure partition function $Z(\mathbf{a})$, and so *scaling* must be used in the implementation. Let $C > 2$ be a user-defined constant. By a slight modification of the previous recursions, we actually compute $\frac{Z^*(i,j;X,Y)}{C^{j-i+1}}$, and hence $\frac{Z^*(s_0)}{C^n}$. This modification does not affect properties of sequences sampled from the low energy ensemble.

6.3.4 Controlling GC-content

The GC-content of an RNA sequence $\mathbf{a} = s_1, \ldots, s_n$ is the number of nucleotides that are either G or C. Instead of computing $Z^*(i,j;X,Y)$ and $Z^*(s_0)$, we can compute $Z^*(i,j;X,Y;num)$ and $Z^*(s_0,num)$, defined to be the corresponding partition *dual partition functions*, restricted to sequences having GC-content of **num**.

We describe two particular subcases, to provide the idea of how modifications need to be undertaken.

6.3.4.1 Triloop

Note that the number of RNA *sequences* of length m having GC-content of α is $\binom{m}{\alpha} \cdot 2^{\alpha} \cdot 2^{m-\alpha} = \binom{m}{\alpha} \cdot 2^{m} \leq 4^{m}$.

Assume that $|\{X,Y\} \cap \{G,C\}| = \beta$. Then

$$\begin{aligned} Z^{*}(i,j;X,Y;\alpha) &= e^{(-\frac{e^{I}_{AU}(X,Y)}{RT})} \cdot \exp(-\frac{H(j-i-1)+e_{AU}(xy)}{RT}) \cdot \\ &\left(\binom{j-i-1}{(\alpha-\beta)} \cdot 2^{j-i-1} - |TriLoop_{x,y}| + \sum_{abc \in TriLoop_{x,y}} \exp(-\frac{triloopE(xabcy)}{RT}) \right) \end{aligned}$$

6.3.4.2 Multiloop and external loop

Assume that (i,j) closes a multiloop, which is a $(k+1)$-way junction with ℓ unpaired nucleotides. Assume that the ordered multiset of potential dangle positions is $D = [a_1,b_1,\ldots,a_{k+1},b_{k+1}]$, where $a_r = i_r - 1$ and $b_r = j_r + 1$ for $r = 1,\ldots,k$, and $a_{k+1} = i$ and $b_{k+1} = j$, and assume that there are m unpaired positions that are not adjacent to a base pair in the multiloop. If $\mathbf{r}$ denotes an RNA sequence of arbitrary length, then let the function $\gamma(\mathbf{r})$ denote the GC-count in $\mathbf{r}$. Given an assignment of nucleotide base pairs $U_1V_1,\ldots,U_kV_k$ to $(i_1,j_1),\ldots,(i_k,j_k)$, where $U_rV_r \in \{GC,CG,AU,UA,GU,UG\}$, and given an assignment $A_1,B_1,\ldots,A_k,B_k$ of dangle nucleotides, where $A_r,B_r \in \mathcal{N}$, for $r = 1,\ldots,k$, we let

$$\gamma(\mathbf{UV},\mathbf{AB}) = \gamma(U_1,V_1,\ldots,U_{k-1},V_{k-1},A_1,\ldots,A_k,B_1,\ldots,B_k).$$

Then, the *dual partition function* of a multiloop with a GC-content of α is defined by

$$
\begin{aligned}
Z^*(i,j;X,Y;\alpha) \quad = \quad & e^{(-\frac{e^I_{AU}(X,Y)}{RT})} \cdot \sum_{\{U_r,V_r \in \mathcal{B}: r=1,\ldots,k\}} \sum_{\{A_1,B_1,\ldots,A_k,B_k \in \mathcal{N}^{2k}\}} \\
& \exp(-\frac{a+b\cdot(k+1)+c\ell}{RT}) \cdot \begin{pmatrix} (\ell-m) \\ (\alpha-\gamma(\mathbf{UV},AB)) \end{pmatrix} \cdot 2^{\ell} \cdot \exp(-\frac{e_{AU}(i,j,X,Y)}{RT}) \cdot \\
& \prod_{r=1}^{k} \left(Z^*(i_r,j_r;U_r,V_r) \cdot \exp(-\frac{E_{d_5}(i_r,j_r,a_r;U_r,V_r,A_r) + E_{d_3}(i_r,j_r,b_r;U_r,V_r,B_r)}{RT}) \right) \cdot \\
& \exp(-\frac{E_{d_3}(j,i,a_{k+1};Y,X,A_{k+1}) + E_{d_5}(j,i,b_{k+1};Y,X,B_{k+1})}{RT})
\end{aligned}
$$

Since the modification required in the remaining cases follows similar reasoning as in the treatment of the hairpin and external loop just described, the details for these remaining cases are not given..

An additional challenge of computing the *dual partition function* with GC-content control is the combinatorial problem of efficiently counting the number N of instantiations of the external loop, consisting of all positions external to every base pair, with GC-content k, where the user can stipulate that certain positions are constrained to contain nucleotides consistent with IUPAC codes. To this end, we implemented the combinatorial algorithm defined in Appendix E.

6.3.4.3 Sampling with GC-content

The implementation of sampling with GC-content is similar to the definitions given in Section 6.3.2. However, there are some important differences.

First, the sampling function is redefined as $\gamma = sample(k,T,i,j,\beta)$, where k indicates the *base pair index* defined from equation 6.11, T indicates the type of structural element closed by base pair (i,j) in the target RNA secondary structure, as shown in Table 6.1; β designates the GC-content

of the sequences to be sampled, and the function returns a value γ indicating the number of Gs and Cs sampled for the given RNA element closed by (i,j).

Second, `RNAdualPF` stores a conditional *dual partition function* table for each base pair (i,j) and possible number of Gs and Cs from 0 to j-i+1, where the values of each conditional *dual partition function* table are computed only accounting for those sequences with an exact GC-content between the positions i and j, as just described.

Therefore $sample(k,T,I,J,\beta)$ samples from the conditional *dual partition function* of those sequences which have exactly β Gs and Cs between the positions i and j.

Let α be the desired GC-content of sequences $\mathbf{a} = a_1,\ldots,a_n$ to be sampled from a target secondary structure of ℓ base pairs, and denote $I[k],J[k],T[k]$ respectively the first position, last position and type of the base pair with *base pair index k* (as shown in Table 6.1). The pseudocode to sample sequences with exact α GCs is as follows:

```
1. targetGC=α
2. for k = ℓ to 0
3.    sampledGC = sample(k,T[k],I[k],J[k],targetGC)
4.    targetGC = targetGC - sampledGC
```

6.4 Benchmarking

As explained in Chapter 2, the software `IncaRNAtion` uses a similar approach to sample from the low energy ensemble of sequences of a given structure. However, `IncaRNAtion` uses a simplified energy model that only accounts for the contribution of for stacked base pairs, without accounting hairpins, bulges, internal loops or multiloops. Another difference with

IncaRNAtion is the GC-content control strategy. While IncaRNAtion introduces a parameter to heuristically adjust the GC-content after each sequence that is sampled, thus approximately targeting a desired GC-content. In contrast, RNAdualPF computes the exact partition function $Z^*(k)$ for sequences of GC-content k, for each value of k, and hence RNAdualPF samples sequences of an exact user-specified GC-content or content range.

In order to measure the importance of loop energy contributions, we performed a benchmarking of RNAdualPF against IncaRNAtion using the same test sets described in Section 2.4 of Chapter 2. For each target structure we ran IncaRNAtion and RNAdualPF 5 times, generating 2,500 sequences in each run. Since RNAdualPF allows sampling with and without accounting for the contribution of dangling positions, we tested both methods, which are denoted respectively by d2 and do using the notation from Vienna RNA Package for dangle treatment.

Table 6.2 summarizes the results of the benchmarking, and shows that RNAdualPF returns 1.5 times more sequences than IncaRNAtion whose MFE structure is the target structure. In addition, sequences returned by RNAdualPF have higher probability to fold into the target structure and lower GC-content. However, our results show that there is small difference in other measures of structural diversity such as *ensemble defect*, *expected base pair distance*, *Vienna structural diversity* and *Morgan-Higgs structural diversity* (see Appendix A). On the other hand, RNAdualPF is three orders of magnitude faster when dangling positions are not included in the computation, and four times faster when dangling positions are included.

Measure	RNAdualPF (do)	RNAdualPF (d2)	IncaRNAtion
% MFE	0.31/0.30	0.33/0.29	0.21/0.22
Free energy	-0.39/-0.35	-0.39/-0.35	-0.44/-0.40
Probability of structure	0.0016/0.0016	0.0017/0.0015	0.0013/0.0013
Average BP distance MFE-target	0.45/0.44	0.44/0.41	0.42/0.40
Positional entropy	0.44/0.43	0.43/0.42	0.39/0.38
Morgan-Higgs structural diversity	0.23/0.22	0.22/0.22	0.20/0.20
Vienna structural diversity	0.16/0.16	0.16/0.16	0.15/0.14
Expected base pair distance	0.29/0.31	0.28/0.30	0.26/0.28
Ensemble defect	0.43/0.45	0.41/0.44	0.39/0.41
Expected number of base pairs	0.34/0.34	0.34/0.34	0.34/0.34
Expected proportion of native contacts	0.58/0.55	0.59/0.55	0.64/0.61
GC-content	0.67	0.67	0.70
GC-content in base paired regions	0.80	0.80	0.88
GC-content in unpaired regions	0.52	0.51	0.50
Run time (in seconds)	0.65	152.41	672.36

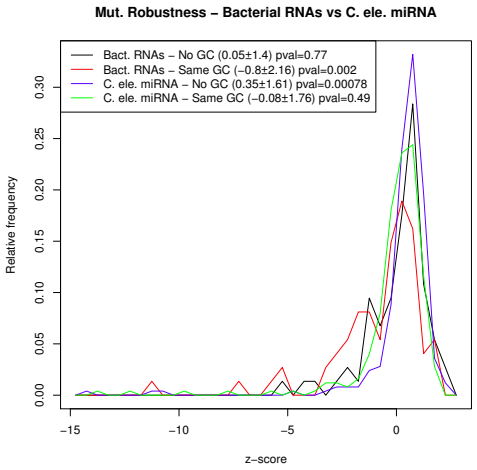
TABLE 6.2: Summary of benchmarking of `RNAdualPF` and `IncaRNAtion` showing averages of: *(% MFE)*: Percentage of structures whose MFE structure is the target structure; *Free energy*: free energy of the sequences when folded into the target structure; *Probability of the target structure* for the sequence; *Average BP distance MFE-target*: Average base pair distance between the MFE structure of the sequences and the target structure; *Positional entropy*; *Morgan-Higgs structural diversity*; *Vienna structural diversity*; *expected base pair distance*; *Expected number of base pairs*; *Expected proportion of native contacts*; (see Appendix A); GC-content of the sequences; GC-content in base paired regions; GC-content in unpaired regions; *Run time* required to generate 2,500 sequences with each method. All measures are length-normalized except *(% MFE)*, *Probability of structure*, *Run time* and *Expected proportion of native contacts*, where the latter is normalized by the number of base pairs in the target structure. In addition, measures were calculated using Turner 2004 energy model without(do) and with(d2) dangling positions, where both values (do/d2) are shown (where applicable) separated by a slash. Benchmarking statistics were obtained by sampling 12,500 sequences with each method, performing 5 runs of 2,500 sequences for each one of the 63 sequences on test sets 1,2,3 described in Section 2.4 of Chapter 2. Benchmarking was performed on a Core2Duo PC (2.8 GHz; 2 Gbyte memory; CentOS 5.5).

6.5 Applications

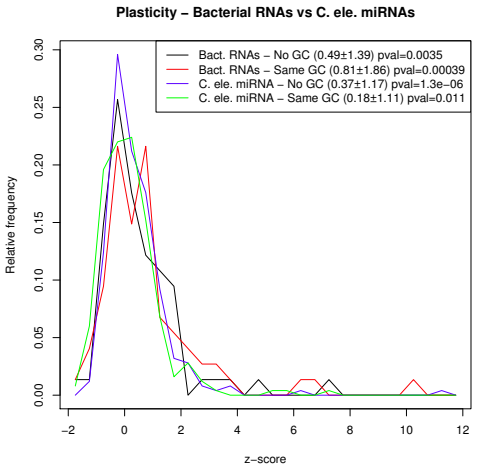
6.5.1 Robustness and plasticity of *C. elegans* miRNAs and *E. coli* sncRNAs

In agreement with [206], we find that *C. elegans* miRNA is significantly robust (Z-score 0.35 ± 1.61) *provided* that GC-content is *not* controlled (1-tailed T-test p-value 0.00039). However, in contrast to [206], when GC-content is controlled, we find that *C. elegans* miRNA is *not* significantly robust (Z-score -0.08 ± 1.76, 2-tailed T-test p-value 0.49). In agreement with [207], we find that bacterial sncRNAs are not significantly robust (Z-score 0.05 ± 1.4) *provided* that GC-content is not controlled (2-tailed T-test p-value 0.77). When GC-content is controlled, we obtain the even stronger result that bacterial sncRNAs are significantly *non-robust* (Z-score -0.8 ± 2.16, 1-tailed T-test p-value 0.001). Figure 6.3a summarizes our findings that precursor microRNAs [resp. bacterial sncRNAs] are *not* significantly robust [resp. are significantly *non-robust*] with respect to a control set of 1000 sequences having similar GC-content, as generated by `RNAdualPF`.

Additionally, we find that when GC-content is *not* controlled, *C. elegans* microRNAs significantly exhibit *plasticity* (Z-score 0.37 ± 1.17, 2-tailed p-value $1.3 \cdot 10^{-06}$), while the amount of plasticity decreases when GC-content is controlled (Z-score 0.18 ± 1.11, 2-tailed p-value 0.11). As well, when GC-content is *not* controlled, bacterial sncRNAs significantly exhibit *plasticity* (Z-score 0.49 ± 1.39, 2-tailed p-value $1.76 \cdot 10^{-3}$), while the amount of plasticity decreases when GC-content is controlled (Z-score 0.81 ± 1.86, 2-tailed p-value $1.97 \cdot 10^{-4}$). Figure 6.3b summarizes these last findings.



(A) Average mutant ensemble distance



(B) Plasticity, or normalized ensemble diversity

FIGURE 6.3: Z-scores of *mutational robustness* (a) and of *plasticity* (b) are presented for the bacterial small noncoding RNA collection from [207] and for *C. elegans* microRNA from Rfam 12.0. For each sncRNA and miRNA structure from the data sets, we used `RNAdualPF` to sample 1000 sequences that approximately fold into the target structure. Additionally, GC-content of the sampled sequences was either required to be exactly that of the initially given sequence, or not, as indicated in the legend. Sampled sequences were used to compute the mutational robustness and plasticity, explained later in this caption. Note that *C. elegans* miRNA is signicantly robust, but only provided that GC-content is not controlled. As well, when GC-content is not controlled, it appears that bacterial sncRNAs are not signicantly robust. When GC-content is controlled, a stronger result is possible – namely that bacterial sncRNAs are significantly non-robust. For this figure, mutational robustness of RNA sequence $\mathbf{a} = a_1, \ldots, a_n$ is defined by $1 - \frac{\langle D_{\mathrm{BP}} \rangle}{n}$, where ensemble distance $D_{\mathrm{bp}}(\mathbf{a},\mathbf{b})$ [210] between two length n sequences $\mathbf{a}$ and $\mathbf{b}$ is defined in equation (A.15) of Appendix A, and the average ensemble distance from all single-point mutants of $\mathbf{a}$ is defined by $\langle D_{\mathrm{BP}} \rangle = \sum_{\mathbf{b}} \frac{D_{\mathrm{BP}}(\mathbf{a},\mathbf{b})}{3n}$ where the sum is taken over all single-point mutants $\mathbf{b}$ of $\mathbf{a}$. We use this notation of mutational robustness, rather than the notion defined in [207], since the latter notion is not a true metric (see Section 6.2.1). The plasticity $P = \frac{\langle D_{\mathrm{V}} \rangle}{n/2} = \sum_{i<j} \frac{p_{i,j}(1-p_{i,j})}{n}$ is defined in [207] as normalized *ensemble diversity*, where ensemble diversity [210] (Vienna structural diversity) D_{V} is defined by equation (A.7) in Appendix A.

6.5.2 Structural RNA has higher free energy than expected

In Figure 2.5 of Chapter 2, it is shown that the average free energy of sequences that fold into the consensus secondary structure for the Rfam family RF00005 of tRNAs, as determined by `RNAiFold` [55, 56], is much lower than the minimum free energy (MFE) structure of *E. coli* val-tRNA (accession RV1600 from `Sprinzl` database [61], tdbR00000454 from `tRNAdb` [62]), a natural tRNA found by `RNAiFold` to fold into the Rfam consensus structure. Here, we show that this is a general phenomenon for structural RNA. Before presenting results, we need some definitions.

For the Turner nearest neighbor energy model [59], the free energy of a secondary structure s of an RNA sequence $\mathbf{a} = a_1,\ldots,a_n$ depends on the (absolute) temperature T_0. To indicate this dependence, we write $E(\mathbf{a},s,T_0)$, where in the sequel, T_0 will be designated as *table temperature*, i.e. the temperature for which parameters from the Turner energy tables are applied. For an arbitrary, but fixed secondary structure s_0 of length n, the *dual partition function* at temperature T_0 is defined by

$$Z(s_0,T_0,T) \quad = \quad \sum_{\mathbf{a}} \exp(-E(\mathbf{a},s_0,T_0)/RT) \tag{6.20}$$

where the sum is taken over all RNA sequences $\mathbf{a} = a_1,\ldots,a_n$ of length n. Note that T_0 indicates the (table) temperature at which the energy of a structure s_0 and nucleotide sequence $\mathbf{a}$ is evaluated using the Turner parameters, while all other occurrences of the temperature variable are designated by T, which we call *formal temperature*. The distinction between formal and table temperature is made to allow us to use finite difference approximations to derivatives with respect to the *formal temperature* when when we compute *dual expected energy* and *dual*

conformational entropy below (see Chapter 5 for more explanation). When table temperature T_0 equals formal temperature T, and the temperature is clear from the context, we write $Z^*(s_0)$; if the target structure s_0 is also clear from the context, then we write Z^*. A similar remark applies to the other thermodynamic functions p^*,G^*,$\langle E^* \rangle$,S^*, which we now define.

The *dual Boltzmann probability* $p^*(\mathbf{a})$ is defined by

$$p^*(\mathbf{a},s_0,T_0,T) \quad = \quad \frac{\exp(-E(\mathbf{a},s_0,T_0))}{Z^*(s_0,T_0,T)} \tag{6.21}$$

The *dual ensemble free energy* $G^*(s_0)$ is defined by

$$G^* \quad = \quad G^*(s_0) = G(s_0,T_0,T) = -RT \ln Z^*(s_0,T_0,T) \tag{6.22}$$

where $R \approx 1.987_{cal/(mol \cdot K)}$ is the universal gas constant. The *dual expected (free) energy* $\langle E^*(s_0) \rangle$ is defined by

$$\langle E^*(s_0,T_0,T) \rangle \quad = \quad \sum_{\mathbf{a}} E(\mathbf{a},s_0,T_0) \cdot p(\mathbf{a},s_0,T_0,T) \tag{6.23}$$

The *dual conformational entropy* $S^*(s_0)$ is defined by

$$S^*(s_0,T_0,T) \quad = \quad -R \sum_{\mathbf{a}} p(\mathbf{a},s_0,T_0,T) \cdot \ln p(\mathbf{a},s_0,T_0,T) \tag{6.24}$$

Note that if the free energy $E(\mathbf{a},s_0,T_0)$ of every sequence $\mathbf{a}$ compatible with structure s_0 is zero, or if the formal temperature T is infinite, then entropy S^* is equal to the universal gas constant R times the logarithm of the number of sequences that are compatible with s_0.

Straightforward derivations analogous to those in Chapter 5 prove the following:

$$\langle E^*(s_0,T_0,T)\rangle = RT^2 \cdot \frac{\partial}{\partial T}\Big(\ln Z^*(s_0,T_0,T)\Big)_{T=T_0} \tag{6.25}$$

$$S^*(s_0,T_0,T) = \frac{\langle E(\mathbf{a},s_0,T_0,T)\rangle - G^*(s_0,T_0,T)}{T} \tag{6.26}$$

Using the *dual partition function*, it is possible to compute the *dual heat capacity* (C_p^*), defined in [211] as

$$C_p^*(s_0,T) = \frac{\partial^2}{\partial T^2}\Big(G^*(s_0,T)\Big) \tag{6.27}$$

where $G^*(T) = -RT \cdot ln(Z^*)$. Note that in this case, the formal temperature is not uncoupled from the table temperature, since $C_p^* = \frac{1}{RT^2} \cdot$ *dual variance of enthalpy.* Also note that C_p^* is different from the *dual variance of free energy over* RT^2 *(denoted by* V^**)*, which does require uncoupling formal and table temperature, defined by

$$V^*(s_0,T_0,T) = \frac{\langle (E^*(s_0,T_0,T))^2\rangle - \langle E^*(s_0,T_0,T)\rangle^2}{RT^2} \tag{6.28}$$

At our web site, we provide scripts to compute *dual conformational entropy* S^*, *dual expected energy* $\langle E^*\rangle$, *dual heat capacity* C_p^*, *dual free energy variance* divided by RT^2 (V^*) using the software `RNAdualPF`. For example, pseudocode to compute *dual heat capacity* at temperatures 0 to 100 in Celsius is given as follows.

Algorithm: Heat capacity for 0° to 100°C.

---

1. $K_0 = 273.15$
2. `for` t `= 0 to 100`
3. $\quad T = K_0 + t$

4. $x_0 = G^*(s_0,T)$

5. $x_1 = G^*(T + \Delta T)$

6. $x_{-1} = G^*(T - \Delta T)$

7. $C_p^*(T) = -T \cdot \frac{x_1 + x_{-1} - 2x_0}{(\Delta T)^2}$

8. `output` $C_p^*(T)$

Figure 6.4 shows that structural RNAs have *higher* free energy with respect to their native structure, hence are thermodynamically *less stable*, than most RNAs that approximately fold into the same structure – even when these sequences are required to have the same (exact) GC-content. We believe that this insight could be important when designing functional synthetic RNAs. To generate Figure 6.4, we proceeded as follows.

For each family from the Rfam 12.0 database [65], we took the family consensus structure s_c, and computed $\langle E(s_c)\rangle$. Additionally, for each Rfam family, we selected that sequence $\mathbf{a}_0$, whose minimum free energy (MFE) structure s_0 has smallest base pair distance to the consensus structure s_c. We computed the expected energy $\langle E(s_0)\rangle$, as well as the free energies $E(\mathbf{a},s_c)$ and $E(\mathbf{a},s_0)$. Figure 6.4 displays box-and-whiskers plots for the fold change $\frac{\langle E(s_c)\rangle}{E(\mathbf{a}_0,s_c)}$ for the consensus structure and the fold change $\frac{\langle E(s_0)\rangle}{E(\mathbf{a}_0,s_0)}$ for the minimum free energy structure. Since the dual Boltzmann probability $p^*(\mathbf{a},s_0)$ is generally larger for sequences $\mathbf{a}$ having higher GC-content (as stacked base pairs involving GC,CG have lower free energy than those involving AU,UA,GU,UG), `RNAdualPF` computes as well the *dual partition function* for GC-content k, defined by

$$Z^*(s_0,k) \quad = \quad \sum_{\substack{\mathbf{a} \text{ such that} \\ \text{GC-content}=k}} \exp(-E(\mathbf{a},s_0)/RT) \tag{6.29}$$

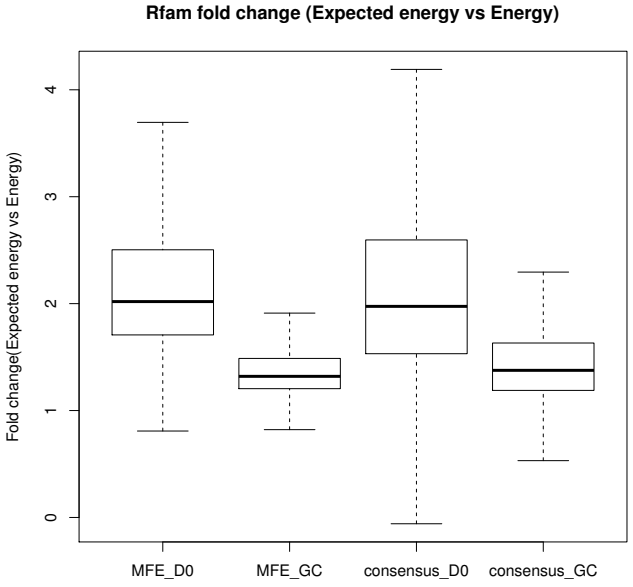
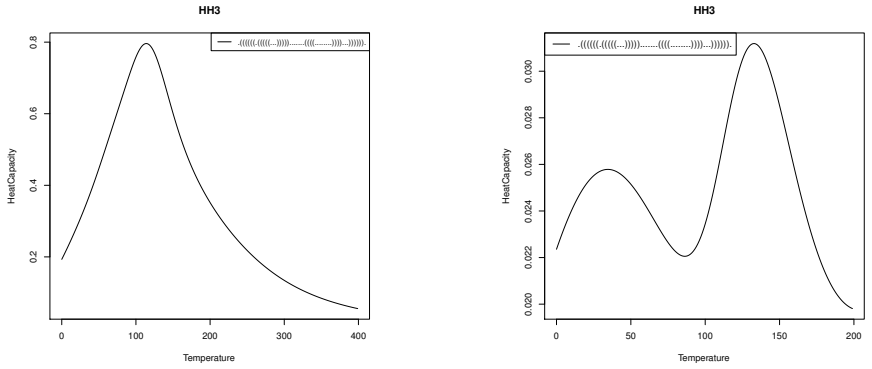


FIGURE 6.4: Analysis of expected free energy $\langle E \rangle$ for structures in Rfam 12.0 [65]. Given a secondary structure s, the expected free energy of all sequences $\mathbf{a}$ with respect to s is defined by $\langle E(s) \rangle = \sum_{\mathbf{a}} E(\mathbf{a},s) \cdot \frac{\exp(-E(\mathbf{a},s)/RT)}{Z^*(\mathbf{a},s)}$, where Z^* is the *dual partition function* defined in equation (6.8). For each Rfam family, we took the family consensus structure s_c, and computed $\langle E(s_c) \rangle$. Additionally, for each Rfam family, we selected that sequence $\mathbf{a}_0$, whose minimum free energy (MFE) structure s_0 has smallest base pair distance to the consensus structure s_c. The expected energy $\langle E(s_0) \rangle$ was computed, as well as the free energies $E(\mathbf{a},s_c)$ and $E(\mathbf{a},s_0)$. The fold change $\frac{\langle E(s_c) \rangle}{E(\mathbf{a}_0,s_c)}$ for the consensus structure and the fold change $\frac{\langle E(s_0) \rangle}{E(\mathbf{a}_0,s_0)}$ for the minimum free energy structure were computed. The box-and-whiskers plots show the mean, 25th and 75th percentile, minimum and maximum values. As indicated in the legend, these computations were performed either with respect to all sequences or with respect to all sequences having the same (exact) GC-content. These data clearly indicate that natural RNA sequences, whose MFE structures most closely resemble the Rfam consensus structures, have *higher* free energy than average.

In this fashion, we can exactly compute the *dual expected energy* $\langle E^*(s_0,k)\rangle$ of all sequences having GC-content k which approximately fold into target structure s_0. Figure 6.4 clearly indicates that natural RNA sequences, whose MFE structures most closely resemble the Rfam consensus structures, have *higher* free energy than average.



(A) Dual heat capacity (C_p^*) (B) Dual energy variance/RT^2 (V^*)

FIGURE 6.5: Dual heat capacity C_p^* for the secondary structure of Peach Latent Mosaic Viroid (PLMVd) hammerhead ribozyme AJ005312.1/282-335, graphed as a function of temperature in degrees Celsius. *(a)* Heat capacity $C_p^* = -\left(\frac{dH}{dT}\right)$, in units of kcal/mol K, approximated by $-T \cdot \frac{G^*(T+\Delta T)+G^*(T-\Delta T)-2G^*(T)}{(\Delta T)^2}$, where $G^*(T) = -RT \ln Z^*(T)$ and $Z^*(T)$ is the *dual partition function* for the given hammerhead structure s_0, and $\Delta T = 1$. *(b)* Variance of free energy of all sequences compatible with given hammerhead structure s_0, divided by RT^2 (V^*), in units of kcal/mol K; i.e. $\frac{\langle\left(E^*(s_0)\right)^2\rangle-\left(\langle E^*(s_0)\rangle\right)^2}{RT^2}$. The formulas in (a) and (b) are well-known to be equivalent if the free energy $E(\mathbf{a},s_0)$ is temperature-independent, for RNA sequence $\mathbf{a} = a_1,\ldots,a_n$ and secondary structure s_0. Since secondary structure free energy using the Turner parameters is temperature-dependent, curves from (a) and (b) are quite different; moreover, values in (b) are about two orders of magnitude smaller than those in (a).

Figure 6.5 displays the *dual heat capacity* C_p^* and *dual free energy variance* divided by RT^2 (V^*) of all sequences compatible with the secondary structure of Peach Latent Mosaic Viroid (PLMVd) hammerhead ribozyme AJ005312.1/282-335, graphed as a function of temperature in degrees Celsius. At present, we have not yet found an interesting application of *dual heat capacity* for RNA design.

6.6 Conclusion

In this chapter we present the program `RNAdualPF`, which computes the *dual partition function* Z^*, defined as the sum of Boltzmann factors $\exp(-E(\mathbf{a},s_0)/RT)$ of all *sequences* $\mathbf{a}$ with respect to the target structure s_0.

Using `RNAdualPF`, we efficiently sample RNA sequences that (approximately) fold into s_0, where additionally the user can specify IUPAC sequence constraints at certain positions, and whether to include dangles (energy terms for stacked, single-stranded nucleotides). Moreover, the user can require that all sampled sequences have a precise, specified GC-content, since, optionally, we compute the *dual partition function* $Z^*(k)$ simultaneously for all values $k = G + C$. This sampling strategy is complementary to the use of `RNAiFold`, since it allows the study of the properties of long RNA structures whose number of solutions for the inverse folding problem is astronomically large.

Our benchmarking results show that `RNAdualPF` is not only faster than the state-of-the art software `IncaRNAtion`, but also samples a higher percentage of sequences whose MFE structure is the target structure. On average, sequences returned by `RNAdualPF` have a higher probability of folding into the target structure than those returned by `IncaRNAtion`.

We use `RNAdualPF` to corroborate previous studies [206] by confirming that *C. elegans* microRNA is significantly mutationally robust; however, in contrast to [206], *C. elegans* microRNA appears *not* to be significantly robust when GC-content is controlled. Moreover, when GC-content is controlled, bacterial small noncoding RNAs are significantly non-robust.

In addition, we show that natural RNAs from the Rfam 12.0 database have *higher* minimum free energy than expected, thus supporting our results in Chapter 2 which suggest that functional RNAs are under evolutionary pressure to be only marginally thermodynamically stable.

Finally, we use `RNAdualPF` to compute the *dual expected energy* $\langle E^* \rangle$, *dual ensemble free energy* G^*, *dual conformational entropy* S^* and *dual heat capacity* C_p^* for the collection of *sequences* that (approximately) fold into target structure s_0. The thermodynamic parameters Z^*,C_p^*,G^*,H^*,S^* of the ensemble of sequences that approximately fold into a target structure s_0, together with sampled sequences from this ensemble, either with or without strict control over GC-content, provide a novel description of the universe of possible sequences that fold into a given structure. These aspects make `RNAdualPF` a unique tool for the field of molecular evolution.

Chapter 7

Discussion

Over the course of this thesis, we have described a collection of tools for RNA synthetic design and the analysis of the properties that differentiate known functional RNA structures. The use of these algorithms in an RNA design strategy, which consists of generating many solutions which are subsequently prioritized for experimental validation by applying various computational filters, produced promising results, which supports the effectiveness of this approach. In addition, we provided insights into different aspects of natural known RNA sequences which we consider will be of interest for the synthetic biology research community.

In particular, Chapter 2 provides a detailed description of the implementation of our software `RNAiFold`, the first complete inverse folding algorithm which also includes a wide range of design constraints. As shown in our benchmarking, `RNAiFold` performance is at least comparable to other state-of-art inverse folding methods for the task of finding a single solution for a given structure. On the other hand, the superiority of `RNAiFold` is clear when the objective is to generate a large number of solutions, which is a key component of the proposed design strategy. Moreover, the wide range of design constraints makes `RNAiFold` a very versatile

tool, as shown by the variety of applications described in this chapter, which include computational analysis of known RNAs, discovery of functional non coding RNAs, determining the relevance of structural motifs, and re-engineering of messenger RNAs to code the same or similar proteins and to contain desired RNA structural motifs.

In Chapter 3 we describe in detail the process of design of synthetic *cis*-cleaving hammerhead ribozymes from `Rfam` alignments using a purely computational pipeline. Indeed, all ten hammerhead ribozyme candidates were functional, despite the fact that the machine learning algorithm `Infernal` does not identify most of them as hammerhead ribozymes. In addition, we investigated the structural implications of the conserved GUH motif at the cleavage site of type III hammerheads using `RNAiFold`, where our software determined that no solution of inverse folding exists (with the given sequence constraints extracted from `Rfam` alignments) such as the GUH motif can be replaced by GUG and fold into the target active structure. This provides computational evidence that there are structural reasons that prevent the occurrence of a GUG motif at the hammerhead cleavage site.

The modifications introduced in `RNAiFold` to create `RNAiFold2T`, described in Chapter 4, illustrate the advantages of the modular implementation of our software. Indeed, the changes introduced allow to solve the k-temperature inverse folding and the use of *local structural constraints* without compromising the performance of the software. Using `RNAiFold2T` in a similar design strategy, we created moderately functional synthetic *thermo*-IRES elements, and generated promising RNA *molecular scissor* candidates for experimental validation.

The final part of this thesis concerns the development of algorithms to analyze the properties that differentiate known functional RNAs. In Chapter 5 we introduce `RNAentropy`, a tool to compute RNA structural entropy. This structural diversity measure is orthogonal to other

properties of RNA molecules, and can be used to improve computational predictions, as shown by the improvement in the correlation between free energy and hammerhead ribozymes cleavage activity by taking secondary structure conformational entropy into consideration.

Finally, in Chapter 6 we present `RNAdualPF`, which can compute the *dual partition function* of a given target secondary structure and generate unbiased samples from the low energy ensemble of sequences. Moreover, `RNAdualPF` implements IUPAC sequence constraints and exact control over GC-content. Allowing to obtain unbiased representations of the sequence ensemble of a structure which facilitate the analysis of intrinsic properties of RNA molecules. Indeed, the high structural conservation among functional non coding RNA families supports the relation between secondary structure and phenotype, and for this reason RNA has been used as a model to analyze properties such as robustness and plasticity. Using `RNAdualPF`, we confirm the results of previous studies on RNA robustness and plasticity. However, we show that these results can be misleading if they are not analyzed into the appropriate GC-content context. In addition, the *dual partition function* can be used to compute other features such as *dual expected energy*, *dual ensemble free energy*, *dual conformational entropy* and *dual heat capacity*. In fact, the analysis of *dual expected energy* over `Rfam` sequences supports of findings of Chapter 2, which suggests that natural RNAs have higher free energy than expected.

Taken as a whole, the different algorithms presented in this thesis represent a unified set of tools, which we expect will contribute to the advance of the field of synthetic biology, a research area poised to make revolutionary contributions to the 21st century.

Appendix A

Appendix A

A.1 Structural Diversity Measures

In this appendix, we define measures of structural diversity, all of which depend only on the computation of the base pairing probabilities

$$p_{i,j} = \sum_{\{S:(i,j)\in S\}} P(S) = \frac{\sum_{\{S:(i,j)\in S\}} \exp(-E(S)/RT)}{Z} \tag{A.1}$$

where $P(S)$ is the Boltzmann probability of structure S of a given RNA sequence $a = a_1, \ldots, a_n$, $E(S)$ is the Turner energy of secondary structure S [31, 32], $R \approx 0.001987_{kcal/(mol\cdot K)}$ is the universal gas constant, T is absolute temperature, and the *partition function* $Z = \sum_S \exp(-E(S)/RT)$, where the sum is taken over all secondary structures S of a. As explained in [175, 212], probability $p_{i,j}$ of base pair (i,j), where $1 \leq i < j \leq n$, can be computed in cubic time and quadratic space. For each fixed position $1 \leq i \leq n$, we define the probability distribution $p^*_{i,j}$, for j varying in $[1, n+1]$, by symmetrizing p for values $1 \leq i,j \leq n$, and then define $p^*_{i,n+1} = 1 - \sum_{j>i} p_{i,j} - \sum_{j<i} p_{j,i}$ [52, 213].

Expected (full) *positional entropy*. For a given RNA sequence $a = a_1,\ldots,a_n$ and fixed position $1 \le i \le n$, the (Shannon) entropy of the probability distribution $p_{i,j}$, as j varies in $[1,n+1]$ is defined by $H_i(a) = -\sum_{j=1}^{n+1} p_{i,j} \cdot \ln p_{i,j}$. As previously defined in [115], given an RNA sequence $a = a_1,\ldots,a_n$, we define the positional entropy $\langle H(a)\rangle$ by

$$\langle H(a)\rangle = -\sum_{i=1}^{n}\sum_{j=1}^{n+1}\frac{p_{i,j}\cdot \ln p_{i,j}}{n} \tag{A.2}$$

Clearly, if all low energy secondary structure of the RNA sequence $a = a_1,\ldots,a_n$ closely resemble the minimum free energy (MFE) structure, then the *positional entropy* is close to 0.

Expected (binary) positional entropy. *Binary positional entropy*, defined by

$$\langle H_b(\mathbf{a})\rangle = \sum_{i=1}^{n}\frac{H_b(\mathbf{a},i)}{n} \tag{A.3}$$

where the *binary positional entropy* at position i is defined by

$$H_b(\mathbf{a},i) = -\left(q_i \cdot \ln q_i + (1-q_i)\cdot \ln(1-q_i)\right) \tag{A.4}$$

Here, $q_i = p^*_{i,i} = 1 - \sum_{j\neq i} p^*_{i,j}$, and $0 \cdot \ln 0$ is taken to be 0.

***Expected base pair distance* from a structure.** Let S_0 be an arbitrary secondary structure of the RNA sequence $a_1,\ldots,a_n$. The *expected base pair distance* to S_0 is defined by

$$E[\{d_{\mathrm{BP}}(S,S_0) : S \in \mathbb{S}(a_1,\ldots,a_n)\}] = \sum_{S} P(S)\cdot d_{\mathrm{BP}}(S,S_0). \tag{A.5}$$

For brevity, we will write $E[\text{BP-distance to } S_0]$, or even $E[d_{\mathrm{BP}}(S_0)]$, to abbreviate $E[\{d_{\mathrm{BP}}(S,S_0) : S \in \mathbb{S}(a_1,\ldots,a_n)\}]$, defined in equation (A.5). We have the following.[1]

$$\begin{aligned} E[d_{\mathrm{BP}}(S_0)] &= \sum_S P(S)\cdot d_{\mathrm{BP}}(S,S_0) = \sum_S P(S)\cdot\left[\sum_{(i,j)\in S-S_0} 1 + \sum_{(i,j)\in S_0-S} 1\right] \\ &= \sum_{1\le i<j\le n} I[(i,j)\notin S_0]\cdot\sum_S P(S) + \sum_{1\le i<j\le n} I[(i,j)\in S_0]\cdot\sum_{\{S:(i,j)\notin S\}} P(S) \\ &= \sum_{1\le i<j\le n} I[(i,j)\notin S_0]p_{i,j} + \sum_{1\le i<j\le n} I[(i,j)\in S_0]\cdot(1-p_{i,j}) \\ &= \sum_{1\le i<j\le n} I[(i,j)\notin S_0]\cdot p_{i,j} + I[(i,j)\in S_0]\cdot(1-p_{i,j}) \end{aligned} \tag{A.6}$$

In this derivation, $I[(i,j)\notin S_0]$ denotes the indicator function for whether the base pair (i,j) does *not* belong to S_0. Although this notion, and the derivation (A.6) both appear to be new, there is a clear relation to the notion of *structural diversity*, $\langle D_v\rangle$, defined in the source code of `Vienna RNA Package` [11, 28] as follows: $\langle D_v\rangle = \sum_{S,T} P(S)\cdot P(T)\cdot d_{\mathrm{BP}}(S,T) = \sum_{i=1}^n\sum_{j=1}^n p_{i,j}\cdot(1-p_{i,j})$.

Vienna structural diversity. The *structural diversity* $\langle D_v\rangle$, defined in the source code of `Vienna RNA Package` [11, 28] is given by :

$$\langle D_v\rangle = \sum_{i=1}^{n}\sum_{j=1}^{n+1}(p_{i,j})\cdot(1-p_{i,j}) \tag{A.7}$$

Morgan-Higgs structural diversity. *Structural diversity*, $\langle D_{mh}\rangle$, as defined by Morgan and Higgs [213] and computed by Lorenz and Clote [214] in the context of the ensemble of locally optimal (kinetically trapped) secondary structures is defined as follows:

$$\langle D_{mh}\rangle = n - \sum_{i=1}^{n}\sum_{j=1}^{n+1}(p^*_{i,j})^2 \tag{A.8}$$

[1] To the best of our knowledge, the observation in equation (A.6), that *expected base pair distance* to a target structure S_0 can be computed in $O(n^3)$ time, seems to be new.

Ensemble defect. This distance measure is clearly motivated by the notion of Morgan-Higgs structural diversity. Given RNA sequence $a = a_1,\ldots,a_n$ and target structure S_0, Dirks et al. [52] define the *ensemble defect*, denoted by $ED(a,S_0)$, to be the expected number of nucleotides whose base pairing status differs from target structure S_0, taken over the ensemble of secondary structures of a. Formally, we recall that

$$ED(a,S_0) = n - \sum_{1\le i,j\le n} p^*_{i,j}\cdot I[(i,j)\in S_0] - \sum_{1\le i\le n} p^*_{i,n+1}\cdot I[i \text{ unpaired in } S_0] \tag{A.9}$$

where p^* is defined above, and I is the indicator function.

Expected proportion of native contacts. Unlike the previous distance measures, the following measure is a similarity measure, which takes values in the real interval [0,1]. Let s_0 be a given *target* structure of length n, for instance the *native* structure of RNA sequence $\mathbf{a} = a_1,\ldots,a_n$ as determined experimentally or by comparative sequence analysis. The *expected proportion of native contacts* $\langle NC(\mathbf{a},s_0)\rangle$ is defined by

$$\begin{aligned}\langle NC(\mathbf{a},s_0)\rangle &= \frac{1}{|s_0|}\sum_{s\in S\mathbb{S}[1,n]}\sum_{(i,j)\in s\cap s_0} P(s) \\ &= \sum_{\substack{i<j\\(i,j)\in s_0}} \frac{p_{i,j}}{|s_0|}.\end{aligned} \tag{A.10}$$

By using `RNAsubopt -p` to sample $m = 1000$ structures $s_1,\ldots,s_m$ from the Boltzmann ensemble of structures of RNA sequence $\mathbf{a} = a_1,\ldots,a_n$, the approximation $\frac{1}{m}\cdot\sum_{i=1}^{m} P(s_i)\cdot\frac{|s_i\cap s_0|}{|s_0|}$ to this measure was defined in [215], where it was used to compute the *sampled ensemble neutrality* (SEN), defined as the average, taken over all $3n$ single-point mutants of $\mathbf{a}$, of the (sampled approximation to the) expected proportion of native contacts. Clearly, it is possible to compute sampled ensemble neutrality both more accurately and substantially more rapidly by computing the *expected proportion of native contacts* (A.10) for each mutant.

Ensemble Hamming distance. Given $\mathbf{a} = a_1,\ldots,a_n$ and $\mathbf{b} = b_1,\ldots,b_n$, define *ensemble Hamming distance* between $\mathbf{a}$ and $\mathbf{b}$ as follows. Let $P(s|a)$ denote the Boltzmann probability of structure s in the ensemble of structures of $\mathbf{a}$, and similarly define $P(t|b|)$. Also, let $p_{i,j}(a) = \sum_s P(s|a) \cdot I[(i,j) \in s]$, i.e. the probability that (i,j) is paired in the ensemble of structures of $\mathbf{a}$, and similarly define $p_{i,j}(b)$. Similarly, define $p^*_{i,j}(a)$ to be the symmetrized base pairing probabilities with respect to $\mathbf{a}$, and similarly $p^*_{i,j}(b)$.

$$\begin{aligned} \langle D_{\mathrm{MH}}(\mathbf{a},\mathbf{b})\rangle &= \sum_{s\in SS(a)} \sum_{t\in SS(b)} P(s|a)\cdot P(t|b)\cdot D_{\mathrm{H}}(s,t) \qquad \text{(A.11)} \\ &= n - \sum_{i=1}^{n} \sum_{s\in SS(a)} \sum_{t\in SS(b)} P(s)\cdot P(t)\cdot I[s[i]=t[i]] \\ &= n - \sum_{ij} p^*_{i,j}(a)\cdot p^*_{i,j}(b) \end{aligned}$$

(A.12)

To remove the contribution of Morgan-Higgs structural diversity for each sequence $\mathbf{a}$, $\mathbf{b}$, the term $\frac{1}{2}\cdot(\langle D_{\mathrm{MH}}(a)\rangle + \langle D_{\mathrm{MH}}(b)\rangle)$ is subtracted. Finally, by taking the square root, we obtain the following proper metric:

$$\sqrt{\frac{1}{2}\cdot\left(\sum_{i,j}(p^*_{i,j}(a) - p^*_{i,j}(b))^2\right)} \qquad \text{(A.13)}$$

Ensemble Hamming distance appears to be a new measure, analogous to the measure dubbed *ensemble distance* defined in [210], and defined next.

Ensemble base pair distance. Ensemble distance was defined in [210] by equation (A.14) below. To distinguish the current notion from ensemble Hamming distance defined in equation (A.13), we will use the term *ensemble base pair distance.*

Given $\mathbf{a} = a_1,\ldots,a_n$ and $\mathbf{b} = b_1,\ldots,b_n$, define *ensemble base pair distance* between $\mathbf{a}$ and $\mathbf{b}$ as

follows.

$$\begin{aligned}\langle D_{\mathrm{V}}(a,b)\rangle &= \sum_{s\in S\mathbb{S}(a)}\sum_{t\in S\mathbb{S}(b)} P(s)\cdot P(t)\cdot D_{\mathrm{BP}}(s,t) \\ &= \sum_{i<j} p_{i,j}(a)\cdot(1-p_{i,j}(b)) + (1-p_{i,j}(a))\cdot p_{i,j}(b)\end{aligned} \tag{A.14}$$

To remove the contribution of Vienna structural diversity for each sequence **a**, **b**, the term $\frac{1}{2}\cdot(\langle D_{\mathrm{V}}(a)\rangle + \langle D_{\mathrm{V}}(b)\rangle)$ is subtracted. Finally, by taking the square root, we obtain the following proper metric:

$$D_{\mathrm{BP}}(a,b) = \sqrt{\sum_{i<j}(p_{i,j}(a)-p_{i,j}(b))^2} \tag{A.15}$$

For a related statistical mechanics study of RNA folding see [115].

A.2 Relative Structural Diversity Measures

The candidate selection of the design of synthetic hammerheads in Chapter 3 is based on the discrepancy some of the structural diversity measures described above with respect to the type III hammerhead found in Peach Latent Mosaic Viroid (*PLMVd*) AJ005312.1/282-335 (isolate LS35, variant ls16b), taken from Rfam[193]. Here we briefly describe the measures used:

EntropyDistAll: This is the maximum discrepancy between the (full) structural *positional entropy* of wild type PLMVd and that of the current sequence **s**, defined by

$$\max_{i=1}^{n} |H(\mathbf{s},i) - H(PLMVd,i)| \tag{A.16}$$

where $PLMVd$ denotes the RNA sequence of PLMVd AJ005312.1/282-335. The analogue for *binary positional entropy* is defined by

$$\max_{i=1}^{n} |H_b(\mathbf{s},i) - H_b(PLMVd,i)| \tag{A.17}$$

EBPDDistAll: This is the maximum discrepancy between expected *expected base pair distance* of wild type PLMVd and that of the current sequence **s**, defined by

$$\max_{i=1}^{n} |EBPD(\mathbf{s},i)/2 - EBPD(PLMVd,i)/2| \tag{A.18}$$

where $PLMVd$ denotes the RNA sequence of PLMVd AJ005312.1/282-335. Division by 2 in equation (A.18) occurs, since values are counted twice (it was earlier mentioned that $EBPD(\mathbf{s})$ was twice the value defined in [55].

EnsDefectAll: This is the maximum discrepancy between positional *ensemble defect* of wild type PLMVd and that of the current sequence **s**, defined by

$$\max_{i=1}^{n} |ED(\mathbf{s},i) - ED(PLMVd,i)| \tag{A.19}$$

where $PLMVd$ denotes the RNA sequence of PLMVd AJ005312.1/282-335.

StructDivAll: This is the maximum discrepancy between Vienna positional structural diversity of wild type PLMVd and that of the current sequence **s**, defined by

$$\max_{i=1}^{n} |SD(\mathbf{s},i) - SD(PLMVd,i)| \tag{A.20}$$

where $PLMVd$ denotes the RNA sequence of PLMVd AJ005312.1/282-335.

StructDivMHAll: This is the maximum discrepancy between Morgan-Higgs positional structural diversity of wild type PLMVd and that of the current sequence **s**, defined by

$$\max_{i=1}^{n} |SDMH(\mathbf{s},i)/2 - SDMH(PLMVd,i)/2| \tag{A.21}$$

where $PLMVd$ denotes the RNA sequence of PLMVd AJ005312.1/282-335. Division by 2 in equation (A.21) to avoid double counting, as in equation (A.18).

EntropyDistActive: This is the maximum discrepancy between the full structural positional structural entropy of wild type PLMVd and that of the current sequence **s**, defined by

$$\max_{i \in AS} |H(\mathbf{s},i) - H(PLMVd,i)| \tag{A.22}$$

where $PLMVd$ denotes the RNA sequence of PLMVd AJ005312.1/282-335, and where AS designates the positions in the 'conserved site', here defined to be the following 16 positions of PLMVd: 6-8, 22-25, 27-29, 44-49. In summary, this value is given by the restriction of equation (A.16) to the conserved site. The analogue for *binary positional entropy* is given by

$$\max_{i \in AS} |H_b(\mathbf{s},i) - H_b(PLMVd,i)| \tag{A.23}$$

EBPDDistActive: Restriction of equation (A.18) to the conserved site; i.e.

$$\max_{i \in AS} |EBPD(\mathbf{s},i)/2 - EBPD(PLMVd,i)/2| \tag{A.24}$$

EnsDefectActive: Restriction of equation (A.19) to the conserved site, i.e.

$$\max_{i \in AS} |ED(\mathbf{s},i) - ED(PLMVd,i)| \tag{A.25}$$

StructDivActive: Restriction of equation (A.20) to the conserved site; i.e.

$$\max_{i \in AS} |SD(\mathbf{s},i) - SD(PLMVd,i)| \tag{A.26}$$

StructDivMHActive: Restriction of equation (A.21) to the conserved site; i.e.

$$\max_{i \in AS} |SDMH(\mathbf{s},i)/2 - SDMH(PLMVd,i)/2| \tag{A.27}$$

Appendix B

Appendix B

B.1 Extended Helix and Extended Helix with Dangles

In this section, we give extend the of *extended helices* (*EH*) and *extended helices with dangles* (*EH*), that appear in the decomposition tree used in the `RNAiFold` algorithm to solve the RN inverse folding problem.

We start by identifying a given secondary structure S by its dot-bracket notation $s_1,\ldots,s_n$. `RNAiFold` instantiates the RNA sequence $a_1,\ldots,a_n$, whose minimum free energy structure at temperature T_1 [resp. T_2] is S_1 [resp. S_2] by assigning nucleotides to base-paired positions and unpaired positions of S_1 and S_2 in a particular order, defined by the helix and value heuristics applied to a structure decomposition tree for S_1 and S_2. We define two types of decomposition trees: (1) Extended Helix (*EH*) decomposition tree, and (2) Extended Helix with Dangles (*EHwD*) decomposition tree. See Figure 2.2 in Chapter 2 for an illustration of the *EH* decomposition tree for Rhizobiaceae group bacterium NR64, with EMBL accession number Z83250.

The convex subword $S' = s_i, \ldots, s_j$ of the dot-bracket representation of a secondary structure S is defined to be a *substructure* of S if the dot-bracket expression $s_i, \ldots, s_j$ is a valid secondary structure – i.e. S' is a well-balanced parenthesis expression. An *extended helix* (*EH*) [resp. *extended helix with dangles* (*EHwD*)] is a maximal substructure S' of S, with closing base pair (i,j) [resp. closing base pair (i,j), including flanking left and right dangle positions $i-1$ and $j+1$, provided the dangles exist in S], defined by the following inductive process, which is motivated by the definition of *order* of a secondary structure [216].

In the base case 0 of the induction, any maximal stem-loop substructure of S [resp. maximal stem-loop substructure of S with flanking left and right dangle, provided the dangle exists in S] is an *EH* [resp. *EHwD*], provided no bulge or internal loop has more than 2 adjacent unpaired positions; i.e. bulges have size at most 2, and internal loops are of the form 1×1, 2×1, 1×2, or 2×2. To define *EH* [rep. *EHwD*] in the $(k+1)$st inductive step, temporarily modify the structure S by replacing all left and right parentheses by a dot for those positions that belong to an *EH* [resp. *EHwD*] defined at a previous inductive step $\leq k$. Then an *EH* [resp. *EHwD*] in the $(k+1)$st inductive step is any maximal stem-loop substructure of the temporarily modified version of S [resp. maximal stem-loop substructure of the temporarily modified version of S with flanking left and right dangle, if the dangle exists in S], provided no bulge or internal loop has more than 2 adjacent unpaired positions.

The previous definition can be formalized by the following inductive definition. In the base case, define an *extended helix* of S to be a subsequence of the form $S' = s_i, \ldots, s_j$ of S of maximal length, such that: (1) S' is a substructure of S; (2) if $s_x, \ldots, s_y$ is a maximal length subsequence of S' that consists only of dots, then either *(i)* $x, \ldots, y$ are the unpaired positions of a hairpin

loop with closing base pair at $(x-1, y+1)$, or *(ii)* $y - x < 2$, which occurs in a bulge, or one side of an interior loop, of size 1 or 2.

In the inductive $(k+1)$st step, define an *extended helix* of S to be a subsequence of the form $S' = s_i, \ldots, s_j$ of S of maximal length, such that: (1) S' is a substructure of S; (2) if $s_x, \ldots, s_y$ is a maximal length subsequence of S' that consists only of dots, then either *(i)* $x, \ldots, y$ are the unpaired positions of a hairpin loop with closing base pair at $(x-1, y+1)$, or *(ii)* $y - x < 2$, which occurs in a bulge, or one side of an interior loop, of size 1 or 2, or *(iii)* position $y+1$ belongs to an *EH* defined at some step $\leq k$, and $x, \ldots, y$ correspond to the unpaired positions in a left bulge, or left portion of interior loop, of size greater than 2, or *(iv)* position $x-1$ belongs to an *EH* defined at some step $\leq k$, and $x, \ldots, y$ correspond to the unpaired positions in a right bulge, or right portion of interior loop, of size greater than 2, or *(v)* $x, \ldots, y$ correspond to the unpaired positions in a multiloop or external loop, each of whose components constitutes an *EH* defined at some step $\leq k$.

An *extended helix with dangles* is analogously defined, except that the leftmost and rightmost positions may constitute a dangle in the original structure S. Leaves of the *EH* [resp. *EHwD*] decomposition tree $\mathcal{T}$ consist of all *extended helix* [resp. *extended helix with dangles*] of S that are defined in the base case of the induction, arranged in left-to-right order. Inductively, an *EH* [resp. *EHwD*] is defined to be the parent of all proper maximal *EH*s [resp. *EHwD*s] that it contains, and these are ordered as daughter nodes in left-to-right order. Finally, the *root* of the decomposition tree $\mathcal{T}$ is the initially given structure S.

Appendix C

Appendix C

C.1 Selection of PLMVd: consensus structure and `RNAiFold'99`

Figure C.1 displays the minimum free energy (MFE) structure of type III hammerhead ribozyme from Peach Latent Mosaic Viroid (PLMVd) AJ005312.1/282-335 (isolate LS35, variant ls16b), taken from Rfam [50] family RF00008. The left [resp. right] panel is the MFE structure computed by `Vienna RNA Package 1.8.5` [resp. `2.0.7`]. The 54 nt structure in the left panel closely resembles the 56 nt family consensus structure illustrated at `http://rfam.sanger.ac.uk/family/RF00008#tabview=tab3`.

For an RNA sequence `s` in the Rfam seed alignment of an Rfam family, such as RF00008, we take the *consensus* structure of `s` to be that structure, obtained in the following manner:

1. Place a left [resp right] parenthesis `(` [resp. `)`] in a position in which the Rfam Stockholm format file has a left angle bracket < [resp. >] in the same column. All other

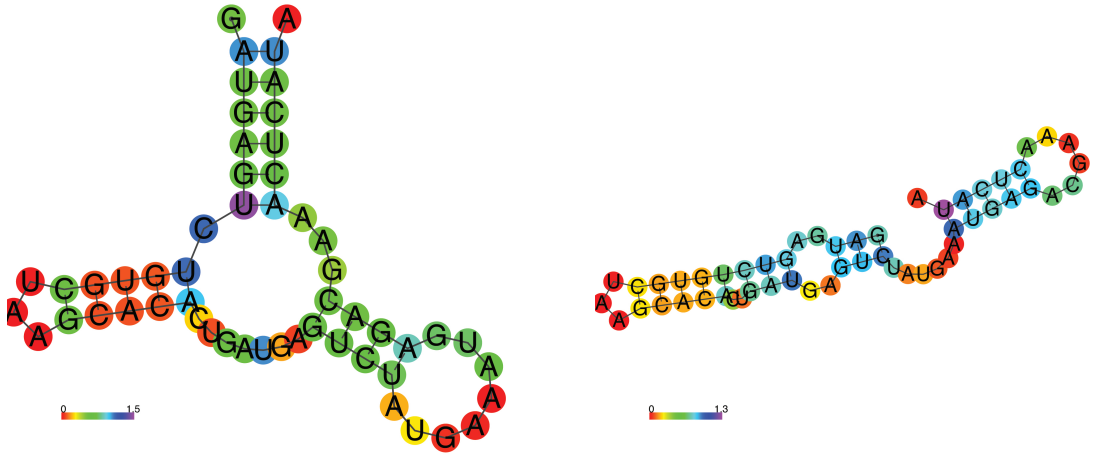


FIGURE C.1: Predicted secondary structure of the 54 nt Peach Latent Mosaic Viroid (PLMVd) AJ005312.1/282-335, colored by (full) structural *positional entropy*. (*Left*) `Vienna RNA Package 1.8.5`, employing the Turner 1999 free energy parameters. (*Right*) `Vienna RNA Package 2.0.7`, employing the Turner 2004 free energy parameters. Since the Vienna 1.8.5 MFE structure is identical to the Rfam consensus structure of PLMVd AJ005312.1/282-335, we used `RNAiFold'99` rather than `RNAiFold'04` for this work.

positions should contain a dot •.

2. Remove a left,right parenthesis pair at position i,j if $j - i \leq 3$.

3. Remove a left,right parenthesis pair at position i,j if the nucleotides in positions i,j of **s** do not constitute a Watson-Crick or wobble pair.

Using these steps, we wrote a simple script, `parseRfam.py`, to produce such consensus structures for any Rfam sequence (script available upon request).

The MFE structure of Peach Latent Mosaic Viroid (PLMVd) AJ005312.1/282-335 is identical to its Rfam consensus structure, as just defined, provided that `RNAiFold'99` is used; the right panel of Figure C.1 shows that the MFE structure produced by `RNAiFold'04` is radically different

than the Rfam consensus structure. For this reason, throughout this paper, we have used Turner 1999 energies (`RNAiFold'99`), rather than Turner 2004 energies (`RNAiFold'04`).

Additionally, PLMVd AJ005312.1/282-335 is the only one of the 84 sequences in the seed alignment of RF00008, whose MFE structure is identical to its Rfam consensus structure. Since we wished to apply `RNAiFold` to solve the inverse folding problem for a biologically functional type III hammerhead ribozyme structure, we selected the secondary structure of PLMVd AJ005312.1/282-335 as target structure. If the MFE structure of every sequence in the seed alignment of RF00008 had been distinct from the Rfam consensus structure, then we would have selected that MFE structure most closely resembling the Rfam consensus structure.

C.1.1 Dependence on sequence identity threshold

In running the software `RNAiFold`, sequence constraints were imposed for those positions in the Rfam seed alignment of PLMVd type III hammerhead ribozyme, for which sequence identity exceeded 96%. Subsequently, ten hammerhead candidates were selected according to various criteria concerning either (1) structural diversity or (2) matching the structural flexibility/stability of the wild type structure. What is the dependence of this protocol on the sequence identity threshold used to set constraints?

To answer this question, we ran `RNAiFold` to generate RNA sequences that (1) fold into the target structure of PLMVd, (2) have GC-content ranging from 35-55% (GC-content of wild type is 45%), and (3) have the same nucleotides in all positions whose sequence conservation in the Rfam seed alignment exceed either 90% (251,537 solutions), 96% (324,203 solutions) and 98% (349,508 solutions). We analyzed the three sets of solutions with respect to all the measures

considered in the paper, but here present only a few sample figures – see Figures C.2 C.3 and C.4 as well as `http://bioinformatics.bc.edu/clotelab/SyntheticHammerheads/`.

By increasing the conservation threshold from 96% to 98%, 11 positions are constrained, rather than 15 (plus H8 constraint), and by decreasing the conservation threshold from 96% to 90%, 19 positions are constrained – see SI Table 1. Note that by design, any position, which is not constrained to be a particular nucleotide, is nevertheless constrained to be *different* than the nucleotide present in wild type PLMVd (as explained in Chapter 3, this is to prevent selection of sequences which happen to have more sequence identity than defined).

Clearly, by adding more nucleotide constraints (e.g. decreasing the threshold from 96% to 90%), there is likely to be less sequence variation, hence the average sequence entropy is likely to be decreased. (Recall that all sequences fold into the correct target structure, so there is not a free range of nucleotide choices). However, structural entropy, *ensemble defect* and other measures of structural diversity tend to increase as more nucleotides are constrained to be identical to those in wild type PLMVd – i.e. structural diversity decreases for the ensemble of low energy structures of sequences, which less closely resemble the wild type sequence. At present, we have no convincing explanation for this phenomenon, which appears to be independent of the algorithm used to generate candidate hammerheads.

Rank	Position (1-54)	Nucleotide	Frequency	Percentage
1	7	U	1	100%
2	23	U	1	100%
3	27	G	1	100%
4	22	C	1	100%
5	48	A	1	100%
6	47	A	1	100%
7	28	A	1	100%
8	25	A	1	100%
9	24	G	0.988095	99%
10	46	A	0.988095	99%
11	6	G	0.987342	99%
12	45	G	0.97619	98%
13	49	C	0.97561	98%
14	29	G	0.964286	96%
15	44	C	0.964286	96%
16	38	A	0.957746	96%
17	8	C	0.949367	95%
18	35	G	0.942857	94%
19	42	G	0.928571	93%
20	31	C	0.892857	89%

TABLE C.1: Top 20 most conserved positions of PLMVd AJ005312.1/282-335 with the corresponding frequency in the seed alignment of family RF00008 from Rfam 11.0. Note the conservation rate of 95% for cleavage site C8.

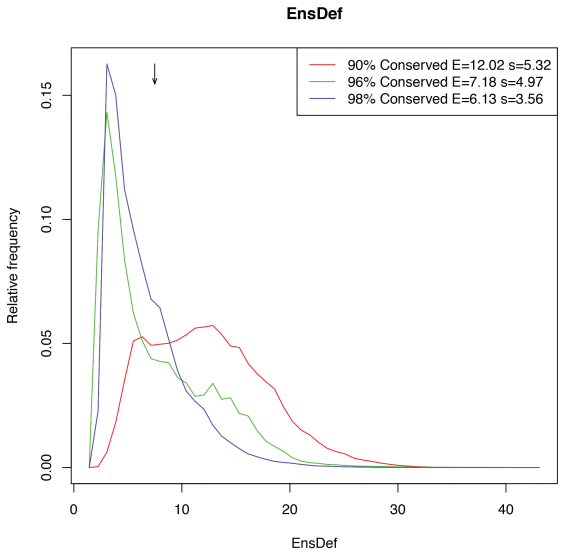


FIGURE C.2: *Ensemble defect* for `RNAiFold` sequences at varying thresholds for sequence identity.

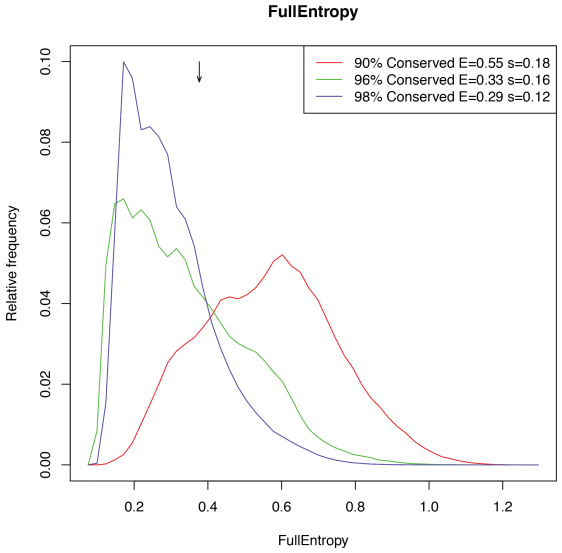


FIGURE C.3: Full structural *positional entropy* for `RNAiFold` sequences at varying thresholds for sequence identity.

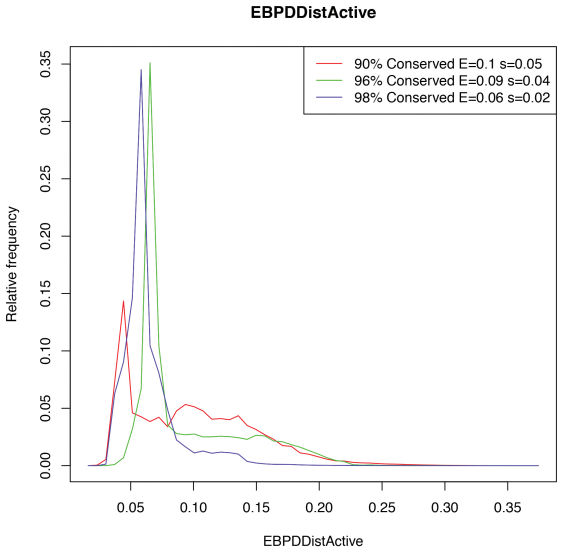


FIGURE C.4: *Expected base pair distance* discrepancy for `RNAiFold` sequences, measured with respect to the 'active site' (i.e. the 15 constrained positions having 96% sequence identity in the Rfam seed alignment of PLMVd hammerhead).

Appendix D

Appendix D

D.1 Pseudocode for value ordering for m-temperature inverse folding

The following pseudocode describes the value type assignment in `RNAiFold2T`. In the pseudocode, `StrList` is a 0-indexed array $[S_1,S_2]$ (resp. $[S_1,\ldots,S_m]$) of structures in the 2-temperature (resp. m-temperature) inverse folding problem. Let `index[S]` denote the index in `StrList` for structure S.

```
for (S in StrList)
    T = folding temperature of S
    for each bp(i,j) in S
        if ((i-1 and i+1 is unpaired in S) or (j-1 and j+1 is unpaired in S))
            Type7
        else
            if (S == StrList[m-1])
                S'= StrList[0]
            else
                S'=StrList[index[S]+1]
            T' = folding temperature of S'
            if (i,j) unpaired in S'
                if (T<T')
                    Type1
                else
                    Type 2
```

```
else if (bp[i]=j in S')
    Type 0
else if (i and j paired in S')
    Type 3
else if (i is paired in S')
    if (bp[i] in S' adjacent to j)
        Type 6
    else
        Type 4
else if (j is paired in S')
    if (bp[j] in S' adjacent to i)
        Type 6
    else
        Type 5
```

D.2 Cost functions

In this section, we define the cost function first described in equation (7) of [145], as well as a new variant defined from the notion of ensemble defect. To define the cost function of [145], we require some notation. For RNA sequence $\mathbf{a} = a_1, \ldots, a_n$, secondary structure S and temperature T, let $G_T(\mathbf{a})$ denote the ensemble free energy $-RT \ln Z(\mathbf{a})$, and let $E_T(\mathbf{a},S)$ denote the free energy of $\mathbf{a}$ with respect to structure S at temperature T – both of these values can be computed by `Vienna RNA Package`. Given sequence $\mathbf{a}$ and target structures S_1 resp. S_2 for temperatures T_1 resp. T_2, the cost function of [145] is defined by

$$\begin{aligned} &\left[E_{T_1}(\mathbf{a},S_1) - G_{T_1}(\mathbf{a})\right] + \left[E_{T_2}(\mathbf{a},S_2) - G_{T_2}(\mathbf{a})\right] + \\ &c \cdot \left\{\left(E_{T_1}(\mathbf{a},S_1) - E_{T_2}(\mathbf{a},S_1)\right) + \left(E_{T_2}(\mathbf{a},S_2) - E_{T_1}(\mathbf{a},S_2)\right)\right\} \end{aligned} \tag{D.1}$$

where $c > 0$ is a constant to weight the relative importance that the solution has low free energy with respect to target structure versus having high free energy with respect to the non-target structure.

To define a new ensemble defect based cost function, we require some additional notation. For a given RNA sequence $\mathbf{a} = a_1,\ldots,a_n$ and indices $1 \leq i < j \leq n$, recall that the base-pairing probability $p_{i,j}$ is defined by

$$p_{i,j} = \sum_{\substack{S \text{ such that} \\ (i,j)\in S}} \frac{\exp(-E(S)/RT)}{Z} \tag{D.2}$$

$$= \frac{\sum\limits_{\substack{S \text{ such that} \\ (i,j)\in S}} \exp(-E(S)/RT)}{\sum_S \exp(-E(S)/RT)} \tag{D.3}$$

where $E(S)$ is the Turner energy of secondary structure S [59], and Z is the *partition function*, defined by $Z = \sum_s \exp(-E(s)/RT)$, where the sum is taken over all secondary structures s of $\mathbf{a}$. The base pairing probabilities $p_{i,j}$ are computed in `RNAfold` [139], which implements McCaskill's algorithm [175]. Now for each fixed position $1 \leq i \leq n$, define the probability distribution $p^*_{i,j}$, for $j \in [1,n]$,

$$p^*_{i,j} = \begin{cases} p_{i,j} & \text{if } i < j \\ p_{j,i} & \text{if } j < i \\ 1 - \sum_{k>i} p_{i,k} - \sum_{k<i} p_{k,i} & \text{if } i = j \end{cases} \tag{D.4}$$

A secondary structure S of an RNA sequence $\mathbf{a} = a_1,\ldots,a_n$ is defined to be a set of base pairs (i,j) satisfying the following: (1) If $(i,j) \in S$ then a_i,a_j constitute a Watson-Crick or GU wobble base pair. (2) If $(i,j) \in S$ then $j > i + 3$, a condition that requires at least three unpaired bases in each hairpin loop. (3) If $(i,j) \in S$ and $(x,y) \in S$, and if $\{i,j\} \cap \{x,y\} \neq \emptyset$, then $i = x$ and $j = y$, a condition that disallows base triple formation. (4) If $(i,j) \in S$ and $(x,y) \in S$ are distinct base pairs, then either $i < x < y < j$ or $x < i < j < y$ or $i < j < x < y$ or $x < y < i < j$, a condition that disallows pseudoknot formation. Another possible data structure to represent a secondary

structure S is an array $s[1],\ldots,s[n]$ of integers, such that $s[i] = i$ when i is unpaired in S, while $s[i] = j \neq i$ when $(i,j) \in S$ or $(j,i) \in S$. Define the Hamming distance between structures s,t as $d_{\mathrm{H}}(s,t) = |\{i : s[i] \neq t[i]\}|$, i.e. the number of positions i in $[1,n]$ where $s[i] \neq t[i]$.

Given a secondary structure S_0 with array representation s_0, the *ensemble defect* $ED(S_0)$ is the expected Hamming distance to s_0 [52] defined by

$$ED(S_0) = \sum_S \frac{\exp(-E(S)/RT)}{Z} \cdot |\{i : s[i] \neq s_0[i]\}| \tag{D.5}$$

$$= n - \sum_{i \neq j} p^*_{i,j} I[(i,j) \in s_0] - \sum_i p^*_{i,i} I[s_0[i] = i] \tag{D.6}$$

where I denotes the indicator function. When the sequence $\mathbf{a} = a_1,\ldots,a_n$ and temperature T need to be indicated, we use the notation $ED(\mathbf{a},S_0,T)$ to denote ensemble defect of $\mathbf{a}$ for target structure S_0 at temperature T. We now define *ensemble defect based cost* as follows:

$$ED(\mathbf{a},S_1,T_1) + ED(\mathbf{a},S_2,T_2) - \tag{D.7}$$

$$\xi\left[(E_{T_1}(\mathbf{a},S_1) - E_{T_1}(\mathbf{a},S_2) + (E_{T_2}(\mathbf{a},S_2) - E_{T_2}(\mathbf{a},S_1)\right]$$

where $\xi > 0$ is a constant to weight the free energy of folding into the intended structure S_1 [resp. S_2] at temperature T_1 [resp. T_2].

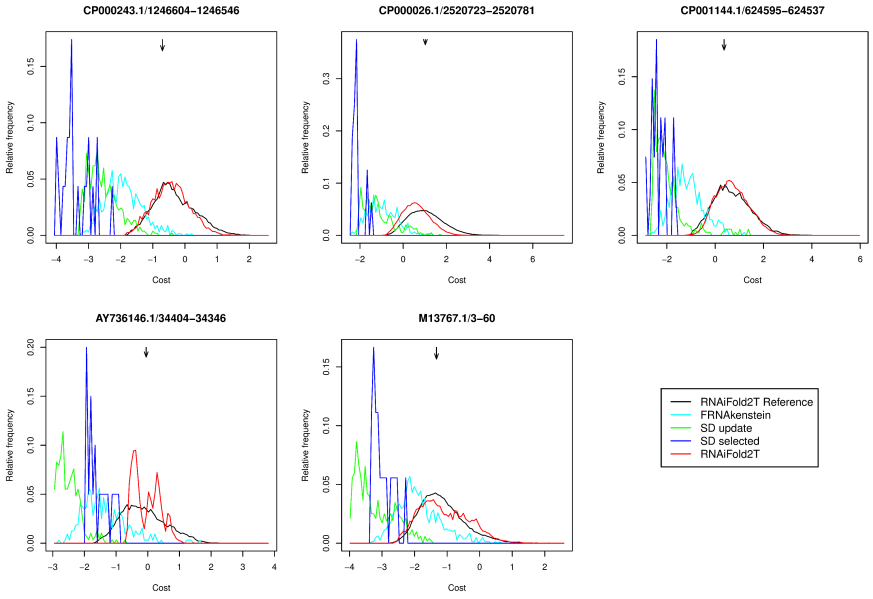


FIGURE D.1: Relative histogram for the cost, as defined in equation (D.1), for the solutions returned by `RNAiFold2T`, `SD` and `FRNA`, given target structure S_1 [resp. S_2] at temperature T_1 [resp. T_2] for λ phage CIII thermoregulators from Rfam family RF01804 – the number of solutions returned for each method is indicated in column A of Table 1, which also gives EMBL accession codes. To produce a reference distribution, the black curve for `RNAiFold2T`. Reference was produced by running `RNAiFold2T` for several days. Remaining curves are for `FRNA` (light green), `SD` (dark green and purple) and `RNAiFold2T` (red). Arrows indicate the cost values for the real λ phage CIII thermoregulators from Rfam RF01804. This figure is similar to Figure 4.3 from Chapter4, except that the histograms of `SD` and `FRNA` are created from the output of these programs, *without* adding 1-point mutants that also fold into the target structures.

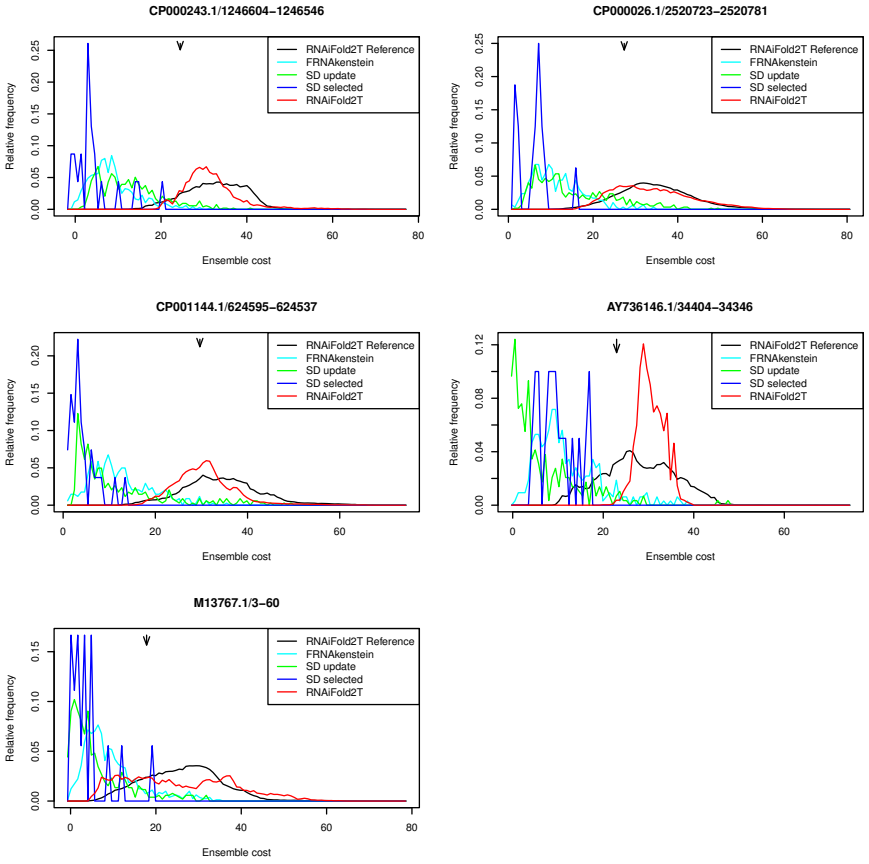


FIGURE D.2: Relative histogram for ensemble defect cost, defined in equation (D.7), for the solutions returned by `RNAiFold2T`, `SD` and `FRNA`, given target structure S_1 [resp. S_2] at temperature T_1 [resp. T_2] for λ phage CIII thermoregulators from Rfam family RF01804 – the number of solutions returned for each method is indicated in column A of Table 1, which also gives EMBL accession codes. To produce a reference distribution, the black curve for `RNAiFold2T`. Reference was produced by running `RNAiFold2T` for several days. Remaining curves are for `FRNA` (light green), `SD` (dark green and purple) and `RNAiFold2T` (red). Arrows indicate the cost values for the real λ phage CIII thermoregulators from Rfam RF01804. The Pearson correlation coefficient between `SD` cost and ensemble defect based cost is 0.6545963

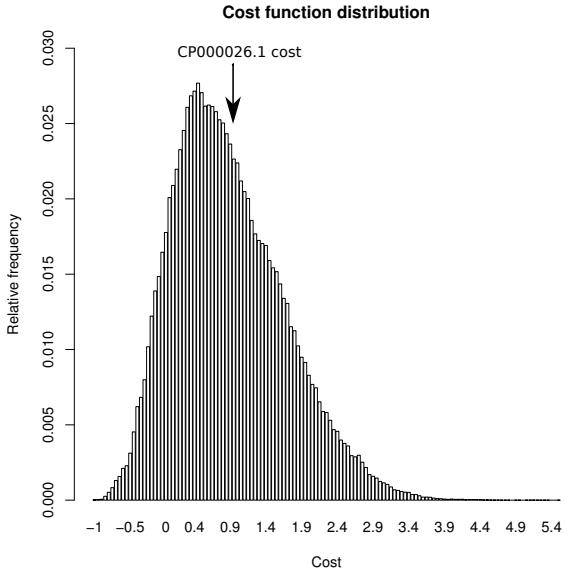


FIGURE D.3: Distribution of the cost function for all possible solution sequences of lambda phage RNA [EMBL:CP000026.1/2520723-2520781] MFE structure at 32°C and 55°C. The arrow indicates the cost of the original sequence for the given structures.

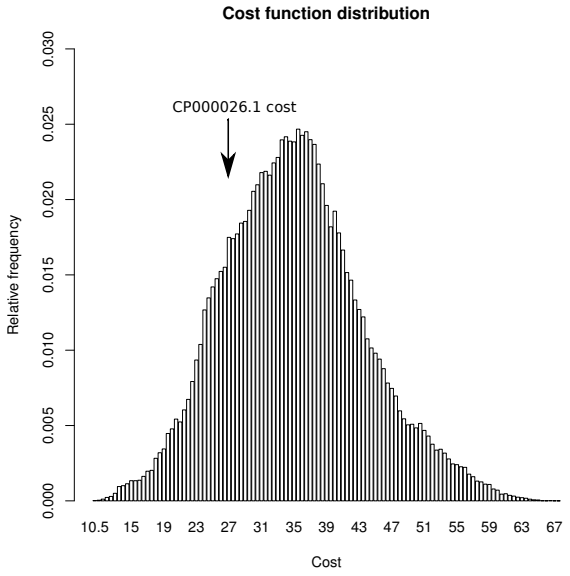


FIGURE D.4: Distribution of the cost function based on ensemble defect for all possible solution sequences of lambda phage RNA [EMBL:CP000026.1/2520723-2520781] MFE structure at 32°C and 55°C. The arrow indicates the cost of the original sequence for the given structures.

D.3 Sequences used in RNAiFold2T benchmark

Accession number, family	Structures S_1,S_2	Temp T_1,T_2	Size
CP000026.1/2520723-2520781	.((((.(((.((((.(((((........(((....)))..)))))))))))))))...	32	59
Lambda thermo	...............................(((..(((.......)))..)))....	55	59
CP001144.1/624595-624537	.((((.(((.((((.(((((.((..((....))..))...)))))))))))))))...	32	59
Lambda thermo	...............................(((..(((.......)))..)))....	55	59
AY736146.1/34404-34346	.((((.(((.((((.(((((.((((((....))))))...)))))))))))))))...	32	59
Lambda thermo	...............................(((..(((.......)))..)))....	55	59
M13767.1/3-60	.((((.(((..(((.((((((.((((((....)))))).))))))))).)))))))..	32	58
Lambda thermo	..............................((((..(((.......)))..))))...	62	58
CP000243.1/1246604-1246546	.((((.(((..(((.((((((.((((((....)))))).))))))))).)))))))...	32	59
Lambda thermo	..............................((((..(((.......)))..))))....	62	59
CP001144.1/2031534-2031470	((.((((((...)))))))).....(((((.((((((((....)))))))).)))))........	37	65
FourU	..............((((.(.((......((..(((((....)))))..))....)).).)))).	58	65
CP001127.1/1302123-1302187	((.((((((...)))))))).....(((((.((((((((....)))))))).)))))........	37	65
FourU	..............((((.....(((((.((..(((((....)))))..)).)))))...)))).	58	65
CP000647.1/1773227-1773291	((.((((((...)))))))).....(((((.((((((((....)))))))).)))))........	37	65
FourU	................((((.....(((((.((((((((....)))))))).)))))...)))).	45	65
CP000653.1/14627-14699	((((((((.((((((.(((.....)))))))))...)))...))))).......((..........))...	20	73
ROSE 2	...(((((..((((((.(((.....)))))))))).........(((....))))))))..............	42	73
ABWL02000023.1/393416-393344	....((.....(((((.(((.....))))))))...((..(((.(((....))))))....)).....))...	20	73
ROSE 2	..(((((.((((((((.(((.....))))))))))......((.....))....)))))))..............	42	73
ACDJ01000026.1/381061-380989	....((.....(((((.(((.....)))))))).((((..(((.(((....))))))....))))...))...	20	73
ROSE 2	..(((((...((((((.(((.....)))))))))..........(((....))).)))))..............	42	73
CP000036.1/3699544-3699616	..(((((.(((((...(((....(((((((......).))).))).))).))))))))))..............	20	73
ROSE 2	..(((((.(((((...(((....((((((.(.....).))).))).))).))))))))))..............	42	73
AE017220.1/3951363-3951290	....(((.((..(((((.(((.....))))))))).))(((.(((.(((....))))))...)))....)))...	20	74
ROSE 2	....(((.....(((((.(((.....))))))))...(((.(((.(((....))))))...)))....)))...	42	74
CP000026.1/3798554-3798481	....((((((..(((((.(((.....))))))))....(..(((.(((....)))))))..)..)))))).....	20	74
ROSE 2	....((((((..(((((.(((.....)))))))).......(((.(((....)))))).....)))))).....	42	74
BAAW01000185.1/6674-6747	.((.((((((..(((((.(((.....))))))))..(((..(((.(((....))))))..)))))))).))..	20	74
ROSE 2	.((.(((((((.(((((.(((.....)))))))))........(((.(((....)))))).....))))))).))..	42	74
CP000647.1/4480191-4480116	.(((....))).(((((.(((.....))))))))....(((.(((.(((.....)))))))..)))...........	20	76
ROSE 2	.(((....))).(((((.(((.....))))))))........(((.(((.....)))))).................	42	76
CP000009.1/1450710-1450627	((((((.((...))))))))....((((((.....)))))).(((((((.((((((.(((....)))))))))).))))))).	20	84
ROSE	((((((.(....).))))))....((((((.....)))))).(((..((.(((((((.(((....)))))))))).))..))).	42	84
AP003017.1/94542-94451	...((((....))))........((((((((((.....)))))))))).((((((..((((((.(((....))))))))).))))))..	20	92
ROSE	...((((....))))........((((((((((.....)))))))))).((..((..((((((.(((....))))))))).))..))..	42	92
AE007872.2/51225-51317	((((((.(((....)))))))))...((((((((((((((.....)))))))))))).((.((((((.(((....)))))))))).)).)))).	20	93
ROSE	((((((.(((....)))))))))..(((((((((((((((.....))))))))))....((((((.(((....)))))))))....)))))	42	93
AE007872.2/441983-442075	((((((.(((....))).))))))((.....))..((((.(.....).)))).(((((.((((((.(((....))))))))).)))))...	20	93
ROSE	((((((.(((....))).))))))..........((((.(.....).)))).(((((.((((((.(((....))))))))).)))))...	42	93
AL591985.1/872145-872052	.(((((.((.....))))))((.((((((((.......))))))))))..(((((.(.((((((.(((...))))))))).))))))..	20	94
ROSE	.(((((.((.....)))))))......((((((.(((.......))).)))))).((((((((((.(((...))))))))))..))))...	42	94
BA000012.4/1943819-1943723	((((((.((......))))))))....((.((((((((((.......)))))))))).((((.((((((.(((....))))))))).)))).)).	20	97
ROSE	((((((.((......))))))))......((((((((((.......))))))))))((((.((((((.(((....))))))))).))))....	42	97
AJ003064.1/2697-2806	(((((((....)))))))(((((((.((....))))))))...(..(((((((.....)))))))..)(((.((.((((((.(((....))))))))).)).)))..	20	110
ROSE	(((((((....)))))))(((((((.((....)))))))))......(((((((.....)))))))...(((.((.((((((.(((....))))))))).)).)))..	42	110
U55047.1/3106-3215	.(((((.....)))))((((((.((....))))))))......((((((.((((....)))).))))))((((((.(((..((.(((....)))))..))).))))))..	20	110
ROSE	.(((((.....))))).((((((((...((.(((((.......))))).))...))))))))........((((((.(((...(.(((....))).)..))).))))))..	42	110
U55047.1/5180-5291	.(((((.....)))))((((((.(((....)))))))))..(((.((((((.((((....)))).)))))))))...(((((...((.(((........))))).))).)).	20	112
ROSE	.(((((.....)))))((((((.(((....)))))))))......((((((.((((....)))).))))))((((((.(((...(.(((....))).)..))).))))))..	42	112
AJ010144.1/622-738	((((((....))))))(((((((((.((....))))))))...))).......((((((((......))))))))((.((((.(((..((.(((....)))))..))).)))).)).	20	117
ROSE	((((((....))))))...((((((.((....))))))))............((((((((......))))))))((.((((.(((...(.(((....))).)..))).)))).)).	42	117
AJ003064.1/2430-2312	.((((((....))))))...(((((((.((....)))))))).((.(((.(((.(((.((((((......))))))))))))))))....((((........)))).)))))..((.....)).	20	119
ROSE	.((((((....))))))((.((((((.((....)))))))).))..........((((((((......))))))))((.((((.(((...(.(((....))).)..))).)))).)).	42	119

TABLE D.1: **RNA thermometers of length at most 130 nt used in benchmarking.** Each RNA corresponds to two successive rows, where the first row contains the EMBL accession code, target structure S_1, temperature T_1 for S_1, and structure length, while the second row contains the type of thermosensor, target structure S_2, temperature T_2 for S_2, and structure length.

Accession number, family	Structures S_1, S_2	Temp T_1, T_2	Size
AJ002742.1/161-290	[illegible]	20	130
PrfA	[illegible]	37	
M55160.1/297-426	[illegible]	20	130
PrfA	[illegible]	37	
X72685.1/1303-1435	[illegible]	20	133
PrfA	[illegible]	37	
AAPQ01006501/448359-448507	[illegible]	22	149
Hsp90_CRE	[illegible]	36	
AAEU02000132l/310682-310830	[illegible]	22	149
Hsp90_CRE	[illegible]	36	
X03811l/879-1030	[illegible]	22	152
Hsp90_CRE	[illegible]	36	
AY1220801/193-344	[illegible]	22	152
Hsp90_CRE	[illegible]	36	
L23115.1/459-834	[illegible]	20	376
CspA	[illegible]	42	
AF017276.1/479-855	[illegible]	20	377
CspA	[illegible]	42	
ABJD02000101.1/494629-495045	[illegible]	20	417
CspA	[illegible]	42	
CP000647.1/4296745-4297166	[illegible]	20	422
CspA	[illegible]	42	
ABWM02000027.1/244578-245001	[illegible]	20	424
CspA	[illegible]	42	
CP000653.1/190938-191361	[illegible]	20	424
CspA	[illegible]	42	
ABWL02000023.1/204545-204970	[illegible]	20	426
CspA	[illegible]	42	
ABEH02000004.1/260631-260204	[illegible]	20	428
CspA	[illegible]	42	
CP000946.1/174765-174338	[illegible]	20	428
CspA	[illegible]	42	
CP000822.1/4602942-4603373	[illegible]	20	432
CspA	[illegible]	42	
ACCI02000028.1/45583-45145	[illegible]	20	439
CspA	[illegible]	42	
AAOS02000014.1/93711-93269	[illegible]	20	443
CspA	[illegible]	42	
ABXW01000053.1/281917-281471	[illegible]	20	447
CspA	[illegible]	42	
AALD02000025.1/10362-9916	[illegible]	20	447
CspA	[illegible]	42	

TABLE D.2: **RNA thermometers longer than 130 nt used in benchmarking.** Each RNA corresponds to two successive rows, where the first row contains the EMBL accession code, target structure S_1, temperature T_1 for S_1, and structure length, while the second row contains the type of thermosensor, target structure S_2, temperature T_2 for S_2, and structure length.

Appendix E

Appendix E

E.1 Number of external loops with given GC-content and IUPAC constraints

Here we describe a simple, yet tricky, combinatorial algorithm to efficiently count the number N of external loops of size n with GC-content k, where the user can stipulate that certain positions are constrained to contain nucleotides consistent with IUPAC codes. If there are no IUPAC constraints, then clearly $N = \binom{n}{k} \cdot 2^k \cdot 2^{n-k} = \binom{n}{k} \cdot 2^n$; however, with IUPAC constraints for uncertain data, the situation is a good deal more complicated. In order to explain and justify the algorithm, we introduce some definitions which may appear pedantic at this point, but most certainly are not and will simplify presentation of the algorithm.

Class $A = \{R,Y,M,K\}$ is defined to be the set consisting of IUPAC codes R,Y,M,K, where R is A or G, Y is C or U, M is A or C, K is G or U. Note that if m nucleotide positions are constrained by an IUPAC code belonging to class A, and k of these positions are required

to have GC-content of k, then the number of sequences satisfying this requirement is $\binom{m}{k} \cdot 1^k \cdot 1^{m-k} = \binom{m}{k}$.

Class $B = \{B,V\}$ is defined to be the set consisting of IUPAC codes B,V where B is C or G or U (i.e. not A), V is A or C or G (i.e. not U). Note that if m nucleotide positions are constrained by an IUPAC code belonging to class B, and k of these positions are required to have GC-content of k, then the number of sequences satisfying this requirement is $\binom{m}{k} \cdot 2^k \cdot 1^{m-k} = \binom{m}{k} \cdot 2^k$.

Class $C = \{D,H\}$ is defined to be the set consisting of IUPAC codes D,H, where D is A or G or U (i.e. not C), H is A or C or U (i.e. not G). Note that if m nucleotide positions are constrained by an IUPAC code belonging to class C, and k of these positions are required to have GC-content of k, then the number of sequences satisfying this requirement is $\binom{m}{k} \cdot 1^k \cdot 2^{m-k} = \binom{m}{k} \cdot 2^{m-k}$.

Class $D = \{N\}$ is defined to be the set consisting of IUPAC code N, where N is A or C or G or U (i.e. any nucleotide). Note that if m nucleotide positions are constrained by an IUPAC code belonging to class D, and k of these positions are required to have GC-content of k, then the number of sequences satisfying this requirement is $\binom{m}{k} \cdot 2^k \cdot 2^{m-k} = \binom{m}{k} \cdot 2^m$.

Class $E = \{S\}$ is defined to be the set consisting of IUPAC code S, where S is G or C (i.e. strong). Note that if m nucleotide positions are constrained by an IUPAC code belonging to class E, then there are 2^m many such sequences satisfying this constraint.

Class $F = \{W\}$ is defined to be the set consisting of IUPAC code W, where W is A or U (i.e. weak). Note that if m nucleotide positions are constrained by an IUPAC code belonging to class F, then there are 2^m many such sequences satisfying this constraint.

Class $G = \{A,C,G,U\}$ is defined to be the set consisting of IUPAC codes A,C,G,U (i.e. the data is certain).

Note that there are 15 IUPAC codes, of which 11 concern *uncertain* data; indeed, only IUPAC codes in class G concern *certain* data.

Algorithm: Number of external loops of size n having GC-content of k, allowing IUPAC constraints

INPUT: Integer $n \geq 1$ denoting the length of the external loop, integer $k \geq 0$ denoting the desired GC-content of the external loop, length n sequence of IUPAC constraints specified by $\mathbf{a} = a_1,\ldots,a_n$; i.e. for each $i = 1,\ldots,n$, we have $a_i in \{A,C,G,U,R,Y,M,K,B,V,D,H,N,S,W\}$.

OUTPUT: Number of external loops of size n having GC-content of k, which satisfy the specified IUPAC constraints.

Define the following.

- Let $n_a,n_b,n_c,n_d,n_e,n_f,n_g$ denote the number of positions in the external loop that are constrained by an IUPAC code belonging respectively to class A,B,C,D,E,F,G.
- Let $n_o = n - (n_e + n_f + n_g)$. Note that n_o is the number of positions in the external loop that may be assigned to contain G or C, or equally well may be assigned to contain A or U. We must have that $n_o = n_a + n_b + n_c + n_d$, hence $n_d = n_o - (n_a + n_b + n_c)$.
- Let num_C [resp. num_G] denote the number of positions in the external loop that are constrained by IUPAC code C [resp. G]. Note that a position constrained to be C [resp. G] will contribute both to the count of n_g as well as to the count of num_C [resp. num_G].
- Let $k_o = k - (num_C + num_G + n_e)$. Note that k_o is the number of C's or G's that must be assigned among the n_o positions, taking into consideration that we have already taken care of assignments of C's and G's to positions that are constrained to be C (there are num_C many), or G (there are num_G many), or either C or G (there are n_e many).

- Let k_a,k_b,k_c,k_d denote the number of positions constrained by IUPAC codes that belong respectively to class A,B,C,D that will be set to contain either C or G. Although k_a,k_b,k_c,k_d will take on different values, we will always ensure that $k_0 = k_a + k_b + k_c + k_d$, hence $k = k_a + k_b + k_c + k_d + n_e + num_C + num_G$; in particular, $k_d = k_0 - (k_a + k_b + k_c)$. As well, it clearly must always hold that $0 \leq k_a \leq n_a$, $0 \leq k_b \leq n_b$, $0 \leq k_c \leq n_c$, $0 \leq k_d \leq n_d$.

Careful scrutiny justifies the fact that the number N of external loops of size n having GC-content of k, which satisfy the specified IUPAC constraints, must satisfy the following.

$$
\begin{aligned}
N &= \sum_{k_a=0}^{n_a} \sum_{k_b=0}^{n_b} \sum_{k_c=0}^{n_c} \binom{n_a}{k_a} \cdot \binom{n_b}{k_b} \cdot \binom{n_c}{k_c} \cdot \binom{n_d}{k_d} \cdot \\
&\quad \left(1^{k_a} \cdot 1^{n_a-k_a}\right) \cdot \left(2^{k_b} \cdot 1^{n_b-k_b}\right) \cdot \left(1^{k_c} \cdot 2^{n_c-k_c}\right) \cdot 2^{n_e} \cdot 2^{n_f} \cdot 2^{k_0-(k_a+k_b+k_c)} \cdot 2^{n_d-(k_0-(k_a+k_b+k_c))} \\
&= \sum_{k_a=0}^{n_a} \sum_{k_b=0}^{n_b} \sum_{k_c=0}^{n_c} \binom{n_a}{k_a} \cdot \binom{n_b}{k_b} \cdot \binom{n_c}{k_c} \cdot \binom{n_d}{k_d} \cdot 2^{n_c+n_d+n_e+n_f} \cdot 2^{k_b-k_c}
\end{aligned}
$$

This leads to the following pseudocode.

```
def f(a) //compute n_a,n_b,n_c,n_d,n_e,n_f,n_g,num_C,num_G from IUPAC constraints
1. //a = a_1,...,a_n stipulates IUPAC constraints at all positions of the external loop
2. n_a = n_b = n_c = n_d = n_e = n_f = n_g = num_C = num_G = 0
3. for i = 1 to n
4.   if a_i ∈ {R,Y,M,K}
5.     n_a += 1
6.   else if a_i ∈ {B,V}
7.     n_b += 1
8.   else if a_i ∈ {D,H}
9.     n_c += 1
10.  else if a_i = N
11.    n_d += 1
12.  else if a_i = S
13.    n_e += 1
14.  else if a_i = W
15.    n_f += 1
16.  else if a_i ∈ {A,C,G,U}
17.    n_g += 1
18.    if a_i = C
```

```
19.        num_C += 1
20.      else //a_i must be G
21.        num_G += 1
22. return n_a,n_b,n_c,n_d,n_e,n_f,n_g,num_C,num_G


def computeNumberExternalLoops(n,k,a)
//number external loops of size n with GC-content k given IUPAC constraints a
 1. n_a,n_b,n_c,n_d,n_e,n_f,n_g,num_C,num_G = f(a)
 2. k_o = k − (num_C + num_G + n_e)
 3. N = 0
 4. for k_a = 0 to n_a
 5.   for k_b = 0 to n_b
 6.     for k_c = 0 to n_c
 7.       k_d = k_o − (k_a + k_b + k_c)
 8.       C = (n_a choose k_a) · (n_b choose k_b) · (n_c choose k_c) · (n_d choose k_d) · 2^(n_c+n_d+n_e+n_f) · 2^(k_b−k_c)
 9.       N += C
10. return N
```

Bibliography

[1] A. M. Poole, D. C. Jeffares, and D. Penny, "The path from the RNA world," *Journal of Molecular Evolution*, vol. 46, pp. 1–17, 1998.

[2] L. Lim, M. Glasner, S. Yekta, C. Burge, and D. Bartel, "Vertebrate microRNA genes," *Science*, vol. 299(5612), p. 1540, 2003.

[3] M. Mandal, B. Boese, J. Barrick, W. Winkler, and R. Breaker, "Riboswitches control fundamental biochemical pathways in Bacillus subtilis and other bacteria," *Cell*, vol. 113(5), pp. 577–586, 2003.

[4] M. T. Cheah, A. Wachter, N. Sudarsan, and R. R. Breaker, "Control of alternative RNA splicing and gene expression by eukaryotic riboswitches," *Nature*, vol. 447, pp. 497–500, May 2007.

[5] S. J. Brouns, M. M. Jore, M. Lundgren, E. R. Westra, R. J. Slijkhuis, A. P. Snijders, M. J. Dickman, K. S. Makarova, E. V. Koonin, and J. Van der Oost, "Small CRISPR RNAs guide antiviral defense in prokaryotes," *Science*, vol. 321, pp. 960–964, August 2008.

[6] G. Varani and W. H. McClain, "The G·U wobble base pair," *EMBO reports*, vol. 1, no. 1, pp. 18–23, 2000.

[7] C. Hammann and E. Westhof, "Searching genomes for ribozymes and riboswitches," *Genome Biol*, vol. 8, p. 210, 2007.

[8] J. Doudna and T. Cech, "The chemical repertoire of natural ribozymes," *Nature*, vol. 418, no. 6894, pp. 222–228, 2002.

[9] M. Zuker, D. H. Mathews, and D. H. Turner, "Algorithms and thermodynamics for RNA secondary structure prediction: A practical guide," in *RNA Biochemistry and Biotechnology* (J. Barciszewski and B. Clark, eds.), NATO ASI Series, pp. 11–43, Kluwer Academic Publishers, 1999.

[10] N. R. Markham and M. Zuker, "UNAFold: software for nucleic acid folding and hybridization," *Methods Mol. Biol.*, vol. 453, pp. 3–31, 2008.

[11] I. Hofacker, "Vienna RNA secondary structure server," *Nucleic Acids Res.*, vol. 31, pp. 3429–3431, 2003.

[12] D. Mathews, M. Disney, J. Childs, S. Schroeder, M. Zuker, and D. Turner, "Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure," *Proc. Natl. Acad. Sci. USA*, vol. 101, pp. 7287–7292, 2004.

[13] J. Gorodkin, L. J. Heyer, and G. D. Stormo, "Finding common sequence and structure motifs in a set of RNA sequences," *Proc Int Conf Intell Syst Mol Biol*, vol. 5, pp. 120–123, 1997.

[14] M. Zytnicki, C. Gaspin, , and T. Schiex, "Darna weighted constraint solver for RNA motif localization," *Constraints*, vol. 13, pp. 91–109, 2008.

[15] D. Mathews and D. Turner, "Dynalign: An algorithm for finding the secondary structure common to two RNA sequences," *J. Mol. Biol.*, vol. 317, pp. 191–203, 2002.

[16] J. Gorodkin, L. J. Heyer, and G. D. Stormo, "Finding the most significant common sequence and structure motifs in a set of RNA sequences," *Nucleic Acids Res.*, vol. 25, no. 18, pp. 3724–3732, 1997.

[17] T.-H. Chang, H.-D. Huang, L.-C. Wu, C.-T. Yeh, B.-J. Liu, and J.-T. Horng, "Computational identification of riboswitches based on RNA conserved functional sequences and conformations," *RNA*, vol. 15, no. 7, 2009.

[18] C. Xue, F. Li, T. He, G. P. Liu, Y. Li, and X. Zhang, "Classification of real and pseudo microRNA precursors using local structure-sequence features and support vector machine," *BMC. Bioinformatics*, vol. 6, p. 310, 2005.

[19] S. Washietl and I. L. Hofacker, "Identifying structural noncoding RNAs using RNAz," *Curr Protoc Bioinformatics*, vol. 0, p. O, September 2007.

[20] E. S. Andersen, "Prediction and design of DNA and RNA structures," *N. Biotechnol.*, vol. 27, pp. 184–193, July 2010.

[21] J. N. Zadeh, B. R. Wolfe, and N. A. Pierce, "Nucleic acid sequence design via efficient ensemble defect optimization," *J. Comput. Chem.*, vol. 32, pp. 439–452, February 2011.

[22] Y. Y. Chen, M. C. Jensen, and C. D. Smolke, "Genetic control of mammalian T-cell proliferation with synthetic RNA regulatory systems," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 107, pp. 8531–8536, May 2010.

[23] R. P. Shetty, D. Endy, and T. F. Knight, Jr, "Engineering BioBrick vectors from BioBrick parts," *J. Biol. Eng.*, vol. 2, no. 1, p. 5, 2008.

[24] R. Bellman, "On the approximation of curves by line segments using dynamic programming," *Communications of the ACM*, vol. 4, no. 6, p. 284, 1961.

[25] P. V. Hentenryck and L. Michel, *Constraint-Based Local Search.* MIT Press, 2005. ISBN-10: 0-262-22077-6 ISBN-13: 978-0-262-22077-4.

[26] Google Inc., "Google's or-tools vehicle routing library. https://github.com/google/or-tools," 2012.

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

[28] A. Gruber, R. Lorenz, S. Bernhart, R. Neubock, and I. Hofacker, "The Vienna RNA websuite," *Nucleic Acids Research*, vol. 36, pp. 70–74, 2008.

[29] M. Zuker, "Mfold web server for nucleic acid folding and hybridization prediction," *Nucleic Acids Res.*, vol. 31(13), pp. 3406–3415, 2003.

[30] M. Zuker and P. Stiegler, "Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information," *Nucleic Acids Res.*, vol. 9, pp. 133–148, 1981.

[31] D. Matthews, J. Sabina, M. Zuker, and D. Turner, "Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure," *J. Mol. Biol.*, vol. 288, pp. 911–940, 1999.

[32] T. Xia, J. J. SantaLucia, M. Burkard, R. Kierzek, S. Schroeder, X. Jiao, C. Cox, and D. Turner, "Thermodynamic parameters for an expanded nearest-neighbor model for formation of RNA duplexes with Watson-Crick base pairs," *Biochemistry*, vol. 37, pp. 14719–35, 1999.

[33] R. B. Lyngso and C. N. Pedersen, "RNA pseudoknot prediction in energy-based models," *J. Comput. Biol.*, vol. 7, no. 3-4, pp. 409–427, 2000.

[34] A. Banerjee, J. Jaeger, and D. Turner, "Thermal unfolding of a group I ribozyme: The low-temperature transition is primarily disruption of tertiary structure," *Biochemistry*, vol. 32, pp. 153–163, 1993.

[35] M. H. Bailor, X. Sun, and H. M. Al-Hashimi, "Topology links RNA secondary structure with global conformation, dynamics, and adaptation," *Science*, vol. 327, pp. 202–206, January 2010.

[36] S. S. Cho, D. L. Pincus, and D. Thirumalai, "Assembly mechanisms of RNA pseudoknots are determined by the stabilities of constituent secondary structures," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 106, pp. 17349–17354, October 2009.

[37] K. Wilkinson, E. Merino, and K. Weeks, "RNA SHAPE chemistry reveals nonhierarchical interactions dominate equilibrium structural transitions in $tRNA^{Asp}$," *J. Am. Chem. Soc.*, vol. 127, pp. 4659–4667, 2005.

[38] M. Wu and I. Tinoco Jr, "RNA folding causes secondary structure rearrangement," *PNAS*, vol. 95, no. 20, pp. 11555–11560, 1998.

[39] I. Hofacker, W. Fontana, P. Stadler, L. Bonhoeffer, M. Tacker, and P. Schuster, "Fast folding and comparison of RNA secondary structures," *Monatsch. Chem.*, vol. 125, pp. 167–188, 1994.

[40] M. Andronescu, A. Fejes, F. Hutter, H. Hoos, and A. Condon, "A new algorithm for RNA secondary structure design," *J Mol Biol.*, vol. 336, pp. 607–624, 2004.

[41] A. Busch and R. Backofen, "INFO-RNA, a fast approach to inverse RNA folding," *Bioinformatics*, vol. 22, no. 15, pp. 1823–1831, 2006.

[42] A. Taneda, "MODENA: a multi-objective RNA inverse folding," *Advances and Applications in Bioinformatics and Chemistry*, vol. 4, no. 1, 2011.

[43] J. Gao, L. Li, and C. Reidys, "Inverse folding of RNA pseudoknot structures," *Algorithms for Molecular Biology*, vol. 5, no. 27, 2010.

[44] R. B. Lyngso, J. W. Anderson, E. Sizikova, A. Badugu, T. Hyland, and J. Hein, "Frnakenstein: multiple target inverse RNA folding," *BMC. Bioinformatics*, vol. 13, p. 260, 2012.

[45] A. Esmaili-Taheri and M. Ganjtabesh, "ERD: a fast and reliable tool for RNA design including constraints," *BMC. Bioinformatics*, vol. 16, p. 20, January 2015.

[46] L. Weinbrand, A. Avihoo, and D. Barash, "RNAfbinv: an interactive Java application for fragment-based design of RNA sequences," *Bioinformatics*, vol. 29, pp. 2938–2940, November 2013.

[47] J. Höner zu Siederdissen, S. Hammer, I. Abfalter, I. Hofacker, C. Flamm, and P. Stadler, "Computational design of RNAs with complex energy landscapes," *Biopolymers*, vol. 99, no. 12, pp. 1124–1136, 2013.

[48] J. Lee, W. Kladwang, M. Lee, D. Cantu, M. Azizyan, H. Kim, A. Limpaecher, S. Gaikwad, S. Yoon, A. Treuille, R. Das, and E. Participants, "RNA design rules from a massive open laboratory," *Proceedings of the National Academy of Sciences*, vol. 111, no. 6, pp. 2122–2127, 2014.

[49] V. Reinharz, Y. Ponty, and J. Waldispuhl, "A weighted sampling algorithm for the design of RNA sequences with targeted secondary structure and nucleotide distribution," *Bioinformatics*, vol. 29, pp. i308–i315, July 2013.

[50] 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.

[51] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley and Sons, 2001.

[52] R. Dirks, M. Lin, E. Winfree, and N. Pierce, "Paradigms for computational nucleic acid design," *Nucleic Acids Res.*, vol. 32, no. 4, pp. 1392–1403, 2004.

[53] F. W. Huang, W. W. Peng, and C. M. Reidys, "Folding 3-noncrossing RNA pseudoknot structures.," *J. Comput. Biol.*, vol. 16, pp. 1549–1575, November 2009.

[54] K. Darty, A. Denise, and Y. Ponty, "VARNA: Interactive drawing and editing of the RNA secondary structure," *Bioinformatics*, vol. 25, pp. 1974–1975, Aug. 2009.

[55] J. A. Garcia-Martin, P. Clote, and I. Dotu, "RNAiFold: a constraint programming algorithm for RNA inverse folding and molecular design," *J. Bioinform. Comput. Biol.*, vol. 11, p. 1350001, April 2013.

[56] 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 Research*, vol. 43, no. W1, pp. W513–W521, 2015.

[57] W. Babcock, "Intermodulation interference in radio systems/frequency of occurrence and control by channel selection," *Bell System Technical Journal*, vol. 31, pp. 63–73, 1953.

[58] S. Sidon, "Ein Satz über trigonometrische Polynome und seine Anwendungen in der Theorie der Fourier-Reihen," *Mathematische Annalen*, vol. 106, pp. 536–539, 1932.

[59] 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, January 2010.

[60] M. Andronescu, A. Condon, H. H. Hoos, D. H. Mathews, and K. P. Murphy, "Efficient parameter estimation for RNA secondary structure prediction," *Bioinformatics*, vol. 23, pp. i19–i28, July 2007.

[61] M. Sprinzl, C. Horn, M. Brown, A. Ioudovitch, and S. Steinberg, "Compilation of tRNA sequences and sequences of tRNA genes," *Nucleic Acids Research*, vol. 26, no. 1, pp. 148–153, 1998.

[62] 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, Jan. 2009.

[63] Y. Wan, K. Qu, Q. C. Zhang, R. A. Flynn, O. Manor, Z. Ouyang, J. Zhang, R. C. Spitale, M. P. Snyder, E. Segal, and H. Y. Chang, "Landscape and variation of RNA secondary structure across the human transcriptome," *Nature*, vol. 505, pp. 706–709, Jan. 2014.

[64] E. P. Nawrocki and S. R. Eddy, "Infernal 1.1: 100-fold faster RNA homology searches," *Bioinformatics*, vol. 29, pp. 2933–2935, Nov. 2013.

[65] 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, November 2015.

[66] I. Dotu, G. Lozano, P. Clote, and E. Martinez-Salas, "Using RNA inverse folding to identify IRES-like structural subdomains," *RNA. Biol.*, vol. 10, pp. 1842–1852, December 2013.

[67] J. Fernandez-Chamorro, G. Lozano, J. A. Garcia-Martin, J. Ramajo, I. Dotu, P. Clote, and E. Martinez-Salas, "Designing synthetic RNAs to determine the relevance of structural motifs in picornavirus ires elements," *Scientific Reports*, vol. 6, p. 24243, 2016.

[68] V. G. Kolupaeva, C. U. Hellen, and I. N. Shatsky, "Structural analysis of the interaction of the pyrimidine tract-binding protein with the internal ribosomal entry site of encephalomyocarditis virus and foot-and-mouth disease virus RNAs," *RNA*, vol. 2, no. 12, pp. 1199–1212, 1996.

[69] C. Carrillo, E. R. Tulman, G. Delhon, Z. Lu, A. Carreno, A. Vagnozzi, G. F. Kutish, and D. L. Rock, "Comparative genomics of foot-and-mouth disease virus," *Journal of Virology*, vol. 79, no. 10, pp. 6487–6504, 2005.

[70] S. R. Stewart and B. L. Semler, "Pyrimidine-rich region mutations compensate for a stem-loop v lesion in the 5' noncoding region of poliovirus genomic RNA," *Virology*, vol. 264, no. 2, pp. 385 – 397, 1999.

[71] S. L. de Quinto and E. Martínez-Salas, "Involvement of the aphthovirus RNA region located between the two functional AUGs in start codon selection," *Virology*, vol. 255, no. 2, pp. 324 – 336, 1999.

[72] S. L. de Quinto and E. Martínez-Salas, "Parameters influencing translational efficiency in aphthovirus IRES-based bicistronic expression vectors," *Gene*, vol. 217, no. 1-2, pp. 51–6, 1998.

[73] G. Lozano, N. Fernandez, and E. Martinez-Salas, "Magnesium-dependent folding of a picornavirus IRES element modulates RNA conformation and eIF4G interaction," *FEBS Journal*, vol. 281, no. 16, pp. 3685–3700, 2014.

[74] O. Fernández-Miragall, R. Ramos, J. Ramajo, and E. Martínez-Salas, "Evidence of reciprocal tertiary interactions between conserved motifs involved in organizing RNA structure essential for internal initiation of translation," *RNA*, vol. 12, no. 2, pp. 223–234, 2006.

[75] K. Ochs, R. Rust, and M. Niepmann, "Translation initiation factor eIF4B interacts with a picornavirus internal ribosome entry site in both 48S and 80S initiation complexes independently of initiator AUG location," *Journal of Virology*, vol. 73, no. 9, pp. 7505–7514, 1999.

[76] N. Luz and E. Beck, "Interaction of a cellular 57-kilodalton protein with the internal translation initiation site of foot-and-mouth disease virus," *Journal of Virology*, vol. 65, no. 12, pp. 6486–6494, 1991.

[77] S. H. Bernhart, I. L. Hofacker, and P. F. Stadler, "Local RNA base pairing probabilities in large sequences," *Bioinformatics*, vol. 22, no. 5, pp. 614–615, 2006.

[78] S. Wuchty and W. Fontana and I.L. Hofacker and P. Schuster, "Complete suboptimal folding of RNA and the stability of secondary structures," *Biopolymers*, vol. 49, pp. 145–164, 1999.

[79] S. Commans and A. Böck, "Selenocysteine inserting tRNAs: an overview," *FEMS Microbiology Reviews*, vol. 23, pp. 333–351, 1999.

[80] E. Grundner-Culemann, G. Martin, J. W. Harney, and M. J. Berry, "Two distinct SECIS structures capable of directing selenocysteine incorporation in eukaryotes," *RNA.*, vol. 5, pp. 625–635, May 1999.

[81] Z. Liu, M. Reches, I. Groisman, and H. Engelberg-Kulka, "The nature of the minimal 'selenocysteine insertion sequence' (SECIS) in Escherichia coli," *Nucleic Acids Res.*, vol. 26, pp. 896–902, February 1998.

[82] K. E. Sandman, D. F. Tardiff, L. A. Neely, and C. J. Noren, "Revised Escherichia coli selenocysteine insertion requirements determined by in vivo screening of combinatorial libraries of SECIS variants," *Nucleic Acids Res.*, vol. 31, pp. 2234–2241, April 2003.

[83] T. R. Cech, A. J. Zaug, and P. J. Grabowski, "In vitro splicing of the ribosomal RNA precursor of Tetrahymena: involvement of a guanosine nucleotide in the excision of the intervening sequence," *Cell*, vol. 27, no. 3, pp. 487–496, 1981.

[84] 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.

[85] C. L. Peebles, P. S. Perlman, K. L. Mecklenburg, M. L. Petrillo, J. H. Tabor, K. A. Jarrell, and H. L. Cheng, "A self-splicing RNA excises an intron lariat," *Cell*, vol. 44, no. 2, pp. 213–223, 1986.

[86] S. C. Darr, J. W. Brown, and N. R. Pace, "The varieties of ribonuclease P," *Trends Biochem. Sci.*, vol. 17, no. 5, pp. 178–182, 1992.

[87] H. Pley, K. Flaherty, and D. McKay, "Three-dimensional structure of a hammerhead ribozyme," *Nature*, vol. 372, pp. 68–74, 1994.

[88] J. Murray, D. Terwey, L. Maloney, A. Karpeisky, N. Usman, L. Beigelman, and W. Scott, "The structural basis of hammerhead ribozyme self-cleavage," *Cell*, vol. 92, pp. 665–673, 1998.

[89] T. J. Wilson, M. Nahas, T. Ha, and D. M. Lilley, "Folding and catalysis of the hairpin ribozyme," *Biochem. Soc. Trans.*, vol. 33, no. Pt, pp. 461–465, 2005.

[90] F. J. Isaacs, D. J. Dwyer, and J. J. Collins, "RNA synthetic biology," *Nat. Biotechnol.*, vol. 24, no. 5, pp. 545–554, 2006.

[91] J. Collins, "Synthetic Biology: Bits and pieces come to life.," *Nature*, vol. 483, no. 7387, pp. S8–S10, 2012.

[92] A. D. Ellington and J. W. Szostak, "In vitro selection of RNA molecules that bind specific ligands," *Nature*, vol. 346, no. 6287, pp. 818–822, 1990.

[93] C. Tuerk and L. Gold, "Systematic evolution of ligands by exponential enrichment: RNA ligands to bacteriophage T4 DNA polymerase.," *Science*, vol. 249, no. 4968, pp. 505–510, 1990.

[94] H. Gu, K. Furukawa, and R. R. Breaker, "Engineered allosteric ribozymes that sense the bacterial second messenger cyclic diguanosyl 5'-monophosphate," *Anal. Chem.*, vol. 84, no. 11, pp. 4935–4941, 2012.

[95] N. Piganeau, "In vitro selection of allosteric ribozymes," *Methods Mol. Biol.*, vol. 535, pp. 45–57, 2009.

[96] J. Sinha, S. J. Reyes, and J. P. Gallivan, "Reprogramming bacteria to seek and destroy an herbicide," *Nat. Chem. Biol.*, vol. 6, no. 6, pp. 464–470, 2010.

[97] S. J. Goldfless, B. J. Belmont, A. M. de Paz, J. F. Liu, and J. C. Niles, "Direct and specific chemical control of eukaryotic translation with a synthetic RNA-protein interaction," *Nucleic Acids Research*, vol. 40, pp. e64–e64, May 2012.

[98] B. J. Belmont and J. C. Niles, "Engineering a direct and inducible protein-RNA interaction to regulate RNA biology," *ACS. Chem. Biol.*, vol. 5, no. 9, pp. 851–861, 2010.

[99] E. A. Schultes and D. P. Bartel, "One sequence, two ribozymes: implications for the emergence of new ribozyme folds," *Science*, vol. 289, no. 5478, pp. 448–452, 2000.

[100] F. J. Isaacs, D. J. Dwyer, C. Ding, D. D. Pervouchine, C. R. Cantor, and J. J. Collins, "Engineered riboregulators enable post-transcriptional control of gene expression," *Nat. Biotechnol.*, vol. 22, no. 7, pp. 841–847, 2004.

[101] T. S. Bayer and C. D. Smolke, "Programmable ligand-controlled riboregulators of eukaryotic gene expression," *Nat. Biotechnol.*, vol. 23, no. 3, pp. 337–343, 2005.

[102] C. Zhou, I. C. Bahner, G. P. Larson, J. A. Zaia, J. J. Rossi, and E. B. Kohn, "Inhibition of HIV-1 in human T-lymphocytes by retrovirally transduced anti-tat and rev hammerhead ribozymes," *Gene.*, vol. 149, no. 1, pp. 33–39, 1994.

[103] G. Bauer, P. Valdez, K. Kearns, I. Bahner, S. F. Wen, J. A. Zaia, and D. B. Kohn, "Inhibition of human immunodeficiency virus-1 (HIV-1) replication after transduction of granulocyte colony-stimulating factor-mobilized $CD34^+$ cells from HIV-1-infected donors using retroviral vectors containing anti-HIV-1 genes," *Blood*, vol. 89, no. 7, pp. 2259–2267, 1997.

[104] B. A. Shapiro, E. Bindewald, W. Kasprzak, and Y. Yingling, "Protocols for the in silico design of RNA nanostructures," *Methods Mol. Biol.*, vol. 474, pp. 93–115, 2008.

[105] E. Bindewald, C. Grunewald, B. Boyle, M. O'Connor, and B. A. Shapiro, "Computational strategies for the automated design of RNA nanoscale structures from building blocks using NanoTiler," *J. Mol. Graph. Model.*, vol. 27, no. 3, pp. 299–308, 2008.

[106] K. A. Afonin, W. W. Grabow, F. M. Walker, E. Bindewald, M. A. Dobrovolskaia, B. A. Shapiro, and L. Jaeger, "Design and self-assembly of siRNA-functionalized RNA nanoparticles for use in automated nanomedicine," *Nat. Protoc.*, vol. 6, no. 12, pp. 2022–2034, 2011.

[107] P. Yin, H. M. Choi, C. R. Calvert, and N. A. Pierce, "Programming biomolecular self-assembly pathways," *Nature*, vol. 451, no. 7176, pp. 318–322, 2008.

[108] K. F. Blount and O. C. Uhlenbeck, "The structure-function dilemma of the hammerhead ribozyme," *Annu. Rev. Biophys. Biomol. Struct.*, vol. 34, pp. 415–440, 2005.

[109] M. Martick and W. G. Scott, "Tertiary contacts distant from the active site prime a ribozyme for catalysis," *Cell*, vol. 126, no. 2, pp. 309–320, 2006.

[110] J. A. Nelson and O. C. Uhlenbeck, "Hammerhead redux: does the new structure fit the old biochemical data?," *RNA*, vol. 14, no. 4, pp. 605–615, 2008.

[111] W. H. Pan, P. Xin, V. Bui, and G. A. Clawson, "Rapid identification of efficient target cleavage sites using a hammerhead ribozyme library in an iterative manner," *Mol. Ther.*, vol. 7, no. 1, pp. 129–139, 2003.

[112] M. A. Gonzalez-Carmona, S. Schussler, M. Serwe, M. Alt, J. Ludwig, B. S. Sproat, R. Steigerwald, P. Hoffmann, M. Quasdorff, O. Schildgen, and W. H. Caselmann, "Hammerhead ribozymes with cleavage site specificity for NUH and NCH display significant anti-hepatitis C viral effect in vitro and in recombinant HepG2 and CCL13 cells," *J. Hepatol.*, vol. 44, no. 6, pp. 1017–1025, 2006.

[113] G. E. Crooks, G. Hon, J. M. Chandonia, and S. E. Brenner, "WebLogo: a sequence logo generator," *Genome Res.*, vol. 14, no. 6, pp. 1188–1190, 2004.

[114] E. P. Nawrocki, D. L. Kolbe, and S. R. Eddy, "Infernal 1.0: inference of RNA alignments," *Bioinformatics*, vol. 25, no. 10, pp. 1335–1337, 2009.

[115] M. Huynen, R. Gutell, and D. Konings, "Assessing the reliability of RNA folding using statistical mechanics," *J. Mol. Biol.*, vol. 267, pp. 1104–1112, April 1997.

[116] P. Higgs, "Overlaps between RNA secondary structures," *Phys. Rev. Lett.*, vol. 76, pp. 704–707, 1996.

[117] J. A. Garcia-Martin, P. Clote, and I. Dotu, "RNAiFold: a web server for RNA inverse folding and molecular design," *Nucleic Acids Res.*, vol. 41, pp. W465–W470, July 2013.

[118] M. Wieland and J. S. Hartig, "Improved aptazyme design and in vivo screening enable riboswitching in bacteria," *Angew. Chem. Int. Ed. Engl.*, vol. 47, no. 14, pp. 2604–2607, 2008.

[119] A. Saragliadis, S. S. Krajewski, C. Rehm, F. Narberhaus, and J. S. Hartig, "Thermozymes: Synthetic RNA thermometers based on ribozyme activity," *RNA. Biol.*, vol. 10, no. 6, pp. 1010–1016, 2013.

[120] 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," *Chem. Biol.*, vol. 11, no. 12, pp. 1729–1741, 2004.

[121] E. Freyhult, V. Moulton, and P. Clote, "Boltzmann probability of RNA structural neighbors and riboswitch detection," *Bioinformatics*, vol. 23, no. 16, pp. 2054–2062, 2007. doi: 10.1093/bioinformatics/btm314.

[122] E. Senter, S. Sheik, I. Dotu, Y. Ponty, and P. Clote, "Using the Fast Fourier Transform to accelerate the computational search for RNA conformational switches," *PLoS One*, vol. 7, no. 12, p. e50506, 2012.

[123] B. Clouet-d'Orval and O. C. Uhlenbeck, "Hammerhead ribozymes with a faster cleavage rate," *Biochemistry.*, vol. 36, no. 30, pp. 9087–9092, 1997.

[124] 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, February 2015.

[125] A. Carbonell, M. De la Pena, R. Flores, and S. Gago, "Effects of the trinucleotide preceding the self-cleavage site on eggplant latent viroid hammerheads: differences in co- and post-transcriptional self-cleavage may explain the lack of trinucleotide AUC in most natural hammerheads," *Nucleic Acids Res.*, vol. 34, no. 19, pp. 5613–5622, 2006.

[126] H. James and I. Gibson, "The therapeutic potential of ribozymes," *Blood*, vol. 91(2), pp. 371–381, 1998.

[127] Z. Weinberg and R. R. Breaker, "R2R–software to speed the depiction of aesthetic consensus RNA secondary structures," *BMC. Bioinformatics*, vol. 12, p. 3, 2011.

[128] A. Nocker, T. Hausherr, S. Balsiger, N. P. Krstulovic, H. Hennecke, and F. Narberhaus, "A mRNA-based thermosensor controls expression of rhizobial heat shock genes," *Nucleic Acids Res.*, vol. 29, pp. 4800–4807, Dec. 2001.

[129] T. Waldminghaus, N. Heidrich, S. Brantl, and F. Narberhaus, "FourU: a novel type of RNA thermometer in Salmonella," *Mol. Microbiol.*, vol. 65, pp. 413–424, July 2007.

[130] Z. Torok, P. Goloubinoff, I. Horvath, N. M. Tsvetkova, A. Glatz, G. Balogh, V. Varvasovszki, D. A. Los, E. Vierling, J. H. Crowe, and L. Vigh, "Synechocystis HSP17 is an amphitropic protein that stabilizes heat-stressed membranes and binds denatured proteins for subsequent chaperone-mediated refolding," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 98, pp. 3098–3103, Mar. 2001.

[131] J. Kortmann, S. Sczodrok, J. Rinnenthal, H. Schwalbe, and F. Narberhaus, "Translation on demand by a simple RNA-based thermosensor," *Nucleic Acids Res.*, vol. 39, pp. 2855–2868, Apr. 2011.

[132] J. Kortmann and F. Narberhaus, "Bacterial RNA thermometers: molecular zippers and switches," *Nat. Rev. Microbiol.*, vol. 10, pp. 255–265, Apr. 2012.

[133] J. Johansson, P. Mandin, A. Renzoni, C. Chiaruttini, M. Springer, and P. Cossart, "An RNA thermosensor controls expression of virulence genes in Listeria monocytogenes," *Cell*, vol. 110, pp. 551–561, Sept. 2002.

[134] S. Altuvia, D. Kornitzer, D. Teff, and A. B. Oppenheim, "Alternative mRNA structures of the cIII gene of bacteriophage lambda determine the rate of its translation initiation," *J. Mol. Biol.*, vol. 210, pp. 265–280, Nov. 1989.

[135] W. Bae, B. Xia, M. Inouye, and K. Severinov, "Escherichia coli CspA-family RNA chaperones are transcription antiterminators," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 97, pp. 7784–7789, July 2000.

[136] T. Waldminghaus, L. C. Gaubig, and F. Narberhaus, "Genome-wide bioinformatic prediction and experimental evaluation of potential RNA thermometers," *Mol. Genet. Genomics.*, vol. 278, pp. 555–564, Nov. 2007.

[137] A. Chursov, S. J. Kopetzky, G. Bocharov, D. Frishman, and A. Shneider, "RNAtips: Analysis of temperature-induced changes of RNA secondary structure," *Nucleic Acids Res.*, vol. 41, pp. W486–W491, July 2013.

[138] A. Churkin, A. Avihoo, M. Shapira, and D. Barash, "RNAthermsw: direct temperature simulations for predicting the location of RNA thermometers," *PLoS. One*, vol. 9, no. 4, p. e94340, 2014.

[139] R. Lorenz, S. H. Bernhart, C. Honer 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.

[140] M. Wieland and J. S. Hartig, "RNA quadruplex-based modulation of gene expression," *Chem. Biol.*, vol. 14, pp. 757–763, July 2007.

[141] T. Waldminghaus, J. Kortmann, S. Gesing, and F. Narberhaus, "Generation of synthetic RNA-based thermosensors," *Biol. Chem.*, vol. 389, no. 10, pp. 1319–1326, 2008.

[142] J. Lee and J. Kotov, "Thermometer design at the nanoscale," *Nano Today*, vol. 2, no. 1, pp. 48–51, 2007.

[143] J. Neupert, D. Karcher, and R. Bock, "Design of simple synthetic RNA thermometers for temperature-controlled gene expression in Escherichia coli," *Nucleic Acids Res.*, vol. 36, p. e124, Nov. 2008.

[144] A. Hoynes-O'Connor, K. Hinman, L. Kirchner, and T. S. Moon, "De novo design of heat-repressible RNA thermosensors in E. coli," *Nucleic Acids Res.*, vol. 43, pp. 6166–6179, July 2015.

[145] C. Flamm, I. L. Hofacker, S. Maurer-Stroh, P. F. Stadler, and M. Zehl, "Design of multi-stable RNA molecules," *RNA*, vol. 7, pp. 254–265, 2001.

[146] S. Gilbert, R. Montange, C. Stoddard, and R. Batey, "Structural studies of the purine and SAM binding riboswitches," *Cold Spring Harbor Symposia on Quantitative Biology*, vol. 71, pp. 259–268, 2006.

[147] A. Haller, U. Rieder, M. Aigner, S. C. Blanchard, and R. Micura, "Conformational capture of the SAM-II riboswitch," *Nat Chem Biol*, vol. 7, pp. 393–400, Jun 2011.

[148] A. Reining, S. Nozinovic, K. Schlepckow, F. Buhr, B. Furtig, and H. Schwalbe, "Three-state mechanism couples ligand and temperature sensing in riboswitches," *Nature*, vol. 499, pp. 355–359, Jul 2013. Letter.

[149] E. R. Lee, J. L. Baker, Z. Weinberg, N. Sudarsan, and R. R. Breaker, "An allosteric self-splicing ribozyme triggered by a bacterial second messenger," *Science*, vol. 329, no. 5993, pp. 845–848, 2010.

[150] J. C. Cochrane, S. V. Lipchock, and S. A. Strobel, "Structural investigation of the glms ribozyme bound to its catalytic cofactor," *Chemistry & Biology*, vol. 14, no. 1, pp. 97 – 105, 2007.

[151] A. Ogawa, "Rational design of artificial riboswitches based on ligand-dependent modulation of internal ribosome entry in wheat germ extract and their applications as label-free biosensors," *RNA*, vol. 17, pp. 478–488, Mar. 2011.

[152] M. Wachsmuth, S. Findeiss, N. Weissheimer, P. F. Stadler, and M. Morl, "De novo design of a synthetic riboswitch that regulates transcription termination," *Nucleic Acids Res.*, vol. 41, pp. 2541–2551, February 2013.

[153] M. Wachsmuth, G. Domin, R. Lorenz, R. Serfling, S. Findeiß, P. F. Stadler, and M. Mörl, "Design criteria for synthetic riboswitches acting on transcription," *RNA Biology*, vol. 12, no. 2, pp. 221–231, 2015. PMID: 25826571.

[154] M. Mandal, M. Lee, J. E. Barrick, Z. Weinberg, G. M. Emilsson, W. L. Ruzzo, and R. R. Breaker, "A glycine-dependent riboswitch that uses cooperative binding to control gene expression," *Science*, vol. 306, no. 5694, pp. 275–279, 2004.

[155] T. Endoh and N. Sugimoto, "Rational design and tuning of functional RNA switch to control an allosteric intermolecular interaction," *Analytical Chemistry*, vol. 87, no. 15, pp. 7628–7635, 2015. PMID: 26122192.

[156] A. Espah Borujeni, D. M. Mishler, J. Wang, W. Huso, and H. M. Salis, "Automated physics-based design of synthetic riboswitches from diverse RNA aptamers," *Nucleic Acids Research*, vol. 44, no. 1, pp. 1–13, 2016.

[157] J. S. Reuter and D. H. Mathews, "RNAstructure: software for RNA secondary structure prediction and analysis," *BMC. Bioinformatics*, vol. 11, p. 129, 2010.

[158] J. Rinnenthal, B. Klinkert, F. Narberhaus, and H. Schwalbe, "Modulation of the stability of the Salmonella fourU-type RNA thermometer," *Nucleic Acids Res.*, vol. 39, pp. 8258–8270, Oct. 2011.

[159] S. Chowdhury, C. Ragaz, E. Kreuger, and F. Narberhaus, "Temperature-controlled structural alterations of an RNA thermometer," *J. Biol. Chem.*, vol. 278, pp. 47915–47921, Nov. 2003.

[160] R. Kuhn, N. Luz, and E. Beck, "Functional analysis of the internal translation initiation site of foot-and-mouth disease virus," *J. Virol.*, vol. 64, pp. 4625–4631, Oct. 1990.

[161] N. Fernandez, O. Fernandez-Miragall, J. Ramajo, A. García-Sacristan, N. Bellora, E. Eyras, C. Briones, and E. Martinez-Salas, "Structural basis for the biological relevance of the invariant apical stem in IRES-mediated translation," *Nucleic Acids Res.*, vol. 39, pp. 8572–8585, 2011.

[162] G. Lozano and E. Martinez-Salas, "Structural insights into viral IRES-dependent translation mechanisms," *Curr. Opin. Virol.*, vol. 12, pp. 113–120, June 2015.

[163] G. Lozano, A. Trapote, J. Ramajo, X. Elduque, A. Grandas, J. Robles, E. Pedroso, and E. Martinez-Salas, "Local RNA flexibility perturbation of the IRES element induced by a novel ligand inhibits viral RNA translation," *RNA Biol.*, vol. 12, no. 5, pp. 555–568, 2015.

[164] D. Pineiro, N. Fernandez, J. Ramajo, and E. Martinez-Salas, "Gemin5 promotes IRES interaction and translation control through its C-terminal region," *Nucleic. Acids. Res.*, vol. 41, pp. 1017–1028, Jan. 2013.

[165] J. Fernandez-Chamorro, D. Pineiro, J. M. Gordon, J. Ramajo, R. Francisco-Velilla, M. J. Macias, and E. Martinez-Salas, "Identification of novel non-canonical RNA-binding sites in Gemin5 involved in internal initiation of translation," *Nucleic. Acids. Res.*, vol. 42, pp. 5742–5754, May 2014.

[166] R. Jenison, S. Gill, A. Pardi, and B. Polisky, "High-resolution molecular discrimination by RNA," *Science*, vol. 263, no. 5152, pp. 1425–1429, 1994.

[167] C. Flamm and I.L. Hofacker and P.F. Stadler and M. Wolfinger, "Barrier trees of degenerate landscapes," *Z. Phys. Chem.*, vol. 216, pp. 155–173, 2002.

[168] S. Bernhart, H. Tafer, U. Mückstein, C. Flamm, P. Stadler, and I. Hofacker, "Partition function and base pairing probabilities of RNA heterodimers," *Algorithms Mol Biol*, vol. 1, no. 1, p. 3, 2006.

[169] C. Reidys, P. Stadler, and P. Schuster, "Generic properties of combinatory maps: neutral networks of RNA secondary structures," *Bull Math Biol.*, vol. 59(2), pp. 339–397, 1997.

[170] M. S. Marlow, J. Dogan, K. K. Frederick, K. G. Valentine, and A. J. Wand, "The role of conformational entropy in molecular recognition by calmodulin," *Nat. Chem. Biol.*, vol. 6, pp. 352–358, May 2010.

[171] A. J. Wand, "The dark energy of proteins comes to light: conformational entropy and its role in protein function revealed by NMR relaxation," *Curr. Opin. Struct. Biol.*, vol. 23, pp. 75–81, Feb. 2013.

[172] M. Karplus, T. Ichiye, and B. M. Pettitt, "Configurational entropy of native proteins," *Biophys. J.*, vol. 52, pp. 1083–1085, Dec. 1987.

[173] K. W. Harpole and K. A. Sharp, "Calculation of configurational entropy with a Boltzmann-quasiharmonic model: the origin of high-affinity protein-ligand binding," *J. Phys. Chem. B.*, vol. 115, pp. 9461–9472, Aug. 2011.

[174] J. I. Tinoco and M. Schmitz, "Thermodynamics of formation of secondary structure in nucleic acids," in *Thermodynamics in Biology* (E. D. Cera, ed.), pp. 131–176, Oxford University Press, 2000.

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

[176] M. Hammell, D. Long, L. Zhang, A. Lee, C. S. Carmack, M. Han, Y. Ding, and V. Ambros, "mirWIP: microRNA target prediction based on microRNA-containing ribonucleoprotein-enriched transcripts," *Nat. Methods*, vol. 5, pp. 813–819, Sept. 2008.

[177] H. M. Choi, J. Y. Chang, A. Trinh le, J. E. Padilla, S. E. Fraser, and N. A. Pierce, "Programmable in situ amplification for multiplexed imaging of mRNA expression," *Nat. Biotechnol.*, vol. 28, pp. 1208–1212, Nov. 2010.

[178] T. D. Schneider and R. M. Stephens, "Sequence logos: a new way to display consensus sequences," *Nucleic Acids Res.*, vol. 18, pp. 6097–6100, October 1990.

[179] E. Bindewald, T. D. Schneider, and B. A. Shapiro, "Correlogo: an online server for 3D sequence logos of RNA and DNA alignments," *Nucleic Acids Res.*, vol. 34, pp. W405–W411, July 2006.

[180] J. Gorodkin, L. J. Heyer, S. Brunak, and G. D. Stormo, "Displaying the information contents of structural RNA alignments: the structure logos," *Comput. Appl. Biosci.*, vol. 13, pp. 583–586, December 1997.

[181] H. Kazan, D. Ray, E. T. Chan, T. R. Hughes, and Q. Morris, "RNAcontext: a new method for learning the sequence and structure binding preferences of RNA-binding proteins," *PLoS. Comput. Biol.*, vol. 6, p. e1000832, 2010.

[182] H. Kazan and Q. Morris, "RBPmotif: a web server for the discovery of sequence and structure preferences of RNA-binding proteins," *Nucleic Acids Res.*, vol. 41, pp. W180–W186, July 2013.

[183] D. H. Mathews, "Using an RNA secondary structure partition function to determine confidence in base pairs predicted by free energy minimization," *RNA*, vol. 10, pp. 1178–1190, August 2004.

[184] A. Manzourolajdad, Y. Wang, T. I. Shaw, and R. L. Malmberg, "Information-theoretic uncertainty of SCFG-modeled folding space of the non-coding RNA," *J. theor. Biol.*, vol. 318, pp. 140–163, February 2013.

[185] Z. Sukosd, B. Knudsen, J. W. Anderson, A. Novak, J. Kjems, and C. N. Pedersen, "Characterising RNA secondary structure space using information entropy," *BMC. Bioinformatics*, vol. 14, p. S22, 2013.

[186] J. W. Anderson, A. Novak, Z. Sukosd, M. Golden, P. Arunapuram, I. Edvardsson, and J. Hein, "Quantifying variances in comparative RNA secondary structure prediction," *BMC. Bioinformatics*, vol. 14, p. 149, 2013.

[187] Z. Sukosd, 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, p. 103, 2011.

[188] R. D. Dowell and S. R. Eddy, "Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction," *BMC. Bioinformatics*, vol. 5, p. 71, June 2004.

[189] R. Nussinov and A. B. Jacobson, "Fast algorithm for predicting the secondary structure of single stranded RNA," *Proceedings of the National Academy of Sciences, USA*, vol. 77, no. 11, pp. 6309–6313, 1980.

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

[191] K. Dill and S. Bromberg, *Molecular Driving Forces: Statistical Thermodynamics in Chemistry and Biology*. Garland Publishing Inc., 2002. 704 pages.

[192] Y. Shao, S. Wu, C. Y. Chan, J. R. Klapper, E. Schneider, and Y. Ding, "A structural analysis of in vitro catalytic activities of hammerhead ribozymes," *BMC. Bioinformatics*, vol. 8, p. 469, 2007.

[193] S. W. Burge, J. Daub, R. Eberhardt, J. Tate, L. Barquist, E. P. Nawrocki, S. R. Eddy, P. P. Gardner, and A. Bateman, "Rfam 11.0: 10 years of RNA families," *Nucleic Acids Res.*, vol. 41, pp. D226–D232, January 2013.

[194] R. Giegerich, D. Haase, and M. Rehmsmeier, "Prediction and visualization of structural switches in RNA," in *Pacific Symposium on Biocomputing* (R. Altman, A. Dunker, L. Hunter, and T. Klein, eds.), pp. 126–137, World Scientific, 1999.

[195] S. Altschul and B. Erikson, "Significance of nucleotide sequence alignments: A method for random sequence permutation that preserves dinucleotide and codon usage," *Mol. Biol. Evol*, vol. 2(6), pp. 526–538, 1985.

[196] P. Clote, F. Ferré, E. Kranakis, and D. Krizanc, "Structural RNA has lower folding energy than random RNA of the same dinucleotide frequency," *RNA*, vol. 11(5), pp. 578–591, 2005.

[197] S. Griffiths-Jones, H. K. Saini, S. Van Dongen, and A. J. Enright, "miRBase: tools for microRNA genomics," *Nucleic Acids Res.*, vol. 36, pp. D154–D158, Jan. 2008.

[198] E. Rivas and S. Eddy, "Secondary structure alone is generally not statistically significant for the detection of noncoding RNA," *Bioinformatics*, vol. 16, pp. 573–585, 2000.

[199] K. L. Ng and S. K. Mishra, "De novo SVM classification of precursor microRNAs from genomic pseudo hairpins using global and intrinsic folding measures," *Bioinformatics*, vol. 23, pp. 1321–1330, June 2007.

[200] E. Rivas, R. Lang, and S. R. Eddy, "A range of complex probabilistic models for RNA secondary structure prediction that includes the nearest-neighbor model and more," *RNA*, vol. 18, pp. 193–212, February 2012.

[201] W. A. Lorenz, Y. Ponty, and P. Clote, "Asymptotics of RNA shapes," *J. Comput. Biol.*, vol. 15, no. 1, pp. 31–63, 2008.

[202] E. Fusy and P. Clote, "Combinatorics of locally optimal RNA secondary structures," *J. Math. Biol.*, vol. 68, pp. 341–375, January 2014.

[203] Y. Ding, C. Y. Chan, and C. E. Lawrence, "Sfold web server for statistical folding and rational design of nucleic acids," *Nucleic Acids Res.*, vol. 32, p. 0, 2004.

[204] R. Dimitrov and M. Zuker, "Prediction of hybridization and melting for double-stranded nucleic acids," *Biophys J*, vol. 87, no. 1, pp. 215–226, 2004.

[205] K. Lu, X. Heng, and M. F. Summers, "Structural determinants and mechanism of HIV-1 genome packaging," *J. Mol. Biol.*, vol. 410, pp. 609–633, July 2011.

[206] 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.

[207] G. Rodrigo and M. A. Fares, "Describing the structural robustness landscape of bacterial small RNAs," *BMC Evolutionary Biology*, vol. 12, no. 1, pp. 1–12, 2012.

[208] "Los Alamos HIV database." `http://www.hiv.lanl.gov/`, 2015. Accessed: 2015-12-30.

[209] J. Krol, K. Sobczak, U. Wilczynska, M. Drath, A. Jasinska, D. Kaczynska, and W. J. Krzyzosiak, "Structural features of microRNA (miRNA) precursors and their relevance to mirna biogenesis and small interfering RNA/short hairpin RNA design," *Journal of Biological Chemistry*, vol. 279, no. 40, pp. 42230–42239, 2004.

[210] A. R. Gruber, S. H. Bernhart, I. L. Hofacker, and S. Washietl, "Strategies for measuring evolutionary conservation of RNA secondary structures," *BMC Bioinformatics*, vol. 9, no. 1, pp. 1–19, 2008.

[211] N. R. Markham, *Algorithms and software for nucleic acid sequences.* PhD thesis, Rensselaer Polytechnic Institute, Troy, New York, 5 2006.

[212] M. Zuker, "On finding all suboptimal foldings of an RNA molecule," *Science*, vol. 244, pp. 48–52, April 1989.

[213] 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.

[214] W. A. Lorenz and P. Clote, "Computing the partition function for kinetically trapped RNA secondary structures," *PLoS. One*, vol. 6, no. 1, p. e16178, 2011.

[215] S. Pei, J. S. Anthony, and M. M. Meyer, "Sampled ensemble neutrality as a feature to classify potential structured RNAs," *BMC Genomics*, vol. 16, no. 1, pp. 1–12, 2015.

[216] M. S. Waterman, "Secondary structure of single-stranded nucleic acids," *Studies in Foundations and Combinatorics, Advances in Mathematics Supplementary Studies*, vol. 1, pp. 167–212, 1978.

Index of terms