Boston College

Doctoral Thesis

On the use of Coarse Grained Thermodynamic Landscapes to Efficiently Estimate Kinetic Pathways for RNA Molecules

---

Author:
Evan Senter

Supervisor:
Dr. Peter Clote

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

in the

Clote Lab
Department of Biology

Wednesday 1st July, 2015

Contents

Chapter 1

Ribofinder

1.1 Introduction

In this chapter, we present the `Ribofinder` program—a pipeline to facilitate the detection of putative guanine riboswitches across genomic data. The `Ribofinder` tool operates in three stages. First we use `Infernal` and `TransTermHP` to detect putative aptamers and expression platforms, two distinct components of riboswitches described in section 1.2. After coalescing this data into a pool of candidate riboswitches, we use `RNAfold` with constraints based on experimental data to compute the two distinct structural conformations—'gene on' and 'gene off'. In the third and final stage, we leverage `FoldAlign` to measure the similarity between our candidate pool and a canonical guanine riboswitch well studied in the literature, the xpt G-box riboswitch from Bacillus subtilis.

1.1.1 Organization

This chapter is organized in the following fashion. After providing background on the structural components of a riboswitch alongside their biological significance, we outline the deficiencies in the 'state of the art' software when as it relates specifically to riboswitch detection. We then move on to outline the three stages of `Ribofinder`: candidate selection, structural prediction, and candidate curation. Having described the approach of the software, we move on to present our findings in using `Ribofinder` to detect guanine riboswitches across the bacterial RefSeq database. Finally, we provide

brief commentary on possible extensions of the algorithm to locate other flavors of riboswitches, of which adenine-sensitive aptamers are a straightforward extension.

1.2 Background

Riboswitches are regulatory mRNA elements that modulate gene expression via structural changes induced by the direct sensing of a small-molecule metabolite. Most often found in bacteria, riboswitches regulate diverse pathways including the metabolism and transport of purines, methionine, and thiamin amongst others. The structure of a riboswitch includes an aptamer domain—involved in the direct sensing of the small-molecule—and an expression platform whose structure changes upon the aptamer binding the metabolite. Because of the discriminatory nature of metabolite sensing, groups have had great success in finding representative examples of aptamers across a diverse collection of bacterial species; RFam 12.0 currently contains 26 different families of aptamers involved in different metabolic pathways. Whereas there exists strong sequence and structural similarity within the aptamer of a riboswitch family, the expression platform is highly variable, and thus challenging to capture using traditional SCFG-based approaches. For this reason databases such as RFam only contain the aptamer portion of the riboswitch, and there exists no database providing sequences including expression platforms, necessary for capturing the 'on' and 'off' conformations of this regulatory element. We have developed a new pipeline—called `Ribofinder`—which can detect putative riboswitches including their expression platforms and likely conformational structures across a wide collection of genomic sequences.

1.3 The `Ribofinder` pipeline

At the time of our retrieval (Tuesday 25$^{\text{th}}$ November, 2014 at 09:14), the RefSeq database hosted by NCBI comprised 5,121 complete bacterial genomes with corresponding genomic annotations. In order to both detect putative full riboswitches across this collection of data as well as filter the candidates down to a number tractable for experimental validation, we developed a novel pipeline which takes a three-tiered approach to candidate selection. Our approach is to *a*) identify a pool of candidate riboswitches across genomic data; *b*) perform a coarse-grained filtering of the candidate pool based on structural characteristics; and finally *c*) fine-grained curation of the candidates based on a collection of measures and pairwise similarity.

In the following discussion, we describe the application of `Ribofinder` to identify unannotated G-box purine riboswitches; guanine-sensing cis-regulatory elements which modulate the expression of genes involved in purine biosynthesis.

1.3.1 Step 1: Candidate selection

The RefSeq data we used for analysis contains 5,121 annotated bacterial genomes across 2,732 different organisms, totaling over $9.5 * 10^9$ bases. We used the program `Infernal` to determine the coordinates of putative aptamer structures within the RefSeq genomes, and `TransTermHP` to locate candidate rho-independent transcription terminators.

1.3.1.1 Detecting Aptamers with `Infernal`

`Infernal` uses a stochastic context-free grammar (SCFG) with a user-provided multiple sequence alignment (MSA) to efficiently scan genomic data for RNA homologs, taking into consideration both sequence and structural conservation. Using the purine aptamer MSA from RFam 12.0 (RF00167), `Infernal` (v1.1.1, default options) detects 1,537 significant hits having E-value <= 0.01. Because `Infernal` leverages the concept of a 'local end'—a large insertion or deletion in the alignment at reduced cost—it is possible for the software to return a significant hit whose aligned structure does not have the canonical three-way junction observed in all purine riboswitches. `Ribofinder` prunes these truncated `Infernal` hits by converting the alignment structure into a parse tree, and only permitting trees of sufficient complexity to contain a multiloop (described further in 1.3.2.1). The pyrimidine residue abutted next to the P1 stem in the J3-1 junction differentiates between guanine and adenine-sensing riboswitches by binding the complimentary purine ligand; for our interest in G-box riboswitches exclusively we require the presence of a cytosine at this residue. In total, using `Infernal` with these additional filters yields 1,280 G-box aptamers across 555 unique organisms (note: here and elsewhere I define a 'unique organism' as having a unique taxonomy ID).

1.3.1.2 Detecting Expression Platforms with `TransTermHP`

`TransTermHP` detects rho-independent terminators in bacterial genomes in a context-sensitive fashion by leveraging the protein annotations available in PTT data. These terminator sequences canonically have a stable hairpin loop structure immediately preceding a run of 5+ uracil residues, the combination of which causes the ribosomal machinery to stall and dissociate from the transcript. `TransTermHP` performs a genomic

scan to determine candidate loci with this motif, and returns scored hits. The scoring system considers both structural homology and the genomic contextual information available in the PTT file. Across our collection of bacterial genomes acquired from NCBI RefSeq data, `TransTermHP` identified 2,752,469 rho-independent terminators using the default filters.

Due to the spatially-mediated structural regulation of purine riboswitches, whereby ligand interaction with the aptamer domain induces local structural rearrangement in the expression platform, we paired aptamers with corresponding terminators by minimizing the genomic distance, with an upper bound of 200 nucleotides between the end of the aptamer domain and start of the terminator. This approach yields 577 candidate riboswitches, 81 of which have multiple rho-independent terminators within range of a putative aptamer produced by `Infernal`. For these, we simply pair the closest `TransTermHP` hit with the aptamer domain.

1.3.2 Step 2: Structural prediction

1.3.2.1 Notation for Representing Abstract RNA Shapes

Given an RNA sequence $\mathbf{s} = a_1, a_2, \cdots, a_n$, where positions a_i are drawn from the collection of single-letter nucleotide codes, i.e. $\{\mathrm{A}, \mathrm{U}, \mathrm{G}, \mathrm{C}\}$, it is possible to describe a corresponding secondary structure $\mathcal{S}$ compatible with $\mathbf{s}$ using the dot-bracket notation. In this notation, each nucleotide a_i has a corresponding state s_i, where s_i is denoted as a '.' if unpaired and a '(' [resp. ')'] if the left [resp. right] base in a basepair. Given any two basepairs (i, j) and (k, l) in $\mathcal{S}$, then $i < k < j \iff i < l < j$; pseudoknots are not permitted in the structure. A secondary structure taking this form is said to have balanced parentheses, and can additionally be represented using a context-free grammar such as:

$$S \to S\,. \mid .\,S \mid (S) \mid SS \mid \epsilon \tag{1.1}$$

The grammar from (1.1) can be used to generate a parse tree $\mathcal{T}$ for $\mathcal{S}$. The benefit of working with $\mathcal{T}$ over $\mathcal{S}$ is that the parse tree offers an abstract representation of secondary structure shape independent of sequence length, permitting us to classify and eventually constrain a large collection of sequences having variable length which are all expected to have the same abstract tree shape. This is analogous to what the Giegerich

lab refers to as their 'type 5' structural abstraction using the `RNAshapes` tool. Every node in $\mathcal{T}$ represents a helix in $\mathcal{S}$, and internally tracks the indices of both its beginning (i, j) and closing (k, l) basepair. We use a level-order naming convention to refer to helices within the parse tree, whereby a position $\mathtt{p}_1$ references the first child of the root node, $\mathtt{p}_{1,2}$ references the second child of $\mathtt{p}_1$, and generally $\mathtt{p}_{i_1, i_2, \cdots, i_n}$ refers to the ${i_n}^{\text{th}}$ child of $\mathtt{p}_{i_1, i_2, \cdots, i_{n-1}}$. To reference specific nucleotides in the context of their location relative to a helix, we use the opening and closing basepairs (i, j) and (k, l) as landmarks. Thus, $\mathtt{p}_1(l)$ is the index in $\mathcal{S}$ of the right-hand side closing basepair of $\mathtt{p}_1$. We use the notation $\mathtt{t}_i$ to refer to the subtree of $\mathcal{T}$ whose root is $\mathtt{p}_i$.

Finally, we introduce the concept of a tree signature. The tree signature for a tree $\mathcal{T}$ is a list of the node depths when traversed in a depth-first pre-order fashion. To provide a concrete example, consider the following experimentally validated xpt G-box riboswitch from Bacillus subtilis subsp. subtilis str. 168 (NC_000964.3 2320197-2320054) with corresponding gene-off structure:

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

The `RNAshapes` 'type 5' representation for this structure is `[[][]][][]` (note the coalesced left bulge in the hairpin immediately downstream the closing multiloop stem, at helix $\mathtt{p}_2$) and the tree signature for this parse tree of the structure is `[0, 1, 2, 2, 1, 1]`.

We leverage the notion of abstract structural filtering initially to ensure that all `Infernal` aptamer hits have a tree signature of `[0, 1, 2, 2]`, which represents a three-way junction, and that the binding site for the guanine ligand $\mathtt{p}_1(l-1) = \mathrm{C}$. These filters, in combination with the proximal terminator hairpins produced by `TransTermHP` yield the aforementioned 577 candidate guanine riboswitches for which we then try to produce reasonable gene-on and off structures.

1.3.2.2 Constrained Folding to Predict Switch Structures

To restrict our search to unannotated G-box riboswitches, and further ensure that we are not re-detecting sequences based off the RFam covariance model provided to `Infernal`, we constrain our search to those RefSeq organisms not represented in the RFam seed alignment. 503 of the 577 candidates, or 87.18% represent putative unannotated riboswitches not represented by RF00167.

The gene-off structure $\mathcal{S}_{\text{off}}$ for a G-box riboswitch is the easier of the two to find computationally, since the terminator loop is exceptionally thermodynamically stable. In the gene-on conformation $\mathcal{S}_{\text{on}}$, the P1 stem of the multiloop partially dissociates and an anti-terminator loop forms between the region immediately 3' of the P1 stem and what was the left-hand side of the terminator loop. This truncated P1 stem, which closes the three-way junction in the aptamer, is exceptionally unstable based on present energy models available for structural folding, and requires special treatment to reconstitute in our final structures.

The software `RNAfold` (v2.1.8) allows for the folding of RNA molecules with 'loose' constraints. In this model of constrained folding, the resulting structure produced by the software guarantees not to explicitly invalidate any user-provided constraints, but does not guarantee all constraints will be satisfied in the resulting structure. For each of the candidate guanine riboswitches, having $\mathcal{T}_{\texttt{Infernal}}$ and $\mathcal{T}_{\texttt{TransTermHP}}$, we build the following constraint masks:

G-box gene-off constraint mask	G-box gene-on constraint mask
Prohibit basepairing upstream of $\mathtt{p}_1(i)$ and downstream of $\mathtt{p}_2(l)$. Force basepairs and unpaired regions in $\mathtt{t}_1$, with the exception of $\mathtt{p}_1$. Prohibit formation of $\mathtt{p}_1$ stem, which closes the three-way junction.	
Force basepairs and unpaired regions in $\mathtt{t}_2$.	Require m nucleotides starting from $\mathtt{p}_1(l+3)$ to pair to the right, where $m = len(\mathtt{p}_2)$, and require the left-hand side of the $\mathtt{p}_2$ helix to pair to the left. Disallow pairing downstream of $\mathtt{p}_2(j)$.

These constraint masks are run using the command-line flags `-d 0 -P rna_turner1999.par` to disable dangles and use the Turner 1999 energies respectively. Experimental evidence using inline probing suggests that the 'on' conformation of the G-box riboswitch has a reduced P1 stem length of 3 base pairs; in practice we were unable to force `RNAfold` to respect this constraint regardless of command-line options specified. For this reason we reconstitute the P1 stem in both structures after constrained folding, having length equivalent to it the `Infernal` P1 stem (resp. 3 basepairs) in the gene-off (resp. gene-on) structure.

This difficulty with `RNAfold` can be shown by using the constraint-produced structures as exhaustive constraints themselves. All unpaired nucleotides in $\mathcal{S}_{\text{off}}$ and $\mathcal{S}_{\text{on}}$ are notated

by a '`x`' and all base pairs by '`()`' for the 5' and 3' side of the pair respectively to form new constraints mask $\mathcal{C}_{\text{off}}$ and $\mathcal{C}_{\text{on}}$, having all bases' state explicitly specified. By refolding all 577 candidate sequences with $\mathcal{C}_{\text{off}}$ and $\mathcal{C}_{\text{on}}$ using the same options as before, only 463 (or 80.24%) of the resulting structures from $\mathcal{C}_{\text{off}}$ have the tree signature prefix `[0, 1, 2, 2, 1]`, and just 21 (or 3.64%) of the $\mathcal{C}_{\text{on}}$ structures correctly re-fold their multiloop.

1.3.3 Step 3: Candidate curation

Until now, we have described our approach for generating the 503 guanine riboswitch candidates in RefSeq, alongside their gene-on and off structures. Unfortunately the experimental validation of all 503 candidates is not tractable, so it was necessary to reduce this collection again to a more manageable size, while only keeping the most promising candidates.

1.4 Using **Ribofinder** against the RefSeq database

1.5 Extending beyond guanine riboswitches

Chapter 2

FFTbor

2.1 Introduction

In this chapter, we present the `FFTbor` algorithm and accompanying software. `FFTbor` is a novel algorithm developed with the intent of efficiently computing the Boltzmann probability of those structures whom, for a given input RNA sequence $\mathbf{s}$, differ by k base pairs. By leveraging polynomial interpolation via the Fast Fourier Transform, this algorithm runs in $O(n^4)$ time and $O(n^2)$ space, a significant improvement over its predecessor. The accompanying software which implements this algorithm has been used to predict the location of expression platforms for putative riboswitches in genomic data, and to evaluate the correlation between kinetic folding speed and landscape ruggedness.

2.1.1 Organization

This chapter is organized in the following fashion. First, we provide background on the problem which `FFTbor` aims to address, as well as a brief overview of existing approaches. We follow by a formal explanation of the problem, and proceed to describe how the energy landscape is coarsified into discrete bins. We then develop the recursions for the parameterized partition function using the Nussinov-Jacobson energy model, which allows us to exposé the novel aspects of the algorithm. After developing the recursions, we indicate how they can be reformulated as a polynomial whose coefficients $c_k = \mathbf{Z}^k_{1,n}$. We then describe how the Fast Fourier Transform can be employed to efficiently compute the coefficients c_k, finishing our description of the underlying algorithm. Then we proceed to present two applications of `FFTbor`, an application to RNA kinetics and another

to riboswitch detection in genomic data. Finally, we give reference to the full recursions using the more accurate Turner energy model, which the underlying implementation actually uses.

2.2 Background

2.3 Formalization of the problem

`FFTbor` aims to compute the coefficients $p_0, \ldots, p_{n-1}$ of the polynomial

$$p(x) = p_0 + p_1 x + p_2 x^2 + \cdots + p_{n-1} x^{n-1}, \tag{2.1}$$

where p_k is defined as $p_k = \frac{Z_k}{Z}$. We employ the Fast Fourier Transform to compute the inverse Discrete Fourier Transform on values $y_0, \ldots, y_{n-1}$, where $y_k = p(\omega^k)$ and $\omega = e^{2\pi i/n}$ is the principal nth complex root of unity and $p(x)$ is defined in (2.1). By leveraging complex roots of unity in conjunction with the inverse Discrete Fourier Transform the we subvert numeric instability issues observed with both Lagrange interpolation and Gaussian elimination.

Consider an RNA sequence $\mathbf{s} = s_1, \ldots, s_n$, where $s_i \in \{A, C, G, U\}$, i.e. a sequence of nucleotides. We can describe a secondary structure $\mathcal{S}$ which is compatible with $\mathbf{s}$ as a collection of base pair tuples (i, j), where $1 \leq i \leq i + \theta < j \leq n$ and $\theta \geq 0$—generally taken to be 3, the minimum number of unpaired bases in a hairpin loop due to steric constraints.

To more simply develop the underlying recursions for `FFTbor`, we introduce a number of constraints on the base pairs within $\mathcal{S}$. Firstly, we require that each base pair is either a Watson-Crick or G-U wobble, i.e. base pair (i, j) for sequence $\mathbf{s}$ has corresponding nucleotides (s_i, s_j), which are restricted to the set

$$\mathbb{B} = \{(A, U), (U, A), (G, C), (C, G), (G, U), (U, G)\}.$$

With this constraint satisfied we say that $\mathcal{S}$ is *compatible* with $\mathbf{s}$, and for the remainder of this chapter will only consider those structures which are compatible with $\mathbf{s}$. Secondly, we insist that given two base pairs $(i, j), (x, y)$ from $\mathcal{S}$, $i = x \iff j = y$—bases have at most one partner. Finally, we require that $i < x < j \iff i < y < j$, no pseudoknots are allowed. While pseudoknots have been shown to be present in some biologically

relevant RNAs, their inclusion greatly complicates the recursive decomposition of the structure, and thus it is common to ignore them.

Provided two secondary structures $\mathcal{S}, \mathcal{T}$, we can define a notion of distance between them. There are a number of different definitions of distance used across the literature; we will use *base pair distance* for `FFTbor`. Base pair distance is defined as the symmetric difference between the sets $\mathcal{S}$,$\mathcal{T}$

$$d_{\mathrm{BP}}(\mathcal{S}, \mathcal{T}) = |\mathcal{S} \cup \mathcal{T}| - |\mathcal{S} \cap \mathcal{T}|. \tag{2.2}$$

Given this definition of distance, two structures $\mathcal{S}$ and $\mathcal{T}$ are said to be *k-neighbors* f $d_{\mathrm{BP}}(\mathcal{S}, \mathcal{T}) = k$. It is important to note that the notion of base pair distance is also applicable to restrictions of secondary structures on the subsequence $\mathbf{s}_{i,j}$, i.e. $\mathcal{S}_{i,j} = \{(x, y) : i \leq x < y \leq j, (x, y) \in \mathcal{S}\}$.

For a restriction of base pairs for a given structure $\mathcal{S}_{i,j}$, $\mathcal{T}_{i,j}$ is said to be a *k-neighbor* f $\mathcal{S}_{i,j}$ if

$$d_{\mathrm{BP}}(\mathcal{S}_{i,j}, \mathcal{T}_{i,j}) = |\{(x, y) : i \leq x < y \leq j, (x, y) \in \mathcal{S} - \mathcal{T} \text{ or } (x, y) \in \mathcal{T} - \mathcal{S}\}| = k.$$

2.4 Derivation of the `FFTbor` algorithm

Given an RNA sequence $\mathbf{s} = s_1, \ldots, s_n$ and compatible secondary structure $\mathcal{S}^*$, let $\mathbf{Z}^k$ denote the sum of the Boltzmann factors $\exp(-E(\mathcal{S})/RT)$ of all k-neighbors $\mathcal{S}$ of $\mathcal{S}^*$; i.e.

$$\mathbf{Z}^k = \mathbf{Z}^k_{1,n} = \sum_{\substack{\mathcal{S} \text{ such that} \\ d_{\mathrm{BP}}(\mathcal{S}, \mathcal{S}^*) = k}} \exp^{\frac{-E(\mathcal{S})}{RT}}$$

where $E(\mathcal{S})$ denotes the Turner (nearest neighbor) energy of $\mathcal{S}$, $R = 0.00198$ kcal/mol denotes the universal gas constant and T denotes absolute temperature. From this, it follows that the full partition function is defined as

$$\mathbf{Z} = \mathbf{Z}_{1,n} = \sum_{k=0}^{n} \mathbf{Z}^k_{1,n} \tag{2.3}$$

since the base pair distance between $\mathcal{S}^*$ and $\mathcal{S}$ is at most

$$d_{\mathrm{BP}}(\mathcal{S}^*, \mathcal{S}) \leq |\mathcal{S}^*| + \lfloor \frac{n - \theta}{2} \rfloor \leq n. \tag{2.4}$$

We can then define the Boltzmann probability of all *k-neighbors* of $\mathcal{S}^*$ as

$$p(k) = \frac{\mathbf{Z}_{1,n}^k}{\mathbf{Z}_{1,n}}. \tag{2.5}$$

By visualizing the probabilities p_k as a function of k, we generate a coarse grained view of the one-dimensional energy landscape of $\mathbf{s}$ with respect to $\mathcal{S}^*$. When $\mathcal{S}^*$ is taken to be the minimum free energy structure for example, one would anticipate to see a peak at $k = 0$, with additional peaks implying additional metastable structures; local energy minima which could suggest an energetic trap while folding.

2.4.1 Definition of the partition function $\mathbf{Z}_{1,n}^k$

2.4.2 Recursions to compute the polynomial $\mathcal{Z}_{i,j}(x)$

2.4.3 Polynomial interpolation to evaluate $\mathcal{Z}_{i,j}(x)$

2.5 Coarse-grained kinetics with `FFTbor`

2.6 Riboswitch detection with `FFTbor`

2.7 Benchmarking and performance considerations

Chapter 3

FFTbor2D

3.1 Introduction

In this chapter, we present the `FFTbor2D` algorithm and accompanying software. `FFTbor2D`, like `FFTbor` described in Chapter 2, is an algorithm which computes the paramerized partition function for an input RNA sequence `s`. `FFTbor2D` computes the two-dimensional coarse energy landscape for `s` given two compatible input secondary structures $\mathcal{A}$ and $\mathcal{B}$, where position (x, y) on the discrete energy landscape corresponds to the Boltzmann probability for those structures $\mathcal{S}$ which have $d_{BP}(\mathcal{S}, \mathcal{A}) = x$ and $d_{BP}(\mathcal{S}, \mathcal{B}) = y$ (where d_{BP} is as defined in equation 2.2). By again leveraging the Fast Fourier Transform, `FFTbor2D` runs in $O(n^5)$ time and only uses $O(n^2)$ space—a significant improvement over previous approaches. This permits the output energy landscape to be used in a high-throughput fashion to analyze folding kinetics; a topic covered in detail in Chapter 4.

3.1.1 Organization

This chapter is organized in the following fashion. Because the history for this work arises naturally from the background described in section 2.2, we forego reiteration and instead fall directly into a technical discussion of the underlying algorithm. We first develop the recursions for the Nussinov energy model for expository clarity, the underlying implementation uses the more complicated and robust Turner energy model. Recursions in place, we then move to show how these lead to a single variable polynomial

$P(x)$ whose coeffecients can be computed by the inverse Discrete Fourier Transform, and map to the 2D energy landscape. We describe two exploitations of $P(x)$, a parity condition and complex conjugates which further reduce the runtime by a factor of 4. Finally, we contrast this software against `RNA2Dfold`, and outline the performance characteristics of both softwares and highlight the benefits and drawbacks of both.

3.2 Derivation of the `FFTbor2D` algorithm

3.2.1 Definition of the partition function $\mathbf{Z}_{1,n}^{x,y}$

3.2.2 Recursions to compute the polynomial $\mathcal{Z}_{i,j}(x)$

3.2.3 Polynomial interpolation to evaluate $\mathcal{Z}_{i,j}(x)$

3.3 Acceleration of the `FFTbor2D` algorithm

3.3.1 Optimization due to parity condition

3.3.2 Optimization due to complex conjugates

3.4 Benchmarking and performance considerations

3.5 Applications of the `FFTbor2D` algorithm

Chapter 4

Hermes

4.1 Introduction

In this chapter, we present the `Hermes` software suite—a collection of programs aimed at evaluating the kinetic properties of RNA molecules. Provided a coarse-grained energy landscape generated by `FFTbor2D` (described in Chapter 3), we present software which computes both the mean first passage time and equilibrium time for this discretized energy landscape. We also provide software which computes the exact kinetics for an RNA molecule, however since this requires exhaustive enumeration of all secondary structures—which is known to be an exponential quantity for the length of the RNA in consideration—the full kinetics are not expected to be practical for anything beyond a sequence of trivial length. The software in `Hermes` presents a practical application of the energy landscapes computed by the `FFTbor2D` algorithm. Contrasted against the other kinetics software in the field, `Hermes` offers similar accuracy with unparalleled performance which opens up the possibility for large-scale kinetic analysis *in silico*, which we expect to be of use for synthetic design.

4.1.1 Organization

This chapter is organized in the following fashion. We begin by providing background on the state-of-the-art approaches for kinetic analysis of RNAs. From there, we move into a technical discussion of two traditional approaches for kinetics, computation of the mean first passage time and the equilibrium time. With this foundation in place, we proceed

to discuss the high-level organization of the `Hermes` software package, and describe in detail each of the four underlying programs which comprise the kinetics suite. We then move on to present comparitive benchmarking of `Hermes` against other methods, before finally concluding with some remarks on the accuracy and applicability of `Hermes` to computational RNA design.

4.2 Background

4.3 Traditional approaches for kinetics

4.3.1 Mean first passage time

4.3.2 Equilibrium time

4.4 Software within the `Hermes` suite

4.4.1 Exact mean first passage time with `RNAmfpt`

4.4.2 Approximate mean first passage time with `FFTmfpt`

4.4.3 Exact equilibrium time with `RNAeq`

4.4.4 Approximate equilibrium time with `FFTeq`

4.4.4.1 Population occupancy curves with `FFTeq`

4.5 Correlations of kinetics data across software