Simulate co-transcriptional folding kinetics by genetic algorithm

Zhuoran Qiao

August 9, 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 (foldons) 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: $D^{foldon}_{A,\,B}$ = MFE(sequence[A,B]). Note that $'.'$ is a trival example of foldon. Our assumption 1 can be rewritten as

$$D_{A,\,B} = D^{foldon}_{A,i_1} \oplus D^{foldon}_{i_1,i_2} \oplus ... \oplus D^{foldon}_{i_n,B} \tag{1}$$

Where $\oplus$ represents a link operation. Note that all structural information of $D_{A,\,B}$ is encoded by the sequential representation $[A,\, i_1,\, ...,\, i_n,\, B]$; 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 each adjacent foldons is not a foldon: $\forall k,\, D^{foldon}_{i_k,i_{k+1}} \oplus D^{foldon}_{i_{k+1},i_{k+2}} \neq D^{foldon}_{i_k,i_{k+2}}$. Then the sufficient and nessary condition for structural rearrangement is

$$\langle D^u_{A,\,B}|\hat{\mathbf{T}}|D^v_{A,\,B}\rangle \neq 0 \text{ if and only if } \exists\, i,\, j \text{ satisfies}$$
$$i,\, j \in D^u_{A,\,B}.\text{IFR},\; i,\, j \in D^v_{A,\,B}.\text{IFR};$$
$$D^u_{A,\,i} = D^v_{A,\,i},\; D^u_{j,\,B} = D^v_{j,\,B};$$
$$D^u_{i,\,j} = D^{foldon}_{i,\,j} \text{ or } D^v_{i,\,j} = D^{foldon}_{i,\,j}.$$
$$\text{Then } \langle D^u_{A,\,B}|\hat{\mathbf{T}}|D^v_{A,\,B}\rangle = \langle D^u_{i,\,j}|\hat{\mathbf{T}}|D^v_{i,\,j}\rangle.$$

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 $D_{0,\,\mathrm{L}}^{Strand}$, all indices in its IFR is identified as possible rearrangement site, then its child strands is generated by linking partial domains $D_{0,\,\mathrm{Site}}^{Strand}$ with a foldon $D_{\mathrm{Site},L+\Delta L}^{foldon}$ that terminated at $L+\Delta L$.

We assume that elongation will not change the inital 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 structual 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

Initalize ActivePool
while sequence length > current length **do**
  OldPool ← ActivePool
  renew ActivePool
  Current length ← Current length + dL
  dt ← dL / Transcription rate
  for left boundary ∈ {0, dL, 2dL, ..., Current length - dL} **do** ▷ Get all new foldons
    $D^{foldon}_{\text{left boundary, Current length}} \leftarrow$ numpy.mfe(sequence[left boundary, Current length])
  end for
  for Strand ∈ OldPool **do** ▷ Recombination
    for Site ∈ Strand.IFR **do**
      $D^{Candidate}_{0,\text{Current length}} \leftarrow D^{Strand}_{0,\text{ Site}} \oplus D^{foldon}_{\text{Site, Current length}}$
      if $D^{Candidate}_{0,\text{Current length}} \in$ ActivePool **then**
        update $D^{Candidate}_{0,\text{Current length}}$.IFR
      else
        add $D^{Candidate}_{0,\text{Current length}}$ to ActivePool
      end if
      if site = Current length − dL **then**
        $\langle$ActivePool.**population** $|D^{Candidate}_{0,\text{Current length}}\rangle \leftarrow \langle$OldPool.**population** $|D^{Strand}_{0,\text{ Site}}\rangle$
      end if
    end for
  end for
  for $D^{\text{u}}_{0,\text{Current length}} \neq D^{\text{v}}_{0,\text{Current length}} \in$ ActivePool **do** ▷ Calculate new rate matrix
    calculate $D^{u}_{\text{rearrange}}$, $D^{v}_{\text{rearrange}}$ ▷ Find all helices involved in rearrangement
    $\langle D^{u}_{\text{rearrange}}|\hat{\mathbf{T}}|D^{v}_{\text{rearrange}}\rangle \leftarrow k_0 \exp\left(-\frac{1}{RT}(\Delta G^{Stack}_{u} + \Delta G^{Loop}_{v})\right)$
  end for
  $\langle$ActivePool.**population** $| \leftarrow \langle$ActivePool.**population** $| \exp\left(\hat{\mathbf{T}}\right)$ ▷ Master equation
  reserve top N populated strands in ActivePool ▷ Selection
  renormalize $\langle$ActivePool.**population** $|$
end while