

Simulate co-transcriptional folding kinetics by genetic algorithm

Zhuoran Qiao

August 15, 2018

We developed a genetic algorithm based approach to simulate kinetics of co-transcriptional folding. Our method is built on two following assumptions:

- 1 All populated RNA secondary structures (SS) are linkage of locally optimal or sub-optimal structures at various folding sites;
- 2 Global structural rearrangement of a partial RNA segment is permitted only if it's folding to the optimal SS on that segment.

Formally, we denote a domain $D_{A,B}$ as a segment between base A and B that all contacts on that segment are local. For simplicity, we denote **foldon** as domains with optimal secondary structures: $D_{A,B}^{foldon} = \text{MFE}(\text{sequence}[A,B])$. Note that $'.'$ is a trivial example of foldon. Our assumption 1 can be rewritten as

$$D_{A,B} = D_{A,i_1}^{foldon} \oplus D_{i_1,i_2}^{foldon} \oplus \dots \oplus D_{i_n,B}^{foldon} \quad (1)$$

Where \oplus represents a link operation. Note that all structural information of $D_{A,B}$ is encoded by the sequential representation $[A, i_1, \dots, i_n, B]$; as a foldon is also a linkage of smaller foldons, there could be multiple way to represent $D_{A,B}$. Here we introduce **Irreducible Foldon Representation** (IFR) as sequential representations for which linkage of every adjacent foldons is not another foldon: $\forall k, D_{i_k,i_{k+1}}^{foldon} \oplus D_{i_{k+1},i_{k+2}}^{foldon} \neq D_{i_k,i_{k+2}}^{foldon}$. Then the sufficient and nessary condition for structural rearrangement is

$$\begin{aligned} \langle D_{A,B}^u | \hat{\mathbf{T}} | D_{A,B}^v \rangle \neq 0 \text{ if and only if } \exists i, j \text{ satisfies} \\ i, j \in D_{A,B}^u \cdot \text{IFR}, i, j \in D_{A,B}^v \cdot \text{IFR}; \\ D_{A,i}^u = D_{A,i}^v, D_{j,B}^u = D_{j,B}^v; \\ D_{i,j}^u = D_{i,j}^{foldon} \text{ or } D_{i,j}^v = D_{i,j}^{foldon}. \\ \text{Then } \langle D_{A,B}^u | \hat{\mathbf{T}} | D_{A,B}^v \rangle = \langle D_{i,j}^u | \hat{\mathbf{T}} | D_{i,j}^v \rangle. \end{aligned}$$

1 Algorithm procedure

During every elongation step, an active species pool of strands with unique SS and diffrent population is updated. New candidate strands $D_{0,L+\Delta L}^{Candidate}$ with length $L + \Delta L$ are generated by a recombination process: for every old strand $D_{0,L}^{Strand}$, all indices in its IFR is identified as possible rearrangement site, then its child strands is generated by linking partial domains $D_{0,\text{Site}}^{Strand}$ with a foldon $D_{\text{Site},L+\Delta L}^{foldon}$ that terminated at $L + \Delta L$.

We assume that elongation will not change the initial population distribution of secondary structures: child strands with the exact parental SS on $[0, L]$ ($D_{0,L+\Delta L}^{child} = D_{0,L}^{strand} \oplus D_{L,L+\Delta L}^{foldon}$) will also inherit the population of their parents.

After structural generation the rate matrix among all candidate strands within the new active species pool is calculated. Then the population distribution of strands after elongation is computed by chemical master equation, and one iterative elongation step is finished.

Pseudocodes of the procedure are as follows:

Algorithm 1 Co-transcriptional folding elongation procedure

```

1: Initialize ActivePool
2: while sequence length > current length do
3:   OldPool  $\leftarrow$  ActivePool
4:   renew ActivePool
5:   Current length  $\leftarrow$  Current length +  $dL$ 
6:    $dt \leftarrow dL / \text{Transcription rate}$ 
7:   for left boundary  $\in \{0, dL, 2dL, \dots, \text{Current length} - dL\}$  do       $\triangleright$  Get all new foldons
8:      $D_{\text{left boundary}, \text{Current length}}^{\text{foldon}} \leftarrow \text{numpy.mfe}(\text{sequence}[\text{left boundary}, \text{Current length}])$ 
9:   end for
10:  for Strand  $\in$  OldPool do       $\triangleright$  Recombination
11:    for Site  $\in$  Strand.IFR do
12:       $D_{0, \text{Current length}}^{\text{Candidate}} \leftarrow D_{0, \text{Site}}^{\text{Strand}} \oplus D_{\text{Site}, \text{Current length}}^{\text{foldon}}$ 
13:      if  $D_{0, \text{Current length}}^{\text{Candidate}} \in \text{ActivePool}$  then
14:        update  $D_{0, \text{Current length}}^{\text{Candidate}}.\text{IFR}$ 
15:      else
16:        add  $D_{0, \text{Current length}}^{\text{Candidate}}$  to ActivePool
17:      end if
18:      if site = Current length -  $dL$  then
19:         $\langle \text{ActivePool.population} | D_{0, \text{Current length}}^{\text{Candidate}} \rangle \leftarrow \langle \text{OldPool.population} | D_{0, \text{Site}}^{\text{Strand}} \rangle$ 
20:      end if
21:    end for
22:  end for
23:  for  $D_{0, \text{Current length}}^u \neq D_{0, \text{Current length}}^v \in \text{ActivePool}$  do       $\triangleright$  Calculate new rate matrix
24:    calculate  $D_{\text{rearrange}}^u, D_{\text{rearrange}}^v$        $\triangleright$  Find all helices involved in rearrangement
25:     $\langle D_{\text{rearrange}}^u | \hat{\mathbf{T}} | D_{\text{rearrange}}^v \rangle \leftarrow k_0 \exp\left(-\frac{1}{RT}(\Delta G_u^{\text{Stack}} + \Delta G_v^{\text{Loop}})\right)$ 
26:  end for
27:   $\langle \text{ActivePool.population} | \leftarrow \langle \text{ActivePool.population} | \exp(\hat{\mathbf{T}}) \rangle$        $\triangleright$  Master equation
28:  reserve top  $N$  populated strands in ActivePool       $\triangleright$  Selection
29:  renormalize  $\langle \text{ActivePool.population} |$ 
30: end while

```
