JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 20, Number 11, 2013

Pp. 905–919
DOI: 10.1089/cmb.2013.0085

Using Structural and Evolutionary Information to Detect and Correct Pyrosequencing Errors in Noncoding RNAs

VLADIMIR REINHARZ,[1] YANN PONTY,[2] and JÉRÔME WALDISPÜHL[1]

ABSTRACT

The analysis of the sequence–structure relationship in RNA molecules is not only essential for evolutionary studies but also for concrete applications such as error-correction in next generation sequencing (NGS) technologies. The prohibitive sizes of the mutational and conformational landscapes, combined with the volume of data to process, require efficient algorithms to compute sequence–structure properties. In this article, we address the correction of NGS errors by calculating which mutations most increase the likelihood of a sequence to a given structure and RNA family. 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 scores and apply our techniques to correct pointwise errors in 5s and 16s rRNA sequences. Our results suggest that RNApyro is a promising algorithm to complement existing tools in the NGS error-correction pipeline.

Key words: mutations, RNA, secondary structure.

1. INTRODUCTION

Ribonucleic acids (RNAs) are found in every living organism and exhibit a broad range of functions, ranging from catalyzing chemical reactions, as the RNase P or the group II introns, hybridizing messenger RNA to regulate gene expressions, 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 proxies to understanding evolution (Zuckerkandl and Pauling, 1965), and because of their fundamental functions and remarkably conserved structures, rRNAs have always been a prime candidate for phylogenetic studies (Olsen et al., 1986; Olsen

[1]School of Computer Science, McGill University, Montreal, Canada.
[2]Laboratoire d'informatique, École Polytechnique, Palaiseau, France.

and Woese, 1993). In recent years, studies such as the Human Microbiome Project (Turnbaugh et al., 2007) benefited from 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 (aka environmental genomics) studies challenging. For instance, pyrosequencing as implemented by Roche's 454 may have an 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 (Quince et al., 2009). 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 that there is not enough similarity to differentiate between the sequencing errors and the natural polymorphisms that we want to observe, often leading to artificially inflated diversity estimates (Kunin et al., 2010). A few techniques have been developed to remedy to this problem (Quinlan et al., 2008; Medvedev et al., 2011), 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 article, we introduce `RNApyro`, a novel algorithm that enables 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 pointwise errors in RNA molecules with conserved secondary structure. In particular, we hypothesize that conserved consensus secondary structures combined with sequence profiles provide information that allows us to identify and fix sequencing errors.

Here, we expand the range of algorithmic techniques previously introduced with the `RNAmutants` software (Waldispühl et al., 2008; Waldispühl and Ponty, 2011). 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 in 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 maximal number of mutations).

We design a new scoring scheme combining nearest-neighbor models (Turner and Mathews, 2010) to isostericity metrics (Stombaugh et al., 2009). Classical approaches use a Boltzmann distribution whose weights are estimated using a nearest-neighbor energy model (Turner and Mathews, 2010). However, the latter only accounts for canonical and wobble base pairs. As was shown by Leontis and Westhof (2001), 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 (Stombaugh et al., 2009), 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 pointwise error corrections on the 16S and 5S ribosomal RNAs. We choose the latter since it has been extensively used for phylogenetic reconstructions (Hori and Osawa, 1987), 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 pointwise 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 Section 2. Details of the implementation and benchmarks are in Section 3. Finally, we discuss future developments and applications in Section 4.

2. METHODS

We introduce a probabilistic model, which aims at capturing both the stability of the folded RNA and its affinity toward a predefined 3D conformation. To that purpose, a Boltzmann weighted distribution is assumed, based on a pseudo-energy function $E(\cdot)$, which includes contributions for both the free energy and its putative isostericity toward 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 (Lari and Young, 1990) in time, which scales linearly with the sequence length.

Definitions. Let $\mathcal{B} := \{A, C, G, U\}$ be the set of nucleotides. Given an RNA sequence $s \in \mathcal{B}^n$, let s_i be the nucleotide at position i. Let Ω be a set of ungapped RNA sequences of length n. S is a secondary structure without pseudoknots, denoted by a dot-parenthesis string (well-parenthesized expression with dots). In any such expression, matched parentheses induce an unambiguous set of corresponding positions, associated with base-pairing positions, mediated by hydrogen bonds. It follows that 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 finally denote by $\delta : \mathcal{B}^* \times \mathcal{B}^* \to \mathbb{N}^+$ the Hamming distance, that is, the number of differing positions between two sequences s' and s'' such that $|s'| = |s''|$.

2.1. Probabilistic model

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

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

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.

Energy Contribution. The free-energy contribution in our pseudo-energy model corresponds to an additive stacking-pairs model, using values from the Turner 2004 model retrieved from the NNDB (Turner and Mathews, 2010). 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., neither GU, UG, CG, GC, AU, nor UA). This latter parameter allows us 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. For instance, multiloops do not consider base-specific contributions, and therefore their consideration would constitute a criterion for preferring a sequence over another. Furthermore, it greatly eases the design and implementation of dynamic-programming equations.

Isostericity Contribution. The concept of the isostericity score is based on the geometric discrepancy (superimposability) of two base pairs, using individual additive contributions computed by Stombaugh *et al.* (2009). 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 isostericity contribution 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.* (2009). Isostericity scores range between 0 and 9.7, 0 corresponding to a perfect isostericity, and a penalty of 10 is used for missing entries. The isostericity contribution exponentially favors sequences that are likely to adopt a similar local conformation as the sequences contained in the alignment.

Combining contributions. Let us remark that any of the individual contributions can be associated with (a subset of) the base pairs occurring in the structure, possibly complemented, in the case of stacking pairs, with the knowledge of flanking base-pairing nucleotides. Dropping the implicit dependency on Ω and β, let us denote by $E^{(i,j)}_{xy\to a'b'}$ the local contribution of a base pair (i,j) of nucleotides (a', b'), surrounded by a stacking pair (x, y) (or $\varnothing$ otherwise), to the pseudo-energy:

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

2.2. Computing the 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, let $\mathcal{D}_{s,M} = \{s' | \delta(s, s') \leq M\}$ be the set of admissible sequences, one aims at computing the probabilities

$$\mathbb{P}(s_i = x \mid s, \Omega, S, M) = \frac{\sum_{\substack{s'\in\mathcal{D}_{s,M} \\ \text{s.t.}\, s'_i = x}} B(s')}{\sum_{s''\in\mathcal{D}_{s,M}} B(s'')} \tag{2}$$

Clearly, the number of sequences in $\mathcal{D}_{s,M}$ grows exponentially with the sequence length; therefore, one cannot realistically rely on an exhaustive enumeration to compute the mutational profile. To work around this issue, we propose a linear-time variant of the Inside—Outside algorithm (Lari and Young, 1990) to compute this probability, based on two sets of dynamic programming equations.

The former, defined in Equations (3) and (4)–(6), is analogous to an *inside* computation: Considering a given substructure of the input structure, it computes the accumulated contributions of any possible sequences that have suitable Hamming distance within the interval. It is therefore similar to a partition function, that is, 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 (7) and (8)–(11), computes the *outside* algorithm, the partition function over sequences within m mutations of s outside the interval (restricted to two intervals $[0, i] \cup [j, n\text{-}\ 1]$), knowing that flanking inner positions $(i+1, j-1)$ contain nucleotides a and b respectively. A suitable combination of these terms, computed as shown in Equations (13)–(15), gives the total weight of every possible sequence that supports a given base-pair (or unpaired position) and, in turn, the probability of seeing a specific base at a given position.

Inside computation. The *inside* function $\mathcal{Z}^{x,y}_{S,m}$ is simply the partition function, that is, the sum of Boltzmann factors over all sequences for a substructure S (implicitly attached to an interval $[i, j]$ of the sequence), featuring m mutations/errors compared to s, and having flanking nucleotides x and y. Such terms can be computed recursively, using the following equation for the initial case:

$$\forall x, y \in \mathcal{B}\times\mathcal{B}, m \in [0, M] : \mathcal{Z}^{x,y}_{\varepsilon,m} = \begin{cases} 1 & \text{If } m = 0 \\ 0 & \text{Otherwise.} \end{cases} \tag{3}$$

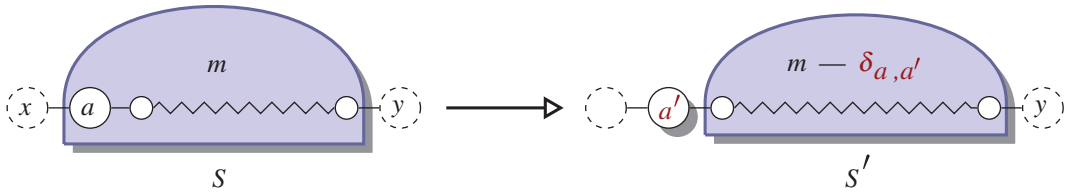
In other words, either there is no sequence at distance $m > 0$ of the empty sequence, or the only allowed sequence is the empty sequence ($m = 0$), having energy 0. Since the contributions only depend on base pairs, they do not appear in the initial conditions.

The main recursion considers a general structure S, flanked by two nucleotides (outside the region of interest) x and y, respectively, on the $5'$ and $3'$ end of the sequence. As illustrated by Figure 1, it is computed, for each subinterval $[i, j]$, by considering one of the three following cases, dependent on the base-pairing status and context of the leftmost position in the sequence/structure:

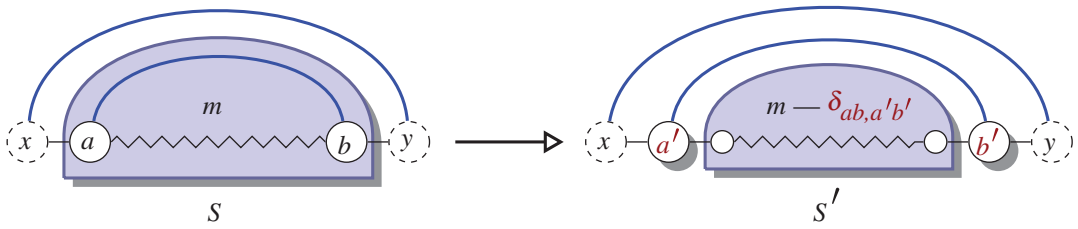
- **Case 1: Unpaired leftmost position.** If the first position is unpaired in the structure, then S can be further decomposed using a dot-parenthesis notation, as $S = \bullet S'$. Let $a \in B$ be the nucleotide found at the leftmost position in the initial structure, then one has:

$$\mathcal{Z}^{x,y}_{S,m} = \sum_{\substack{a'\in\mathcal{B}, \\ \delta_{a,a'}\leq m}} \mathcal{Z}^{a',y}_{S',\,m-\delta_{a,a'}}. \tag{4}$$

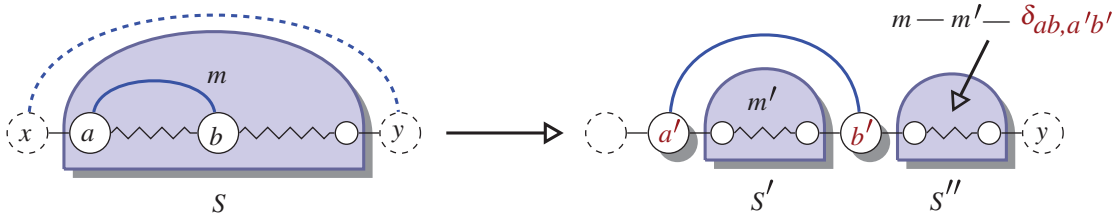
  Indeed, any suitable sequence is a concatenation of a, possibly mutated, nucleotide a' at the first position, followed by a sequence over the remaining interval, having $m - \delta_{a,a'}$ mutations (accounting for a possible mutation at first position), and having flanking nucleotides a' and y.
- **Case 2: Paired ends, stacking onto another base-pair.** If both ends of the considered interval form a base-pair ($S = (S')$), stacking onto another base pair just outside the whole region, then the isosteric contribution of the base-pair must be supplemented with a specific ''stacking-pairs'' bonus. Let a and b be the nucleotides found on both ends of the interval (positions i and j), then one has



Case 1: First position is unpaired.



Case 2: Extremities are paired, nested within a consecutive base-pair, forming a stacking base-pair.



Case 3: First position in paired to some position, but not involved in a stacking pair.

FIG. 1. Principle of the inside computation (partition function). Any sequence (mutated) can be decomposed as a sequence preceded by a possibly mutated base (unpaired case), a sequence surrounded by a possibly mutated base-pair (stacking-pair case), or as two sequences segregated by a possibly mutated base-pair (general base-pairing case). In this latter case, mutations must be distributed between subsequences.

$$\mathcal{Z}^{x,y}_{S,m} = \sum_{\substack{a',b' \in \mathcal{B}^2, \\ \delta_{ab,a'b'} \leq m}} e^{\frac{-E^{(i,j)}_{xy \to a'b'}}{RT}} \cdot \mathcal{Z}^{a',b'}_{S',m-\delta_{ab,a'b'}}. \tag{5}$$

Any sequence generated here consists of two possibly mutated nucleotides a' and b', flanking a sequence over the remaining portion. In order for the total distance to sum to m, this portion must feature $m-\delta_{ab,a'b'}$ additional pointwise mutations.

– **Case 3: Paired leftmost position + absence of stacking pairs.** In this case, the structure is split into two parts by the base pair $(S=(S')S'')$. Let us denote by k the partner of position i, and by a, b, and c the bases found at positions i, k, and j respectively, then one has:

$$\mathcal{Z}^{x,y}_{S,m} = \sum_{\substack{a',b' \in \mathcal{B}^2, \\ \delta_{ab,a'b'} \leq m}} \sum_{m'=0}^{m-\delta_{ab,a'b'}} e^{\frac{-E^{(i,k)}_{\emptyset \to a'b'}}{RT}} \cdot \mathcal{Z}^{a',b'}_{S',m-m'-\delta_{ab,a'b'}} \cdot \mathcal{Z}^{b',y}_{S'',m'}. \tag{6}$$

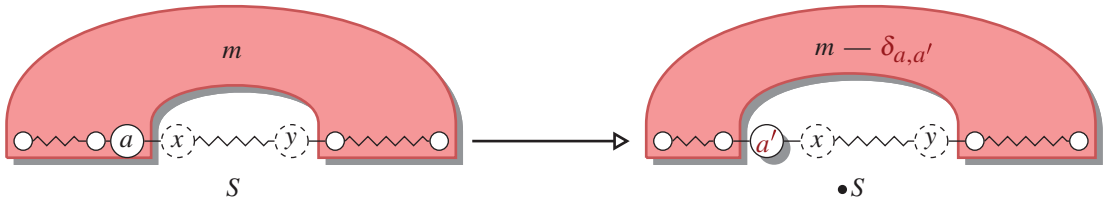
In other words, if the leftmost position is paired, and the base pair is not stacking onto another base pair, then the only term contributing directly to the energy is the isostericity of the base pair. Admissible sequences for S consist of two paired nucleotides a' and b' at positions i and k respectively, flanking a sequence for S' (over an interval $[i+1, k-1]$) and followed by a (possibly empty) sequence for S'' (over $[k+1, j]$). Since the total number of mutations sums to m, a parameter m' is introduced to distribute the remaining mutations between the two sequences.

Outside computation. The *outside* function $\mathcal{Y}^{x,y}_{S,m}$ is the partition function considering only the contributions of subsequences excluding a given structure/interval S, occupying the open interval $]i, j[$ in the sequence, at Hamming distance exactly m to the initial sequence s and assuming that nucleotides x and y were previously chosen for $i+1$ and $j-1$, the outermost portions of the excluded structure. The associated terms $\mathcal{Y}^{x,y}_{S,m}$ can then be computed recursively, initially considering the case of any prefix S' of the complete structure:

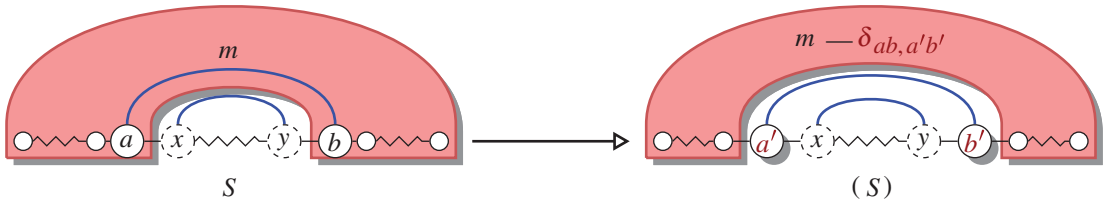
$$\forall x, y \in \mathcal{B}\times\mathcal{B}, \forall S' \text{ s.t. } S = S'.S'', m \in [0, M] : \mathcal{Y}^{x,y}_{S',m} = \mathcal{Z}^{y,z}_{S'',m} \tag{7}$$

where $z \in \mathcal{B}$ can be any nucleotide, and provably does not affect further computations. In other words, the sequences explored by an outside computation, excluding a prefix of S, are exactly the sequences generated on the corresponded suffix. This set of sequences is also the inside term on the suffix structure. It is also worth pointing out that $\mathcal{Y}^{x,y}_{S,m} = \mathcal{Z}^{x,y}_{\varepsilon,m} = 1_{m=0}$.

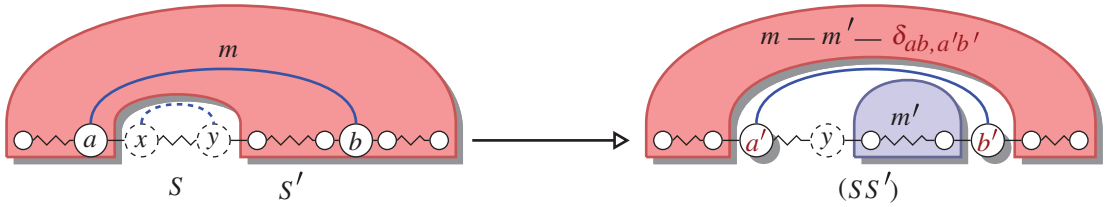
In general, the main recurrence below works by extending the excluded structure S (covering the $]i, j[$ interval) in the leftward direction. As shown in Figure 2, four cases must be considered depending on the base-pairing status and directionality of the next considered position:



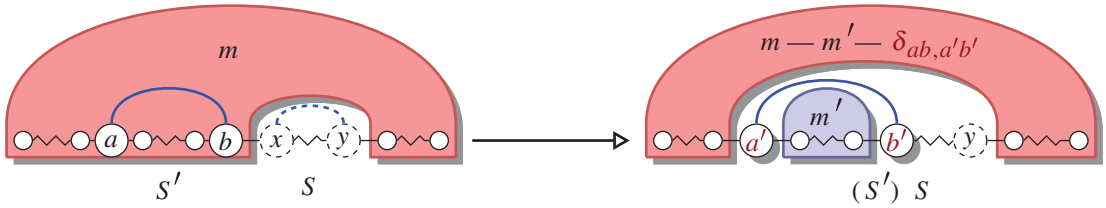
Case 1: Left innermost position is unpaired



Case 2: Paired innermost positions, leading to stacking base-pairs.



Case 3: Next leftward position is paired to the right, but no stacking pairs.



Case 4: Next leftward position is paired to the left.

FIG. 2. Principle of the outside computation. Note that the outside algorithm uses intermediate results from the inside algorithm; therefore, its efficient implementation requires the precomputation and storage of the inside contributions.

– **Case 1: Unpaired position.** When the innermost leftward base is unpaired, any base $a \in \mathcal{B}$ may be chosen for this position, and the rest of the sequence is generated recursively. The number of remaining mutations may be decreased by the choice of a, leading to the following recurrence:

$$\mathcal{Y}_{S,m}^{x,y} = \sum_{\substack{a' \in \mathcal{B}, \\ \delta_{a,a'} \leq m}} \mathcal{Y}_{\bullet S, m-\delta_{a,a'}}^{a',y} \tag{8}$$

– **Case 2: Stacking base-pair.** If both ends of the excluded structure are paired together and are nested within another base pair in the remaining structure, then an additional contribution, stemming from a stacking-pairs energy, has to be considered. The outside terms are then computed by simulating any pair of base-pairing nucleotides for i, j and by proceeding recursively on the remaining portion, as follows:

$$\mathcal{Y}_{S,m}^{x,y} = \sum_{\substack{a'b' \in \mathcal{B}^2, \\ \delta_{ab,a'b'} \leq m}} e^{\frac{-E^{(i,j)}_{xy \to a'b'}}{RT}} \cdot \mathcal{Y}_{(S),m-\delta_{ab,a'b'}}^{a',b'} \tag{9}$$

– **Case 3: Next position paired rightward + absence of stacking pair.** In this case, the innermost leftward position i is paired to the right at some position k. Let us assume that its partner resides outside the excluded structure S. (This assumption is provably without loss of generality and directly follows from the fact that S is well-parenthesized). The structure within the base pair may be described by the expression (SS'), where S' may possibly be empty (except if S is enclosed within corresponding brackets to avoid stacking pairs). Therefore, any sequence considered by the outside computation consists of three independent parts: two nucleotides for the paired positions, a sequence for the region excluding S'', and a sequence for S'. It follows that:

$$\mathcal{Y}_{S,m}^{x,y} = \sum_{\substack{a'b' \in \mathcal{B}^2, \\ \delta_{ab,a'b'} \leq m}} \sum_{m'=0}^{m-\delta_{ab,a'b'}} e^{\frac{-E^{(i,k)}_{\emptyset \to a'b'}}{RT}} \cdot \mathcal{Y}_{(SS'),m-\delta_{ab,a'b'}-m'}^{a',b'} \cdot \mathcal{Z}_{S',m'}^{y,b'}. \tag{10}$$

– **Case 4: Next position paired leftward.** If the innermost, leftward position i is paired to some position $k < i$, delimiting a substructure S', then any sequence considered by the outside computation consists in three parts: two nucleotides a and b, a sequence for S', and a sequence for the region excluding $(S')S$. Consequently, one has:

$$\mathcal{Y}_{S,m}^{x,y} = \sum_{\substack{a'b' \in \mathcal{B}^2, \\ \delta_{ab,a'b'} \leq m}} \sum_{m'=0}^{m-\delta_{ab,a'b'}} e^{\frac{-E^{(k,i)}_{\emptyset \to a'b'}}{RT}} \cdot \mathcal{Y}_{(S')S,m-\delta_{ab,a'b'}-m'}^{a',b} \cdot \mathcal{Z}_{S',m'}^{a',b'} \tag{11}$$

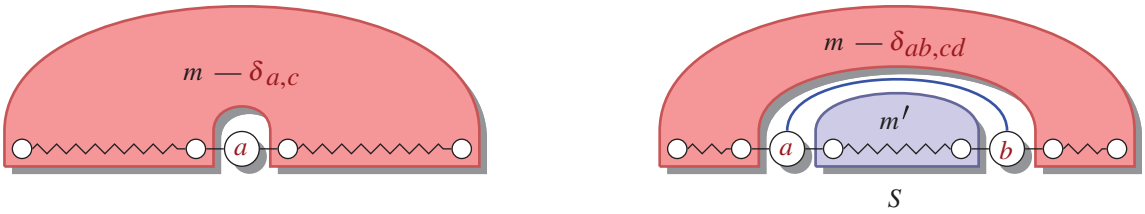
Combining inside and outside computations into pointwise mutations probabilities. We are now left to compute the probability that a given nucleotide $a \in \mathcal{B}$ is found at a given position i. This quantity can also be expressed as the ratio of $\mathcal{W}_{i,[a]}^{M}$, the total Boltzmann weight of the set of sequences featuring the nucleotide, and $\mathcal{Z}_{S,\leq M}^{X,X}$, the total weight of sequences having at most M mutations:

$$\mathbb{P}(s_i = a | M) := \frac{\mathcal{W}_{i,[a]}^{M}}{\mathcal{Z}_{S,\leq M}^{X,X'}} = \frac{\mathcal{W}_{i,[a]}^{M}}{\sum_{m=0}^{M} \mathcal{Z}_{S,m}^{X,X'}} \tag{12}$$

where $X, X' \in \mathcal{B}$ may be any nucleotides (no impact on energy/weights).

To that purpose, one leverages the *inside–outside* construction as illustrated in Figure 3. Namely, while computing the total contribution of sequences featuring a base $a' \in \mathcal{B}$ at position i, one must consider the three following situations:

– **Case 1: Unpaired position.** In this case, the supporting sequences are simply those excluding the structure • at position i, where a base $c \in \mathcal{B}$ was formerly found. Summing over any possible number of mutations for the *outside* region, one obtains:



Case 1: Unpaired position. **Case 2 & 3:** Paired positions.

FIG. 3. Combining the inside and outside contributions to compute the total Boltzmann weight of all sequences having a given base (case 1) or base pair (cases 2 and 3) at a given position.

$$\mathcal{W}^{M}_{i,[a]} = \sum_{m=\delta_{a,c}}^{M} \mathcal{Y}^{a,a}_{\bullet,\, m-\delta_{a,c}}. \tag{13}$$

– **Case 2: Position is paired rightward.** Here, position i is paired to some position $k > i$, whose content impacts the base-pair contribution. Any sequence having a at position i can be decomposed as a base pair, nesting a sequence for the structure S' on the interval $[i+1, k-1]$, and an outside sequence, excluding the structure (S'). Consequently, letting c and d be the original nucleotides at positions i and k, one has:

$$\mathcal{W}^{M}_{i,[a]} = \sum_{m=0}^{M} \sum_{\substack{b\in\mathcal{B} \\ \delta_{ab,cd}\le m}} \sum_{m'=0}^{m-\delta_{ab,cd}} e^{\frac{-E^{(i,k)}_{\emptyset\to ab}}{RT}} \cdot \mathcal{Y}^{a,b}_{(S'),\, m-\delta_{ab,cd}} \cdot \mathcal{Z}^{a,b}_{S',m'} \tag{14}$$

– **Case 3: Position is paired leftward.** This case is symmetrical to the previous one, with the exception that $k < i$. Consequently, letting c and d be the original nucleotides at positions i and k, one has:

$$\mathcal{W}^{M}_{i,[a]} = \sum_{m=0}^{M} \sum_{\substack{b\in\mathcal{B} \\ \delta_{ba,cd}\le m}} \sum_{m'=0}^{m-\delta_{ba,cd}} e^{\frac{-E^{(k,i)}_{\emptyset\to ba}}{RT}} \cdot \mathcal{Y}^{b,a}_{(S'),\, m-\delta_{ba,cd}} \cdot \mathcal{Z}^{b,a}_{S',m'} \tag{15}$$

2.3. *Complexity considerations*

Using dynamic programming, Equations (4)–(6) and (8)–(11) can be computed in linear time and memory. Namely, the $\mathcal{Z}^{*}_{*}$ and $\mathcal{Y}^{*}_{*}$ terms are computed starting from smaller values of m and structure lengths, storing the results as they become available to ensure constant-time access during later stages of the computation. Furthermore, energy terms $E(\cdot)$ may be accessed in constant time after a simple pre-computation of the isostericity contributions in $\Theta(n \cdot |\Omega|)$. Computing any given term therefore requires $\Theta(m)$ operations due to the explicit distribution of the number of mutations.

In principle, $\Theta(M \cdot n^2)$ terms, identified by different (S, m) 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. A similar property holds for the outside computation, following from the linearity of the outside recurrence (i.e., the computation of the outside term for a given excluded structure only relies on the computation of another, strictly larger, structure).

These properties drastically limit 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* (Johansson et al., 2010) library for arbitrary floating point precision. The source code is freely available online.

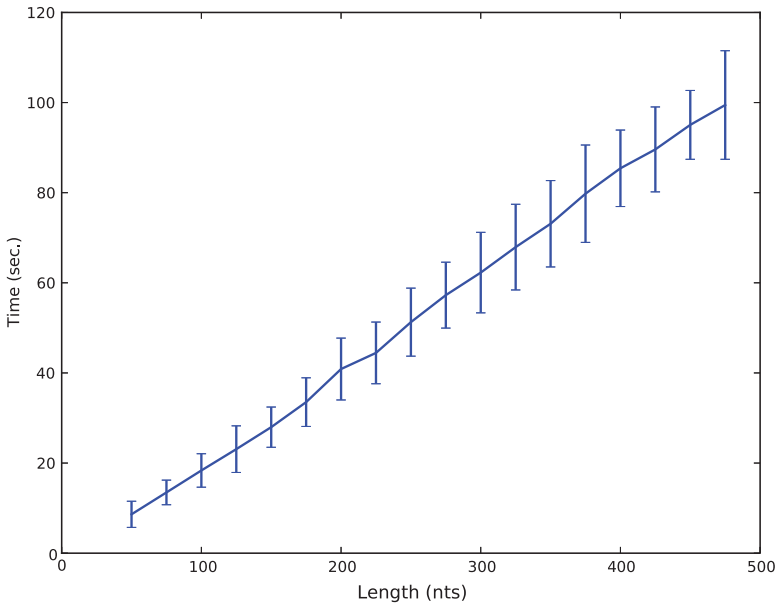


FIG. 4. Typical runtimes required by the computation of mutational profiles averaged on 50 random sequences for each length ranging from 100 and 400 nts, while allowing for a maximal number of mutations, $M = 10$. For each sequence, a random multiple sequence alignment was generated, consisting ozf 51 aligned sequences, each compatible with a randomly generated consensus secondary structure.

The time benchmarks were performed on an AMD Opteron Processor 6278 at 2.4 GHz with cache of 512 KB, (20 cores, and 175 GB ram. Since typical use-cases of `RNApyro` require efficiency and scalability, we present in Figure 4 typical runtimes required to compute the probabilities for every nucleotide at every position for a vast set of parameters. For those tests, both the multiple sequence alignment and the reference sequence were generated uniformly at random, based on a realistic random secondary structure, generated as described in Levin et al. (2012).

3.2. Homogenous error correction in 5s rRNAs

To illustrate the potential of our algorithm, we applied our techniques to identify and correct pointwise 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` (Li and Godzik, 2006) to remove sequences with more than 80% of sequence similarity. This operation resulted in a data set of 45 sequences.

We designed our benchmark using a leave-one-out strategy. We randomly picked a single sequence from our data set and performed 12 random mutations, corresponding to an error-rate of 10%. We repeated this operation 10 times. The value of β was set to 15 (larger values gave similar results). To estimate the impact on the distribution of the relative weights of energy and isostericity, we used four different values of $\alpha = 0,0.5,0.8,1.0$. Similarly, we also investigated the impact of an under- and overestimate of the number of errors, by setting the presumed number of errors to 50% (6 mutations) and 200% (24 mutations) of their exact number (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 $P(i, x)$ of a nucleotide x exceeds this threshold. To correct the errors we used the set of nucleotides having probability greater than λ, that is

$$C(i) = \{x | x \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}\}, P(i, x) > \lambda \text{ and } n \neq \omega[i]\},$$

where $\omega[i]$ is the nucleotide at position i in the input sequence. We note that, for lower thresholds, multiple nucleotides may be available in $C(i)$ to correct the sequence. Here, we remind the reader that our aim is to estimate the potential of error-correction of `RNApyro` and not to develop a full-fledged error-correction pipeline, which shall 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). Our results are reported in Figure 5.

Our data demonstrates that our algorithm shows interesting potential 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 consideration of stacking energies 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 of it. Finally, our algorithm seems robust to the number of presumed mutations. Indeed, good AUC values are achieved even with conservative estimates for the number of errors (cf, 50% of the errors,

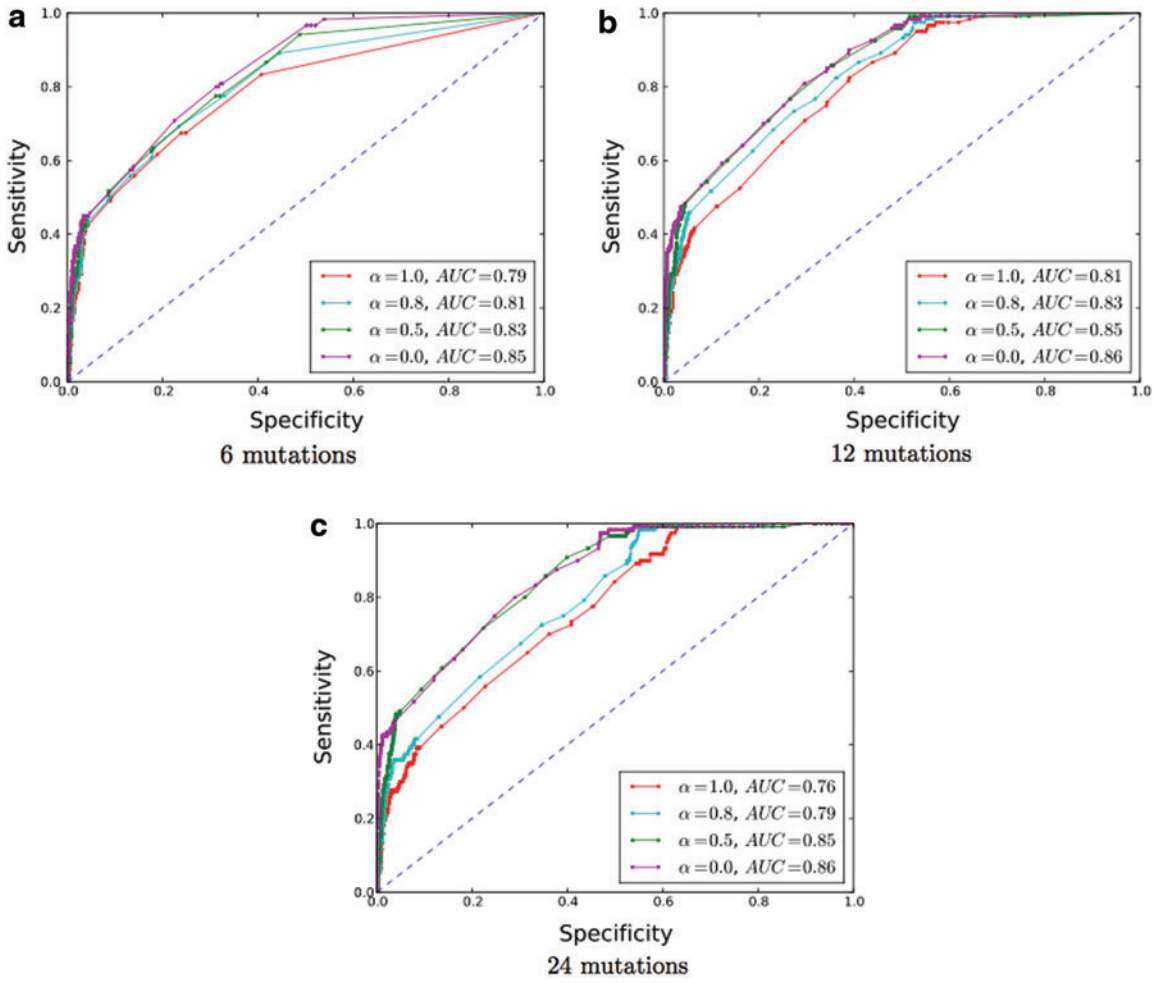


FIG. 5. Performance of error-correction. Subfigures show accuracy with underestimated error rates (6 mutations), exact estimates (12 mutations), and overestimates (24 mutations). We also analyze the impact of the parameter α distributing the weights of stacking pair energies versus isostericity scores 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 represents the average performance over the 10 experiments.

leading to Fig. 5a), as well as with large values (cf, 200% of the errors in Fig. 5c). It is worth noting that scoring schemes giving a larger weight on the isostericity scores (i.e., for low α values) seem more robust to under- and overestimating the number of errors.

3.3. *Correcting Illumina sequencing errors in 16s rRNAs*

To complete our benchmark, we turned to the small subunit ribosomal RNA in bacteriae, a molecule which is of particular interest in metagenomics and phylogenetics. Our aim was to get as close as possible to a pyrosequencing context in which reads are produced nonuniformly by an Illumina sequencer, impacting the distribution of errors in the sequence. We chose this setting both because of the popularity of Illumina sequencing in metagenomics, and since the underlying sequencing technique only considers base

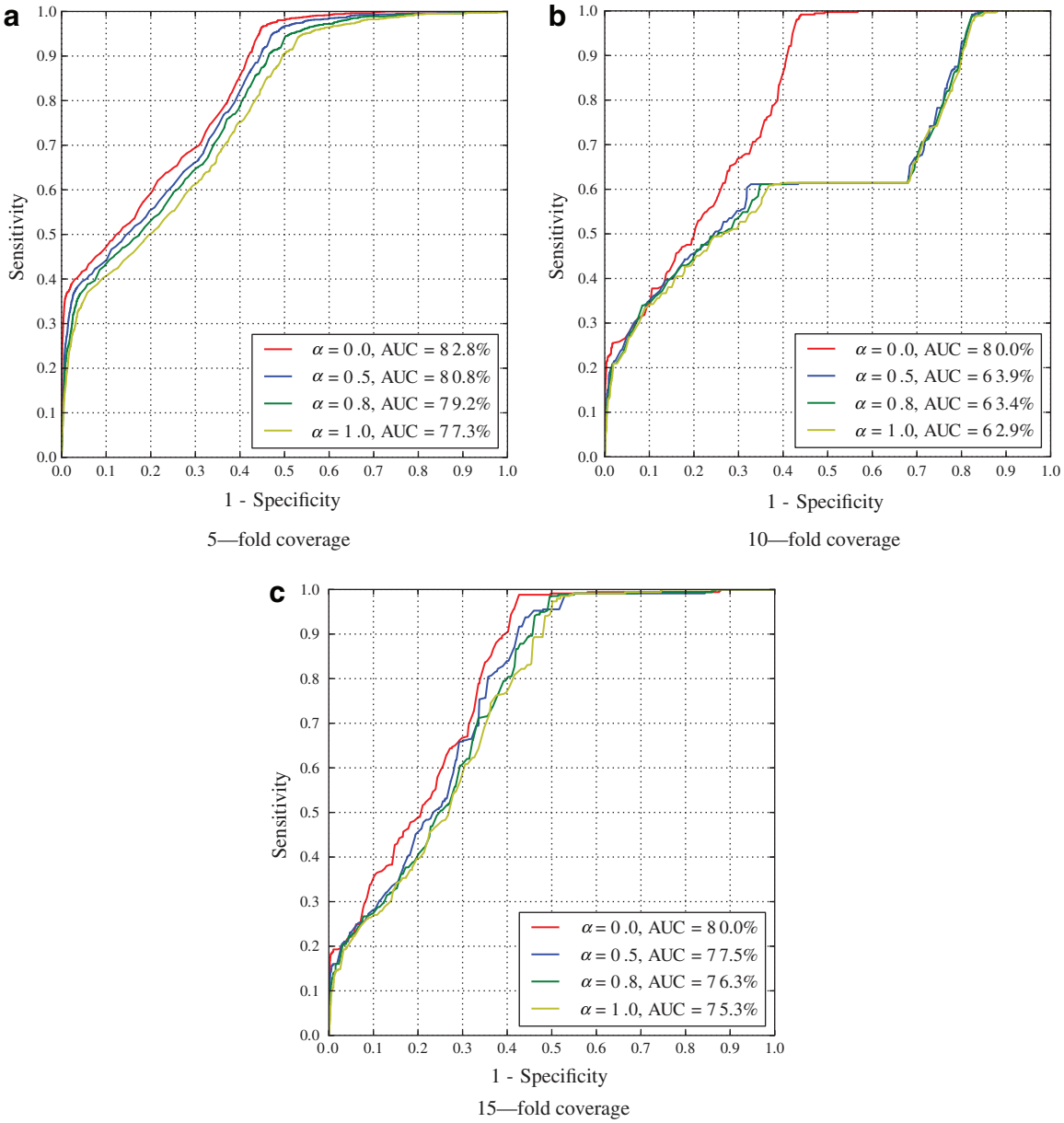


FIG. 6. Performance of error-correction over all positions. Subfigures show accuracy when ART fold parameter is set to 5-, 10-, and 15-fold coverage. We also analyze the impact of the parameter α distributing the weights of stacking pair energies versus isostericity scores and use values ranging from $\alpha = \{0,0.5,0.8,1.0\}$. The AUC is indicated in the legend of the figures. Each individual ROC curve represents the average performance over 30 experiments.

substitutions (no insertions), the only type of errors currently detected by RNApyro. To that purpose, we used simulated Illumina reads, mapped the reads back to a reference alignment, and ran RNApyro on a consensus sequence derived from the mapped reads, estimating the maximal amount of mutations from both the length of the sequence and the sequencing depth.

We gathered the seed sequences of the bacterial multiple sequence alignment (MSA) retrieved from the RFAM Database 11.0 (Rfam id: RF00177) (Gardner et al., 2011). This alignment is composed of 93 sequences, whose length varies between 1461 and 1568 nucleotides and has an average pairwise sequence identity of 69%. We used the pseudoknot-free version of the consensus secondary structure. A secondary structure for a specific reference sequence was obtained by simply mapping the structure back to sequence, that is, by removing any base pair having at least one partner involved in a gapped position. For similar reasons, we locally excluded from our calculation of the isostericity contribution for a given base pair, described in Section 2.1, any sequence featuring at least a gap on the corresponding positions.

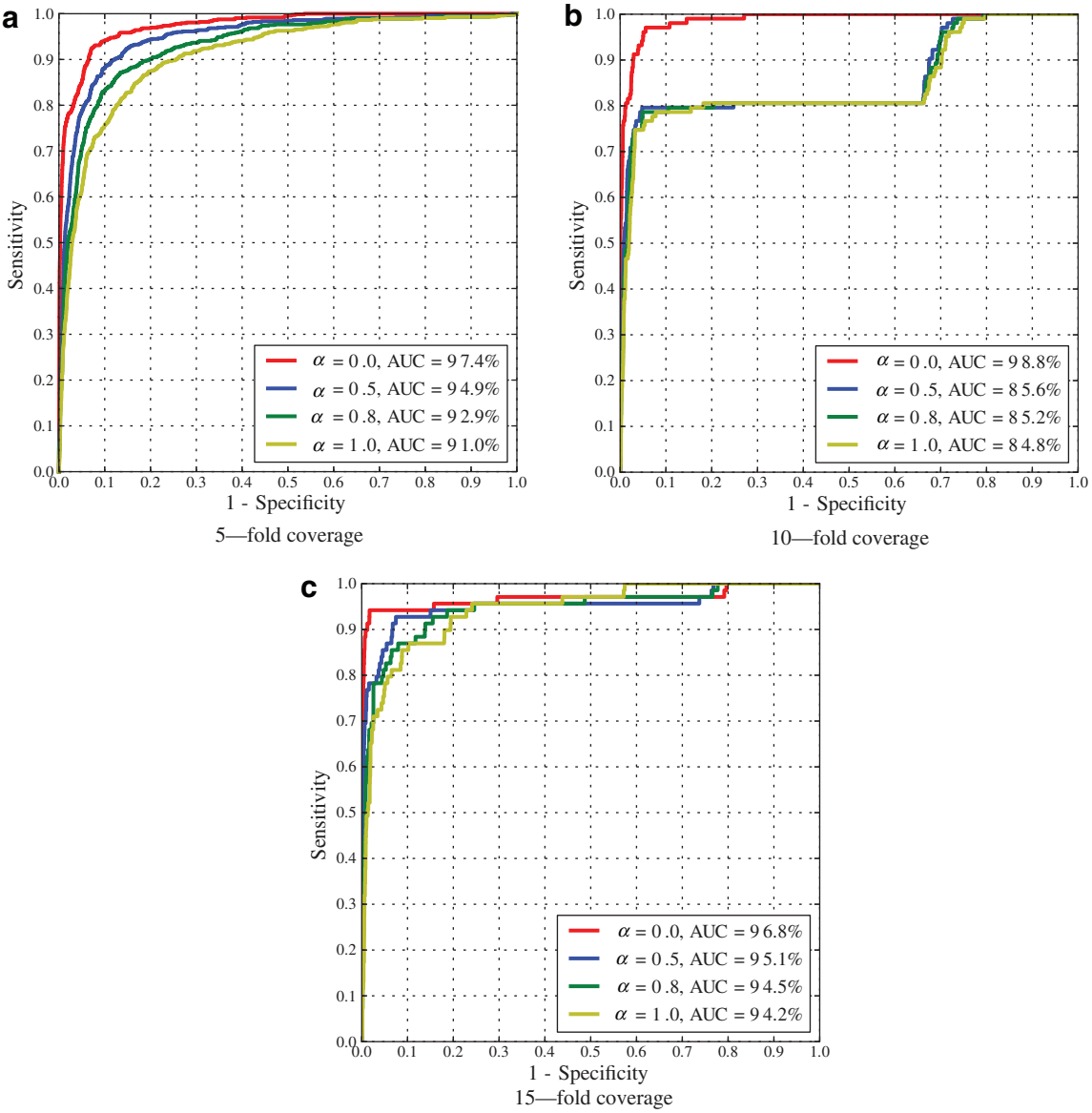


FIG. 7. Performance of error-correction restricted to structured positions (i.e., involved in a canonical base pair). Subfigures show accuracy when ART fold parameter is set to 5-, 10-, and 15-fold coverage. We also analyze the impact of the parameter α distributing the weights of stacking pair energies versus isostericity scores and use values ranging from $\alpha = \{0,0.5,0.8,1.0\}$. The AUC is indicated in the legend of the figures. Each individual ROC curve represents the average performance over at least 10 experiments.

To simulate sequencing errors, we used the next-generation sequencing read simulator ART (Huang et al., 2012). The *Illumina technology* setting was chosen as the main error mode, generating reads of 75 bps, featuring mostly base substitution errors. Reads were mapped back to the original sequence, and a consensus sequence was determined from the sequencing output by a simple majority vote in the case of multiple coverage. Uncovered regions were simply generated at random. The average rate of errors observed in the final consensus sequence was empirically estimated to represent 2.4%, 0.9%, and 0.6% of the reference sequence for prescribed sequencing coverages of 5-, 10-, and 15-fold respectively.

As in Section 3.2, we evaluated the predictive power of RNApyro-computed mutational profiles using a leave-one-out strategy. We picked a sequence at random from the MSA, sequenced/mutated it as above, and ran RNApyro to establish its mutational profile. In this execution, we used values of $\beta = 15$ and $\alpha \in [0, 0.5, 0.8, 1.0]$, and set the presumed number of mutations to twice the average error rate made by ART, i.e., 4.8%, 1.8%, and 1.2% for fold coverages of 5, 10, and 15 respectively. We repeated the whole procedure 30 times for each (5/10/15-fold) coverage. We evaluated the predictive power of the profile by computing a joint ROC curve for each value of α, and each coverage, as described in Section 3.2. Figure 6 shows the ROC curves, computed over all positions, while Figure 7 only focuses on positions that are paired in the consensus secondary structure.

Our data demonstrates that even on long sequences our algorithm shows interesting potential for error-correction applications. First, the AUC values (up to 0.81) when looking at all positions, in Figure 6, indicate that a signal has been successfully extracted. When restraining to structured regions (i.e., base-paired regions), we obtain AUC values up to 0.988. Since contributions to the energy and isostericity only arise from structured regions, this was an expected result.

An interesting feature is that the best results are almost always obtained when $\alpha = 0$, i.e., when all the contribution comes from the isostericity.

A final observation is related to all positions as in Figure 6. The first 20% to 30% of sensitivities are obtained almost without any errors and correspond to the nucleotides in structured regions. The rest of the predictions are done on unpaired nucleotides.

4. CONCLUSIONS

In this article, we presented a new and efficient way of exploring the mutational landscape of RNA under structural constraints and applied our techniques to identify and fix sequencing errors. In addition, we introduced a new scoring scheme to measure the likelihood of sequencing errors that combines the classical nearest-neighbor energy model parameters (Turner and Mathews, 2010) to the recently introduced isostericity matrices (Stombaugh et al., 2009). The latter accounting for geometrical discrepancies occurred during base-pair replacements.

We combined our algorithm for exploring the mutational neighborhood of an input sequence with known secondary structure to this new pseudo energy model and created a tool to predict pointwise sequencing errors in structured RNA. Importantly, our 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. This achievement enables us to envision applications to high-throughput sequencing pipelines.

We validated our model on the 5s rRNA and 16s rRNA (see Section 3) and showed that our technology enables us to recover mutational errors with high accuracy (area under the ROC curve between 0.95 and 0.99 when restricting predicted mutations to base-paired regions). Interestingly, we observed that using the isostericity matrices alone yields higher performance than with the nearest-neighbor energy model alone. This finding supports our hypothesis that isostericity matrices provide a valuable source of information that can be efficiently used for RNA sequence and structure analysis. Nonetheless, we also found that the nearest-neighbor model seems to provide a valuable signal when the number of errors in the input sequences is underestimated.

Our techniques are designed to correct pointwise errors in structured regions (i.e., base-paired nucleotides). However, our software RNApyro can be easily combined with other methodologies previously developed for correcting other types of sequencing errors such as indels in unstructured regions or repeats (Quinlan et al., 2008; Quince et al., 2009).

In future works, we also intend to include our model errors stemming from insertions or deletions. It is indeed theoretically possible to consider these scenarios within our dynamic programming scheme

(Waldispühl et al., 2002), with minor impacts to the complexity. Finally, we hope that integrating our software to current sequencing pipelines used in metagenomics studies will permit us to improve the estimate of microbial diversity.

ACKNOWLEDGMENTS

The authors would like to thank Rob Knight for his suggestions and comments. This work was funded by the French Agence Nationale de la Recherche (ANR) through the MAGNUM ANR 2010 BLAN 0204 project (to Y.P.), the FQRNT team grant 232983, and NSERC Discovery grant 219671 (to J.W.). This article has been developed as a result of a mobility stay funded by the Erasmus Mundus Programme of the European Commission under the Transatlantic Partnership for Excellence in Engineering TEE Project (V.R.).

AUTHOR DISCLOSURE STATEMENT

The authors declare that no competing financial interests exist.

REFERENCES

Gardner, P.P., Daub, J., Tate, J., et al. 2011. Rfam: Wikipedia, clans and the ''decimal'' release. *Nucleic Acids Res.* 39, D141–D145.

Hori, H., and Osawa, S. 1987. Origin and evolution of organisms as deduced from 5s ribosomal RNA sequences. *Mol. Biol. Evol.* 4, 445–472.

Huang, W., Li, L., Myers, J.R., and Marth, G.T. 2012. Art: a next-generation sequencing read simulator. *Bioinformatics* 28, 593–594.

Johansson, F., et al. 2010. mpmath: a Python library for arbitrary-precision floating-point arithmetic (version 0.14). Available at http://code.google.com/p/mpmath/

Kunin, V., Engelbrektson, A., Ochman, H., and Hugenholtz, P. 2010. Wrinkles in the rare biosphere: pyrosequencing errors can lead to artificial inflation of diversity estimates. *Environ. Microbiol.* 12, 118–123.

Lari, K., and Young, S. 1990. The estimation of stochastic context-free grammars using the Inside-Outside algorithm. *Computer Speech & Language* 4, 35–56.

Leontis, N.B., and Westhof, E. 2001. Geometric nomenclature and classification of RNA base pairs. *RNA* 7, 499–512.

Levin, A., Lis, M., Ponty, Y., et al. 2012. A global sampling approach to designing and reengineering RNA secondary structures. *Nucleic Acids Res.* 40, 10041–10052.

Li, W., and Godzik, A. 2006. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics* 22, 1658–1659.

Medvedev, P., Scott, E., Kakaradov, B., and Pevzner, P. 2011. Error correction of high-throughput sequencing datasets with non-uniform coverage. *Bioinformatics (Oxford, England)* 27, i137–41.

Olsen, G., and Woese, C. 1993. Ribosomal RNA: a key to phylogeny. *FASEB J.* 7, 113–123.

Olsen, G.J., Lane, D.J., Giovannoni, S.J., et al. 1986. Microbial ecology and evolution: a ribosomal RNA approach. *Annu. Rev. Microbiol.* 40, 337–365.

Quince, C., Lanzén, A., Curtis, T.P., et al. 2009. Accurate determination of microbial diversity from 454 pyrosequencing data. *Nat. Methods* 6, 639–641.

Quinlan, A., Stewart, D., Strömberg, M., and Marth, G. 2008. Pyrobayes: an improved base caller for SNP discovery in pyrosequences. *Nat. Methods* 5, 179–81.

Stombaugh, J., Zirbel, C.L., Westhof, E., and Leontis, N.B. 2009. Frequency and isostericity of RNA base pairs. *Nucleic Acids Res.* 37, 2294–2312.

Turnbaugh, P.J., Ley, R.E., Hamady, M., et al. 2007. The Human Microbiome Project. *Nature* 449, 804–810.

Turner, D.H., and Mathews, D.H. 2010. NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. *Nucleic Acids Res.* 38, D280–D282.

Waldispühl, J., Behzadi, B., and Steyaert, J.-M. 2002. An approximate matching algorithm for finding (sub-)optimal sequences in s-attributed grammars. *Bioinformatics* 18, S250–S259.

Waldispühl, J., Devadas, S., Berger, B., and Clote, P. 2008. Efficient Algorithms for probing the RNA mutation landscape. *PLoS Comput. Biol.* 4, e1000124.

Waldispühl, J., and Ponty, Y. 2011. An unbiased adaptive sampling algorithm for the exploration of RNA mutational landscapes under evolutionary pressure. *J. Comp. Biol.* 18, 1465–1479.

Zuckerkandl, E., and Pauling, L. 1965. Molecules as documents of evolutionary history. *J. Theor. Biol.* 8, 357–366.

Address correspondence to:
Dr. Yann Ponty
LIX/CNRS UMR 7161
Ecole Polytechnique
Palaiseau 91128
France

E-mail: yann.ponty@lix.polytechnique.fr

and

Dr. Jerome Waldispühl
School of Computer Science
McGill University
3480 University Street
Montreal, Quebec
H3A 0E9 Canada

E-mail: jeromew@cs.mcgill.ca