On the use of polynomial interpolation to improve the performance of dynamic programming algorithms with discrete distance metrics

Evan Senter

2015

Outline

Goal of Presentation

Structure of talk

- Provide motivation for synthetic RNA design
- Overview of thermodynamic-based computational analysis
- Describe algorithm to generate a discretized, coarse-grained energy landscape
- Show how polynomial interpolation improves asymptotics
- Highlight practical applications of energy landscapes

Why RNA?

- The central dogma of DNA is a lie
- RNA has been shown to regulate many aspects of the cellular machinery
- What was once considered ‘junk DNA’ is now appreciated as non-coding RNA ‘ncRNA’

Why RNA?

- RNA is an enzymatically active molecule (hydroxyl group on 2' carbon is highly reactive)
- Secondary structure is more mathematically tractable than proteins
- Interesting applications of *cis*-regulation via motifs in the 5' untranslated region of coding RNAs

Outline

RNA Representation

Sequence

An RNA sequence is a string $\mathbf{s} = s_1, \ldots, s_n$, where $s_i \in \{\text{A}, \text{U}, \text{G}, \text{C}\}$

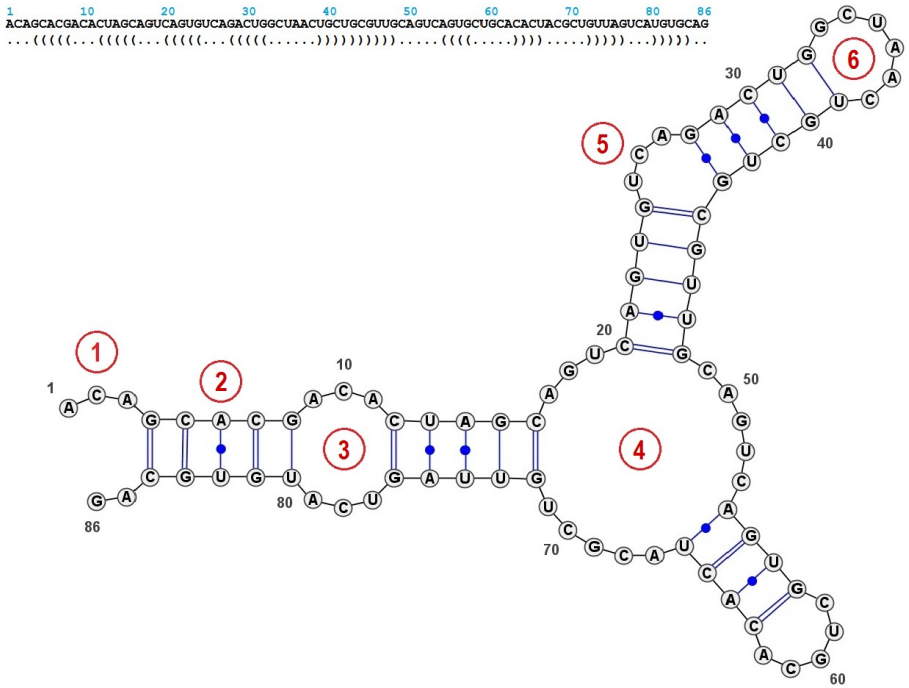
Structure

An secondary structure $\mathcal{S}$ compatible with $\mathbf{s}$ is a collection of base pair tuples such (i, j), such that:

- $(\mathbf{s}_i, \mathbf{s}_j) \in \mathbb{B}$
- $1 \leq i \leq i + \theta < j \leq n$ where $\theta \geq 0$
- Given $(i, j), (x, y)$ from $\mathcal{S}$, $i = x \iff j = y$
- Given $(i, j), (x, y)$ from $\mathcal{S}$, $i < x < j \iff i < y < j$

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

Structural Motifs



Structural Motifs

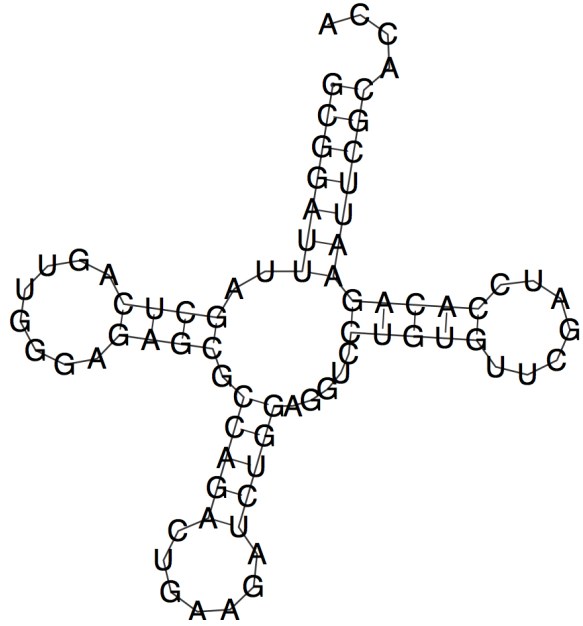
1. Exterior loop
2. Stack
3. Interior loop
4. Multiloop
5. Bulge
6. Hairpin

RNA Notation

Yeast tRNA^{phe} dot-bracket notation

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

Yeast tRNA^{phe} structural diagram



Outline

Problem Definition

Desire

Given an input sequence $\mathbf{s}$ and two input structures $\mathcal{A}$, $\mathcal{B}$, we would like to compute all possible structures $\mathcal{S}$ compatible with $\mathbf{s}$, and bin them into discrete sets based on their *distance* to $\mathcal{A}$ and $\mathcal{B}$

Issue

Consider $\mathbb{S}$ to be the set of all structures compatible with $\mathbf{s}$. It has been shown that $|\mathbb{S}|$ grows exponentially with sequence length n

Refinement

Rather than store $\mathbb{S}$ at any point in time, we will use dynamic programming to compute the thermodynamic properties of these bins

Concrete Example

Input

$\texttt{GGAAACC} = \mathbf{s}$

$\texttt{.......} = \mathcal{A}$

$\texttt{.(...).} = \mathcal{B}$

Structures

$\texttt{.......}\ \ 0.00\frac{\text{kcal}}{\text{mol}}, [0, 1]$

$\texttt{.(...).}\ \ 4.40\frac{\text{kcal}}{\text{mol}}, [1, 0]$

$\texttt{(....).}\ \ 2.30\frac{\text{kcal}}{\text{mol}}, [1, 2]$

$\texttt{.(....)}\ \ 4.10\frac{\text{kcal}}{\text{mol}}, [1, 2]$

$\texttt{(.....)}\ \ 4.20\frac{\text{kcal}}{\text{mol}}, [1, 2]$

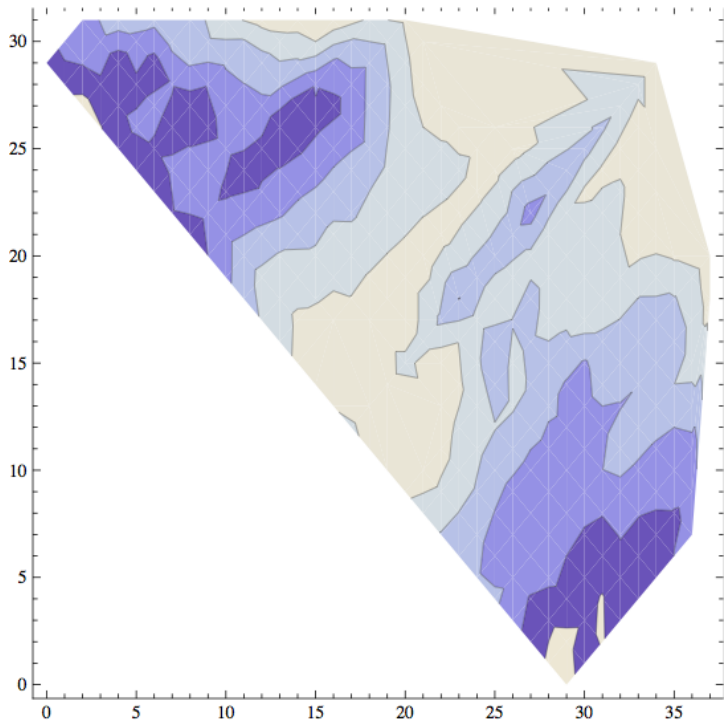
$\texttt{((...))}\ \ 2.10\frac{\text{kcal}}{\text{mol}}, [2, 1]$

Output

$$\begin{bmatrix} 0 & 0.9595 & 0 \\ 0.0001 & 0 & 0.0086 \\ 0 & 0.0318 & 0 \end{bmatrix}$$

Concrete Example

Energy landscape between two metastable structures of *L.collosoma* spliced leader RNA



Base Pair Distance

Symmetric distance

$$d_{\mathsf{BP}}(\mathcal{S}, \mathcal{T}) = |\mathcal{S} \cup \mathcal{T}| - |\mathcal{S} \cap \mathcal{T}|$$

Distance between two structures

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

Parameterized Partition Function, 1D

$\mathbf{Z}$ binned by k

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

Recursions to compute $\mathbf{Z}_{i,j}^{k}$

Structural decomposition from one target

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

Parameterized Partition Function, 2D

$\mathbf{Z}$ binned by x, y pairs

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

Recursions to compute $\mathbf{Z}_{i,j}^{x,y}$

Structural decomposition from two targets

$$\mathbf{Z}_{i,j}^{x,y} = \mathbf{Z}_{i,j-1}^{x-\omega_0,y-\beta_0} + \sum_{\substack{s_k s_j \in \mathbb{B}, \\ i \leq k < j}} \left(e^{\frac{-E_0(k,j)}{RT}} \sum_{u+u'=x-\omega(k)} \sum_{v+v'=y-\beta(k)} \mathbf{Z}_{i,k-1}^{u,v} \cdot \mathbf{Z}_{k+1,j-1}^{u',v'} \right)$$

Partition function of a variable x

Only compute $\mathcal{Z}_{i,j}(x)$x instead of $\mathbf{Z}_{i,j}^{x,y}$

$$\mathcal{Z}_{i,j}(x) = \mathcal{Z}_{i,j-1}(x) \cdot x^{\omega_0 n + \beta_0} + \sum_{\substack{s_k s_j \in \mathbb{B}, \\ i \leq k < j}} \left(e^{\frac{-E_0(k,j)}{RT}} \cdot \mathcal{Z}_{i,k-1}(x) \cdot \mathcal{Z}_{k+1,j-1}(x) \cdot x^{\omega(k) n + \beta(k)} \right)$$

Outline

FFT background

Complex *k*th roots of unity

$$\omega_0 = \exp(\frac{0 \cdot 2\pi i}{n^2}), \omega_1 = \exp(\frac{1 \cdot 2\pi i}{n^2}), \ldots, \omega_{n^2-1} = \exp(\frac{(n^2-1) \cdot 2\pi i}{n^2})$$

Evaluate $\mathcal{Z}_{i,j}(x)$x for all n^2 roots of unity

$$y_0 = \mathcal{Z}(\omega_0), \ldots, y_{n^2-1} = \mathcal{Z}(\omega_{n^2-1}))$$

Represent results of evaluation in column form

$$\mathbf{Y} = (y_0, \ldots, y_{n^2-1})^{\mathsf{T}}$$

Vandermonde matrix

Matrix construction

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

Definition

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

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n^2-1} \end{pmatrix} = V_n \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{n^2-1} \end{pmatrix}$$

Since we defined $\mathbf{Y} = (y_0, \ldots, y_{n-1})^{\mathsf{T}}$, where:

$$y_0 = \mathcal{Z}(\omega_0), \ldots, y_{n^2-1} = \mathcal{Z}(\omega n^2 - 1))$$

and $\omega_k = \exp(\frac{2\pi ki}{n^2})$, it follows that the coefficients $c_{rn+s} = \mathbf{Z}_{1,n}^{rn+s}$ in the polynomial:

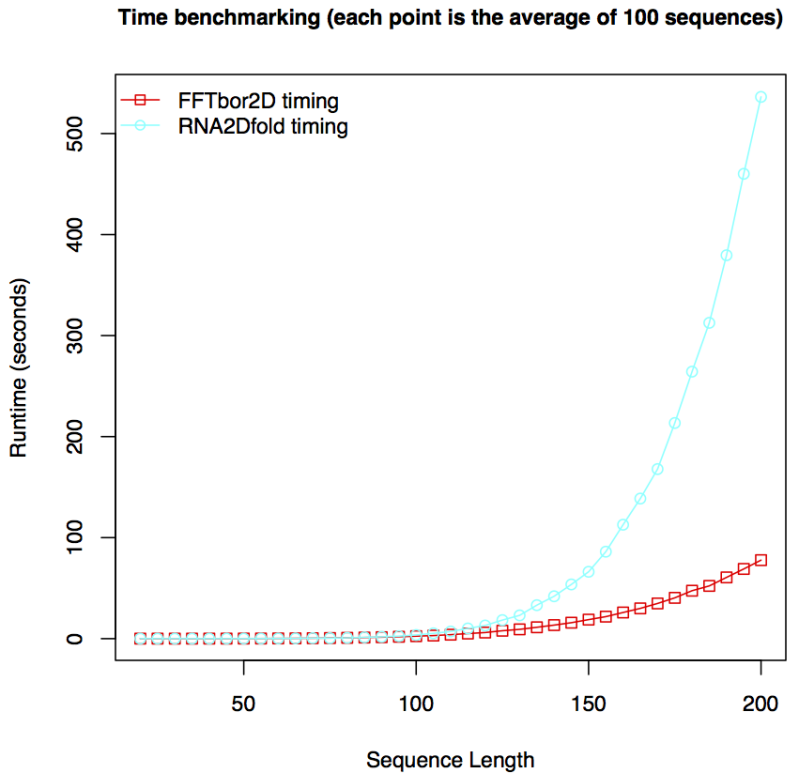
$$\mathcal{Z}(x) = c_0 + c_1 x + \cdots + c_{n^2-1} x^{n^2-1}$$

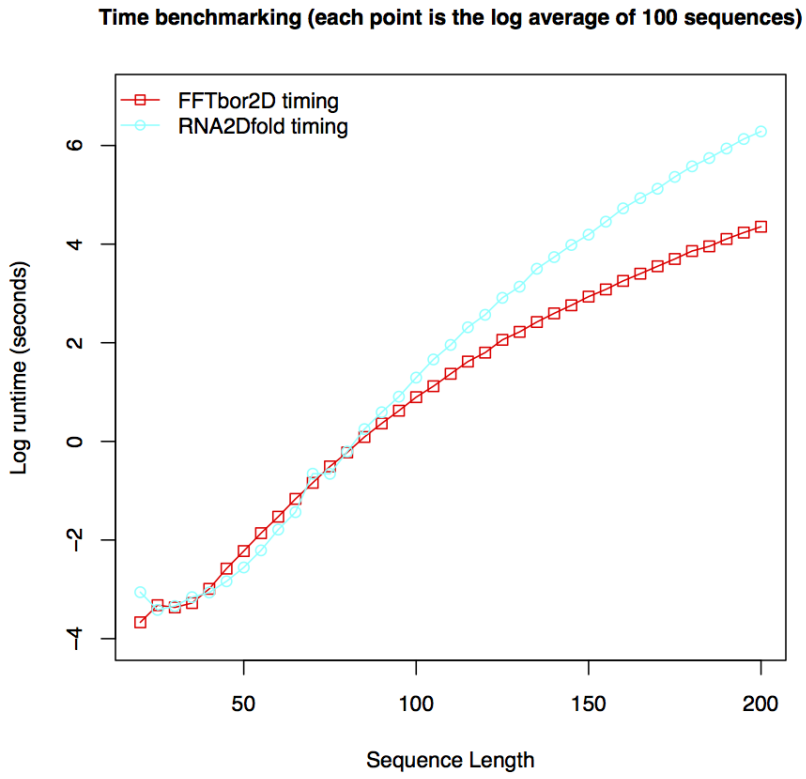
can be computed using the Fast Fourier Transform, and:

$$c_{rn+s} = = \sum_{\substack{\mathcal{S} \text{ such that} \\ d_{\mathsf{BP}}(\mathcal{S}, \mathcal{A}) = r,\, d_{\mathsf{BP}}(\mathcal{S}, \mathcal{B}) = s}} e^{\frac{-E(\mathcal{S})}{RT}}$$

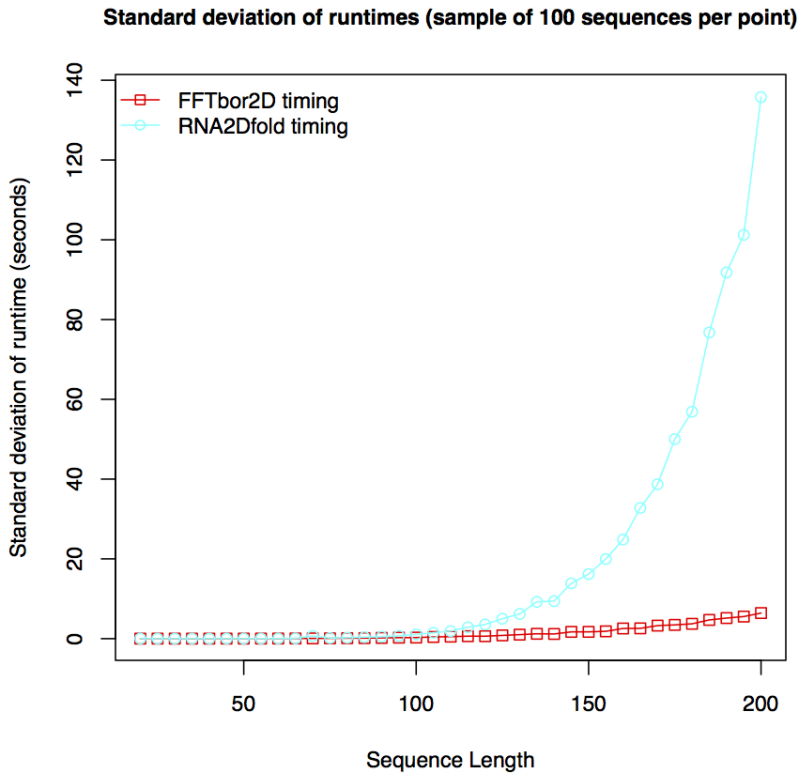
Outline

Performance Characteristics





Performance Characteristics



- Approach using FFT goes from $O(n^7)$ to $O(n^5)$
- We observe a real performance gain in line with 100x speedup
- Memory requirements drop from $O(n^4)$ to $O(n^2)$
- More consistent performance characteristics

Questions?

Thanks for your time!