An inside-outside algorithms for mutant RNAs with applications to error detections in structured RNAs

Abstract

Some stuff [**?**]
Key words: RNA

1 Introduction

2 Methods

Let Ω be an un-gapped RNA alignment, S its associated secondary structure, s an RNA sequence and $m \geq 0$. S is considered as one derivation of the SCFG generating all secondary structures of length $|s|$. We are interested in the probability of a given position being a specific nucleotides, over all sequences having at most m mutations from s, under the SCFG derivation S (formally $\mathbb{P}(s_i = x \mid s, \Omega, S, m)$). We define as a variant of the *Inside-Outside algorithm* [**?**], allowing us to obtain the desired probability, the two functions $\mathcal{Z}^m_{\substack{(i,j)\\[a,b]}}$ and $\mathcal{Y}^m_{\substack{(i,j)\\[a,b]}}$. The former presented in Eq. 1 and Eq. 2, our version of the *inside*, computes the partition function between i and j included, knowing that position $i-1$ is composed of nucleotide a (resp. $j+1$ is b) and containing m mutations. The latter in Eq. 3 and Eq.4, computes the *outside*, in particular, the partition function considering only between $[0, i] \cup [j, n-1]$ knowing that position $i+1$ is composed of a (resp. $j-1$ is b) and containing m mutations outside.

The Boltzmann weights are a combination of the base pairs stacking energy, using as values those of the NNDB [**?**], and the average isostericity difference between the mutant and Ω, and s with Ω, using the isostericity values as defined in [**?**]. The value of 10 was used for the isostericity of any base pair compared with *GG* given that the latter base pair was not found in the [**?**] tables, and the range of values is $[0, 9.7]$, 0 for a perfect isostericity.

2.1 Definitions

Let be $B := \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{U}\}$, the set of nucleotides. Given $s \in B^n$ an RNA sequence, let s_i be the nucleotide at position i. Let Ω be a set of un-gapped RNA sequences of length n, and S a secondary structure without pseudoknots. Formally, if (i, j) and (k, l) are base pairs in S, there is no overlapping extremities $\{i, j\} \cap \{k, l\} = \varnothing$ and either the intersection is empty ($[i, j] \cap [k, l] = \varnothing$) or one is included in the other ($[k, l] \subset [i, j]$ or $[i, j] \subset [k, l]$). Let R be the Boltzmann constant, T the temperature in Kelvin and the function δ such that: $\forall a, a' \in B, \delta_{a,a'} := \begin{cases} 1 & \text{If } a \equiv a' \\ 0 & \text{Else} \end{cases}$

2.2 Energy Model

The energy used is composed of two function, $\mathrm{ES}^{\beta}_{ab \to a'b'}$ and $\mathrm{EI}^{\Omega}_{(i,j),ab}$. The former is equal to the stacking energy of the base pair with nucleotides ab on top of the base pair with nucleotides $a'b'$, as set in the NNDB [**?**]. If one of the base pair is not valid (i.e. not in $\{\mathrm{GU}, \mathrm{UG}, \mathrm{CG}, \mathrm{GC}, \mathrm{AU} \text{ or } \mathrm{UA}\}$, the value is a parameter $\beta \in [1, \infty]$. This allows to completely forbid a sequence where a base pair is non valid, when $\beta = \infty$ or only penalize it. $\mathrm{EI}^{\Omega}_{(i,j),a'b'}$ is the average of the sum of differences between the isostericity of base pairs at positions (i, j) in Ω and ab, and the isostericity of base pairs at positions (i, j) in Ω and $s_i s_j$. It gives us an estimation of which base pair, between ab in the mutant sequence and $s_i s_j$, is in average more isosteric to Ω. Formally, given s, Ω, two positions (i, S_i) and two nucleotides $a, b \in B$:

$$\mathrm{EI}^{\Omega}_{(i,j),ab} := \frac{\sum_{s' \in \Omega} \big(\mathrm{ISO}((s'_i, s'_j), (a, b))\big) - \big(\mathrm{ISO}((s'_i, s'_j), (s_i, s_j))\big)}{|\Omega|}$$

Where $\mathrm{ISO}((a', b'), (a, b))$ is the isostericity value between the canonical base pairs (a', b') and (a, b) as defined in [**?**]. Let be $\alpha \in [0, 1]$, it will be used to balance the weight given to the stacking

energy and the isostericity, by considering from now on $\alpha \mathrm{ES}^{\beta}_{ab\to a'b'}$ and $(1-\alpha)\mathrm{EI}^{\Omega}_{(i,j),ab}$.

2.3 Inside

The *Inside* function $\mathcal{Z}^{m}_{\substack{(i,j)\\[a,b]}}$ is the partition function considering only the energy in subsequence $[i,j]$ over mutants of s having exactly m mutations between $[i,j]$ and whose nucleotide at position $i-1$ is a (resp. b in position $j+1$). We define function $\mathcal{Z}^{m}_{\substack{(i,j)\\[a,b]}}$ as a recurrence, and will use as initial conditions:

$$\forall i \in (0,\cdots,n-1):\ \mathcal{Z}^{m}_{\substack{(i+1,i)\\[a,b]}} = \begin{cases} 1 & \text{If } m=0 \\ 0 & \text{Else} \end{cases} \tag{1}$$

In other words, when we evaluate the function $\mathcal{Z}$, after exhausting all positions, there is only one possible solution if there is 0 mutations left, and none else. Since the energetic terms only depend on base pairs, they are not involved in the initial conditions.

The recursion itself is composed of four terms:

$$\mathcal{Z}^{m}_{\substack{(i,j)\\[a,b]}} := \begin{cases} \displaystyle\sum_{\substack{a'\in\mathcal{B},\\ \delta_{a',s_i}\le m}} \mathcal{Z}^{m-\delta_{a',s_i}}_{\substack{(i+1,j)\\[a',b]}} & \text{If } S_i = -1 \\ \displaystyle\sum_{\substack{a',b'\in\mathcal{B}^2,\\ \delta_{a'b',s_is_j}\le m}} e^{\frac{-(\alpha\mathrm{ES}^{\beta}_{ab\to a'b'}+(1-\alpha)\mathrm{EI}^{\Omega}_{(i,j),a'b'})}{RT}} \mathcal{Z}^{m-\delta_{a'b',s_is_j}}_{\substack{(i+1,j-1)\\[a',b']}} & \text{Elif } S_i = j \wedge S_{i-1} = j+1 \\ \displaystyle\sum_{\substack{a',b'\in\mathcal{B}^2,\\ \delta_{a'b',s_is_k}\le m}} \sum_{m'=0}^{m-\delta_{a'b',s_is_k}} e^{\frac{-(1-\alpha)\mathrm{EI}^{\Omega}_{(i,k),a'b'}}{RT}} \mathcal{Z}^{m-\delta_{a'b',s_is_k}-m'}_{\substack{(i+1,k-1)\\[a',b']}} \mathcal{Z}^{m'}_{\substack{(k+1,j)\\[b',b]}} & \text{Elif } S_i = k \wedge i < k \le j \\ 0 & \text{Else} \end{cases} \tag{2}$$

The cases can be broked down as follows.

$S_i = -1$: If the nucleotide at position i is not paired, then the value is the same as if we increase the lower interval bound by 1 (i.e. $i+1$), and consider all possible nucleotides a' at position i, updating the number of mutants in function of δ_{a',s_i}.

$S_i = j$ and $S_{i-1} = j+1$: If nucleotide i is paired with j and nucleotide $i-1$ is paired with $j+1$, we are in the only case were stacked base pairs can occur. We thus add the energy of the stacking and of the isostericity of the base pair (i,j). What is left to compute is the *inside* value of the interval $[i+1,j-1]$ over all possible nucleotides $a',b' \in B^2$ at positions i and j respectively.

$S_i = k$ and $i < k \le j$: If nucleotide i is paired with position k but is not stacked outside, the only term contributing directly to the energy is the isostericity of the base pair (i,k). This creates two different intervals for which we must compute the values, $[i+1,k-1]$ and $[k+1,j-1]$, for all possible values $a',b' \in B^2$ for nucleotides at positions i and j respectively.

Else: In all other cases, we are in a derivation of the SCFG that does not correspond to the secondary structure S, and we return 0.

2.4 Outside

The *Outside* function, $\mathcal{Y}$, is the partition function considering only the energy in subsequences $[0,i] \cup [j,n-1]$ over the mutants of s having exactly m mutations between $[0,i] \cup [j,n-1]$ and whose nucleotide at position $i+1$ is a (resp. in position $j-1$ it is b). We define function $\mathcal{Y}^m_{\substack{(i,j)\\[a,b]}}$ as a recurrence, and will use as initial conditions:

$$\mathcal{Y}^m_{\substack{(-1,j)\\[X,X]}} := \mathcal{Z}^m_{\substack{(j,n-1)\\[X,X]}} \tag{3}$$

The recurrence, as shown below, will increase the interval $[i,j]$ by decreasing i when it is not base paired. If it is with a position $k > j$, we increase j to include it. Thus, when we need to evaluate an interval as $(-1,j)$, all stems between $(0,j)$ are taken into account and the structure between $(j,n-1)$ must be a set of independent stems. Therefore, all the outside energy between $[j,n-1]$ is equal to $\mathcal{Z}^m_{\substack{(j,n-1)\\[X,X]}}$, for any $X \in B$. The recursion itself is as follows.

$$\mathcal{Y}^m_{\substack{(i,j)\\[a,b]}} = \begin{cases} \displaystyle\sum_{\substack{a' \in \mathcal{B},\\ \delta_{a',s_i} \leq m}} \mathcal{Y}^{m-\delta_{a',s_i}}_{\substack{(i-1,j)\\[a',b]}} & \text{Elif } S_i = -1 \\ \displaystyle\sum_{\substack{a'b' \in \mathcal{B}^2,\\ \delta_{a'b',s_i s_j} \leq m}} e^{\frac{-(\alpha \mathrm{ES}^{\beta}_{ab \to a'b'} + (1-\alpha)\mathrm{EI}^{\Omega}_{(i,j),a'b'})}{RT}} \mathcal{Y}^{m-\delta_{a'b',s_i s_j}}_{\substack{(i-1,j+1)\\[a',b']}} & \text{Elif } S_i = j \wedge S_{i+1} = j-1 \\ \displaystyle\sum_{\substack{a'b' \in \mathcal{B}^2,\\ \delta_{a'b',s_i s_k} \leq m}} \sum_{m'=0}^{m-\delta_{a'b',s_i s_k}} e^{\frac{-(1-\alpha)\mathrm{EI}^{\Omega}_{(i,k),a'b'}}{RT}} \mathcal{Y}^{m-\delta_{a'b',s_i s_k}-m'}_{\substack{(i-1,k+1)\\[a',b']}} \mathcal{Z}^{m'}_{\substack{(j,k-1)\\[b,b']}} & \text{Elif } S_i = k \geq j \\ \displaystyle\sum_{\substack{a'b' \in \mathcal{B}^2,\\ \delta_{a'b',s_k s_i} \leq m}} \sum_{m'=0}^{m-\delta_{a'b',s_k s_i}} e^{\frac{-(1-\alpha)\mathrm{EI}^{\Omega}_{(k,i),a'b'}}{RT}} \mathcal{Y}^{m-\delta_{a'b',s_k s_i}-m'}_{\substack{(k-1,j)\\[a',b]}} \mathcal{Z}^{m'}_{\substack{(k+1,i-1)\\[a',b']}} & \text{Elif } -1 < S_i = k < i \\ 0 & \text{Else} \end{cases} \tag{4}$$

The five cases can be broked down as follows.

$S_i = -1$**:** If the nucleotide at position i is not paired, then the value is the same as if we decrease the lower interval bound by 1 (i.e. $i-1$), and consider all possible nucleotides a' at position i, correcting the number of mutants in function of δ_{a',s_i}.

$S_i = j$ **and** $S_{i+1} = j-1$**:** If nucleotide i is paired with j and nucleotide $i+1$ is paired with $j-11$, we are in the only case were stacked base pairs can occur. We thus add the energy of the stacking and of the isostericity of the base pair (i,j). What is left to compute is the *outside* value for the interval $[i-1,j+1]$ over all possible nucleotides $a',b' \in B^2$ at positions i and j respectively.

$S_i = k \geq j$**:** If nucleotide i is paired with position $k \geq j$, and is not stacked inside, the only term contributing directly to the energy is the isostericity of the base pair (i,k). We can then consider the outside interval $[i-1,k+1]$ by multiplying it by the the *forward* value of the newly included interval (i.e. $[j,k-1]$), for all possible values $a',b' \in B^2$ for nucleotides at positions i and k respectively.

$-1 < S_i < i$**:** As above but if position i is to pairing with a lower value.

Else: In all other cases, we are in a derivation of the SCFG that does not correspond to the secondary structure S, and we return 0.

3 Inside-Outside

By construction, the partition function over all sequences at exactly m mutations of s can be described in function of the *forward* term as $\underset{[X,X]}{\mathcal{Z}^m_{(0,n-1)}}$, for any nucleotide $X \in B$ or in function of the *backward* term, for any position k such that $S_k = -1$:

$$\underset{[X,X]}{\mathcal{Z}^m_{(0,n-1)}} \equiv \sum_{\substack{a\in\mathcal{B},\\ \delta_{a,s[k]}\leq m}} \underset{[a,a]}{\mathcal{Y}^{m-\delta_{a,s[k]}}_{(k-1,k+1)}}$$

We are now interested in knowing, under our model, the probability that a given position is a given nucleotide. We leverage the *Inside-Outside* construction to immediately obtain the following 3 cases. Given $i \in [0, n-1], x \in B$, and $M \geq 0$ a bound on the number of mutations allowed.

$$\mathbb{P}(s_i = x \mid s, \Omega, S, M) := \begin{cases} \dfrac{\sum_{m=0}^{M} \underset{[x,x]}{\mathcal{Y}^{m-\delta_{x,s_i}}_{(i-1,i+1)}}}{\sum_{m=0}^{M} \underset{[X,X]}{\mathcal{Z}^m_{(0,n-1)}}} & \text{If } S_i = -1 \\ \dfrac{\sum_{m=0}^{M} \sum_{\substack{b\in Bases\\ \delta_{xb,s_is_k}\leq m}} \sum_{m'=0}^{m-\delta_{xb,s_is_k}} e^{\frac{-(1-\alpha)\mathrm{EI}^{\Omega}_{(i,k),xb}}{RT}} \underset{[x,b]}{\mathcal{Y}^{m-\delta_{xb,s_is_k}-m'}_{(i-1,k+1)}} \underset{[x,b]}{\mathcal{Z}^{m'}_{(i+1,k-1)}}}{\sum_{m=0}^{M} \underset{[X,X]}{\mathcal{Z}^m_{(0,n-1)}}} & \text{If } S_i = k > i \\ \dfrac{\sum_{m=0}^{M} \sum_{\substack{b\in Bases\\ \delta_{bx,s_ks_i}\leq m}} \sum_{m'=0}^{m-\delta_{bx,s_ks_i}} e^{\frac{-(1-\alpha)\mathrm{EI}^{\Omega}_{(k,i),bx}}{RT}} \underset{[b,x]}{\mathcal{Y}^{m-\delta_{bx,s_ks_i}-m'}_{(k-1,i+1)}} \underset{[b,x]}{\mathcal{Z}^{m'}_{(k+1,i-1)}}}{\sum_{m=0}^{M} \underset{[X,X]}{\mathcal{Z}^m_{(0,n-1)}}} & \text{If } S_i = k < i \end{cases} \tag{5}$$

In every case, the denominator is the sum of the partitions function of exactly m mutations, for m smaller or equal to our target M. The numerators are divided in the following three cases.

$S_i = -1$**:** If the nucleotide at position i is not paired, we are concerned by the weights over all sequences which have at position i nucleotide x, which is exactly the sum of the values of $\underset{[x,x]}{\mathcal{Y}^{m-\delta_{x,s_i}}_{(i-1,i+1)}}$, for all m between 0 and M.

$S_i = k > i$**:** Since we need to respect the derivation of the secondary structure S, if position i is paired, we must consider the two partition functions. The *outside* of the base pair, and the *inside*, for all possible values for the nucleotide at position k, and all possible distribution of

the mutant positions between the inside and outside of the base pair. We also add the term of isostericity for this specific base pair.

$S_i = k < i$**:** Same as above, but with position i pairing with a lower position.

4 Results

4.1 Implementation

The software was implemented in Python2.7 using the *mpmath* [**?**] library for arbitrary floating point precision. The code at `https://github.com/McGill-CSB/RNApyro` is freely available.

5 Discussion

6 Acknowledgments