Journal of Mathematical Biology manuscript No.
(will be inserted by the editor)

RNA folding pathways and kinetics using 2D energy landscapes

Evan Senter · Ivan Dotu · Peter Clote

Received: date / Accepted: date

Abstract RNA folding pathways play an important role in various biological processes, such as *(i)* the *hok/sok* (host-killing/suppression of killing) system in *E. coli* to check for sufficient plasmid copy number, *(ii)* the conformational switch in spliced leader (SL) RNA from *Leptomonas collosoma*, which controls *trans* splicing of a portion of the 5′ exon, and *(iii)* riboswitches – portions of the 5′ untranslated region of messenger RNA that regulate genes by allostery.

Since RNA folding pathways are determined by the energy landscape, we describe a novel algorithm, `FFTbor2D`, which computes the 2D projection of the energy landscape for a given RNA sequence. Given two metastable secondary structures A, B for a given RNA sequence, `FFTbor2D` computes the Boltzmann probability $p(x, y) = Z_{x,y}/Z$ that a secondary structure has base pair distance x from A and distance y from B. Using polynomial interpolation with the fast Fourier transform, we compute $p(x, y)$ in $O(n^5)$ time and $O(n^2)$ space, which is an improvement over an earlier method, which runs in $O(n^7)$ time and $O(n^4)$ space.

`FFTbor2D` has potential applications in synthetic biology, where one might wish to design bistable switches having target metastable structures A, B with favorable pathway kinetics. By inverting the transition probability matrix determined from `FFTbor2D` output, we show that *L. collosoma* spliced leader RNA has larger mean first passage time from A to B on the 2D energy landscape, than 97.145% of 20,000 sequences, each having metastable structures A, B.

Source code and binaries are freely available for download at `http://bioinformatics.bc.edu/clotelab/FFTbor2D`. The program `FFTbor2D` is implemented in C++, with optional OpenMP parallelization primitives.

Keywords RNA · Boltzmann partition function · FFT · polynomial interpolation · dynamic programming · kinetics

PACS 87.14.gn · 05.10.-a · 82.39.-k · 82.39.Pj

Research supported by National Science Foundation grants DMS-1016618 and DBI-1262439.

E. Senter, I. Dotu and P. Clote
Department of Biology, Boston College, Chestnut Hill, MA 02467, USA. Tel.: +1-617-552-1332.
Fax: +1-617-552-2011. E-mail: clote@bc.edu.

Mathematics Subject Classification (2000) 92B99 · 60J22 · 68N19 · 74P05 · 81V55

1 Introduction

RNA folding pathways play an important role in biological processes. For instance, in the *hok/sok* (host-killing/suppression of killing) system [11], the transition between two metastable RNA structures determines the fate of a cell as follows. The *hok* gene of *E. coli* and other bacteria codes a small (52 amino acid) toxin causing irreversible damage to the cell membrane. It has been shown that *hok*-mRNA is constitutively expressed from a weak promoter, while the rapidly degraded *sok*-RNA is constitutively expressed from a strong promoter. The *hok*-mRNA is initially inactive, since a foldback sequesters the Shine-Dalgarno sequence; however, slow exonucleolytic processing digests the last ≈ 40 nt of the $3'$ end of *hok*-mRNA, transforming the molecule into its active form in which the Shine-Dalgarno sequence is no longer sequestered. If R1 plasmids of *E. coli* are present in sufficient copy number, then a portion of the 64 nt *sok*-RNA, which is complementary to *hok*-mRNA leader region, binds to the active conformation of *hok*-mRNA, thus causing degradation of the complex by RNase III [11]. If plasmids are not present in sufficient copy number, then the cell is killed by *hok* toxin, thus ensuring fitness of the daughter cells.

In the case of spliced leader (SL) RNA from certain trypanosomes and nematodes, a portion of the $5'$ exon is donated to another mRNA by trans splicing. Intermediate structures appear to be important in the process of splicing, as shown by LeCuyer and Crothers [12], who performed stopped-flow rapid-mixing and temperature-jump measurements of the kinetics for the structural transition between two low energy structures of SL RNA from *Leptomonas collosoma*. Conformational switches are thought not only to play a role in such trans splicing, but as well in transcriptional and translational regulation, protein synthesis, and mRNA splicing.

For these reasons, substantial experimental and computational work has been done on folding pathways and the kinetics of RNA folding; below, we cite only a small sample of the work in this area. On the experimental side, Neupane et al. [19] applied single-molecule force spectroscopy of the add adenine riboswitch, to show how folding relates to gene regulation; see also [25,2,17]. On the computational side, Morgan and Higgs [18] appear to have been among the first to have considered the computational problem of determining optimal and near-optimal *folding pathways* between two metastable secondary structures A, B of a given RNA sequence. In [5], Flamm developed an event-driven Monte Carlo simulation, `Kinfold`, to determine expected folding time between two reference structures, including the mean first passage time to fold into the minimum free energy structure. Using `RNAsubopt` [26], Flamm et al. [6] designed the exponential time exact algorithm, `barriers`, that computes the optimal folding pathway between metastable structures A and B. Since `barriers` may not converge, due to its reliance on the enumeration of possibly exponentially many structures, in [4] we developed a local search (tabu) method that rapidly returns near-optimal folding pathways; in benchmarking results [4], local search was shown to outperform various methods, including direct and indirect path methods of Morgan and Higgs [18], breadth-first search with bounded lookahead, `Findpath` [7], and the exponential time, exact method `barriers` [6], in producing near-optimal pathways within reasonable time. There are far too many contributions to kinetics and folding pathways to adequately survey here; the references [27,22,10,29,13] give some idea of the variety of methods.

1.1 Preliminaries

A secondary structure for a given RNA nucleotide sequence $\mathbf{s} = s_1, \ldots, s_n$ is a set S of Watson-Crick or wobble base pairs (i, j), containing neither base triples nor pseudoknots. The number of base pairs in S is denoted by $|S|$. The secondary structure S is *compatible* with $\mathbf{s}$ if for every base pair (i, j) in S, the pair (s_i, s_j) is contained in the set $\mathbb{B}$ of six canonical (Watson-Crick and wobble) base pairs. Throughout this paper, by *structure*, we always mean a secondary structure which is compatible with an arbitrary, but fixed RNA sequence $\mathbf{s}$.

If A, B are secondary structures of $\mathbf{s}$, then the *base pair distance*, $d_{BP}(A, B)$, is defined to be the $|A - B| + |B - A|$, i.e. the number of base pairs belonging to one structure but not the other. Structures S, T are said to be *k-neighbors* if $d_{BP}(S, T) = k$. In [8], we described recursions for an $O(n^5)$ time and $O(n^3)$ algorithm, `RNAbor`, that computes the Boltzmann probability p_k of structures to have base pair distance k from a given reference structure A. In [21], we described an $O(n^4)$ time and $O(n^2)$ space algorithm, by using polynomial interpolation with the fast Fourier transform (FFT). In [14], Lorenz et al. generalized the recursions of `RNAbor` [8] to yield recursions for an $O(n^7)$ time and $O(n^4)$ space algorithm, `RNA2Dfold`, that computes the Boltzmann probability $p(x, y)$ that a structure has base pair distance x from reference structure A, and distance y from another reference structure B. The goal of this paper is to describe a new algorithm, `FFTbor2D`, using polynomial interpolation with the FFT, to reduce the worst case complexity of `RNA2Dfold` to $O(n^5)$ time and $O(n^2)$ space. As well, we provide an illustrative application by computing the mean first passage time between metastable structures A, B of spliced leader RNA from *L. collosoma* on the 2D energy landscape computed by `FFTbor2D`.

The general idea of using interpolation to compute partition function values was first suggested by Waldispühl and Ponty in the context of the `RNAmutants` program [23]. Subsequently, we used the Fast Fourier Transform (FFT) in our algorithm `FFTbor` [21] to interpolate the probabilities p_k that structures from the Boltzmann ensemble have base pair distance k from a target structure S^*. This paper extends the result from [21] to two dimensions; i.e. in the algorithm `FFTbor2D`, we interpolate the probabilities $p(x, y)$ that structures from the Boltzmann ensemble have base pair distance x [resp. y] from target structure A [resp. B].

1.2 Plan of paper

The plan for the rest of this paper is as follows. In Section 2, we develop the quintic $O(n^5)$ time and quadratic $O(n^2)$ space algorithm `FFTbor2D`, which uses dynamic programming to evaluate a complex polynomial $\mathcal{Z}(x)$ at quadratically many complex roots of unity, and then use the fast Fourier transform (FFT) to compute the coefficients of $\mathcal{Z}(x)$ by polynomial interpolation. The coefficients of

$\mathcal{Z}(x)$ yield the 2D energy landscape for a given RNA sequence. Moreover, by exploiting parity and complex conjugates, we obtain an additional reduction of time by a factor of 4. Although this section is rather technical, the main result entails a significant improvement in the algorithm run time. In Section 3, we present bencharking results, comparing the run times of `FFTbor2D` and `RNA2Dfold` (developed by Lorenz et al. [14]). In Section 4, we apply `FFTbor2D` to determine the mean first passage time (MFPT) along the 2D energy grid in folding between two metastable structures of *L. collosoma* spliced leader RNA. Finally, in Secton 5, we we describe differences in the algorithms `RNA2Dfold` and and `FFTbor2D`, and mention relative strengths of each software depending on the envisioned application.

2 Polynomial interpolation using the FFT

For expository clarity, we describe `FFTbor2D` and all recursions in terms of the Nussinov energy model [20], where the energy $E_0(i,j)$ of a base pair (i,j) is defined to be -1, and the energy $E(S)$ of a secondary structure S is -1 times the number $|S|$ of base pairs in structure S. Nevertheless, the implementation of `FFTbor2D` involves the full Turner energy model [28], where free energy $E(S)$ depends on negative, stabilizing energy contributions from base stacking, and positive, destabilizing energy contributions due to loss of entropy in loops.

Given reference secondary structures A, B of a given RNA sequence $\mathbf{s} = s_1, \ldots, s_n$, our goal is to compute

$$\mathbf{Z}_{1,n}^{x,y} = \sum_{\substack{S \text{ such that} \\ d_{BP}(S,A)=x, d_{BP}(S,B)=y}} e^{\frac{-E(S)}{RT}} \tag{1}$$

for all $0 \leq x, y < n$, where R is the universal gas constant, T absolute temperature, $E(S)$ denotes the free energy of S, and S ranges over all secondary structures that are compatible with $\mathbf{s}$. As mentioned, we emphasize that for expository reasons alone, the Nussinov energy model is used in the recursions in this paper, although full recursions and the implementation of `FFTbor2D` involve the Turner energy model.

For any secondary structure S of $\mathbf{s}$, and any values $1 \leq i \leq j \leq n$, the restriction $S_{[i,j]}$ is defined to be the collection of base pairs of S, lying within interval $[i,j]$; i.e. $S_{[i,j]} = \{(k,\ell) : i \leq k < \ell \leq j\}$. In [14], Lorenz et al. generalized the dynamic programming recursions of our earlier work [8], to yield recursions for the partition function $\mathbf{Z}_{i,j}^{x,y}$ in equation (1). In the context of the Nussinov model, $\mathbf{Z}_{i,j}^{x,y}$ is equal to

$$\begin{gathered} \mathbf{Z}_{i,j-1}^{(x-\alpha_0),(y-\beta_0)} + \\ \textstyle\sum_{\substack{s_k s_j \in \mathbb{B}, \\ i \leq k < j}} \left(e^{\frac{-E_0(k,j)}{RT}} \sum_{u+u'=x-\alpha(k)} \sum_{v+v'=y-\beta(k)} \mathbf{Z}_{i,k-1}^{u,v} \mathbf{Z}_{k+1,j-1}^{u',v'} \right) \end{gathered} \tag{2}$$

where $\alpha_0 = 1$ if j is base paired in $A_{[i,j]}$ and 0 otherwise, $\beta_0 = 1$ if j is base paired in $B_{[i,j]}$ and 0 otherwise, $E_0(k,j) = -1$ if k, j can base-pair, and otherwise $E_0(k,j) = +\infty$, and $\alpha(k) = d_{BP}(A_{[i,j]}, A_{[i,k-1]} \cup A_{[k+1,j-1]} \cup \{(k,j)\})$, and $\beta(k) = d_{BP}(B_{[i,j]}, B_{[i,k-1]} \cup B_{[k+1,j-1]} \cup \{(k,j)\})$.

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

Given RNA sequence $\mathbf{s} = s_1, \ldots, s_n$ and two arbitrary, but fixed reference structures A, B, we define the *polynomial*

$$\mathcal{Z}(x) = \sum_{r=0}^{n-1} \sum_{s=0}^{n-1} z_{rn+s} x^{r \cdot n+s} \tag{3}$$

where (constant) coefficients

$$z_{rn+s} = \mathbf{Z}_{1,n}^{r,s} = \sum_{\substack{S \text{ such that} \\ d_{BP}(S,A)=r, d_{BP}(S,B)=s}} e^{\frac{-E(S)}{RT}}$$

where $E(S)$ denotes the free energy of S. If we evaluate the polynomial $\mathcal{Z}(x)$ at n^2 distinct pairs of values $a_0, \ldots, a_{n^2-1}$ in

$$\mathcal{Z}(a_0) = z_0, \ldots, \mathcal{Z}(a_{n^2-1}) = z_{n^2-1}, \tag{4}$$

then Lagrange polynomial interpolation guarantees that we can determine the coefficients c_{rn+s} of $\mathcal{Z}(x)$, for $0 \leq r, s < n$. Due to technical difficulties concerning numerical robustness, we will perform polynomial interpolation by using Vandermonde matrices and the fast Fourier transform (FFT).

The following theorem shows that a recursion, analogous to equation (2), can be used to compute the *polynomial* $\mathcal{Z}_{i,j}(x)$ defined by

$$\mathcal{Z}_{i,j}(x) = \sum_{r=0}^{n-1} \sum_{s=0}^{n-1} z_{rn+s}(i,j) x^{rn+s} = \sum_{k=0}^{n^2-1} z_k(i,j) x^k \tag{5}$$

where

$$z_{rn+s}(i,j) = \mathbf{Z}_{i,j}^{r,s} = \sum_{\substack{S \text{ such that} \\ d_{BP}(S,A_{[i,j]})=r, d_{BP}(S,B_{[i,j]})=s}} e^{\frac{-E(S)}{RT}}.$$

Here, in the summation, S runs over structures on $s_i, \ldots, s_j$, which are r-neighbors of the restriction $A_{[i,j]}$ of reference structure A to interval $[i,j]$, and simultaneously s-neighbors of the restriction $B_{[i,j]}$ of reference structure B to interval $[i,j]$.

Theorem 1: Let $s_1, \ldots, s_n$ be a given RNA sequence. For any integers $1 \leq i < j \leq n$, let

$$\mathcal{Z}_{i,j}(x) = \sum_{r=0}^{n-1} \sum_{s=0}^{n-1} z_{rn+s} x^{rn+s}$$

where

$$z_{rn+s}(i,j) = \mathbf{Z}_{i,j}^{r,s}.$$

Inductively we define $\mathcal{Z}_{i,j}(x)$ to equal

$$\mathcal{Z}_{i,j-1}(x) \cdot x^{\alpha_0 n+\beta_0} + \sum_{\substack{s_k s_j \in \mathbb{B}, \\ i \leq k < j}} \left(e^{\frac{-E_0(k,j)}{RT}} \cdot \mathcal{Z}_{i,k-1}(x) \cdot \mathcal{Z}_{k+1,j-1}(x) \cdot x^{\alpha(k)n+\beta(k)} \right) \tag{6}$$

where $\alpha_0 = 1$ if j is base-paired in $A_{[i,j]}$ and 0 otherwise, $\beta_0 = 1$ if j is base-paired in $B_{[i,j]}$ and 0 otherwise, and $\alpha(k) = d_{BP}(A_{[i,j]}, A_{[i,k-1]} \cup A_{[k+1,j-1]} \cup \{(k,j)\})$, $\beta(k) = d_{BP}(B_{[i,j]}, B_{[i,k-1]} \cup B_{[k+1,j-1]} \cup \{(k,j)\})$. The proof is given in supplementary information.

Note 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 $O(n^7)$ time and $O(n^4)$ space, the same time complexity of [14]. Instead of explicitly performing polynomial expansion in *variable* x, we instantiate x to a complex number $\rho \in \mathbb{C}$, and apply the following recursion, by setting $\mathcal{Z}_{i,j}(\rho)$ equal to

$$\mathcal{Z}_{i,j-1}(\rho) \cdot \rho^{\alpha_0 n + \beta_0} + \sum_{\substack{(s_k,s_j)\in\mathbb{B},\\ i\le k<j}} \left(e^{\frac{-E_0(k,j)}{RT}} \cdot \mathcal{Z}_{i,k-1}(\rho) \cdot \mathcal{Z}_{k+1,j-1}(\rho) \cdot \rho^{\alpha(k)n+\beta(k)} \right) \tag{7}$$

In this fashion, we can compute $\mathcal{Z}(\rho) = \mathcal{Z}_{1,n}(\rho)$ in $O(n^3)$ time and $O(n^2)$ space. For n^2 distinct complex numbers ρ_i where $0 \le i \le n^2 - 1$, we can compute and save only the values $\mathcal{Z}(\rho_0), \ldots, \mathcal{Z}(\rho_{n^2-1})$, each time re-using the $O(n^2)$ space for the next computation of $\mathcal{Z}(\rho_i)$. It follows that the computation resources used to determine the (column) vector

$$\mathbf{Y} = (y_0, \ldots, y_{n^2-1})^T = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n^2-1} \end{pmatrix} \tag{8}$$

where $y_0 = \mathcal{Z}(\alpha_0), \ldots, y_{n^2-1} = \mathcal{Z}(\alpha_{n^2-1})$ are thus quintic time $O(n^5)$ and quadratic space $O(n^2)$.

2.2 Polynomial interpolation

Our plan is to determine the coefficients of the polynomial $\mathcal{Z}(x)$ in equation (3) by polynomial interpolation. For reasons of numerical stability, we instead determine the coefficients of the polynomial $\mathbf{p}(x)$, defined by

$$\mathbf{p}(x) = \sum_{r=0}^{n-1}\sum_{s=0}^{n-1} \mathbf{p}_{rn+s} x^{r\cdot n+s} = \sum_{r=0}^{n-1}\sum_{s=0}^{n-1} \frac{z_{rn+s}}{Z} x^{r\cdot n+s}, \tag{9}$$

where the fast Fourier transform (FFT) is used to implement the interpolation of the coefficients using the inverse discrete Fourier transform (DFT), as described in Section 2.6. The following pseudocode describes how to compute the m most significant digits for probabilities $p_{rn+s} = \frac{\mathbf{Z}_{1,n}^{r,s}}{\mathbf{Z}}$. It is well-known that the FFT requires $O(N \log N)$ time to solve the inverse discrete Fourier transform for a polynomial of degree N. In our case, $N = n^2$, and so line 6 involving the FFT requires time $O(n^2 \log n)$.

The pseudocode for the algorithm to compute $\mathbf{p}(x)$ is given in Figure 1. In the next section, we explain a highly non-trivial improvement of this algorithm to reduce time by a factor of 4.

ALGORITHM for FFTbor2D
INPUT: RNA sequence $\mathbf{s} = s_1, \ldots, s_n$, and distinct secondary structures A, B of $\mathbf{s}$, and integer m.
OUTPUT: Probabilities $p_{rn+s} = p(r, s) = \mathbf{Z}^{r,s}_{1,n}/\mathbf{Z}$ to m significant digits for $x, y = 0, \ldots, n-1$.
Let i denote $\sqrt{-1}$, $\alpha = \exp(\frac{2\pi i}{n^2})$ and $\alpha^k = \exp(\frac{2\pi ik}{n^2})$.

```
for $k = 0, \ldots, n^2 - 1$
  compute the $k$th roots of unity $\alpha^k$
for $k = 0, \ldots, n^2 - 1$
  compute $y_k = \mathcal{Z}(\alpha^k)$
  $y_k = 10^m \cdot \frac{y_k}{\mathbf{Z}}$ //normalize $y_k$
compute $p_0, \ldots, p_{n^2-1}$ by FFT
for $k = 0$ to $n^2 - 1$
  $p_k = \lfloor 10^m \cdot p_k \rfloor \cdot \frac{1}{10^m}$
//truncate to $m$ most significant digits
```

Fig. 1 Pseudocode to compute the m most significant digits for probabilities $p_{rn+s} = \frac{\mathbf{Z}^{r,s}_{1,n}}{\mathbf{Z}}$. In our implementation, due to numerical stability issues in the FFT engine, precision parameter m has an upper bound of 8 – only the $m = 8$ most significant digits are computed with FFTbor2D. (Note that the software actually uses base 2 precision parameter, with maximum of 27, where $2^{27} \approx 10^8$.) It is well-known that the FFT requires $O(N \log N)$ time to solve the inverse discrete Fourier transform for a polynomial of degree N. In our case, $N = n^2$, and so the FFT requires time $O(n^2 \log n)$.

2.3 Speed-up by factor of 4

Recall that if $a + bi$ is a complex number, where a, b are real values and i denotes $\sqrt{-1}$, then the complex conjugate of $a + bi$, denoted by $\overline{a + bi}$ is defined to be $a - bi$. Recall that a complex nth root of unity is a number whose nth power equals one. Moreover, $e^{2\pi i/n}$ is the *principal* complex nth root of unity; i.e. $\{e^{2\pi ik/n} : k = 0, \ldots, n-1\}$ is a set of pairwise distinct nth roots of unity. For notational reasons below, we will write 'n-root of unity' instead of 'nth root of unity'. We have the following.

LEMMA 1: Let A, B denote two distinct, arbitrary but fixed, secondary structures of RNA sequence $\mathbf{s}$, let S range over all secondary structures of $\mathbf{s}$, and let d_0 denote $d_{BP}(A, B)$. If $x = d_{BP}(A, S)$ and $y = d_{BP}(S, B)$, then $y \in \{d_0 - x + 2k : k = 0, \ldots, x\}$.

It follows that if $x = d_{BP}(A, S)$ and $y = d_{BP}(S, B)$, then the only possible values for (x, y) are $(0, d_0), (1, d_0 - 1), (1, d_0 + 1), (2, d_0 - 2), (2, d_0), (2, d_0 + 2), (3, d_0 - 3), (3, d_0 - 1), (3, d_0 + 1), (3, d_0 + 3), \cdots$. As a corollary, we have the *parity condition*, that

$$d_{BP}(A, S) + d_{BP}(S, B) \equiv d_{BP}(A, B) \bmod 2 \tag{10}$$

first noticed in [14], as well as the triangle inequality $d_{BP}(A, S) + d_{BP}(S, B) \geq d_{BP}(A, B)$ for base pair distance, probably folklore. Lorenz et al. [14] exploited the parity condition and the triangle inequality by using sparse matrix methods to improve on the efficiency of the naive implementation of the $O(n^7)$ time and $O(n^4)$ space algorithm to compute the partition function, $\mathbf{Z}^{r,s}_{1,n}$, and minimum free energy structure, $MFE^{r,s}_{1,n}$, over all structures having base pair distance r to A and s to B. The following lemma is not difficult to establish.

Lemma 2: If $\mathcal{Z}(x)$ is the complex polynomial defined in equation (3), then for any complex nth root of unity α, it is the case that $\mathcal{Z}(\overline{\alpha}) = \overline{\mathcal{Z}(\alpha)}$.

Lemma 3: Let $\mathcal{Z}(x)$ be defined by equation (3), and let $\alpha \in \mathbb{C}$ be any complex number. If the base pair distance between reference structures A, B is even, then $Z(-\alpha) = Z(\alpha)$, while if the distance is odd, then $Z(-\alpha) = -Z(\alpha)$.

Lemma 4: Suppose that M is evenly divisible by 4, $\nu = \exp(\frac{2\pi i}{M})$ is the principal M-root of unity, and $\frac{M}{4} < k \leq \frac{M}{2}$. Then

$$\nu^k = -(\nu^{-(M/2-k)}) = -\overline{\nu^{M/2-k}}.$$

Lemma 1 is proved by simple induction; Lemma 2 is proved by a computation involving binomial coefficients; Lemma 3 is immediate by the parity observation above, resulting from Lemma 1; Lemma 4 is elementary, relying on Euler's formula and trigonometric addition formulas. Details proofs of Lemmas 2,3,4 can be found in supplementary information.

Lemma 1 entails that either all even coefficients, or all odd coefficients of $\mathcal{Z}(x)$ are zero, and so by a variable change described in detail below, we require only half the number of evaluations of $\mathcal{Z}(x)$, in order to perform polynomial interpolation. Lemma 2 entails that we require only half again the number of evaluations of $\mathcal{Z}(x)$, since the remainder can be inferred by taking the complex conjugate. Lemma 1 and Lemma 2, along with a precomputation of powers of the complex roots of unity, lead to a large performance speed-up in our implementation of `FFTbor2D` – indeed, by a factor of 4 or more. Though the intuitive idea of how to obtain this speedup by a factor of four may be apparent, the technical details leading to the pseudocode of `FFTbor2D`, presented in Figure 2, are rather tricky. These details are presented in the next two subsections, which can be skipped by the reader wishing to move on to the algorithm itself.

2.4 Time reduction due to Lemma 1

Let n denote the length of RNA sequence $\mathbf{s}$, and let N denote the least *even* integer greater than or equal to n. Since N is even, we have $(r+s) \equiv (r \cdot (N+1)+s) \bmod 2$. For distinct fixed structures A, B, let $\pi_1(k) = \lfloor \frac{k}{N+1} \rfloor$, and $\pi_2(k) = k \bmod (N+1)$, and define the polynomial

$$\begin{aligned}\mathcal{Z}(x) &= \sum_{r=0}^{N} \sum_{s=0}^{N} z_{rN+s} x^{r \cdot N+s} \\ &= \sum_{k=0}^{(N+1)^2-1} z_{\pi_1(k)\cdot(N+1)+\pi_2(k)} x^{\pi_1(k)\cdot(N+1)+\pi_2(k)} \\ &= \sum_{k=0}^{(N+1)^2-1} z_k x^k\end{aligned}$$

where for the last equality, we have used the fact that $k = \pi_1(k) \cdot (N+1) + \pi_2(k)$, well-known from row major order of a 2-dimensional array.

Consider the coefficients of the polynomial

$$\mathcal{Z}(x) = \sum_{r=0}^{N}\sum_{s=0}^{N} z_{rN+s}x^{rN+s} = \sum_{k=0}^{(N+1)^2-1} z_k x^k. \tag{11}$$

Since N is even, the parity of $r+s$ equals the parity of $r(N+1)+s$, hence it follows from the parity condition that either *(i)* all coefficients $z_1, z_3, z_5, \ldots$ of odd parity are zero, or *(ii)* all coefficients $z_0, z_2, z_4, \ldots$ of even parity are zero. To simplify notation, in the remainder of this subsection, let M be the least integer greater than or equal to $(N+1)^2$ that is evenly divisible by 4, and let $M_0 = M/2$. We will assume that $\mathcal{Z}(x) = \sum_{k=0}^{M-1} z_k x^k$, whereupon coefficients $z_k = 0$ for $k > (N+1)^2$.

Case 1: All coefficients z_k of odd parity in equation (11) are zero.

In this case, we have $\mathcal{Z}(x) = \sum_{k=0}^{\frac{M}{2}-1} z_{2k}x^{2k}$. But then $\mathcal{Z}(x) = Y(u) = \sum_{k=0}^{M_0-1} b_k u^k$, where we have made a variable change $u = x^2$, and coefficient changes $b_k = a_{2k}$. By evaluating $M_0 = \frac{M}{2}$ many complex M_0-roots of unity, we can use polynomial interpolation to determine all coefficients b_k of the polynomial

$$Y(u) = \sum_{k=0}^{M_0-1} b_k u^k = \sum_{k=0}^{M_0-1} z_{2k}x^{2k}.$$

Since $Y(x^2) = Z(x)$, we have $Y(\exp(\frac{2\pi ki}{M/2})) = Y(\exp(\frac{4\pi ki}{M})) = Z(\exp(\frac{2\pi ki}{M}))$, hence we use the previous recursions (6) to evaluate $Z(\exp(\frac{2\pi ki}{M})$. Instead of performing M evaluations of $Z(x)$ at M-roots of unity, this requires only $M_0 = M/2$ evaluations of $Y(u)$ at M_0-roots of unity; i.e. only half the number of evaluations of $Z(x)$ are necessary to obtain the coefficients of $Y(x)$. But then, we immediately obtain the full polynomial $\mathcal{Z}(x)$, since its coefficients of odd parity are zero.

Case 2: All coefficients z_k of even parity in equation (11) are zero.

In this case, $z_0, z_2, z_4, \cdots$ are zero, so $\mathcal{Z}(x) = \sum_{k=0}^{M/2-1} z_{2k+1}x^{2k+1}$. But then $\mathcal{Z}(x) = x \cdot Y(u)$, where $Y(u) = \sum_{k=0}^{M_0-1} b_k u^k$, where we have made a variable change $u = x^2$, and coefficient changes $b_k = z_{2k+1}$. Similarly to Case 1, we can interpolate the M_0 coefficients of the polynomial $Y(u) = \sum_{k=0}^{M_0-1} b_k u^k$ by evaluating M_0 many complex M_0-roots of unity. Since $\mathcal{Z}(x) = x \cdot Y(x^2)$, $Y(x^2) = x^{-1} \cdot \mathcal{Z}(x)$, so $Y(\exp(\frac{2\pi ki}{M/2})) = Y(\exp(\frac{4\pi ki}{M})) = \exp(\frac{-2\pi ki}{M}) \cdot \mathcal{Z}(\exp(\frac{2\pi ki}{M}))$, employing the previous recursions (6) to evaluate $\mathcal{Z}(\exp(\frac{2\pi ki}{M})$. Note, that unlike the Case 1, since $Z(x) = x \cdot Y(x^2)$, we have $Y(x^2) = \frac{\mathcal{Z}(x)}{x}$, which explains the presence of additional factor $\exp(\frac{-2\pi ki}{M}))$ in Case 2. Thus, instead of performing M evaluations of $\mathcal{Z}(x)$ at M-roots of unity, we perform only $M_0 = \frac{M}{2}$ evaluations of $Y(u)$ at M_0-roots of unity; i.e. only half the number of evaluations of $\mathcal{Z}(x)$ are necessary to obtain the coefficients of $Y(x)$. But then, we immediately obtain the full polynomial $\mathcal{Z}(x)$, since $\mathcal{Z}(x) = x \cdot Y(x^2)$, and the coefficients of $\mathcal{Z}(x)$ of even parity are zero.

In the following, we will need the observation, that if the parity of base pair distance $d_{BP}(A, B)$ between A, B is even, then

$$Y(x^2) = \mathcal{Z}(x) \tag{12}$$

while if the parity is odd, then

$$Y(x^2) = \frac{1}{x} \cdot \mathcal{Z}(x). \tag{13}$$

2.5 Time reduction due to Lemma 2

As before, let M be the the least number evenly divisible by 4, which is greater than or equal to $(N+1)^2$, let $\nu = \exp(\frac{2\pi i}{M})$ and $\omega = \nu^2 = \exp(\frac{2\pi i}{M})^2 = \exp(\frac{2\pi i}{M/2})$. Clearly, ν is a principal complex M-root of unity, while ω is a principal complex $\frac{M}{2}$-root of unity. Evaluate $Z(\alpha)$ for each $\frac{M}{2}$-root of unity that belongs to the first quadrant, and apply Lemma 2 to infer the values of $Z(\alpha)$ for each $\frac{M}{2}$-root of unity that belongs to the fourth quadrant. More precisely, we compute $Z(\nu^k)$, for $k = 0, \ldots, \frac{M}{4}$, and by Lemmas 2,3,4 infer that for $k = \frac{M}{4} + 1, \ldots, \frac{M}{2} - 1$, we have $Z(\nu^k) = -1^{d_0} \cdot \overline{Z(\nu^{\frac{M}{2}-k})}$, where $d_0 = d_{BP}(A,B)$. This is justified in the following.

By induction on $k = \frac{M}{4} + 1, \ldots, \frac{M}{2} - 1$, we have

$$\begin{aligned} Y(\omega^k) &= Y(\nu^{2k}) \\ &= \begin{cases} Z(\nu^k) & \text{if } d_{BP}(A,B) = 0 \bmod 2 \\ \frac{1}{\nu^k} \cdot Z(\nu^k) & \text{if } d_{BP}(A,B) = 1 \bmod 2 \end{cases} \\ &= \begin{cases} Z(-\overline{\nu^{(\frac{M}{2}-k)}}) & \text{if } d_{BP}(A,B) = 0 \bmod 2 \\ \nu^{-k} \cdot Z(-\overline{\nu^{(\frac{M}{2}-k)}}) & \text{if } d_{BP}(A,B) = 1 \bmod 2 \end{cases} \\ &= \begin{cases} Z(\overline{\nu^{(\frac{M}{2}-k)}}) & \text{if } d_{BP}(A,B) = 0 \bmod 2 \\ \nu^{-k} \cdot -Z(\overline{\nu^{(\frac{M}{2}-k)}}) & \text{if } d_{BP}(A,B) = 1 \bmod 2 \end{cases} \\ &= \begin{cases} \overline{Z(\nu^{(\frac{M}{2}-k)})} & \text{if } d_{BP}(A,B) = 0 \bmod 2 \\ -\nu^{-k} \cdot \overline{Z(\nu^{(\frac{M}{2}-k)})} & \text{if } d_{BP}(A,B) = 1 \bmod 2 \end{cases} \end{aligned}$$

Line 1 follows by definition, since $\omega = \nu^2$; line 2 follows by equations (12) and (13); line 3 follows by Lemma 4; line 4 follows by Lemma 3. Thus if $d_{BP}(A,B)$ is even, then

$$y_k = Y(\omega^k) = \begin{cases} Z(\nu^k) & \text{for } k = 0, \ldots, \frac{M}{4} \\ \overline{Z(\nu^{\frac{M}{2}-k})} & \text{for } k = \frac{M}{4} + 1, \ldots, \frac{M}{2} - 1. \end{cases} \tag{14}$$

while if $d_{BP}(A,B)$ is odd, then

$$y_k = Y(\omega^k) = \begin{cases} \nu^{-k} \cdot Z(\nu^k) & \text{for } k = 0, \ldots, \frac{M}{4} \\ -\nu^{-k} \cdot \overline{Z(\nu^{\frac{M}{2}-k})} & \text{for } k = \frac{M}{4} + 1, \ldots, \frac{M}{2} - 1. \end{cases} \tag{15}$$

It follows that values $y_0, \ldots, y_{M/2-1}$ can be obtained by only $\frac{M}{4}$ evaluations of $\mathcal{Z}(x)$.

Improved Algorithm for FFTbor2D
Input: RNA sequence $\mathbf{s} = s_1, \ldots, s_n$, and distinct secondary structures A, B of $\mathbf{s}$, and integer m.
Output: Probabilities $p(x,y) = \mathbf{Z}^{x,y}_{1,n}/\mathbf{Z}$ to m significant digits for $x, y = 0, \ldots, n-1$. Let N be the least even number greater than or equal to n, M be the least number evenly divisible by 4, which is greater than or equal to $(N+1)^2$, $M_0 = M/2$, $\nu = \exp(\frac{2\pi i}{M})$, $\omega = \nu^2 = \exp(\frac{2\pi i}{M_0})$. For $0 \leq k < M^2$, let $\pi_1(k) = \lfloor \frac{k}{M} \rfloor$, $\pi_2(k) = k - M \cdot \pi_1(k) = k \bmod M$, and note that $k = \pi_1(k) \cdot M + \pi_2(k)$.

```
for $k = 0, \ldots, \frac{M}{2}$
  compute the $M$-roots of unity $\nu^k, \nu^{-k}$
for $k = 0, \ldots, \frac{M}{2} - 1$
  if $d_{BP}(A,B)$ even
    if $k \leq \frac{M}{4}$
      $y_k = Y(\omega^k) = \mathcal{Z}(\nu^k)$ by (14)
    else// $\frac{M}{4} < k < \frac{M}{2}$
      $y_k = Y(\omega^k) = \overline{\mathcal{Z}(\nu^{M/2-k})}$ by (14)
  else // $d_{BP}(A,B)$ is odd
    if $k \leq \frac{M}{4}$
      $y_k = Y(\omega^k) = \nu^{-k} \cdot \mathcal{Z}(\nu^k)$ by (15)
    else// $\frac{M}{4} < k < \frac{M}{2}$
      $y_k = Y(\omega^k) = -1 \cdot \nu^{-k} \cdot \overline{\mathcal{Z}(\nu^{M/2-k})}$ by (15)
//note that $\mathbf{Z} = \sum_{r,s} \mathbf{Z}^{r,s}_{1,n} = y_0 = \mathcal{Z}(\nu^0)$
for $k = 0, \ldots, \frac{M}{2} - 1$
  $y_k = 10^m \cdot \frac{y_k}{\mathbf{Z}}$ //normalize $y_k$
//compute coefficients of $\frac{Y(x)}{\mathbf{Z}}$ using (16)
if $d_{BP}(A,B)$ even then
  for $k = 0$ to $M - 1$
    $r = \pi_1(k)$, $s = \pi_2(k)$
    if $k$ even
      $\frac{\mathbf{Z}^{r,s}_{1,n}}{\mathbf{Z}} = a_{k/2}$ from (16)
    else// $k$ odd
      $\frac{\mathbf{Z}^{r,n}_{1,n}}{\mathbf{Z}} = 0$
else // $d_{BP}(A,B)$ odd
  for $k = 0$ to $M - 1$
    $r = \pi_1(k)$, $s = \pi_2(k)$
    if $k$ even
      $\frac{\mathbf{Z}^{r,n}_{1,n}}{\mathbf{Z}} = 0$
    else// $k$ odd
      $\frac{\mathbf{Z}^{r,n}_{1,n}}{\mathbf{Z}} = a_{(k-1)/2}$ from (16)
 for $k = 0$ to $(N+1)^2$
   $z_k = \lfloor 10^m \cdot z_k \rfloor \cdot \frac{1}{10^m}$
   //truncate to $m$ significant digits
```

Fig. 2 Pseudocode to compute the m most significant digits for probabilities $p_k = \frac{z_k}{\mathbf{Z}} = \frac{\mathbf{Z}^{\pi_1(k),\pi_2(k)}_{1,n}}{\mathbf{Z}}$. Our program, FFTbor2D, supports values of $m = 1, \ldots, 8$ for the precision parameter m. (Note that the software actually uses base 2 precision parameter, with maximum of 27, where $2^{27} \approx 10^8$.)

2.6 Using the fast Fourier transform

Now let $M_0 = \frac{M}{2}$, let $\nu = \exp(\frac{2\pi i}{M})$ be the principal M-root of unity, and $\omega = \nu^2 = \exp(\frac{2\pi i}{M/2}) = \exp(\frac{2\pi \cdot 2i}{M})$ be the principal M_0-root of unity. Recall that the

Vandermonde matrix V_{M_0} is defined to be the $M_0 \times M_0$ matrix, whose i, j entry is $\omega^{i \cdot j} = \nu^{2i \cdot j}$; i.e.

$$V_{M_0} = \begin{pmatrix} 1 & 1 & 1 \cdots & 1 \\ 1 & \omega & \omega^2 \cdots & \omega^{M_0-1} \\ 1 & \omega^2 & \omega^4 \cdots & \omega^{2(M_0-1)} \\ 1 & \omega^3 & \omega^6 \cdots & \omega^{3(M_0-1)} \\ \vdots & \vdots & \vdots \ \vdots & \vdots \\ 1 & \omega^{M_0-1} & \omega^{2(M_0-1)} \cdots & \omega^{(M_0-1)(M_0-1)} \end{pmatrix}$$

The Fast Fourier Transform (FFT) is the $O(n \log n)$ algorithm, which computes the Discrete Fourier Transform (DFT), defined as the matrix product $\mathbf{Y} = V_{M_0}\mathbf{A}$:

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{M_0-1} \end{pmatrix} = V_{M_0} \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{M_0-1} \end{pmatrix}$$

The (i, j) entry of $V_{M_0}^{-1}$ is $\frac{\omega^{-ji}}{M_0}$ and that

$$a_j = \frac{1}{M_0} \sum_{k=0}^{M_0-1} y_k \omega^{-kj} = \frac{1}{M_0} \sum_{k=0}^{M_0-1} y_k \nu^{-2kj} \tag{16}$$

for $j = 0, \ldots, M_0 - 1$ (for more on FFT, see [3]).

Since we defined $\mathbf{Y}$ in (8) by $\mathbf{Y} = (y_0, \ldots, y_{M_0-1})^T$, where $y_0 = \mathcal{Z}(\alpha_0), \ldots, y_{M_0-1} = \mathcal{Z}(\alpha_{M_0-1})$ and $\alpha_k = \omega^k \exp(\frac{k \cdot 2\pi i}{M_0})$, it follows that the coefficients $z_k = \mathbf{Z}_{1,n}^{\pi_1(k),\pi_2(k)}$ in the polynomial $\mathcal{Z}(x) = z_0 + z_1 x + \cdots + z_M x^M$ defined in (3) can be computed, at least in principle, by using the FFT. However, since the values of z_k are astronomically large, numerical instability makes even this approach infeasible for moderate values of n. Nevertheless, we apply this approach to compute the m most significant digits of $\frac{\mathbf{Z}_{1,n}^{\pi_1(k),\pi_2(k)}}{\mathbf{Z}}$, where the partition function $\mathbf{Z} = \sum_S \exp(-E(S)/RT)$ satisfies $\mathbf{Z} = \sum_{x,y} \mathbf{Z}_{1,n}^{x,y}$. This leads to numerical stability, allowing `FFTbor2D` to compute the m most significant digits of $p(x, y) = \frac{\mathbf{Z}_{1,n}^{x,y}}{\mathbf{Z}}$. Pseudocode for the complete algorithm, `FFTbor`, is given in Figure 2.

3 Benchmarking

To perform comparative benchmarking between `RNA2Dfold` and `FFTbor2D`, we took precision parameter $m = 8$, and proceeded as follows. For each sequence length $n = 20, 25, 30, \ldots, 300$, we generated 100 random sequences using probability 0.25 for each nucleotide A,C,G,U. For a given RNA sequence $\mathbf{s}$, the metastable structure A was taken to be the MFE structure of $\mathbf{s}$. Using `RNAbor`, we determined that value $k_0 \geq 10$, for which partition function Z_{k_0} constitutes a visible peak in the graphical output – see Figure 2 and 3 of [8] for an example. Subsequently, metastable structure B was taken to be that structure having minimum free energy over all structures, whose base pair distance from A was k_0.

For all $0 \leq x, y \leq n$, RNA2Dfold and FFTbor2D were benchmarked in the computation of all Boltzmann probabilities $p(x, y) = \frac{Z(x,y)}{Z}$, where x [resp. y] represents base pair distance to metastable structure A [resp. B]. Care was taken for both software to employ the same energy model (Turner99 energy model, no dangles, suppression of minimum free energy structure computations for RNA2Dfold) and the same number of parallel threads (8 threads using OpenMP). Nonetheless, there are slight differences in the energy models – namely, RNA2Dfold includes mismatch penalties for multiloop stems and for exterior loops, while FFTbor2D does not. Even in the computation of the partition function Z, for spliced leader RNA from *L. collosoma* of length 56 nt, RNA2Dfold -d0 obtains a value of -9.660419 kcal/mol, while FFTbor2D obtains -9.660543 kcal/mol; similarly, for attenuator RNA of length 73 nt, RNA2Dfold -d0 obtains a value of -22.171785 kcal/mol, while FFTbor2D obtains -22.173213 kcal/mol. Note that the straightforward calculation of the partition function, following McCaskill's algorithm [15] makes no use of the FFT engine, and thus the differences cannot be due to floating point or precision issues.

For benchmarking purposes, to allow for a fair comparison of FFTbor2D with RNA2Dfold, we restricted the range of x, y in the same manner as done in the source code of RNA2Dfold. In that code, parameters K [resp. L] are defined respectively to be the sum of the number of base pairs in reference structure A [resp. reference structure B] plus the number of base pairs in the maximum matching (Nussinov) structure which contains no base pair of A [resp. B]. For $x \geq K, y \geq L$, both RNA2Dfold and FFTbor2D set $p(x, y) = 0$. For the benchmarking results displayed in Figures 3,4,5, the values x, y are restricted in FFTbor2D to $0 \leq x, y \leq \max(K, L)$, while $0 \leq x \leq K$ and $0 \leq y \leq L$ in RNA2Dfold.

Figure 3 depicts average run time of RNA2Dfold and FFTbor2D as a function of RNA sequence length, for random RNA sequences of lengths $20 - 200$ and their metastable structures A, B, as previously explained. We see that both programs have roughly comparable run times for sequences of length up to approximately 80 nt, while FFTbor2D is demonstrably faster for longer sequences. Figure 4 presents log run time as a function of sequence length, in order to more clearly determine the crossover point in performance. RNA2Dfold is marginally faster for sequences of length up to roughly 80 nt, though the difference is in the millisecond range. Figure 5 shows that the standard deviation of run times on random sequences is tiny for FFTbor2D compared with RNA2Dfold, where standard deviation increases rapidly as a function of sequence length. This figure shows that run time of RNA2Dfold depends on sequence details, as well as sequence length, while the run time of FFTbor2D depends only on sequence length.

4 Kinetics

In this section, we describe folding kinetics along the 2D energy grid, as depicted in Figure 6. Consider the 56 nt *L. collosoma* spliced leader RNA [12], described in the Introduction, having sequence AACUAAAACA AUUUUUGAAG AACAGUUUCU GUACUUCAUU GGUAUGUAGA GACUUC. Let A denote the minimum free energy structure of spliced leader, using Turner 1999 energies as implemented in Vienna RNA Package 1.8.5:

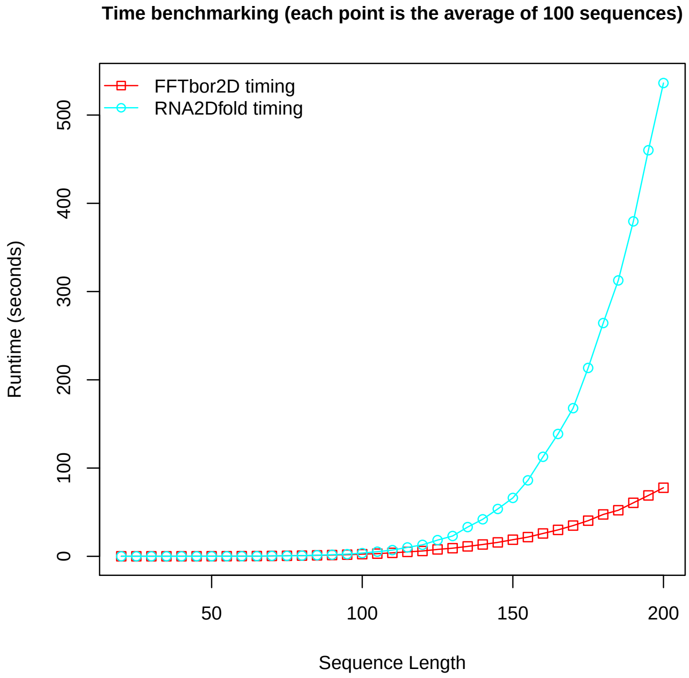

Fig. 3 Run time in seconds for `RNA2Dfold` and `FFTbor2D` on random RNA sequences of length $20-200$ nt, where sequence generation and choice of metastable structures A,B is described in the text. Beyond a length of approximately 80 nt, `FFTbor2D` is demonstrably faster.

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

and let B denote the low energy alternate structure for spliced leader:

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

Using the program *Switch Design* (`switch.pl`) described in [7], we generated 20,000 sequences, for which structures A,B are metastable. For spliced leader RNA as well as for each of these 20,000 sequences, we computed the corresponding probability profile $p(x,y)$ using `FFTbor2D`, and subsequently defined the Markov chain $\mathbb{M}(\mathbf{s}) = (Q,M)$, where $Q = \{(x,y) : 0 \leq x,y \leq n, \text{ and } p(x,y) > 0\}$ is the set of states, and the transition probability matrix $M = (M_{i,j})$ is defined by

$$M_{(x,y),(u,v)} = \begin{cases} \frac{1}{|Q|-1} \cdot \min(1, \frac{p(u,v)}{p(x,y)}) & \text{if } (u,v) \neq (x,y) \\ 1 - \sum\limits_{(u,v)\neq(x,y)} M_{(x,y),(u,v)} & \text{if } (u,v) = (x,y). \end{cases}$$

Let $d_0 = d_{BP}(A,B)$ denote the base pair distance between the metastable structures A,B, and let $M^{-}_{(d_0,0)}$ denote the matrix obtained from M by removing both the row and column corresponding to $(0,d_0)$. For spliced leader and each of the 20,000 sequences obtained from *Switch Design*, we determined the *mean first passage* *time* (MFPT) from state $(0, d_0)$, corresponding to metastable structure A, to state $(d_0, 0)$, corresponding to metastable structure B, by computing $(I - M^{-}_{(d_0,0)})^{-1} \cdot \mathbf{e}$, where I denotes the identity matrix, and $\mathbf{e}$ denotes the column vector composed entirely of ones [16]. Using LAPACK [1] for matrix inversion, we found that *L. collosoma* spliced leader RNA has a MFPT on the 2D energy grid, which is smaller than only 2.855% of the 20,000 sequnces generated by *Switch Design*, thus constituting a Z-score of 1.989 for the kinetics of folding from A to B (see left panel of Figure 7). This result seems to suggest that spliced leader could be under evolutionary pressure for *slow* folding between these metastable structures, if we take MFPT from $(0, d_0)$ to $(d_0, 0)$ as a surrogate for `Kinfold` [6] folding time from A to B – an interpretation which seems to be consistent with the functional role of spliced leader as described in the Introduction. Since accurate `Kinfold` kinetics requires many simulations, each requiring enormous time [24], our method may prove useful in synthetic biology, in prioritizing computationally designed RNA sequences for subsequent experimental validation.

Finally, it should be mentioned that the GC-content of spliced leader RNA is 30.357%, which constitutes a Z-score of +2.50; i.e. the overwhelming majority of

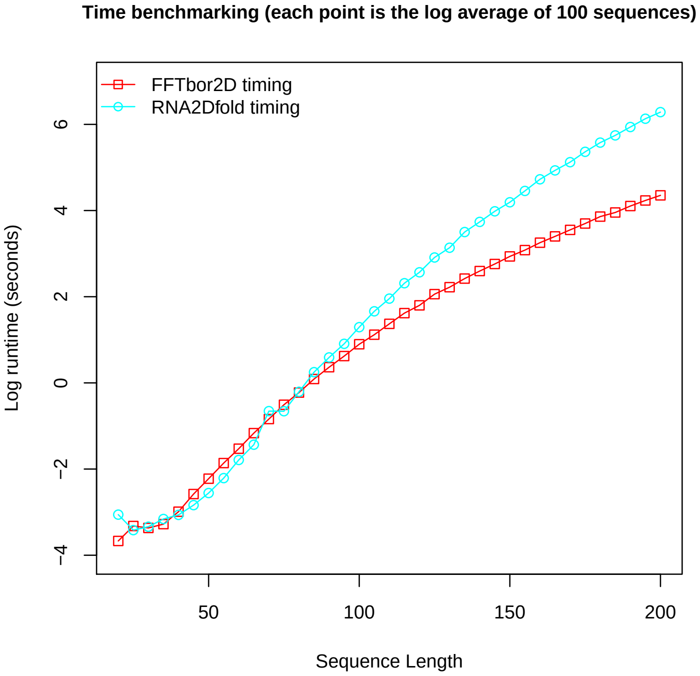

Fig. 4 Logarithm of run time in seconds for `RNA2Dfold` and `FFTbor2D` on random RNA sequences of length less than 200 nt, for same data as that in Figure 3. By taking logarithm of run times, the crossover points are apparent, where `FFTbor2D` is faster than `RNA2Dfold`. For very small sequences, `RNA2Dfold` is faster, though since both programs converge in a fraction of a second, this difference is of no practical consequence.

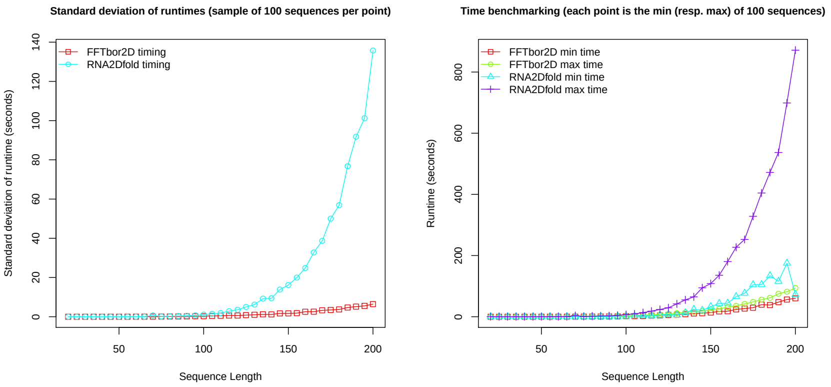

Fig. 5 *(Left)* Standard deviation of run times of `RNA2Dfold` and `FFTbor2D` as a function of sequence length n. *(Right)* Minimum and maximum run times for `RNA2Dfold` and `FFTbor2D`. For each collection of 100 random sequences of length n, the minimum and maximum run time for a sequence of that length was computed. Taken together, these figures clearly show the run time dependence of `RNA2Dfold` on particular sequences, while the run time of `FFTbor2D` depends only on sequence length, rather than sequence details.

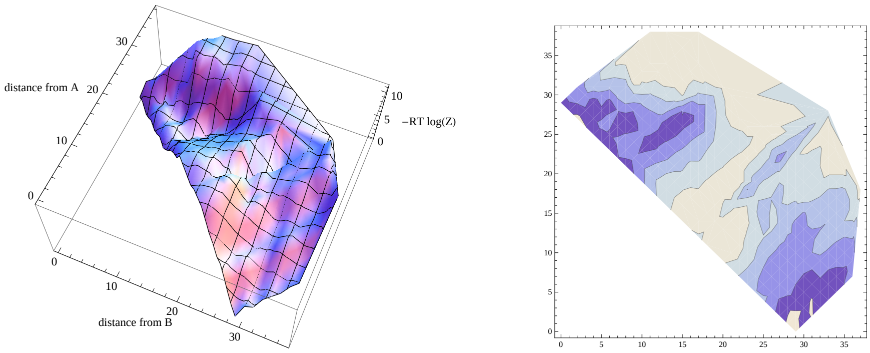

Fig. 6 2D projection of energy landscape for Spliced Leader (SL) RNA from *Leptomonas collosoma*, having sequence **`AACUAAAACA AUUUUUGAAG AACAGUUUCU GUACUUCAUU GGUAUGUAGA GACUUC`** and metastable secondary structures $A =$ ..((...((((((..(((((.((((...)))).)))))..))).)))..))....., and $B =$ ......................((((((((((((.....)))))..)))))))... The x-axis (resp. y-axis) represents base pair distance between metastable structure A (resp. B), while the z-axis represents the ensemble free energy $-RT \log Z_{x,y}$, where $Z_{x,y}$ is computed in `FFTbor2D` by $Z_{x,y} = p(x, y) \cdot Z$. Low energy positions (x, y) correspond to high Boltzmann probability positions. The left panel depicts a heat map of the ensemble free energy, while the right panel depicts a contour map with level curves. In analogy with mountain climbing, one expects an optimal path to follow along the valley regions in traversing the landscape from A to B. Data produced with `FFTbor2D`; graphics produced using Mathematica.

the 20,000 sequences generated by *Switch Design* have higher GC-content than that of spliced leader RNA from *L. collosoma*, as shown in the right panel of Figure 7.

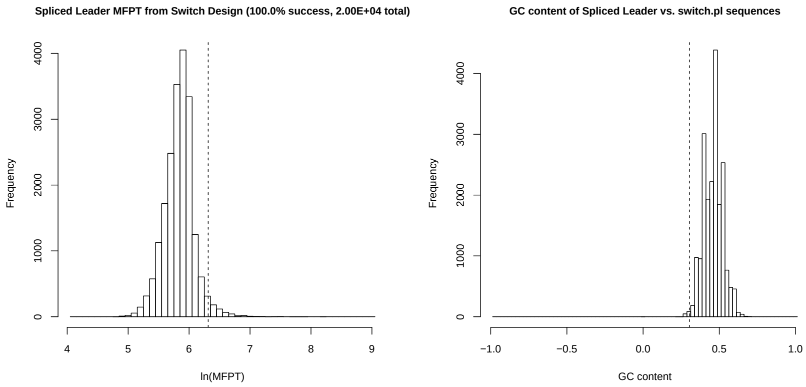

Fig. 7 *(Left)* Histogram of log base 10 mean first passage times, computed by $(I - M^{-}_{(d_0,0)})^{-1} \cdot \mathbf{e}$ (see text), for a collection of 20,000 RNA sequences **s**, each of has a metastable structure at the minimum free energy structure A of *L. collosoma* spliced leader RNA, given by ..((...((((((..(((((.((((...)))).)))))..))).)))..))....., as well as a metastable structure at the alternate structure B, given by ......................((((((((((((.....)))))..)))))))... These sequences were generated by the program *Switch Design* (`switch.pl`) described in [7], using the Turner 1999 energy model without dangles. The log base 10 value of MFPT of *L. collosoma* spliced leader is indicated by the arrow in figure, corresponding to a Z-score of 1.989, which suggests that spliced leader RNA may be under evolutionary pressure for slow folding kinetics from A to B. *(Right)* Histogram of GC-content of the 20,000 sequences generated by *Switch Design.* Note that GC-content of spliced leader RNA is 30.357%, which constitutes a Z-score of +2.50.

5 Discussion

Given an RNA sequence **s** and two reference secondary structures A, B, the algorithm, `FFTbor2D`, computes the partition function $Z(x, y)$, defined to be the sum of Boltzmann factors $\exp(-E(S)/RT)$ over all secondary structures S, having base pair distance x to A and distance y to B, where $0 \leq x, y \leq n$ and n denotes the length **s**. Using polynomial interpolation with the FFT and exploiting the observations of Lemmas 1,2, `FFTbor2D` has worst case complexity $O(n^5)$ time and $O(n^2)$ space. This worst case algorithmic complexity is two orders of magnitude faster and requires two orders of magnitude less space than the worst case complexity of the algorithm `RNA2Dfold` of Lorenz et al. [14]. This run time complexity bound is not only theoretical, but entails a significant practical speedup, as depicted in Figures 3,4 and 5.

An important advantage of `RNA2Dfold` over `FFTbor2D` is that the former can additionally compute the structures $M_{x,y}$ having minimum free energy over all structures that are x-neighbors of metastable A and simultaneously y-neighbors of metastable B. (There is a similar advantage of `RNAbor` [8] over the faster `FFTbor` [21].) As well, `RNA2Dfold` directly computes the partition function values $Z_{x,y}$, while `FFTbor2D` estimates $Z_{x,y}$ by computing $p(x, y) \cdot Z$. This difference entails a significant loss of precision, when depicting the energy landscape.

The right panel of Figure 6 depicts a contour heat map of the 2D energy landscape for spliced leader RNA from *L. collosoma*, as computed by `FFTbor2D`. This figure should be compared with the left panel of Figure 8, which depicts a

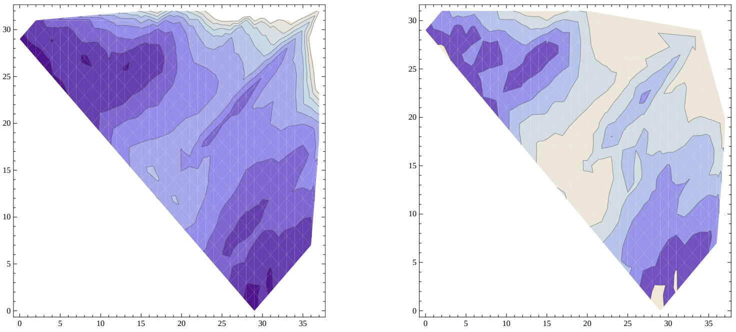

Fig. 8 2D projection of energy landscape for Spliced Leader (SL) RNA from *Leptomonas collosoma*, as in Figure 6, except that in the left panel, ensemble free energy $-RT \log Z_{x,y}$ is computed from the values of $Z_{x,y}$ output by `RNA2Dfold`, while in the right panel, ensemble free energy is computed from the values $Z_{x,y} = p(x,y) \cdot Z$, where values $p(x,y)$ are output by `RNA2Dfold`. The loss of detail in the 2D energy landscape is caused uniquely by working with probabilities $p(x,y)$, rather than partition function values $Z_{x,y}$. Data produced with `RNA2Dfold`; graphics produced using Mathematica.

contour heat map of the 2D energy landscape for the same RNA, as computed by `RNA2Dfold`. Notice the additional detail in this figure, due to the fact that `RNA2Dfold` directly computes $Z_{x,y}$, while `FFTbor2D` computes Boltzmann probabilities $p(x,y)$ by interpolation, allowing one to subsequently compute $Z_{x,y} = p(x,y) \cdot Z$. The additional detail of the energy landscape is lost in the right panel of Figure 8, obtained by computing ensemble free energy by $-RT \log p(x,y) + RT \log \cdot Z$, where $p(x,y)$ is parsed from `RNA2Dfold` output. It follows that the loss of detail in 2D energy landscape is due solely to the fact that probabilities $p(x,y)$ are computed by `FFTbor2D`, rather than partition function values $Z_{x,y}$. Given numerical stability issues involving the FFT engine, `FFTbor2D` can only estimate the probabilities $p(x,y)$ to within $m = 8$ significant places. Nevertheless, our algorithm `FFTbor2D` was developed with the intended application in synthetic biology, where one wishes to prioritize RNA candidate sequences with respect to kinetics. For such applications, the speedup of `FFTbor2D` is an important asset.

Acknowledgments

We would like to thank anonymous reviewers for their critique and suggestions. `FFTbor2D` uses the implementation FFTW of the Fast Fourier Transform, developed by Frigo and Johnson [9] at `http://www.fftw.org/`. The research was funded by National Science Foundation grants DMS-1016618 and DBI-1262439. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

1. E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide.*

Society for Industrial and Applied Mathematics SIAM, 1999.

2. N. J. Baird, S. J. Ludtke, H. Khant, W. Chiu, T. Pan, and T. R. Sosnick. Discrete structure of an RNA folding intermediate revealed by cryo-electron microscopy. *J. Am. Chem. Soc.*, 132(46):16352–16353, November 2010.
3. T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Algorithms*. McGraw-Hill, 1990. 1028 pages.
4. I. Dotu, W. A. Lorenz, P. Van Hentenryck, and P. Clote. Computing folding pathways between RNA secondary structures. *Nucleic. Acids. Res.*, 38(5):1711–1722, 2010.
5. C. Flamm. *Kinetic Folding of RNA*. PhD thesis, University of Vienna, 1998. Department of Chemistry.
6. C. Flamm, W. Fontana, I.L. Hofacker, and P. Schuster. RNA folding at elementary step resolution. *RNA*, 6:325–338, 2000.
7. C. Flamm, I. L. Hofacker, S. Maurer-Stroh, P. F. Stadler, and M. Zehl. Design of multistable RNA molecules. *RNA.*, 7(2):254–265, February 2001.
8. E. Freyhult, V. Moulton, and P. Clote. Boltzmann probability of RNA structural neighbors and riboswitch detection. *Bioinformatics*, 23(16):2054–2062, August 2007.
9. Matteo Frigo and Steven G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. Special issue on "Program Generation, Optimization, and Platform Adaptation".
10. M. Geis, C. Flamm, M. T. Wolfinger, A. Tanzer, I. L. Hofacker, M. Middendorf, C. Mandl, P. F. Stadler, and C. Thurner. Folding kinetics of large RNAs. *J. Mol. Biol.*, 379(1):160–173, May 2008.
11. K. Gerdes, A. P. Gultyaev, T. Franch, K. Pedersen, and N. D. Mikkelsen. Antisense RNA-regulated programmed cell death. *Annu. Rev. Genet.*, 31:1–31, 1997.
12. K.A. Lecuyer and D.M. Crothers. The Leptomonas collosoma spliced leader RNA can switch between two alternate structural forms. *Biochemistry*, 32(20):5301–5311, 1993.
13. Y. Li and S. Zhang. Predicting folding pathways between RNA conformational structures guided by RNA stacks. *BMC. Bioinformatics*, 13:S5, 2012.
14. R. Lorenz, C. Flamm, and I.L. Hofacker. 2D projections of RNA folding landscapes. In I. Grosse, S. Neumann, S. Posch, F. Schreiber, and P.F. Stadler, editors, *German Conference on Bioinformatics 2009*, volume 157 of *Lecture Notes in Informatics*, pages 11–20, 2009.
15. J.S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29:1105–1119, 1990.
16. C.D. Meyer. The role of the group inverse in the theory of finite Markov chains. *SIAM Rev.*, 17(46):443–464, 1975.
17. S. Mitra, A. Laederach, B. L. Golden, R. B. Altman, and M. Brenowitz. RNA molecules with conserved catalytic cores but variable peripheries fold along unique energetically optimized pathways. *RNA.*, 17(8):1589–1603, August 2011.
18. S.R. Morgan and P.G. Higgs. Barrier heights between ground states in a model of RNA secondary structure. *J. Phys. A: Math. Gen.*, 31:3153–3170, 1998.
19. K. Neupane, H. Yu, D. A. Foster, F. Wang, and M. T. Woodside. Single-molecule force spectroscopy of the add adenine riboswitch relates folding to regulatory mechanism. *Nucleic. Acids. Res.*, 39(17):7677–7687, September 2011.
20. 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*, 77(11):6309–6313, 1980.
21. 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*, 7(12):e50506, 2012.
22. X. Tang, S. Thomas, L. Tapia, D. P. Giedroc, and N. M. Amato. Simulating RNA folding kinetics on approximated energy landscapes. *J. Mol. Biol.*, 381(4):1055–1067, September 2008.
23. J. Waldispühl and Y. Ponty. An unbiased adaptive sampling algorithm for the exploration of RNA mutational landscapes under evolutionary pressure. *Journal of Computational Biology*, 18(11):1465–1479, 2011.
24. M. Wolfinger, W.A. Svrcek-Seiler1, C. Flamm, and P.F. Stadler. Efficient computation of RNA folding dynamics. *J Phys. A: Math. Gen.*, 37:4731–4741, 2004.
25. M. T. Woodside, C. Garcia-Garcia, and S. M. Block. Folding and unfolding single RNA molecules under tension. *Curr. Opin. Chem. Biol.*, 12(6):640–646, December 2008.

26. S. Wuchty, W. Fontana, I. L. Hofacker, and P. Schuster. Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers*, 49:145–165, 1999.
27. A. Xayaphoummine, T. Bucher, and H. Isambert. Kinefold web server for RNA/DNA folding path and structure prediction including pseudoknots and knots. *Nucleic. Acids. Res.*, 33(Web):W605–W610, July 2005.
28. T. Xia, Jr. J. SantaLucia, M.E. Burkard, R. Kierzek, S.J. Schroeder, X. Jiao, C. Cox, and D.H. Turner. Thermodynamic parameters for an expanded nearest-neighbor model for formation of RNA duplexes with Watson-Crick base pairs. *Biochemistry*, 37:14719–35, 1999.
29. P. Zhao, W. B. Zhang, and S. J. Chen. Predicting secondary structural folding kinetics for nucleic acids. *Biophys. J.*, 98(8):1617–1625, April 2010.

Appendix

Here, we provide proofs of Theorem 1 and of Lemmas 2,3,4.

Theorem 1: Let $s_1, \ldots, s_n$ be a given RNA sequence. For any integers $1 \leq i \leq j \leq n$, let $\mathcal{Z}_{i,j}(x) = \sum_{r=0}^{n-1} \sum_{s=0}^{n-1} z_{rn+s} x^{rn+s}$ where $z_{rn+s}(i,j) = \mathbf{Z}_{i,j}^{rn+s}$. Then for $i \leq j \leq i+\theta$, $\mathcal{Z}_{i,j}(x) = 1$ and for $j > i+\theta$ we have the recurrence relation

$$\begin{aligned}\mathcal{Z}_{i,j}(x) &= \mathcal{Z}_{i,j-1}(x) \cdot x^{\alpha_0 n+\beta_0} + \\ &\sum_{\substack{s_k s_j \in \mathbb{B}, \\ i \leq k < j}} \left(e^{\frac{-E_0(k,j)}{RT}} \cdot \mathcal{Z}_{i,k-1}(x) \cdot \mathcal{Z}_{k+1,j-1}(x) \cdot x^{\alpha(k)n+\beta(k)} \right)\end{aligned} \tag{17}$$

where $\alpha_0 = 1$ if j is base-paired in $A_{[i,j]}$ and 0 otherwise, $\beta_0 = 1$ if j is base-paired in $B_{[i,j]}$ and 0 otherwise, and $\alpha(k) = d_{BP}(A_{[i,j]}, A_{[i,k-1]} \cup A_{[k+1,j-1]} \cup \{(k,j)\})$, $\beta(k) = d_{BP}(B_{[i,j]}, B_{[i,k-1]} \cup B_{[k+1,j-1]} \cup \{(k,j)\})$.

Proof: First, some notation is necessary. Recall that if F is an arbitrary polynomial [resp. analytic] function, then $[x^{rn+s}]F(x)$ denotes the coefficient of monomial x^{rn+s} in the Taylor expansion of $F(x)$] – for instance, in equation (3) of the main text, $[x^{rn+s}]Z(x) = z_{rn+s}$.

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

$$\begin{aligned}
[x^{rn+s}]\mathcal{Z}_{i,j}(x) &= z_{rn+s}(i,j) = \mathbf{Z}_{i,j}^{rn+s} \\
&= \mathbf{Z}_{i,j-1}^{(r-\alpha_0)n+(s-\beta_0)} + \sum_{k=i}^{j-1} \sum_{u_0+u_1=r-\alpha(k)} \sum_{v_0+v_1=s-\beta(k)} e^{\frac{-E_0(k,j)}{RT}} \cdot \mathbf{Z}_{i,k-1}^{u_0 n+v_0} \cdot \mathbf{Z}_{k+1,j-1}^{u_1 n+v_1} \\
&= [x^{(r-\alpha_0)n+(s-\beta_0)}]\mathcal{Z}_{i,j-1}(x) + \\
&\sum_{k=i}^{j-1} \sum_{u_0+u_1=r-\alpha(k)} \sum_{v_0+v_1=s-\beta(k)} e^{\frac{-E_0(k,j)}{RT}} \cdot \left\{[x^{u_0 n+v_0}]\mathcal{Z}_{i,k-1}(x)\right\} \cdot \left\{[x^{u_1 n+v_1}]\mathcal{Z}_{k-1,j-1}(x)\right\} \\
&= [x^{(r-\alpha_0)n+(s-\beta_0)}]\mathcal{Z}_{i,j-1}(x) + \\
&\sum_{k=i}^{j-1} \sum_{u_0+u_1=r-\alpha(k)} \sum_{v_0+v_1=s-\beta(k)} e^{\frac{-E_0(k,j)}{RT}} \cdot [x^{(u_0+u_1)n+(v_0+v_1)}]\mathcal{Z}_{i,k-1}(x)\mathcal{Z}_{k-1,j-1}(x) \\
&= [x^{(r-\alpha_0)n+(s-\beta_0)}]\mathcal{Z}_{i,j-1}(x) + \sum_{k=i}^{j-1} e^{\frac{-E_0(k,j)}{RT}} \cdot [x^{(r-\alpha(k))n+(s-\beta(k))}]\mathcal{Z}_{i,k-1}(x)\mathcal{Z}_{k-1,j-1}(x) \\
&= [x^{rn+s}] \left(\mathcal{Z}_{i,j-1}(x) \cdot x^{\alpha_0 n+\beta_0} \right) + \sum_{k=i}^{j-1} e^{\frac{-E_0(k,j)}{RT}} \cdot [x^{rn+s}] \left(\mathcal{Z}_{i,k-1}(x)\mathcal{Z}_{k-1,j-1}(x) x^{\alpha(k)n+\beta(k)} \right) \\
&= [x^{rn+s}] \left(\mathcal{Z}_{i,j-1}(x) \cdot x^{\alpha_0 n+\beta_0} + \sum_{k=i}^{j-1} e^{\frac{-E_0(k,j)}{RT}} \cdot \mathcal{Z}_{i,k-1}(x)\mathcal{Z}_{k-1,j-1}(x) x^{\alpha(k)n+\beta(k)} \right).
\end{aligned}$$

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

The following lemma was proved in [21], and is reproduced for the convenience of the reader.

Lemma 2: If $\mathcal{Z}(x)$ is the complex polynomial defined in equation (9) of the main text, 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

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

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 $\mathbf{Z}^k_{1,n}$ over all secondary structures of a given RNA sequence $s_1, \ldots, s_n$, whose base pair distance from initial structure 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.

$$
\begin{aligned}
(a+bi)^m &= \sum_{k=0}^{m} \binom{m}{k} a^{m-k} \cdot (bi)^k \\
(bi)^k &= \begin{cases} b^k & \text{if } k \equiv 0 \bmod 4 \\ ib^k & \text{if } k \equiv 1 \bmod 4 \\ -b^k & \text{if } k \equiv 2 \bmod 4 \\ -ib^k & \text{if } k \equiv 3 \bmod 4 \end{cases}
\end{aligned}
$$

$$
\begin{aligned}
(a-bi)^m &= \sum_{k=0}^{m} \binom{m}{k} a^{m-k} \cdot (-bi)^k \\
(-bi)^k &= \begin{cases} b^k & \text{if } k \equiv 0 \bmod 4 \\ -ib^k & \text{if } k \equiv 1 \bmod 4 \\ -b^k & \text{if } k \equiv 2 \bmod 4 \\ ib^k & \text{if } k \equiv 3 \bmod 4 \end{cases}
\end{aligned}
$$

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 3: Let $d_0 = d_{BP}(A, B)$. Then for any complex number $\alpha \in \mathbb{C}$, $Z(-\alpha) = -1^{d_0} \cdot Z(\alpha)$.

Proof: The lemma states that if the base pair distance between reference structures A, B is even, then $Z(-\alpha) = Z(\alpha)$, while if the distance is odd, then $Z(-\alpha) = -Z(\alpha)$. Suppose first that d_0 is even. By Lemma 1, $Z(x) = z_0 + z_2x^2 + z_4x^4 + \cdots + z_{M-2)}x^{M-2)}$, and so $Z(-\alpha) = Z(\alpha)$. Suppose now that d_0 is odd. By Lemma 1, $Z(x) = z_1x^1 + z_3x^3 + z^5x^5 \cdots + z_{M-1}x^{M-1}$, and so $Z(-\alpha) = -Z(\alpha)$.

Lemma 4: Suppose that $\nu = \exp(\frac{2\pi i}{M})$ is the principal M-root of unity, and that $\frac{M}{4} < k \leq \frac{M}{2}$. Then

$$\nu^k = -(\nu^{-(M/2-k)}) = -\overline{\nu^{M/2-k}}.$$

Proof: Recall Euler's formula in complex analysis: $\exp(ix) = \cos(x) + i\sin(x)$. As well, recall that $\sin(\pi) = 0$, $\cos(\pi) = -1$, and the trigonometric addition formulas:

$$
\begin{aligned}
\cos(\alpha - \beta) &= \cos(\alpha)\cos(\beta) + \sin(\alpha)\sin(\beta) \\
\sin(\alpha - \beta) &= \sin(\alpha)\cos(\beta) - \sin(\beta)\cos(\alpha).
\end{aligned}
$$

Then

$$
\begin{aligned}
\nu^{M/2-k} &= \exp\left(\frac{2\pi i(M/2-k)}{M}\right)\\
&= \cos\left(\frac{2\pi(M/2-k)}{M}\right) + i\sin\left(\frac{2\pi(M/2-k)}{M}\right)\\
&= \cos\left(\pi - \frac{2\pi k}{M}\right) + i\sin\left(\pi - \frac{2\pi k}{M}\right)\\
&= \left[\cos(\pi)\cos\left(\frac{2\pi k}{M}\right) + \sin(\pi)\sin\left(\frac{2\pi k}{M}\right)\right] +\\
&\quad \left[\sin(\pi)\cos\left(\frac{2\pi k}{M}\right) - \sin\left(\frac{2\pi k}{M}\right)\cos(\pi)\right]\\
&= -\cos\left(\frac{2\pi k}{M}\right) + i\sin\left(\frac{2\pi k}{M}\right)\\
&= -1\left[\cos\left(\frac{2\pi k}{M}\right) - i\sin\left(\frac{2\pi k}{M}\right)\right]\\
&= -1\cdot\overline{\cos\left(\frac{2\pi k}{M}\right) + i\sin\left(\frac{2\pi k}{M}\right)}\\
&= -1\cdot\overline{\exp\left(\frac{2\pi i k}{M}\right)} = -\overline{\nu^k}.
\end{aligned}
$$

It follows that $\nu^{M/2-k} = -\overline{\nu^k}$, so $\nu^k = \overline{-\nu^{(M/2-k)}} = -\overline{\nu^{(M/2-k)}}$. This completes the proof of the lemma.