A linear inside-outside algorithm for correcting sequencing errors in structured RNA sequences

Vladimir Reinharz, Yann Ponty, Jérôme Waldispühl

A linear inside-outside algorithm for correcting sequencing errors in structured RNA sequences

Vladimir Reinharz[1], Yann Ponty[2*], Jérôme Waldispühl[1*]

[1] School of Computer Science & McGill Centre for Bioinformatics, McGill University, Montréal, Canada H3C 2J7

[2] CNRS/École Polytechnique/Inria AMIB, LIX, Palaiseau, France

Abstract

Analysis of the sequence-structure relationship in RNA molecules are essential to evolutionary studies but also to concrete applications such as error-correction methodologies in sequencing technologies. The prohibitive sizes of the mutational and conformational landscapes combined with the volume of data to proceed require efficient algorithms to compute sequence-structure properties. More specifically, here we aim to calculate which mutations increase the most the likelihood of a sequence to a given structure and RNA family.

In this paper, we introduce `RNApyro`, an efficient linear-time and space inside-outside algorithm that computes exact mutational probabilities under secondary structure and evolutionary constraints given as a multiple sequence alignment with a consensus structure. We develop a scoring scheme combining classical stacking base pair energies to novel isostericity scales, and apply our techniques to correct point-wise errors in 5s rRNA sequences. Our results suggest that `RNApyro` is a promising algorithm to complement existing tools in the NGS error-correction pipeline.

Key words: RNA, mutations, secondary structure

*To whom correspondence should be addressed

1 Introduction

Ribonucleic acids (RNAs) are found in every living organisms and exhibit a broad range of functions, from catalyzing chemical reactions as the RNase P or the group II introns, hybridizing messenger RNA to regulate gene expression, to ribosomal RNA (rRNA) synthesizing proteins. Those functions require specific structures, encoded in their nucleotide sequence. Although the functions, and thus the structures, need to be preserved through various organisms, the sequences can greatly differ from one organism to another. This sequence diversity coupled with the structural conservation is a fundamental asset for evolutionary studies. To this end, algorithms to analyze the relationship between RNA mutants and structures are required.

For half a century, biological molecules have been studied as a proxy to understand evolution [16], and due to their fundamental functions and remarkably conserved structures, rRNAs have always been a prime candidate for phylogenetic studies [8, 9]. In recent years, studies as the *Human Microbiome Project* [13] benefited of new technologies such as the NGS techniques to sequence as many new organisms as possible and extract an unprecedented flow of new information. Nonetheless, these high-throughput techniques typically have high error rates that make their applications to metagenomics (a.k.a. environmental genomics) studies challenging. For instance, pyrosequencing as implemented by Roche's 454 produces may have a error rate raising up to 10%. Because there is no cloning step, resequencing to increase accuracy is not possible and it is therefore vital to disentangle noise from true sequence diversity in this type of data [10]. Errors can be significantly reduced when large multiple sequence alignments with close homologs are available, but in studies of new or not well known organisms, such information is rather sparse. In particular, it is common to is not enough similarity to differentiate between the sequencing errors and the natural polymorphisms that we want to observe, often inflating the diversity estimates [3]. A few techniques have been developed to remedy to this problem [11, 7] but they do not take into account all the available information. It is therefore essential to develop methods that can exploit any type of signal available to correct errors.

In this paper, we introduce `RNApyro`, a novel algorithm to that enable us to calculate precisely mutational probabilities in RNA sequences with a conserved consensus secondary structure. We show how our techniques can exploit the structural information embedded in physics-based energy models, covariance models and isostericity scales to identify and correct point-wise errors in RNA molecules with conserved secondary structure. In particular, we hypothesize that conserved consensus secondary structures combined with sequence profiles provide an information that allow us to identify and fix sequencing errors.

Here, we expand the range of algorithmic techniques previously introduced with `RNAmutants` [15]. Instead of exploring the full conformational landscape and sample mutants, we develop an inside-outside algorithm that enables us to explore the complete mutational landscape with a *fixed* secondary structure and to calculate exactly mutational probability values. In addition to a gain into the numerical precision, this strategy allows us to drastically reduce the computational complexity ($\mathcal{O}(n^3 \cdot m^2)$ for the original version of `RNAmutants` to $\mathcal{O}(n \cdot m^2)$ for `RNApyro`, where n is the size of the sequence and m the number of mutations).

We design a new scoring scheme combining nearest-neighbor models [14] to isostericity metrics [12]. Classical approaches use a Boltzmann distribution whose weights are estimated using a nearest-neighbour energy model [14]. However, the latter only accounts for canonical and wobble, base pairs. As was shown by Leontis and Westhof [5], the diversity of base pairs observed in tertiary structures is much larger, albeit their energetic contribution remains unknown. To quantify geometrical discrepancies, an isostericity distance has been designed [12], increasing as two base pairs geometrically differ from each other in space. Therefore, we incorporate these scores in the

Boltzmann weights used by `RNApyro`.

We illustrate and benchmark our techniques for point-wise error corrections on the 5S ribosomal RNA. We choose the latter since it has been extensively used for phylogenetic reconstructions [1] and its sequence has been recovered for over 712 species (in the Rfam seed alignment with id `RF00001`). Using a leave one out strategy, we perform random distributed mutations on a sequence. While our methodology is restricted to the correction of point-wise error in structured regions (i.e. with base pairs), we show that `RNApyro` can successfully extract a signal that can be used to reconstruct the original sequence with an excellent accuracy. This suggests that `RNApyro` is a promising algorithm to complement existing tools in the NGS error-correction pipeline.

The algorithm and the scoring scheme are presented in Sec. 2. Details of the implementation and benchmarks are in Sec. 3. Finally, we discuss future developments and applications in Sec. 4.

2 Methods

We introduce a probabilistic model, which aims at capturing both the stability of the folded RNA and its ability to adopt a predefined 3D conformation. To that purpose, a Boltzmann weighted distribution is used, based on a pseudo-energy function $E(\cdot)$ which includes contributions for both the free-energy and its putative isostericity towards a multiple sequence alignment. In this model, the probability that the nucleotide at a given position needs to be mutated (i.e. corresponds to a sequencing error) can be computed using a variant of the *Inside-Outside algorithm* [4].

2.1 Probabilistic model

Let Ω be an gap-free RNA alignment sequence, S its associated secondary structure, then any sequence s is assigned a probability proportional to its Boltzmann factor

$$\mathcal{B}(s) = e^{\frac{-E(s)}{RT}}, \qquad \text{with} \qquad E(s) := \alpha \cdot \mathrm{ES}(s, S) + (1 - \alpha) \cdot \mathrm{EI}(s, S, \Omega),$$

where R is the Boltzmann constant, T the temperature in Kelvin, $\mathrm{ES}(s)$ and $\mathrm{EI}(s, S, \Omega)$ are the free-energy and isostericity contributions respectively (further described below), and $\alpha \in [0, 1]$ is an arbitrary parameter that sets the relative weight for both contributions.

2.1.1 Energy contribution

The free-energy contribution in our pseudo-energy model corresponds to an additive stacking-pairs model, taking values from the Turner 2004 model retrieved from the NNDB [14]. Given a candidate sequence s for a secondary structure S, the free-energy of S on s is given by

$$\mathrm{ES}(s, S) = \sum_{\substack{(i,j)\to(i',j')\in S \\ \text{stacking pairs}}} \mathrm{ES}^{\beta}_{s_i s_j \to s_{i'} s_{j'}}$$

where $\mathrm{ES}^{\beta}_{ab\to a'b'}$ is set to 0 if $ab = \varnothing$ (no base-pair to stack onto), the tabulated free-energy of stacking pairs $(ab)/(a'b')$ in the Turner model if available, or $\beta \in [0, \infty]$ for non-Watson-Crick/Wobble entries (i.e. not in $\{\mathsf{GU}, \mathsf{UG}, \mathsf{CG}, \mathsf{GC}, \mathsf{AU}$ or $\mathsf{UA}\}$). This latter parameter allows to choose whether to simply penalize invalid base pairs, or forbid them altogether ($\beta = \infty$). The loss of precision due to this simplification of the Turner model remains reasonable since the targeted secondary structure is fixed (e.g. multiloops do not account for base-specific contributions). Furthermore, it greatly eases the design and implementation of dynamic-programming equations.

2.1.2 Isostericity contribution

The concept of isostericity score [12] is based on the geometric discrepancy (superimposability) of two base-pairs, using individual additive contributions computed by Stombaugh *et al* [12]. Let s be a candidate sequence for a secondary structure S, given in the context of a gap-free RNA alignment Ω, we define the contribution of the isostericity to the pseudo-energy as

$$\mathrm{ES}(s, S, \Omega) = \sum_{\substack{(i,j)\in S \\ \text{pairs}}} \mathrm{EI}^{\Omega}_{(i,j),s_i s_j}, \qquad \text{where} \qquad \mathrm{EI}^{\Omega}_{(i,j),ab} := \frac{\sum_{s'\in\Omega} \mathrm{ISO}((s'_i, s'_j), (a, b))}{|\Omega|}$$

is the average isostericity of a base-pair in the candidate sequence, compared with the reference alignment. The ISO function uses the Watson-Crick/Watson-Crick cis isostericity matrix computed by Stombaugh *et al* [12]. Isostericity scores range between 0 and 9.7, with 0 being assigned to a perfect isostericity, and a penalty of 10 is used for missing entries. The isostericity contribution will favor exponentially sequences that are likely to adopt a similar local conformation as the sequences contained in the alignment.

2.2 Mutational profile of sequences

Let s be an RNA sequence, S a reference structure, and $m \geq 0$ a desired number of mutations. We are interested in the probability that a given position contains a specific nucleotide, over all sequences having at most m mutations from s (formally $\mathbb{P}(s_i = x \mid s, \Omega, S, m)$). We define a variant of the *Inside-Outside algorithm* [4], allowing us to obtain the desired probability, the two functions $\mathcal{Z}^*_*$ and $\mathcal{Y}^*_*$.

The former, defined in Equations (2) and (3), is analogous to the *inside* algorithm. It is the partition function, i.e. the sum of Boltzmann factors, over all sequences within $[i, j]$, knowing that position $i-1$ is composed of nucleotide a (resp. $j+1$ is b), within m mutations of s. The latter, defined by Equations (4) and (5), computes the *outside* algorithm, i.e. the partition function over sequences within m mutations of s, restricted to two intervals $[0, i] \cup [j, n-1]$, and knowing that position $i+1$ is composed of a (resp. $j-1$ is b). A suitable combination of these terms, given in Equation (7), gives the total weight, and in turn the probability, of seeing a specific base at a given position.

2.2.1 Definitions

Let $B := \{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{U}\}$ be 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 us then remind a Hamming distance function $\delta : B^* \times B^* \to \mathbb{N}^+$, which takes two sequences s' and s'' as input, $|s'| = |s''|$, and returns the number of differing positions. Finally, let us denote by $E^{\Omega,\beta}_{(i,j),ab\to ab'}$ the local contribution of a base-pair (i, j) to the pseudo-energy, such that

$$E^{\Omega,\beta}_{(i,j),ab\to a'b'} = \alpha \cdot \mathrm{ES}^{\beta}_{ab\to a'b'} + (1-\alpha) \cdot \mathrm{EI}^{\Omega}_{(i,j),a'b'}. \tag{1}$$

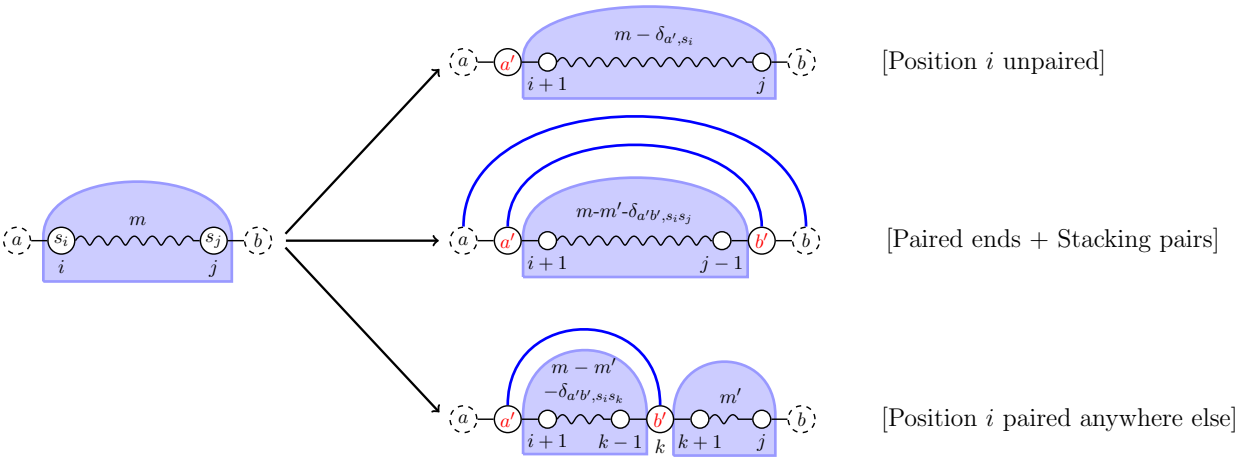


Figure 1: Principle of the inside computation (aka partition function). Any sequence with m mutations over an interval $[i, j]$ can be decomposed as a sequence over $[i + 1, j]$ preceded by a, possibly mutated, base at i (Unpaired case), a sequence over $[i + 1, j - 1]$ surrounded by some base-pair (Stacking-pair case), or as two sequences over $[i + 1, k - 1]$ and $[k + 1, j]$, completed by some base-pair (General base-pairing case). In each case, one has to investigate any possible ways to distribute mutations between the different sequences and locally instanciated bases.

2.2.2 Inside

The *Inside* function $\mathcal{Z}^{m}_{\substack{(i,j)\\[a,b]}}$ is the partition function, i.e. the sum of Boltzmann factors, over all sequences in the interval $[i, j]$, at distance m of $s_{[i,j]}$, and having flanking nucleotides a and b (at positions $i - 1$ and $j + 1$ respectively). Such terms can be defined by recurrence, for which the following initial conditions holds:

$$\forall i \in [0, n-1] : \mathcal{Z}^{m}_{\substack{(i+1,i)\\[a,b]}} = \begin{cases} 1 & \text{If } m = 0 \\ 0 & \text{Otherwise.} \end{cases} \tag{2}$$

In other words, the set of sequences at distance m of the empty sequence is either empty if $m > 0$, or restricted to the empty sequence, having energy 0, if $m = 0$. Since the energetic terms only depend on base pairs, they are not involved in the initial conditions.

The main recursion is composed of four terms:

$$\mathcal{Z}^{m}_{\substack{(i,j)\\[a,b]}} := \begin{cases} \displaystyle\sum_{\substack{a' \in \mathcal{B},\\ \delta_{a',s_i} \leq m}} \mathcal{Z}^{m-\delta_{a',s_i}}_{\substack{(i+1,j)\\[a',b]}} & \text{If } S_i = -1 \\ \displaystyle\sum_{\substack{a',b' \in \mathcal{B}^2,\\ \delta_{a'b',s_is_j} \leq m}} e^{\frac{-E^{\Omega,\beta}_{(i,j),ab\to a'b'}}{RT}} \cdot \mathcal{Z}^{m-\delta_{a'b',s_is_j}}_{\substack{(i+1,j-1)\\[a',b']}} & \text{Elif } S_i = j \wedge S_{i-1} = j+1 \\ \displaystyle\sum_{\substack{a',b' \in \mathcal{B}^2,\\ \delta_{a'b',s_is_k} \leq m}} \sum_{m'=0}^{m-\delta_{a'b',s_is_k}} e^{\frac{-E^{\Omega,\beta}_{(i,k),\varnothing\to a'b'}}{RT}} \cdot \mathcal{Z}^{m-\delta_{a'b',s_is_k}-m'}_{\substack{(i+1,k-1)\\[a',b']}} \cdot \mathcal{Z}^{m'}_{\substack{(k+1,j)\\[b',b]}} & \text{Elif } S_i = k \wedge i < k \leq j \\ 0 & \text{Otherwise} \end{cases} \tag{3}$$

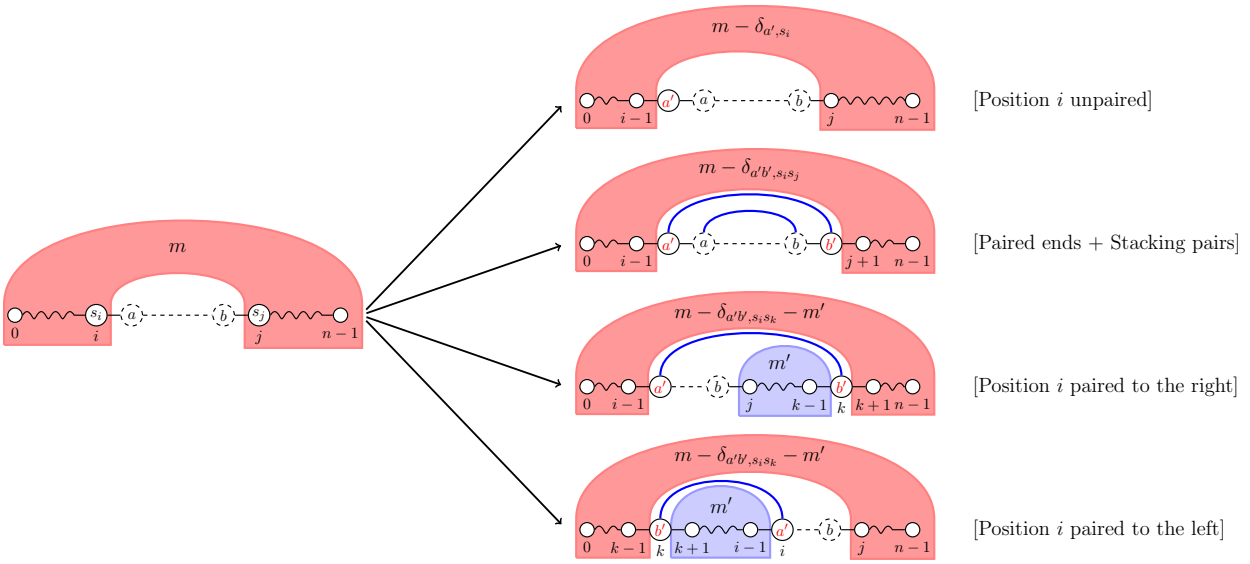


Figure 2: Principle of the outside computation. Note that the outside algorithm uses intermediate results from the inside algorithm, therefore its efficient implementation requires an implementation of the inside computation.

The cases can be broken down as follows:

$S_i = -1$**:** If the nucleotide at position i is unpaired, then any sequence consists in a, possibly mutated, nucleotide a' at position i, followed by a sequence over $[i+1, j]$ having either $m - \delta_{a',s_i}$, accounting for a possible mutation at position i, and having flanking nucleotides a' and b.

$S_i = j$ **and** $S_{i-1} = j+1$**:** Any sequence generated in $[i, j]$ consists of two, possibly mutated, nucleotides a' and b', flanking a sequence over $[i+1, j-1]$ having distance $m - \delta_{a'b',s_i s_j}$ (to avoid exceeding the targeted distance m). Since positions i and $i-1$ are paired with j and $j+1$ respectively, then a stacking energy contribution is added.

$S_i = k$ **and** $i < k \leq j$**:** If position i is paired and not involved in a stacking, then the only term contributing directly to the energy is the isostericity of the base pair (i, k). Any sequence on $[i, j]$ consists of two nucleotide a' and b' at positions i and k respectively, flanking a sequence over interval $[i+1, k-1]$ and preceding a (possibly empty) sequence interval $[k+1, j]$. Since the number of mutations sum to m over the whole sequence must , then a parameter m' is introduced to distribute the remaining mutations between the two sequences.

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

2.2.3 Outside

The *Outside* function, $\mathcal{Y}$, is the partition function considering only the contributions of 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:

$$\underset{[X,X]}{\mathcal{Y}^m_{(-1,j)}} := \underset{[X,X]}{\mathcal{Z}^m_{(j,n-1)}} \tag{4}$$

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 $\underset{[X,X]}{\mathcal{Z}^m_{(j,n-1)}}$, for any $X \in B$. The recursion itself is as follows.

$$\underset{[a,b]}{\mathcal{Y}^m_{(i,j)}} = \begin{cases} \displaystyle\sum_{\substack{a' \in \mathcal{B}, \\ \delta_{a',s_i} \leq m}} \underset{[a',b]}{\mathcal{Y}^{m-\delta_{a',s_i}}_{(i-1,j)}} & \text{Elif } S_i = -1 \\ \displaystyle\sum_{\substack{a'b' \in \mathcal{B}^2, \\ \delta_{a'b',s_i s_j} \leq m}} e^{\frac{-E^{\Omega,\beta}_{(i,j),ab \to a'b'}}{RT}} \cdot \underset{[a',b']}{\mathcal{Y}^{m-\delta_{a'b',s_i s_j}}_{(i-1,j+1)}} & \text{Elif } S_i = j \wedge S_{i+1} = j-1 \\ \displaystyle\sum_{\substack{a'b' \in \mathcal{B}^2, \\ \delta_{a'b',s_i s_k} \leq m}} \sum_{m'=0}^{m-\delta_{a'b',s_i s_k}} e^{\frac{-E^{\Omega,\beta}_{(i,k),\varnothing \to a'b'}}{RT}} \cdot \underset{[a',b']}{\mathcal{Y}^{m-\delta_{a'b',s_i s_k}-m'}_{(i-1,k+1)}} \cdot \underset{[b,b']}{\mathcal{Z}^{m'}_{(j,k-1)}} & \text{Elif } S_i = k \geq j \\ \displaystyle\sum_{\substack{a'b' \in \mathcal{B}^2, \\ \delta_{a'b',s_k s_i} \leq m}} \sum_{m'=0}^{m-\delta_{a'b',s_k s_i}} e^{\frac{-E^{\Omega,\beta}_{(k,i),\varnothing \to a'b'}}{RT}} \cdot \underset{[a',b]}{\mathcal{Y}^{m-\delta_{a'b',s_k s_i}-m'}_{(k-1,j)}} \cdot \underset{[a',b']}{\mathcal{Z}^{m'}_{(k+1,i-1)}} & \text{Elif } -1 < S_i = k < i \\ 0 & \text{Otherwise} \end{cases} \tag{5}$$

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 *inside* 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.

2.2.4 Combining Inside and Outside values into point-wise mutations probabilities

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

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

We are now left to compute 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 allowed mutations.

$$\mathbb{P}(s_i = x \mid s, \Omega, S, M) := \frac{\mathcal{W}(i, x, s, \Omega, S, M)}{\sum_{m=0}^{M} \underset{[X,X]}{\mathcal{Z}^{m}_{(0,n-1)}}} \tag{6}$$

$$\mathcal{W}(i, x, s, \Omega, S, M) = \begin{cases} \sum_{m=0}^{M} \underset{[x,x]}{\mathcal{Y}^{m-\delta_{x,s_i}}_{(i-1,i+1)}} & \text{If } S_i = -1 \\ \displaystyle\sum_{m=0}^{M} \sum_{\substack{b \in B \\ \delta_{xb,s_i s_k} \leq m}} \sum_{m'=0}^{m-\delta_{xb,s_i s_k}} e^{\frac{-E^{\Omega,\beta}_{(i,k),\varnothing \to xb}}{RT}} \cdot \underset{[x,b]}{\mathcal{Y}^{m-\delta_{xb,s_i s_k}-m'}_{(i-1,k+1)}} \cdot \underset{[x,b]}{\mathcal{Z}^{m'}_{(i+1,k-1)}} & \text{If } S_i = k > i \\ \displaystyle\sum_{m=0}^{M} \sum_{\substack{b \in B \\ \delta_{bx,s_k s_i} \leq m}} \sum_{m'=0}^{m-\delta_{bx,s_k s_i}} e^{\frac{-E^{\Omega,\beta}_{(k,i),\varnothing \to bx}}{RT}} \cdot \underset{[b,x]}{\mathcal{Y}^{m-\delta_{bx,s_k s_i}-m'}_{(k-1,i+1)}} \cdot \underset{[b,x]}{\mathcal{Z}^{m'}_{(k+1,i-1)}} & \text{If } S_i = k < i \end{cases} \tag{7}$$

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.

2.3 Complexity considerations

Equations (3) and (5) can be computed using dynamic programming. Namely, the $\mathcal{Z}^{*}_{*}$ and $\mathcal{Y}^{*}_{*}$ terms are computed starting from smaller values of m and interval lengths, memorizing the results as they become available to ensure constant-time access during later stages of the computation. Furthermore, energy terms $E(\cdot)$ can be accessed in constant time thanks to a simple precomputation

$\|s\|$	Number of mutations		
	6	12	24
100	35s	238s	1023s
300	135s	594s	2460s
	25		50
500	5400s		21003s

Table 1: Time to compute all probabilities. The first column indicates the length and the column indexes indicate the number of mutations. α is set at 0.5, β to 1.5 and $|\Omega| = 44$.

(not described) of the isostericity contributions in $\Theta(n \cdot |\Omega|)$. Computing any given term therefore requires $\Theta(m)$ operations.

In principle, $\Theta(m \cdot n^2)$ terms, identified by (m, i, j) triplets, should be computed. However, a close inspection of the recurrences reveals that the computation can be safely restricted to a subset of intervals (i, j). For instance, the inside algorithm only requires computing intervals $[i, j]$ that do not break any base-pair, and whose next position $j + 1$ is either past the end of the sequence, or is base-paired prior to i. Similar constraints hold for the outside computation, resulting in a drastic limitation of the combinatorics of required computations, dropping from $\Theta(n^2)$ to $\Theta(n)$ the number of terms that need to be computed and stored. Consequently the overall complexity of the algorithm is $\Theta(n \cdot (|\Omega| + m^2))$ arithmetic operations and $\Theta(n \cdot (|\Omega| + m))$ memory.

3 Results

3.1 Implementation

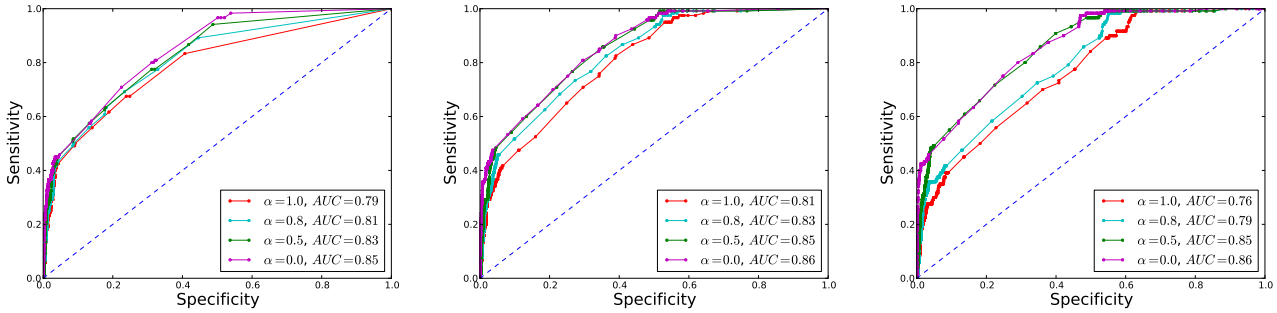
The software was implemented in Python2.7 using the *mpmath* [2] library for arbitrary floating point precision. The source code is freely available at `https://github.com/McGill-CSB/RNApyro`.

The time benchmarks were done on a MacMini 2010, 2.3GHz dual-core Intel Core i5, 8GB of RAM. Since applications of `RNApyro` implies a need for efficiency and scalability, we present in Table 1 typical runtimes required to compute the probabilities for every nucleotide at every positions for a vast set of parameters. For those tests, both the sequences and the target secondary structure were generated at random.

3.2 Error correction in 5s rRNA

To illustrate the potential of our algorithm, we applied our techniques to identify and correct point-wise errors in RNA sequences with conserved secondary structures. More precisely, we used `RNApyro` to reconstruct 5s rRNA sequences with randomly distributed mutations. This experiment has been designed to suggest further applications to error-corrections in pyrosequencing data.

We built our data set from the 5S rRNA multiple sequence alignment (MSA) available in the Rfam Database 11.0 (Rfam id: `RF00001`). Since our software does not currently implement gaps (mainly because scoring indels is a challenging issue that cannot be fully addressed in this work), we clustered together the sequences with identical gap locations. From the 54 MSAs without gap produced, we selected the biggest MSA which contains 130 sequences (out of 712 in the original Rfam MSA). Then, in order to avoid overfitting, we used `cd-hit` [6] to remove sequences with more than 80% of sequence similarity. This operation resulted in a data set of 45 sequences.



(a) 6 mutations (b) 12 mutations (c) 24 mutations

Figure 3: Performance of error-correction. Subfigures accuracy with under-estimated error rate (6 mutations), exact estimates (12 mutations) and over estimates (24 mutations). We also analyze the impact of the parameter α distributing the weights of stacking pair energies vs isostericity scores and use values ranging of $\alpha = \{0, 0.5, 0.8, 1.0\}$. The AUC is indicated in the legend of the figures. Each individual ROC curve represent the average performance over the 10 experiments.

We designed our benchmark using a leave-one-out strategy. We randomly picked one sequence from our data set and performed 12 random mutations. Our sequences have 119 nucleotides, thus the number of mutations corresponds to an error-rate of 10%. We repeated this operation 10 times. The value of β has been set to 1.5 (larger values gave similar results). To estimate the impact of the distribution of the weights between the energy term and isostericity score, we used 4 different values of $\alpha = 0, 0.5, 0.8, 1.0$. Similarly, we also investigated the impact of an under- and over- estimate of the number of errors, and use values equal to 50% (6 mutations) and 200% (24 mutations) of the exact number of errors (i.e. 12).

To evaluate our method, we computed a ROC curve representing the performance of a classifier based on the mutational probabilities computed by `RNApyro`. More specifically, we fixed a threshold $\lambda \in [0, 1]$, and predicted an error at position i in sequence ω if and only if the probability $P(i, n)$ of a nucleotide $n \in \{A, C, G, U\}$ exceeds this threshold. To correct the errors we used the set of nucleotides with a probability higher than this threshold is $C(i) = \{n \mid n \in \{A, C, G, U\}$ and $P(i, n) > \lambda$ and $n \neq \omega[i]\}$, where $\omega[i]$ is the nucleotide at position i in the input sequence. We note that for the lowest thresholds multiple nucleotides can be available in $C(i)$ to correct the sequence. Here, we remind that our aim is to estimate the potential of error-correction of `RNApyro` and not to develop an error-correction pipe-line, which will be the subject of further studies. Finally, we progressively varied λ between 0 and 1 to calculate the ROC curve and the area under the curve (AUC). We report our results in Figure 3.

Our data demonstrate that our algorithm exhibits interesting performance for error-correction applications. First, the AUC values (up to 0.86) indicate that a signal has been successfully extracted. This result has been achieved with errors in loop regions (i.e. without base pairing information) and thus suggests that correction rates in structured regions (i.e. base paired regions) could be even higher. Next, the optimal values of α tend to be close to 0.0. This finding suggests that at this point the information issued from the stacking energy is currently modest. However, specific examples showed improved performance using this energy term. Further studies must be conducted to understand how to make the best use it. Finally, our algorithm seems robust to the number of mutations considered. Indeed, good AUC values are achieved with low estimates of the number of errors in the sequence (c.f. 50% of the errors in Fig. 3a) and with large values (c.f. 200% of the errors in Fig. 3c). It is worth noting that scoring scheme with larger weight on the isostericity

scores (low α values) seem more robust to under- and over-estimate of the number of errors.

4 Conclusion

In this article we presented a new and efficient way of exploring the mutational landscape of an RNA under structural constraints, and apply our techniques to identify and fix sequencing errors. In addition, we introduce a new scoring scheme combining the nearest-neighbour energy model to new isostericity matrices in order to account for geometrical discrepancies occurring during base pair replacements. The algorithm runs in $\Theta(n \cdot (|\Omega| + m^2))$ time and $\Theta(n \cdot (|\Omega| + m))$ memory, where n is the length of the RNA, m the number of mutations and Ω the size of the multiple sequence alignment.

By combining into `RNApyro` these two approaches, the mutational landscape exploration and the pseudo energy model, we created a tool predicting the positions yielding point-wise sequencing error and correcting them. We validated our model with the 5s rRNA, as presented in Sec. 3. We observed that the models with larger weights on the isostericity seems to hold a higher accuracy on the estimation of errors. This indicates that an extractable signal is contained in the isostericity. Importantly, the implementation is fast enough for practical applications.

We must recall that our approach is restricted to the correction of point-wise error in structured regions (i.e. base paired nucleotides). Nonetheless it should supplement well existing tools, by using previously discarded information holding, as shown, a strong signal.

Further research, given the potential of error-correction of `RNApyro`, will evaluate its impact over large datasets with different existing NGS error-correction pipe-line.

5 Acknowledgments

The authors would like to thank Rob Knight for his suggestions and comments.

References

[1] H Hori and S Osawa. Origin and Evolution of Organisms as Deduced from 5s Ribosomal RNA Sequences. *Molecular biology and evolution*, 4(5):445–472, 1987.

[2] Fredrik Johansson et al. *mpmath: a Python library for arbitrary-precision floating-point arithmetic (version 0.14)*, February 2010. `http://code.google.com/p/mpmath/`.

[3] Victor Kunin, Anna Engelbrektson, Howard Ochman, and Philip Hugenholtz. Wrinkles in the rare biosphere: pyrosequencing errors can lead to artificial inflation of diversity estimates. *Environmental microbiology*, 12(1):118–23, January 2010. ISSN 1462-2920. doi: 10.1111/j.1462-2920.2009.02051.x.

[4] K Lari and SJ Young. The estimation of stochastic context-free grammars using the Inside-Outside algorithm. *Computer Speech & Language*, 4(1):35–56, January 1990. ISSN 08852308. doi: 10.1016/0885-2308(90)90022-X.

[5] N B Leontis and E Westhof. Geometric nomenclature and classification of RNA base pairs. *RNA (New York, N.Y.)*, 7(4):499–512, April 2001.

[6] Weizhong Li and Adam Godzik. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22(13):1658–9, Jul 2006. doi: 10.1093/bioinformatics/btl158.

[7] Paul Medvedev, Eric Scott, Boyko Kakaradov, and Pavel Pevzner. Error correction of high-throughput sequencing datasets with non-uniform coverage. *Bioinformatics (Oxford, England)*, 27(13):i137–41, July 2011. ISSN 1367-4811. doi: 10.1093/bioinformatics/btr208.

[8] G J Olsen, D J Lane, S J Giovannoni, N R Pace, and D a Stahl. Microbial ecology and evolution: a ribosomal RNA approach. *Annual review of microbiology*, 40:337–65, January 1986. ISSN 0066-4227. doi: 10.1146/annurev.mi.40.100186.002005.

[9] GJ Olsen and CR Woese. Ribosomal RNA: a key to phylogeny. *The FASEB journal*, 7(1): 113–123, 1993.

[10] Christopher Quince, Anders Lanzén, Thomas P Curtis, Russell J Davenport, Neil Hall, Ian M Head, L Fiona Read, and William T Sloan. Accurate determination of microbial diversity from 454 pyrosequencing data. *Nat Methods*, 6(9):639–41, Sep 2009. doi: 10.1038/nmeth.1361.

[11] AR Quinlan, DA Stewart, MP Strömberg, and GT Marth. Pyrobayes: an improved base caller for SNP discovery in pyrosequences. *Nature methods*, 5(2):179–81, February 2008. ISSN 1548-7105. doi: 10.1038/nmeth.1172.

[12] Jesse Stombaugh, Craig L Zirbel, Eric Westhof, and Neocles B Leontis. Frequency and isostericity of RNA base pairs. *Nucleic acids research*, 37(7):2294–312, April 2009. ISSN 1362-4962. doi: 10.1093/nar/gkp011.

[13] Peter J Turnbaugh, Ruth E Ley, Micah Hamady, Claire M Fraser-Liggett, Rob Knight, and Jeffrey I Gordon. The Human Microbiome Project. *Nature*, 449(7164):804–10, October 2007. ISSN 1476-4687. doi: 10.1038/nature06244.

[14] Douglas H Turner and David H Mathews. NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. *Nucleic acids research*, 38(Database issue):D280–2, January 2010. ISSN 1362-4962. doi: 10.1093/nar/gkp892.

[15] Jérôme Waldispühl, Srinivas Devadas, Bonnie Berger, and Peter Clote. Efficient Algorithms for Probing the RNA Mutation Landscape. *PLoS Computational Biology*, 4(8):e1000124, 2008. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1000124.

[16] Emile Zuckerkandl and Linus Pauling. Molecules as documents of evolutionary history. *Journal of Theoretical Biology*, 8(2):357–366, March 1965. ISSN 00225193. doi: 10.1016/0022-5193(65)90083-4.