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

Tuesday 7th July, 2015

Contents

Chapter 1

FFTbor

1.1 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.

1.1.1 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.

1.2 Background

1.3 Formalization of the problem

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

$$p(x) = p_0 + p_1 x + p_2 x^2 + \cdots + p_{n-1} x^{n-1}, \tag{1.1}$$

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 equation (1.1). 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 $s_i \in \{\mathrm{A}, \mathrm{U}, \mathrm{G}, \mathrm{C}\}$, 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

$$\mathbb{B} = \{(\mathrm{A}, \mathrm{U}), (\mathrm{U}, \mathrm{A}), (\mathrm{G}, \mathrm{C}), (\mathrm{C}, \mathrm{G}), (\mathrm{G}, \mathrm{U}), (\mathrm{U}, \mathrm{G})\}. \tag{1.2}$$

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 $\mathcal{S}, \mathcal{T}$

$$d_{\mathrm{BP}}(\mathcal{S}, \mathcal{T}) = |\mathcal{S} \cup \mathcal{T}| - |\mathcal{S} \cap \mathcal{T}|. \tag{1.3}$$

Given this definition of distance, two structures $\mathcal{S}$ and $\mathcal{T}$ are said to be *k-neighbors* if $d_{\mathrm{BP}}(\mathcal{S}, \mathcal{T}) = k$. 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* of $\mathcal{S}_{[i,j]}$ if

$$d_{\mathrm{BP}}(\mathcal{S}_{[i,j]}, \mathcal{T}_{[i,j]}) = |\{(x,y) : i \leq x < y \leq j, (x,y) \in \mathcal{S} - \mathcal{T} \mathrm{or} (x,y) \in \mathcal{T} - \mathcal{S}\}| = k. \tag{1.4}$$

1.4 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.

$$\mathbf{Z}^k = \mathbf{Z}^k_{1,n} = \sum_{\substack{\mathcal{S} \text{ such that} \\ d_{\mathrm{BP}}(\mathcal{S}, \mathcal{S}^*) = k}} e^{\frac{-E(\mathcal{S})}{RT}} \tag{1.5}$$

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

$$\mathbf{Z} = \mathbf{Z}_{1,n} = \sum_{k=0}^{n} \mathbf{Z}_{1,n}^{k} \tag{1.6}$$

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

$$d_{\mathrm{BP}}(\mathcal{S}^*, \mathcal{S}) \leq |\mathcal{S}^*| + \lfloor \frac{n-\theta}{2} \rfloor \leq n. \tag{1.7}$$

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

$$p(k) = \frac{\mathbf{Z}_{1,n}^{k}}{\mathbf{Z}_{1,n}}. \tag{1.8}$$

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.

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

For the rest of the paper, we consider both $\mathbf{s}$ as well as the secondary structure $\mathcal{S}^*$ on $\mathbf{s}$ to be fixed. We now recall the recursions from Freyhult et al. **?** to determine the partition function $\mathbf{Z}_{i,j}^{k}$ with respect to the Nussinov-Jacobson energy E_0 model **?**, defined by -1 times the number of base pairs; i.e. $E_0(S) = -1 \cdot |S|$. Although we describe here the recursions for the Nussinov-Jacobson model, for the sake of simplicity of exposition, both `RNAbor` **?** as well as our current software `FFTbor`, concern the Turner energy model, consisting of free energy parameters for stacked bases, hairpins, bulges, internal loops and multiloops.

The base case for $\mathbf{Z}_{i,j}^{k}$ is given by

$$\mathbf{Z}_{i,j}^{0} = 1, \text{ for } i \leq j, \tag{1.9}$$

since the only 0-neighbor to a structure $\mathcal{S}^*$ is the structure $\mathcal{S}^*$ itself, and

$$\mathbf{Z}_{i,j}^{k} = 0, \text{ for } k > 0, i \leq j \leq i + \theta, \tag{1.10}$$

since the empty structure is the only possible structure for a sequence shorter than $\theta + 2$ nucleotides, and so there are no *k-neighbors* for $k > 0$. The recursion used to compute $\mathbf{Z}_{i,j}^{k}$ for $k > 0$ and $j > i + \theta$ is

$$\mathbf{Z}_{i,j}^{k} = \mathbf{Z}_{i,j-1}^{k-b_0} + \sum_{\substack{(s_r,s_j)\in\mathbb{B},\\ i\leq r<j}} \sum_{w+w'=k-b(r)} \exp(-E_0(r,j)/RT) \cdot \mathbf{Z}_{i,r-1}^{w} \mathbf{Z}_{r+1,j-1}^{w'}, \tag{1.11}$$

where $E_0(r, j) = -1$ if positions r, j can pair in sequence $\mathbf{s}$, and otherwise $E_0(r, j) = +\infty$. Additionally, $b_0 = 1$ if j is base-paired in $\mathcal{S}^*_{[i,j]}$ and 0 otherwise, and $b(r) = d_{\mathrm{BP}}(\mathcal{S}^*_{[i,j]}, \mathcal{S}^*_{[i,r-1]} \cup \mathcal{S}^*_{[r+1,j-1]} \cup \{(r, j)\})$. This holds since in a secondary structure $T_{[i,j]}$ on $s_i, \ldots, s_j$ that is a *k-neighbor* of $\mathcal{S}^*_{[i,j]}$, either nucleotide j is unpaired in $[i, j]$ or it is paired to a nucleotide r such that $i \leq r < j$. In this latter case it is enough to study the smaller sequence segments $[i, r - 1]$ and $[r + 1, j - 1]$ noting that, except for (r, j), base pairs outside of these regions are not allowed, since there are no pseudoknots. In addition, for $d_{\mathrm{BP}}(\mathcal{S}^*_{[i,j]}, T_{[i,j]}) = k$ to hold, it is necessary for $w + w' = k - b(r)$ to hold, where $w = d_{\mathrm{BP}}(\mathcal{S}^*_{[i,r-1]}, T_{[i,r-1]})$ and $w' = d_{\mathrm{BP}}(\mathcal{S}^*_{[r+1,j-1]}, T_{[r+1,j-1]})$, since $b(r)$ is the number of base pairs that differ between $\mathcal{S}^*_{[i,j]}$ and a structure $T_{[i,j]}$, due to the introduction of the base pair (r, j).

Given RNA sequence $\mathbf{s}$ and compatible initial structure $\mathcal{S}^*$, we define the *polynomial*

$$\mathcal{Z}(x) = \sum_{k=0}^{n} c_k x^k \tag{1.12}$$

where coefficients $c_k = \mathbf{Z}_{1,n}^{k}$. Moreover, because of (1.7) and the fact that the minimum number of unpaired bases in a hairpin loop θ is 3, we know that $c_n = 0$, so that $\mathcal{Z}(x)$ is a polynomial of degree strictly less than n. If we evaluate the polynomial $\mathcal{Z}(x)$ for n distinct values

$$\mathcal{Z}(a_1) = y_1, \ldots, \mathcal{Z}(a_n) = y_n, \tag{1.13}$$

then the Lagrange polynomial interpolation formula guarantees that $\mathcal{Z}(x) = \sum_{k=1}^{n} y_k P_k(x)$, where the polynomials $P_k(x)$ have degree at most $n-1$ and are given by the Lagrange formula

$$P_k(x) = \frac{\prod_{i \neq k}(x - x_i)}{\prod_{i \neq k}(x_k - x_i)}. \tag{1.14}$$

Since the polynomials $P_k(x)$ can be explicitly computed, it follows that we can compute the coefficients c_k of polynomial $\mathcal{Z}(x)$. As we describe below, the evaluation of $\mathcal{Z}(x)$ for a fixed value of x can be done in time $O(n^3)$ and space $O(n^2)$. It follows that the coefficients $c_k = \mathbf{Z}_{1,n}^k$ can be computed after n evaluations of $\mathcal{Z}(x)$, where the space for each evaluation of $\mathcal{Z}(x)$ is re-used; hence these evaluations can be performed in time $O(n^4)$ and space $O(n^2)$. Finally, Lagrange interpolation is clearly computable in time $O(n^3)$. Although this approach is theoretically sound, there are severe numerical stability issues related to the interpolation method **?**, the choice of values $a_1, \ldots, a_n$ in the interpolation, and floating point arithmetic (round-off error) related to the astronomically large values of the partition functions $\mathbf{Z}_{1,n}^k$, for $0 \leq k < n$. After many unsuccessful approaches including scaling we obtained excellent results by interpolating the polynomial $p(x)$, defined in equation (1.1), rather than the polynomial $\mathcal{Z}(x)$, defined in equation (1.12), and performing interpolation with the Fast Fourier Transform (FFT) **?** where $\alpha_0, \ldots, \alpha_{n-1}$ are chosen to be nth complex roots of unity, $\alpha_k = e^{2\pi i k/n}$. One advantage of the FFT is that interpolation can be performed in $O(n \log n)$ time, rather than the cubic time required by using the Lagrange formula (1.14) or by Gaussian elimination. Fewer numerical operations implies increased numerical stability in our application.

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

Given an initial secondary structure $\mathcal{S}^*$ of a given RNA sequence $\mathbf{s}$, our goal is to compute

$$\mathbf{Z}_{1,n}^k = \sum_{\substack{\mathcal{S} \text{ such that} \\ d_{\mathrm{BP}}(\mathcal{S}, \mathcal{S}^*) = k}} e^{\frac{-E_0(\mathcal{S})}{RT}} \tag{1.15}$$

where $\mathcal{S}$ can be any structure compatible with $\mathbf{s}$. As previously mentioned, the recurrence relation for `RNAbor` with respect to the Nussinov energy model E_0 is

$$\mathbf{Z}_{i,j}^{k} = \mathbf{Z}_{i,j-1}^{k-b_0} \; + \sum_{\substack{s_r s_j \in \mathbb{B}, \\ i \le r < j}} \left(e^{\frac{-E_0(r,j)}{RT}} \sum_{w+w'=k-b(r)} \mathbf{Z}_{i,r-1}^{w} \mathbf{Z}_{r+1,j-1}^{w'} \right) \tag{1.16}$$

where $E_0(r,j) = -1$ if r and j can base-pair and otherwise $+\infty$, and $b_0 = 1$ if j is base paired in $\mathcal{S}^*_{[i,j]}$ and 0 otherwise, and $b(r) = d_{\mathrm{BP}}(\mathcal{S}^*_{[i,j]}, \mathcal{S}^*_{[i,r-1]} \cup \mathcal{S}^*_{[r+1,j-1]} \cup \{(r,j)\})$. The following theorem shows that an analogous recursion can be used to compute the *polynomial* $\mathcal{Z}_{i,j}(x)$ defined by

$$\mathcal{Z}_{i,j}(x) = \sum_{k=0}^{n} c_k(i,j)\, x^k \tag{1.17}$$

where

$$c_k(i,j) \; = \; \mathbf{Z}_{i,j}^{k} \; = \sum_{\substack{\mathcal{S} \text{ such that} \\ d_{\mathrm{BP}}(\mathcal{S}, \mathcal{S}^*_{[i,j]}) = k}} e^{\frac{-E_0(\mathcal{S})}{RT}}. \tag{1.18}$$

Here, in the summation, $\mathcal{S}$ runs over structures on $s_i, \ldots, s_j$, which are *k-neighbors* of the restriction $\mathcal{S}^*_{[i,j]}$ of initial structure $\mathcal{S}^*$ to interval $[i,j]$, and $E_0(S) = -1 \cdot |S|$ denotes the Nussinov-Jacobson energy of $\mathcal{S}$.

Theorem 1.1. *Let $s_1, \ldots, s_n$ be a given RNA sequence. For any integers $1 \le i \le j \le n$, let*

$$\mathcal{Z}_{i,j}(x) = \sum_{k=0}^{n} c_k\, x^k \tag{1.19}$$

where

$$c_k(i,j) = \mathbf{Z}_{i,j}^{k}. \tag{1.20}$$

Then for $i \le j \le i + \theta$, $\mathcal{Z}_{i,j}(x) = 1$ and for $j > i + \theta$ we have the recurrence relation

$$\mathcal{Z}_{i,j}(x) = \mathcal{Z}_{i,j-1}(x) \cdot x^{b_0} + \sum_{\substack{s_r s_j \in \mathbb{B}, \\ i \le r < j}} \left(e^{\frac{-E_0(r,j)}{RT}} \cdot \mathcal{Z}_{i,r-1}(x) \cdot \mathcal{Z}_{r+1,j-1}(x) \cdot x^{b(r)} \right). \quad (1.21)$$

where $b_0 = 1$ *if* j *is base-paired in* $\mathcal{S}^*_{[i,j]}$ *and* 0 *otherwise, and* $b(r) = d_{BP}(\mathcal{S}^*_{[i,j]}, \mathcal{S}^*_{[i,r-1]} \cup \mathcal{S}^*_{[r+1,j-1]} \cup \{(r,j)\})$.

Proof. First, some notation is necessary. Recall that if F is an arbitrary polynomial [resp. analytic] function, then $[x^k]F(x)$ denotes the coefficient of x^k [resp. the kth Taylor coefficient in the Taylor expansion of F]. For instance, in equation (1.1), $[x^k]p(x) = p_k$, and in equation (1.12), $[x^k]\mathcal{Z}(x) = c_k(i,j)$.

By definition, it is clear that $\mathcal{Z}_{i,j}(x) = 1$ if $i \le j \le i + \theta$, where we recall that $\theta = 3$ is the minimum number of unpaired bases in a hairpin loop. For $j > i + \theta$, we have

$$\begin{aligned}
[x^k]\mathcal{Z}_{i,j}(x) = c_k(i,j) &= \mathbf{Z}^k_{i,j} \\
&= \mathbf{Z}^{k-b_0}_{i,j-1} + \sum_{r=i}^{j-1} \sum_{k_0+k_1=k-b(r)} e^{\frac{-E_0(r,j)}{RT}} \cdot \mathbf{Z}^{k_0}_{i,r-1} \cdot \mathbf{Z}^{k_1}_{r+1,j-1} \\
&= [x^{k-b_0}]\mathcal{Z}_{i,j-1}(x) \\
&+ \sum_{r=i}^{j-1} \sum_{k_0+k_1=k-b(r)} e^{\frac{-E_0(r,j)}{RT}} \cdot \left\{ [x^{k_0}]\mathcal{Z}_{i,r-1}(x) \right\} \cdot \left\{ [x^{k_1}]\mathcal{Z}_{r-1,j-1}(x) \right\} \\
&= [x^{k-b_0}]\mathcal{Z}_{i,j-1}(x) \\
&+ \sum_{r=i}^{j-1} \sum_{k_0+k_1=k-b(r)} e^{\frac{-E_0(r,j)}{RT}} \cdot [x^{k_0+k_1}]\mathcal{Z}_{i,r-1}(x)\mathcal{Z}_{r-1,j-1}(x).
\end{aligned} \quad (1.22)$$

By induction, the proof of the theorem now follows. □

Notice that if one were to compute all terms of the polynomial $\mathcal{Z}_{1,n}(x)$ by explicitly performing polynomial multiplications, then the computation would require $O(n^5)$ time and $O(n^3)$ space. Instead of explicitly performing polynomial expansion in *variable* x, we instantiate x to a fixed complex number $\alpha \in \mathbb{C}$, and apply the following recursion for this instantiation:

$$\mathcal{Z}_{i,j}(\alpha) = \mathcal{Z}_{i,j-1}(\alpha) \cdot \alpha^{b_0} + \sum_{\substack{(s_r,s_j)\in\mathbb{B},\\ i\le r<j}} \left(e^{\frac{-E_0(r,j)}{RT}} \cdot \mathcal{Z}_{i,r-1}(\alpha) \cdot \mathcal{Z}_{r+1,j-1}(\alpha) \cdot \alpha^{b(r)} \right). \quad (1.23)$$

In this fashion, we can compute $\mathcal{Z}(\alpha) = \mathcal{Z}_{1,n}(\alpha)$ in $O(n^3)$ time and $O(n^2)$ space. For n distinct complex values $\alpha_0, \ldots, \alpha_{n-1}$, we can compute and save only the values $\mathcal{Z}(\alpha_0), \ldots, \mathcal{Z}(\alpha_{n-1})$, each time re-using the $O(n^2)$ space for the next computation of $\mathcal{Z}(\alpha_k)$. It follows that the computation resources used to determine the (column) vector

$$\mathbf{Y} = (y_0, \ldots, y_{n-1})^{\mathrm{T}} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix} \quad (1.24)$$

where $y_0 = \mathcal{Z}(\alpha_0), \ldots, y_{n-1} = \mathcal{Z}(\alpha_{n-1}))$ is thus quartic time $O(n^4)$ and quadratic space $O(n^2)$.

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

Let $\omega = e^{2\pi i/n}$ be the principal nth complex root of unity. Recall that the Vandermonde matrix V_n is defined to be the $n \times n$ matrix, whose i, j entry is $\omega^{i \cdot j}$; i.e.

$$V_n = \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & \omega & \omega^2 & \ldots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \ldots & \omega^{2(n-1)} \\ 1 & \omega^3 & \omega^6 & \ldots & \omega^{3(n-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \ldots & \omega^{(n-1)(n-1)} \end{pmatrix} \quad (1.25)$$

The Fast Fourier Transform (FFT) is defined to be the $O(n \log n)$ algorithm to compute the Discrete Fourier Transform (DFT), defined as the matrix product $\mathbf{Y} = V_n \mathbf{A}$:

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{pmatrix} = V_n \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{n-1} \end{pmatrix} \tag{1.26}$$

On page 837 of **?**, it is shown that the (i, j) entry of V_n^{-1} is $\frac{\omega^{-ji}}{n}$ and that

$$a_j = \frac{1}{n} \sum_{k=0}^{n-1} y_k \, \omega^{-kj} \tag{1.27}$$

for $j = 0, \ldots, n-1$.

Since we defined $\mathbf{Y}$ in (1.24) by $\mathbf{Y} = (y_0, \ldots, y_{n-1})^{\mathrm{T}}$, where $y_0 = \mathcal{Z}(\alpha_0), \ldots, y_{n-1} = \mathcal{Z}(\alpha_{n-1}))$ and $\alpha_k = \omega^k e^{2\pi i k/n}$, it follows that the coefficients $c_k = \mathbf{Z}_{1,n}^k$ in the polynomial $\mathcal{Z}(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}$ defined in (1.12) can be computed, at least in principle, by using the FFT. It turns out, however, that the values of $\mathbf{Z}_{1,n}^k$ are so astronomically large, that the ensuing numerical instability makes even this approach infeasible for values of n that exceed 56 (data not shown). Nevertheless, our approach can be modified as follows. Define $\mathbf{Y}$ by $\mathbf{Y} = (y_1, \ldots, y_n)^{\mathrm{T}}$, where $y_1 = \frac{\mathcal{Z}(\alpha_1)}{\mathcal{Z}(x)}, \ldots, y_n = \frac{\mathcal{Z}(\alpha_n)}{\mathcal{Z}(x)}$, and $\mathbf{Z}$ is the partition function defined in (1.6). Using the FFT to compute the inverse DFT, it follows from (1.27) that we can compute the probabilities $p_0, \ldots, p_{n-1}$ that are coefficients of the polynomial $p(x) = p_0 + p_1 x + \cdots + p_{n-1} x^{n-1}$ defined in equation (1.1). For genomics applications, we are only interested in the m most significant digits of each p_k, as described in the pseudocode that follows.

Algorithm 1 Pseudocode for `FFTbor`

PURPOSE: Computes the m most significant digits of probabilities $p_k = \mathbf{Z}_{1,n}^k/\mathbf{Z}$
INPUT: RNA sequence $\mathbf{s} = s_1, \ldots, s_n$, secondary structure $\mathcal{S}^*$ of $\mathbf{s}$, integer m
OUTPUT: Probabilities $p_k = \mathbf{Z}_{1,n}^k/\mathbf{Z}$ to m significant digits for $k = 0, \ldots, n-1$

```
1: procedure FFTBOR
2:     stringlen ← length of string
3:     i ← patlen
4:     if i > stringlen then return false
5:     end if
6:     j ← patlen
7:     if string(i) = path(j) then
8:         j ← j − 1.
9:         i ← i − 1.
10:        goto loop.
11:        close;
12:    end if
13:    i ← i + max(delta1(string(i)), delta2(j)).
14:    goto top.
15: end procedure
```

1.5 Benchmarking and performance considerations

In this subsection, we show that we need only evaluate the polynomial $\mathcal{Z}(x)$, as defined in equation (**??**), for $n/2$ of the complex nth roots of unity. It is first necessary to recall the definition of complex conjugate. Recall that the complex conjugate of z is denoted by $\overline{z}$; i.e. if $z = a + bi$ where $a, b \in \mathbb{R}$ are real numbers and $i = \sqrt{-1}$, then $\overline{z} = a - bi$.

LEMMA 1: If $\mathcal{Z}(x)$ is the complex polynomial defined in equation (**??**), then for any complex nth root of unity α, it is the case that $\mathcal{Z}(\overline{\alpha}) = \overline{\mathcal{Z}(\alpha)}$. In other words, if α is a complex nth root of unity of the form $a + bi$, where $a, b \in \mathbb{R}$ and $b > 0$, and if $\mathcal{Z}(a + bi) = A + Bi$ where $A, B \in \mathbb{R}$, then it is the case that

$$\mathcal{Z}(a - bi) = A - Bi. \tag{1.28}$$

PROOF: Letting $i = \sqrt{-1}$, if $\theta = \frac{2\pi}{n}$, then $\omega = e^{i\theta} = \cos(\theta) + i\sin(\theta)$ is the principal nth complex root of unity, and $1 = \omega^0, \ldots, e^{(n-1)\cdot i\theta} = \omega^{n-1}$ together constitute the complete collection of all nth complex roots of unity – i.e. the n unique solutions of of the equation $x^n - 1 = 0$ over the field $\mathbb{C}$ of complex numbers. Clearly, for any $1 \leq r < n$, $e^{-ir\theta} = 1 \cdot e^{-ir\theta} = e^{2\pi i} \cdot e^{-ir\theta} = e^{i(2\pi - r\theta)} = e^{i(n\theta - r\theta)} = e^{i\theta(n-r)}$. Moreover, if $\omega^r = e^{ir\theta} = a + bi$ where $b > 0$, then we have $e^{-ir\theta} = a - bi$. It follows that for any nth root of unity of the form $a + bi$, where $b > 0$, the number $a - bi$ is also an nth root of unity.

Recall that $\mathcal{Z}(x) = \sum_{k=0}^{n} c_k x^k$, where $c_k \in \mathbb{R}$ are real numbers representing the partition function $\mathbf{Z}_{1,n}^k$ over all secondary structures of a given RNA sequence $s_1, \ldots, s_n$, whose base pair distance from initial structure $\mathcal{S}^*$ is k. Thus, in order to prove the lemma, it suffices to show that for all values $k = 0, \ldots, n-1$, if $a + bi$ is a complex nth root of unity, where $a, b \in \mathbb{R}$ and $b > 0$, and if $(a + bi)^k = C + Di$ where $C, D \in \mathbb{R}$, *then* $(a - bi)^k = C - Di$. Indeed, we have the following.

$$(a + bi)^m = \sum_{k=0}^{m} \binom{m}{k} a^{m-k} \cdot (bi)^k \tag{1.29}$$

$$(bi)^k = \begin{cases} b^k & \text{if } k \equiv 0 \bmod 4 \\ ib^k & \text{if } k \equiv 1 \bmod 4 \\ -b^k & \text{if } k \equiv 2 \bmod 4 \\ -ib^k & \text{if } k \equiv 3 \bmod 4 \end{cases} \tag{1.30}$$

$$(a - bi)^m = \sum_{k=0}^{m} \binom{m}{k} a^{m-k} \cdot (-bi)^k \tag{1.31}$$

$$(-bi)^k = \begin{cases} b^k & \text{if } k \equiv 0 \bmod 4 \\ -ib^k & \text{if } k \equiv 1 \bmod 4 \\ -b^k & \text{if } k \equiv 2 \bmod 4 \\ ib^k & \text{if } k \equiv 3 \bmod 4 \end{cases} \tag{1.32}$$

It follows that each term of the form $a^{m-k} \cdot (bi)^k$, for $k = 0, \ldots, m$, is the complex conjugate of $a^{m-k} \cdot (-bi)^k$, and thus $(a + bi)^m$ is the complex conjugate of $(a - bi)^m$. Since $\mathcal{Z}(a + bi)$ is a sum of terms of the form $c_k(a + bi)^k$, it follows that $\mathcal{Z}(a - bi)$ is the complex conjugate of $\mathcal{Z}(a + bi)$. This completes the proof of the lemma. □

Lemma 1 immediately entails that we need only evaluate $\mathcal{Z}(x)$ on $n/2$ many of the complex nth roots of unity – namely, those of the form $a + bi$, where $b \geq 0$. The remaining values of $\mathcal{Z}(x)$ are obtained by taking conplex conjugates of the first $n/2$ values. This, along with a precomputation of powers of the complex nth roots of unity, leads to an enormous performance speed-up in our implementation of `FFTbor`.

1.6 Coarse-grained kinetics with `FFTbor`

1.7 Riboswitch detection with `FFTbor`