

Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

ISMB/ECCB 2017 Proceedings – RNA Bioinformatics

# Efficient approximations of RNA kinetics landscape using non-redundant sampling

Juraj Michálik<sup>1,2</sup>, Hélène Touzet<sup>3,4</sup> and Yann Ponty<sup>1,2,\*</sup>

<sup>1</sup>AMIB project, Inria Saclay, 91120 Palaiseau, France

<sup>2</sup>LIX CNRS UMR 7161, Ecole Polytechnique, 91120 Palaiseau, France

<sup>3</sup>CRISTAL (CNRS UMR 9189, University of Lille), 59650 Villeneuve d'Ascq, France

<sup>4</sup>Inria Lille-Nord Europe, France

\*To whom correspondence should be addressed.

Received on XXXXX; revised on XXXXX; accepted on XXXXX

## Abstract

**Motivation:** Kinetics is key to understand many phenomena involving RNAs, such as co-transcriptional folding and riboswitches. Exact out-of-equilibrium studies induce extreme computational demands, leading state-of-the-art methods to rely on approximated kinetics landscapes, obtained using sampling strategies that strive to generate the key landmarks of the landscape topology. However, such methods are impeded by a large level of redundancy within sampled sets. Such a redundancy is uninformative, and obfuscates important intermediate states, leading to an incomplete vision of RNA dynamics.

**Results:** We introduce *RNANR*, a new set of algorithms for the exploration of RNA kinetics landscapes at the secondary structure level. *RNANR* considers locally optimal structures, a reduced set of RNA conformations, in order to focus its sampling on basins in the kinetic landscape. Along with an exhaustive enumeration, *RNANR* implements a novel non-redundant stochastic sampling, and offers a rich array of structural parameters. Our tests on both real and random RNAs reveal that *RNANR* allows to generate more unique structures in a given time than its competitors, and allows a deeper exploration of kinetics landscapes.

**Availability:** *RNANR* is freely available at <https://project.inria.fr/rnaland/rnanr>

**Contact:** [yann.ponty@lix.polytechnique.fr](mailto:yann.ponty@lix.polytechnique.fr)

## 1 Introduction

Ribonucleic Acids (RNAs) are fascinating biopolymers. Beyond their coding capacities, they can serve as a medium for the transmission of genetic information, as in the case of highly-structured RNA viruses such as Ebola or HIV (Wilkinson *et al.*, 2008). They can also perform a large diversity of catalytic and regulatory functions, as demonstrated by the 2474 functional families found in the current release of the RFAM database (Nawrocki *et al.*, 2015). This versatility is such that RNA is currently considered by a whole scientific community as the most parsimonious explanation for the molecular basis of the origin of life (Cech, 2015). This versatility, coupled with the combinatorial specificity of its interactions with other nucleic acids, makes RNA a tool of choice for designing nano-architectures through programmable self-assembly (Li *et al.*, 2011), or in the blooming field of synthetic biology (Kushwaha *et al.*, 2016).

A substantial proportion of the functions performed by RNAs critically relies on the adoption of a stable 3D structure through a pairing of its nucleotides, mediated by hydrogen bonds. A precise structural modeling of

RNA structure, possibly in interaction with other molecules, is thus required to identify binding and catalytic sites, and more generally formulate functional hypotheses Cruz and Westhof (2011). Despite recent progress, such as SHAPE chemistry (Smola *et al.*, 2015), experimental techniques for RNA structure resolution are still lagging behind high-throughput sequencing techniques, leading to a striking asymmetry between the amount of available structure and sequence data. It is a current challenge of RNA bioinformatics, and the object of ongoing efforts for a whole community, to accurately predict the structure of RNA from its sequence by integrating data of various origins (Miao *et al.*, 2015).

RNA folding is inherently stochastic, and governed by the laws of statistical physics (McCaskill, 1990). It is generally believed to be hierarchical (Tinoco and Bustamante, 1999) which, in conjunction with intrinsic computational limitations (Akutsu, 2000; Sheikh *et al.*, 2012), has led to an initial dismissal of complex topological motifs such as pseudoknots within computational methods (Isambert, 2009). The seminal work of McCaskill (1990) has demonstrated the computability in polynomial-time

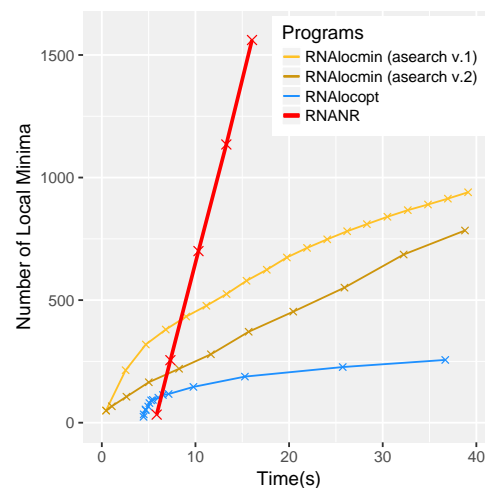
of the partition function and the subsequent derivation of base-pairing probabilities which provide realistic notion of supports for predicted base pairs (Mathews, 2004).

However the assumption of a thermodynamic equilibrium fails to account for the observed behavior of certain RNAs, which strongly suggests the prevalence of kinetics effects in their folding process. Perhaps the most prominent example can be found in riboswitches (Baumstark *et al.*, 1997; Schultes and Bartel, 2000), RNAs that have been found to adopt different conformations depending on the presence/absence of a ligand, despite a significant difference in free energies between the two conformers. This is hardly compatible with the assumption of a thermodynamic equilibrium, which would dictate the main adoption of the Minimal Free Energy (MFE) structure regardless of the presence of the ligand. This is however consistent with a kinetics-inspired model, where the ligand modulates an energy barrier separating the two conformers in the folding landscape, modifying the convergence speed towards the thermodynamic equilibrium (Badelt *et al.*, 2015). The prevalence of kinetics effects can also be suspected in instances of co-transcriptional folding (Watters *et al.*, 2016), or when transcripts undergo a fast degradation and the half-life of some transcript are much shorter than the time taken to converge towards the thermodynamic equilibrium (Sharova *et al.*, 2009).

Computational methods for the study of RNA kinetics essentially fall into two categories. A first category of methods, dubbed *simulation methods*, perform a stochastic simulation of the folding process at the base-pair (Flamm *et al.*, 2000, *Kinfold*) or helix (Danilova *et al.*, 2006, *RNAkinetics*) step resolution, possibly allowing for the presence of pseudoknots (Xayaphoummine *et al.*, 2007, *kinfold*). Sets of generated folding trajectories are analyzed and main conformers, along with the evolution of their concentrations through time, is easily obtained.

However, as noted in Flamm and Hofacker (2008), the number of trajectories required to obtain reproducible results quickly becomes prohibitively large as the size of RNAs increases. For this reason, a second type of computational methods analyze RNA kinetics as a continuous Markov process, adopting a general four-steps program: **Generation** of a representative subset of conformations; **Embedding** of representative conformations into adjacency structure, whose main alternatives are barrier trees used by *barrier* (Flamm *et al.*, 2002), and the basin hopping graphs of *BHGBuilder* (Kucharik *et al.*, 2014); **Estimation** of transition rates from (approximate) energy barriers. The exact computation of this quantity requires solving an NP-hard problem (Mañuch *et al.*, 2011), and available methods rely on direct path heuristics (Morgan and Higgs, 1998), or on upper bounds based on 2D projections of folding landscapes (Lorenz *et al.*, 2009; Senter *et al.*, 2015); **Analysis** of the evolution of concentrations through time, typically through numerical integration as provided by *treekin* (Wolfinger *et al.*, 2004).

The present work pertains to the *generation* step, arguably the most critical aspect of kinetics analysis. A first category of approaches, such that *RNASLOpt* (Li and Zhang, 2011) and *RNAsubopt* (Wuchty *et al.*, 1999), rely on an exhaustive enumeration of suboptimal structures within some predefined energy distance of the MFE. Popular alternatives, such as *RNAlocmin* (Kucharik *et al.*, 2014) and *RNAlocopt* (Lorenz and Clote, 2011), rely on some variation on Boltzmann-Gibbs sampling. Kucharik *et al.* (2014) have noted the difficulties of existing approaches relying on sampling to generate unique conformations, leading to the adoption by *RNAlocmin* of an adaptive heuristics similar to simulated-annealing called  $\xi$ -scheduling to increase the sample diversity as the Boltzmann ensemble of low-energy becomes saturated. Despite such specific efforts, as shown in Fig. 1, the number of distinct structures decreases as the number of sample increases, making it hard to reach alternative structures beyond a few kcal/mol of the MFE structure.



**Fig. 1.** Comparison of local minima production speed for the SV11 RNA switch L07337\_1 (115 nt). Experiment reproduced from Kucharik *et al.* (2014).

## 2 Approach

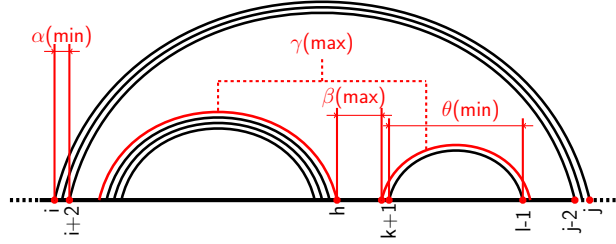
We introduce the concept of **non-redundant sampling** to study RNA kinetics, using **locally optimal secondary structures** as representative structures. Working with a reduced conformation space allows to mitigate the limitations of existing (redundant) sampling approaches. The problem of constructing all locally optimal secondary structures was addressed in Saffarian *et al.* (2012). Here, we describe an alternative generation algorithm which is efficient, and allows the specification of comprehensive structural restrictions (Section 2.2). We then define the first non-redundant sampling algorithm for locally optimal RNA structures (Section 2.3), allowing for the exploration of RNA folding landscape. The two algorithms are implemented within the standalone software *RNANR*.

### 2.1 Definitions

An **RNA sequence**  $w$  is a nucleotide sequence of length  $n$  over the alphabet  $\{A, C, G, U\}$ . The symbol at position  $i$  is denoted by  $w[i]$ . A **secondary structure**  $S$  is set of pairs of positions in  $w$ , called **base pairs**, that are pairwise **juxtaposed** or **nested**. Specifically, if two base pairs  $(i, j)$  and  $(k, \ell)$  are such that  $i \leq k$ , then either  $i < j < k < \ell$  or  $i < k < \ell < j$ . This definition implies that a secondary structure is non-crossing, or **pseudoknot-free**, and each position is involved in at most one base pair. As a consequence, it can be encoded by a *dot-parenthesis* expression, where each base pair is a pair of matching brackets and unpaired positions are represented by a dot. We also require that each pair  $(i, j)$  in  $S$  is **valid**, i.e.  $\{w[i], w[j]\}$  is in  $\{\{A, U\}, \{C, G\}, \{U, G\}\}$ . Given a base pair  $(x, y)$ , we denote  $h^p(x, y)$  the **helix** of length  $p$  stemming from  $(x, y)$ : this is the set of base pairs  $\{(x, y), \dots, (x + p - 1, y - p + 1)\}$ .

**Structural restrictions.** The set of secondary structures on a given sequence can be further restricted by enforcing additional constraints, giving rise to more realistic structures. Those restrictions include: a) the **minimum helix length**  $\alpha$ . In particular, isolated base pairs are forbidden for any value  $\alpha > 1$ ; b) the **maximum number of consecutive unpaired bases** in the structure  $\beta$ ; c) the **maximum number of branches** within a multiloop  $\gamma \geq 1$ , which defines the maximum allowed number of outermost base pairs within another base pair; d) the **minimum length of a hairpin loop**  $\theta$ . These parameters are illustrated on Fig. 2. The special case  $\alpha = 1$ ,  $\beta = n$ ,  $\gamma = n$  and  $\theta = 1$  corresponds to the whole set of secondary structures.

Subsequently, we will use  $S$  as a shorthand for this **search space**, i.e. the restriction of secondary structures that respect those parameters. Within



**Fig. 2. Graphical representation of the structural restrictions supported by RNANR.**  $\alpha$  is the lower bound on the size of helices.  $\beta$  and  $\theta$  is a minimum length of a hairpin.  $\gamma$  limits the maximum number of branches within multiloops. Finally,  $\delta$  is the maximum number of nucleotides in unpaired regions.

this search space, we define the **neighborhood** of a secondary structure  $S$  as the subset of secondary structures on  $w$  that can be obtained by adding or removing a single base pair in  $S$ .

**Energy models.** For a given secondary structure  $S$  on  $w$ , we associate a **free energy**  $E_S$ , computed with respect to a specific energy model. In this work, we consider two energy models. The first model is a simple base-pairing model where the energy is the number of base pairs. We call it the **Nussinov** model in the spirit of (Nussinov and Jacobson, 1980), even if structural restrictions introduced in the preceding paragraph significantly reduce the set of secondary structures. The second model is the 2004 version of the **Turner** thermodynamic model (Turner *et al.*, 1988; Turner and Mathews, 2010). For a given free energy model, we say that a secondary structure is **locally optimal** if, and only if, it has minimal free energy within its neighborhood. We respectively denote by  $\mathcal{L}_N$  and  $\mathcal{L}_T$  the **sets of locally optimal secondary structures (LOSSes)** with respect to the Nussinov and Turner energy models, and will respectively refer to them as Nussinov LOSSes and Turner LOSSes in the following.

**Thermodynamic concepts.** From a free-energy model, one computes the **Boltzmann factor**  $\mathcal{B}_s$  of a secondary structure as  $\mathcal{B}_s = e^{-\frac{E_s}{kT}}$  where  $E_s$  is the free energy of  $s$ ,  $k$  is the Boltzmann constant and  $T$  is the temperature in Kelvin. The **partition function**  $\mathcal{Z}$  is obtained by summing the Boltzmann factors of all the conformations in a set  $\mathcal{S}$ :

$$\mathcal{Z} = \sum_{s \in \mathcal{S}} \mathcal{B}_s \quad (1)$$

The **Boltzmann probability** of a given structure  $s$  in  $\mathcal{S}$  is then simply defined as  $P(s) = \frac{\mathcal{B}_s}{\mathcal{Z}}$ . Finally, given a set of structures  $\mathcal{T} \subset \mathcal{S}$ , we define the **coverage**  $c(\mathcal{T})$  as the accumulated Boltzmann probability in the set, also expressed as

$$c(\mathcal{T}) = \frac{\sum_{t \in \mathcal{T}} \mathcal{B}_t}{\mathcal{Z}} \quad (2)$$

## 2.2 Building LOSSes in the restrained Nussinov model.

In Saffarian *et al.* (2012), it is shown that locally optimal secondary structures without structural restrictions can be built from substructures that are maximal by juxtaposition, called here **flat structures** in short. We elaborate on this idea in order to account for the expressive set of structural restrictions introduced in Subsection 2.1.

**Flat structure.** For any interval  $[i, j]$  in  $w$ , a flat structure  $f$  is a sequence of juxtaposed (non-nested) helices which is **maximal**, meaning that it cannot be completed by a valid base pair between the positions left accessible by the helices. In other words, let  $x_1, y_1, \dots, x_\ell, y_\ell$  be positions in  $w$ , such

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18  
ACUCAGUUCGACGGUAGC

Helices (length  $\geq 2$ )

$((\dots)) \dots \dots \dots h^3(1, 7)$   
 $\cdot (\dots) \dots \dots h^2(2, 14)$   
 $\cdot (\dots) \dots \dots h^2(2, 17)$   
 $\cdot (\dots) \dots \dots h^2(3, 14)$   
 $\dots ((\dots)) \dots \dots h^3(6, 12)$   
 $\dots ((\dots)) \dots \dots h^3(6, 15)$   
 $\dots ((\dots)) \dots \dots h^3(6, 18)$   
 $\dots ((\dots)) \dots \dots h^3(8, 14)$   
 $\dots ((\dots)) \dots \dots h^2(11, 15)$   
 $\dots ((\dots)) \dots \dots h^2(14, 18)$

Flat structures

$((\dots)) \cdot ((\dots)) ((\dots)) \dots f_1 : h^2(1, 7) \oplus h^2(9, 13) \oplus h^2(14, 18)$   
 $((\dots)) \dots ((\dots)) \dots f_2 : h^2(1, 7) \oplus h^2(11, 15)$   
 $\cdot ((\dots)) ((\dots)) \dots f_3 : h^2(2, 6) \oplus h^2(7, 17)$   
 $\cdot ((\dots)) ((\dots)) \cdot ((\dots)) f_4 : h^2(2, 6) \oplus h^2(7, 11) \oplus h^2(14, 18)$   
 $\cdot ((\dots)) \dots ((\dots)) \dots f_5 : h^2(2, 17)$

**Fig. 3. Examples of flat structures.** For the structural parameters  $\alpha = 2, \delta = 3, \theta = 1$  and  $\gamma = 3$ , the set  $\mathcal{F}_{1,18}$  of flat structures associated with the interval  $[1, 18]$  consists of the five flat structures  $f_1 \dots f_5$ . Note that  $h^2(1, 7) \oplus h^2(8, 14)$  does not meet the condition on  $\gamma$ , as pos. 15..18 are left unpaired, and thus is not a valid flat structure.

that  $i \leq x_1 < y_1 < \dots < x_\ell < y_\ell \leq j$  and such that  $h^\alpha(x_k, y_k)$  is a valid helix for each  $k, 1 \leq k \leq \ell$ . This sequence of helices is a flat structure  $f$  on  $[i, j]$  if, and only if: for each secondary structure  $S$  on  $[i, j]$  containing  $f$ , if  $(x, y)$  is in  $S$  and not in  $f$ , then  $(x, y)$  is nested in  $(x_k, y_k)$  for some  $1 \leq k \leq \ell$ . We further assume that  $\ell \leq \theta, y_k - x_k - 2\alpha \geq \beta$  and  $x_{k+1} - y_k \leq \delta$  to meet the requirements on  $\beta, \delta$  and  $\theta$ . We denote by  $\oplus_{k=1}^\ell h^\alpha(x_k, y_k)$  such a flat structure, assuming that  $x_1 < \dots < x_\ell$ . We denote by  $\mathcal{F}_{i,j}$  the **complete set of flat structures** associated with the region  $[i, j]$  in  $w$ , as illustrated by Fig. 3. When an interval  $[i, j]$  is not associated with any flat structure, we have  $\mathcal{F}_{i,j} = \{\varepsilon\}$  ( $\varepsilon$  is the **empty** flat structure) whenever  $j - i + 1 \leq \delta$ , and  $\mathcal{F}_{i,j} = \emptyset$  (empty set) otherwise.  $\mathcal{F}_{i,j}$  can be computed using a dynamic programming scheme adapted from Section 3.1.2, Theorem 2 in Saffarian *et al.* (2012).

**A grammar for  $\mathcal{L}_N$ .** Now comes the crucial observation that any locally optimal secondary structure of  $\mathcal{L}_N$  can be built up from a set of flat structures, completed with **helix extensions**. Given a helix  $h^p(x, y)$  of length  $p$ , an extension is the addition of the valid base pair  $(x + p, y - p)$  to form  $h^{p+1}(x, y)$ . Following The dot-parenthesis notations for the set of all structures of  $\mathcal{L}_N$  associated with an RNA  $w$  can be modeled as a context-free language generated by the grammar  $G_w = (N, T, R, Y)$ , where

- $N := \{A_i^j, H_i^j; 1 \leq i < j \leq n\}$  is the set of non-terminal symbols.  $A_i^j$  represents all locally optimal substructures within the interval  $[i, j]$ , and  $H_i^j$  represents the choice between a helix extension with the base pair  $(i, j)$  or starting a new substructure on the same interval;
- $T := \{(\cdot), \cdot\}$  is the set of terminal symbols;
- $R$  is the set of production rules:

$$H_i^j \rightarrow (H_{i+1}^{j-1}) \text{ if } (i, j) \text{ valid for } w \text{ and } j - i + 1 \geq \theta \quad (\text{P1})$$

$$H_i^j \rightarrow A_i^j \quad (\text{P2})$$

$$A_i^j \rightarrow \cdot x_{1-i} \prod_{1 \leq k \leq \ell} (^\alpha H_{x_k+\alpha}^{y_k-\alpha})^\alpha \cdot x_{k+1}-y_k-1 \quad (\text{P3})$$

for each  $\oplus_{k=1}^\ell h^\alpha(x_k, y_k) \in \mathcal{F}_{i,j}$  such that  
 $(x_1 \neq i) \vee (y_1 \neq j)$  whenever  $(i, j) \neq (1, n)$ ,  
 and where  $x_{\ell+1} = j + 1$

Using classic language notations,  $\prod$  is the concatenation operator and  $l^i$  denotes  $i \geq 0$  copies of the letter  $l$ .  $A_i^j$  represents all locally

```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
ACUCAGUUCGACGGUAGC
(((.))) . ((.)) ((.))
(((.))) . . ((.)) . .
. ((.)) ((.)) . . ((.))
. ((.)) ((. ((.))) .
. ((.)) ((. ((.)) . .))
. ((. ((. ((.))) . .))

```

**Fig. 4.** All locally optimal secondary structures of  $\mathcal{L}_{\mathcal{N}}$  generated by the grammar  $G_w$  for the sequence, and using structural restrictions, described in Fig. 3.

optimal substructures within the interval  $[i, j]$  (P3), and  $H_i^j$  represents the choice between a helix extension with the base pair  $(i, j)$  (P1), or starting a new substructure on the same interval (P2);

- $Y := A_1^n$  is the start symbol.

This grammar has  $\Theta(n^2)$  non-terminal symbols and  $\Theta(n^2 + \sum_{i,j} |\mathcal{F}_{i,j}|)$  productions. The proof of its completeness with respect to  $\mathcal{L}_{\mathcal{N}}$  can be adapted from the proof of Theorem 1 in Saffarian *et al.* (2012).

The condition  $(x_1 \neq i) \vee (y_1 \neq j)$  in P3 ensures that  $G_w$  is unambiguous, and can be used as a conceptual template to derive other algorithms, *e.g.* to compute the partition function (Waldspühl and Clote, 2007) or base-pairing probabilities. Here, we use it on two related applications: exhaustive enumeration of structures and non-redundant statistical sampling of structures. While the former can be implemented in a straightforward fashion using recursive functions, the latter is more involved and is the object of the next section.

### 2.3 Non-redundant sampling algorithm

The large redundancy of stochastic sampling methods has been identified by previous studies as one of the major shortcomings of existing methods. *Ad hoc* heuristics, such as the  $\xi$ -scheduling technique inspired by simulated annealing, have been introduced to circumvent such limitations, sometimes at the cost of a control over the sampled distribution (Kucharik *et al.*, 2014). Here, we propose another approach based on an explicit avoidance of redundancy within the sampling, adapting principles introduced by Lorenz and Ponty (2013).

For the sake of simplicity, we illustrate those ideas by describing a uniform sampling algorithm for structures of  $\mathcal{L}_{\mathcal{N}}$  that are compatible with a given RNA sequence. Starting from the precomputed sets  $\mathcal{F}_{i,j}$  of flat structures, the algorithm computes the number of locally optimal structures for each interval  $[i, j]$  using DP equations. Those equations are isomorphic to the productions of grammar  $G_w$ .

$$\begin{aligned}
 h(i, j) &= \sum \begin{cases} h(i+1, j-1) & \text{if } (i, j) \text{ is valid;} \\ a(i, j) \end{cases} \\
 a(i, j) &= \prod_{h^\alpha(x_k, y_k) \in f} h(x_k + \alpha, y_k - \alpha), \\
 &\quad \text{for all } f \in \mathcal{F}_{i,j} \text{ and when } j - i + 1 \geq \theta.
 \end{aligned}$$

A **stochastic backtrack**, a concept independently introduced in enumerative combinatorics (Flajolet *et al.*, 1994; Denise *et al.*, 2010) and RNA bioinformatics (Ding and Lawrence, 2003), can then be used to generate elements of  $\mathcal{L}_{\mathcal{N}}$  uniformly. Such a procedure chooses at every step one of the possible productions of the grammar, with probability proportional to its contribution to the overall number/weights of words. It considers a triplet  $(i, j, m)$ , where  $[i, j]$  is the current interval and  $m$  is the current matrix, initially starting from  $(1, n, a)$ . At each step, it proceeds as follows, depending on the value of  $m$ :

- $m = h$ : Choose  $a(i, j)$  with probability  $a(i, j)/h(i, j)$  and backtrack over the triplet  $(i, j, a)$ , otherwise append the base-pair  $(i, j)$  to the output, and backtrack over  $(i+1, j-1, h)$ ;
- $m = a$ : Choose a flat structure  $f = (h^\alpha(x_k, y_k))_{k=1}^\ell \in \mathcal{F}_{i,j}$  with probability

$$p_f = \frac{\prod_{h^\alpha(x_k, y_k) \in f} h(x_k + \alpha, y_k - \alpha)}{a(i, j)},$$

append all the helices in the chosen  $f$  to the output, and backtrack over the triplets  $(x_k + \alpha, y_k - \alpha, h)$  (if any).

Note that the sets  $\mathcal{F}_{i,j}$  are explicitly computed, and so are the probabilities  $p_f$  of choosing any flat structure  $f$  during the backtrack. It is thus possible to order the flat structures in  $\mathcal{F}_{i,j}$  by decreasing probability, leading to a substantial speed-up during the backtrack.

**Non-redundant sampling.** The stochastic backtrack becomes much more involved in the presence of a predefined set of **forbidden structures**, *e.g.* singled out to avoid redundancy, as the probabilities of the backtrack on disjoint intervals can no longer be considered independent.

As a minimal illustration, consider two intervals  $I_1$  and  $I_2$  where local sets of substructures  $\{S_1, S'_1\}$  and  $\{S_2, S'_2, S''_2\}$  can be respectively chosen, leading to the generation of 6 different structures. Clearly, in the absence of forbidden sets, one simply needs to choose uniformly within each set to draw each structure with probability  $1/6$ , *i.e.* in the uniform distribution. However, if a given combination of structures has to be avoided, say  $S'_1 S_2$ , then the choice over  $I_1$  now influences the valid combinations, and thus the probabilities, of choosing a structure over  $I_2$ . Namely, choosing  $S'_1$  for  $I_1$  only allows access to 2 viable alternatives for  $I_2$ , while choosing  $S_1$  for  $I_1$  enables 3 alternatives over  $I_2$ . In order to be uniform, a stochastic backtrack must therefore choose  $S_1$  with probability  $3/5$ , and  $S'_1$  with probability  $2/5$ . Once chosen, the remaining choice is uniform over  $\{S_2, S'_2, S''_2\}$  if  $S_1$  is chosen (prob. =  $2/5 \times 1/2 = 1/5$ ), or over  $\{S'_2, S''_2\}$  (prob. =  $3/5 \times 1/3 = 1/5$ ) if  $S'_1$  is chosen.

More generally, in order to pick local alternatives in a way that is consistent with a predetermined distribution, one needs to access (or compute) the overall mass of forbidden structures that can be generated before and after the choice. The idea of Lorenz and Ponty (2013) consists in maintaining a dedicated data structure  $\mu$ , which enables efficient access to the overall count/weight  $\mu(T)$  of accessible forbidden structures from the current state  $f$  the backtracking stack  $T$ .

The modified **non-redundant backtrack** is similar in structure to the classic one, but uses different derivation probabilities, starting from a stack  $T = \{(1, n, a)\}$ . Let  $N(T) := \prod_{(i,j,m) \in T} m(i, j)$  denote the overall number/weight of structures accessible from a given state. At each iteration, it extracts a triplet  $(i, j, m)$  from  $T$ :

- If  $m = h$ , choose  $a(i, j)$  with probability 
$$\frac{a(i, j) \times N(T') - \mu(T' \cup \{(i, j, a)\})}{N(T) - \mu(T)}$$
 where  $T' := T - \{(i, j, m)\}$  and backtrack over  $T' \cup \{(i, j, a)\}$ , or otherwise backtrack over  $T' \cup \{(i+1, j-1, h)\}$ , after adding  $(i, j)$  to the output;
- If  $m = a$ , choose a flat structure  $f = (h^\alpha(x_k, y_k))_{k=1}^\ell \in \mathcal{F}_{i,j}$  with probability

$$p_f = \frac{\prod_{k=1}^{\ell} h(x_k + \alpha, y_k - \alpha) \times N(T') - \mu(T' \cup \{(x_l + \alpha, y_l - \alpha, h)\})}{N(T) - \mu(T)}$$

where  $T' := T - \{(i, j, m)\}$ , and backtrack over the stack  $T' \cup \{(x_k + \alpha, y_k - \alpha, h)\}_k$  after adding the chosen  $f$  to the output.

In practice, both the data structure  $\mu$  and the  $N(T)$  can be updated *on-the-fly* during the backtrack, so that the non-redundancy retains the same asymptotic complexity as the redundant one.

**Turner energy model and expressive structural restrictions.** The various contributions of the loops in the Turner energy model can easily be identified in the grammar. Namely, stacking pairs are generated by rule (P1), while rule (P2) generates terminal loops (hairpins) when  $f = \emptyset$ , internal loops and bulges when  $|f| = 1$ , or multiple loops when  $|f| > 1$ . Finally, the exterior loop corresponds to  $(i, j) := (1, n)$ .

This enables the incorporation of Boltzmann weights, based on the Turner energy model as weights in each of the DP equations. The incorporation of such weights during the stochastic (non-redundant) backtrack leads to an algorithm for Boltzmann sampling, whose details and (sketch of) proof of correctness are provided in Supp. Mat. 1.

### 3 Methods

All experiments were run on a laptop with Intel Core i7 5600 CPU equipped with a quad core at 2.6 GHz with 16GB of RAM under Ubuntu 16.04 LTS.

RNANR was implemented in C, and is freely available. It interfaces the RNALib, using the C++ API provided in the Vienna package (Lorenz *et al.*, 2011), to access the individual contributions of the 2004 version of the Turner energy model. The tests were done on a version compiled with gcc using GNU99 standard.

Gradient walks were performed using the `move_gradient` function of the ViennaRNA package (Lorenz *et al.*, 2011), which performs a gradient descent in the Turner energy landscape and return one of the closest local minima.

#### 3.1 Datasets

Three datasets were considered in our validation effort. The **uniform dataset** consists in random, uniformly-distributed, RNA sequences of length from 10 to 140 nt increasing by 10 nt, with 50 samples per length.

In order to assess the characteristics of real RNAs, we also gathered the **RNASTrand dataset**, which consists of the 154 RNA sequences of length between 120 and 170 nts downloaded from RNASTrand database Andronescu *et al.*, 2008, filtering out undefined symbols.

Finally, since currently available kinetics data is too scarce to allow for a quantitative comparison of tools, we created a data set of 250 **bistable sequences** of length 100 nt. A bistable RNA sequence presents two stable conformations differing by a sufficient number of base pairs. We generated random uniform sequences of length 100 nt, and retained only those whose most stable LOSS (MFE) had free-energy lower than  $-30 \text{ kcal.mol}^{-1}$  according to RNAEval (Lorenz *et al.*, 2011). Remaining sequences were then subjected to non-redundant sampling of 1 000 LOSSes using RNANR. We finally kept the sequences which, within the sampled set, featured an alternative **metastable** LOSS, differing by  $\geq 20$  base-pairs from the MFE structure, and having free-energy  $\leq 5 \text{ kcal.mol}^{-1}$  higher than the MFE.

#### 3.2 Program parameters

Unless noted otherwise, the settings of the different programs used in our comparisons are those described in this section.

For RNANR, the default mode is that of non-redundant sampling, with 20 samples. The structural restrictions include a minimum helix length  $\alpha = 3$ , a minimal base pair distance  $\theta = 3$ , a maximal unpaired region length  $\delta = 7$ , and a maximal number of helices in multiloop  $\beta = 4$ .

For RNALocopt the number of returned samples is the same and the temperature was set to 310.15K.

For RNALocopt, the suboptimality percentage was set to 100%, meaning that all LOSSes with energy ranging between the MFE and 0 are returned. Since we are interested in LOSSes independently of their stability, we set the `barrier_cutoff` to an arbitrarily large value of  $30 \text{ kcal.mol}^{-1}$  to speed up the computations by avoiding the computation of the energy barriers. Likewise, the number of top stable LOSSes can be chosen arbitrarily, here its value is set to 3.

RNALocmin was run using the second version of the built-in *adaptive search* script, referred to as `asearch` and coded in Python (Kucharik *et al.*, 2014). The parameters used are those by default, *i. e.* 10 000, 10 and 0.1 respectively for the number of samples per iteration, number of iterations and convergence parameter.

#### 3.3 Theoretical speedup of non-redundant sampling

We propose a closed-form formula to quantify the **speedup** factor induced by non-redundant sampling, *i. e.* the average number of occurrence of each unique samples. Let us consider a fixed sequence of  $i$  *unique* structures, and let  $R_i$  be the number of structures, generated by a redundant Boltzmann sampling algorithm, before returning a novel  $i + 1$ -th structure. It is easy to show that the expectation of  $R_i$  is:

$$\mathbb{E}(R_i) = \left( 1 - \left( \sum_{j=1}^i \frac{e^{-\frac{E_j}{kT}}}{\mathcal{Z}} \right) \right)^{-1} \quad (3)$$

where  $E_j$  is the energy of  $j$ -th secondary structure,  $k$  the Boltzmann constant and  $T$  the temperature in Kelvin. Since, for a given samples sequence, the  $R_j$  are independent, then the overall number of generations needed to obtain  $k$  distinct elements via redundant sampling is given by  $T(k) = k + \sum_{i=0}^{k-1} \mathbb{E}(R_i)$  and the speed-up factor is simply by  $T(k)/k$ .

#### 3.4 Time benchmarking RNANR against its competitors

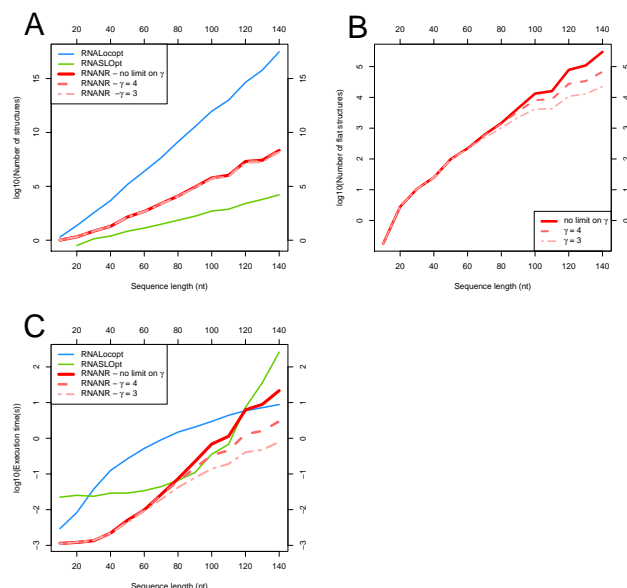
We reproduced, and report in Fig. 1, the benchmark of Kucharik *et al.* (2014) which compares the rates at which different sampling methods produce unique local minima. It focuses on the SV11 L07337\_1 RNA switch, a challenging 115 nts RNA whose landscape is particularly deep and steep. For RNANR, the total number of generated samples was 4 000. RNALocmin was run using both first and second version of `asearch` for maximum of 30 iterations. For both tools, the sampled structures are not necessary LOSSes with respect to the Turner energy model, so a gradient walk is performed, and duplicates were removed. The time spent by the gradient walk is added to the generation time in the benchmark. Finally, the total number of samples generated by RNALocopt was set to 6 000 000. Other parameters were the same as those specified above.

#### 3.5 Comparison of folding landscape analysis efficiency

To compare the quality of sampled sets of structures, we considered an artificial bistable dataset described in Section 3.1, and produced representative sets of structures using the four programs mentioned in Section 3.2. We analyzed sampled sets using a standard kinetic analysis pipeline based on an estimation of energy barriers for each pair of structures, followed by a numerical integration using `treekin`.

For each sequence in the bistable dataset, each program was used to generate  $n_{sam} = 50, 75$  and 100 samples (output truncated if necessary). Gradient walks were performed to each sample set, and duplicated Turner LOSSes were removed. As a control, we also included the results of RNASubopt -e, adjusting  $\Delta E$  to return at least  $n_{sam}$  LOSSes. Next, we estimated the energy barriers using the single path heuristics implemented by the `findpath` tool (Flamm *et al.*, 2001). Due to the high computational cost of this operation, we restricted it to pairs of states having base-pair distance at most  $bp_{lim}$ . For each sequence, the value of  $bp_{lim}$  was set in





**Fig. 5. Comparison of RNANR with RNALocopt and RNASLOpt.** A) Number of structures returned by each program and in case of RNANR, for different upper limits of  $\gamma$  B) Number of flat structures returned by RNANR for different values of  $\gamma$  C) Benchmark for different programs, for the same cases as A).

such a way that landscapes sampled by all tools were connected (with the possible exception of RNASubopt).

We then used Arrhenius rule to estimate the transition rate  $k_{i \rightarrow j}$  from a structure  $i$  to a structure  $j$  as

$$k_{i \rightarrow j} = e^{\frac{-(E_{B_{i \rightarrow j}} - E_i)}{kT}},$$

where  $E_{B_{i \rightarrow j}}$  is the free energy of the barrier,  $E_i$  the free-energy of state  $i$ ,  $k$  the Boltzmann constant and  $T$  the absolute temperature. The rate between unconnected LOSSes was set to be 0. These rates were used to generate transition matrix. For each sampled set, we identified a minimal free-energy (MFE) and a metastable LOSSes as the most similar structures to the reference ones (see SubSection 3.1) for the sequence.

Finally, we considered a scenario where the starting concentration of the metastable structure (or its closest neighbor in the sampled set) is set to 1. We used `treekin` to determine the evolution of the concentration of all LOSSes in the sampled set. Finally, we report the **switching time**, *i.e.* the time at which the MFE structure eventually achieves higher concentration becomes more frequent than the metastable structure.

## 4 Discussion

### 4.1 Validating Nussinov LOSSes as key landmarks of kinetic landscapes

*Nussinov LOSSes are less numerous than Turner LOSSes.* We compared the numbers of LOSSes returned by RNANR, RNALocopt and RNASLOpt. For each sequence in the uniform dataset, these sequences were subjected to runs of each software. For RNANR, we performed three runs, each with different value of  $\gamma$  (no limit, 4 and 3 respectively). For RNALocopt and RNASLOpt we performed one run for each with parameters as specified in 3.2. The values of each run were then aggregated by the sequence length. The results are shown on Fig. 5.

Fig. 5A shows the evolution of number of LOSSes for different programs in function of sequence length. We observe that RNASLOpt returns

the smallest number of results. This is consistent with the primary objective of Li and Zhang (2011) to reduce the number of structures as a way to reduce the complexity. Besides aggressive structural constraints, RNASLOpt returns only LOSSes that have negative free energy, which is not the case for both RNANR and RNALocopt. On the other hand, RNANR presents a lower number of solutions than RNALocopt. This stems both from the reduction of search space by RNANR due to the structural restrictions, and to our restriction to saturated structures while RNALocopt arguably considers a larger neighborhood which includes unsaturated structures (Lorenz and Clote, 2011). Our reduced search space, while not as aggressive as that of RNASLOpt, leads to a substantial reduction of the complexity, both theoretically and practically, as shown in Fig. 5.

Of interesting note is the comparison between the numbers of flat structures returned by RNANR for different values of  $\gamma$  (Fig. 5B). While the number of flat structures noticeably decreases for lower values of  $\gamma$ , the number of LOSSes does not seem to be particularly affected (Fig. 5A). This could be explained by the fact the excluded flat structures participate in few of the complete LOSSes, as substantiated by the fact that, for shorter sequences, the formation of multiloops with high number of branches is improbable. This interpretation is consistent with an analysis of RNAStrand structures (Fig. 6), which shows that very few (0.1%) RNAs of length under 140 nts features multiloops of degree greater than 4 branches. For shorter sequences, it thus seems reasonable to limit  $\gamma$ , which results in lowered time complexity. Naturally, as shown by Fig. 6, this ceases to be the case for longer sequences. While for sequences shorter 300 nt the number of ignored structures for  $\gamma = 4$  is still low (1.92%), for sequences shorter than 400 nt the number of multiloops with at least 5 branches accounts for 15.82% of all structures. Overall, we found that setting  $\gamma$  to 5 constitutes a reasonable tradeoff, reducing the computation time while keeping the number of undetected LOSSes reasonable.

*Nussinov LOSSes are very close to Turner LOSSes.* We used gradient walks to determine how distant are Nussinov LOSSes, output by RNANR, to their counterpart in the Turner energy model. For each sequence in the RNAStrand dataset, a non-redundant sampling of 1000 distinct structures was performed by RNANR. These structures were then subject to a gradient descent, resulting in a Turner LOSS. We tracked the modifications, both in term of energy and base pairs, induced by the walk. Our results are summarized in Table 1.

Overall, 52.49% of the Nussinov LOSSes were already local minima with respect to the Turner energy model. Moreover, our analysis shows that, from a Nussinov LOSS, it is sufficient to add or remove an average of 0.703 base pairs, contributing an average 0.547 kcal/mol, to reach a Turner LOSS. A further analysis reveals differing behaviors of whether or not the Turner LOSS, obtained as the outcome of the gradient descent, belongs to the restricted search space.

Finally, it is worth stressing that the structures obtained after the gradient walk are overwhelmingly unique (99.1%), suggesting a homogenous coverage of the Turner LOSSes by the Nussinov LOSSes. We thus conclude that the structures generated by RNANR can reliably used as representatives for Turner LOSSes within the folding landscape.

### 4.2 Efficiency of sampling methods for LOSSes

A first time benchmark, whose results are given on Fig. 5C was performed on our uniform dataset. We observe that, without limiting  $\gamma$ , RNANR is fastest for sequences of length under 80 nt. Around this length, it is briefly matched by RNASLOpt, whose execution time increases spectacularly around 110nts. RNALocopt is faster than RNASLOpt or RNANR (no limitation on  $\gamma$ ) for longer sequences (more than 120 nt) due to the polynomial nature of the underlying algorithm. However, for values of  $\gamma$  set to 3 or 4, the execution time of RNANR becomes polynomial, and RNANR becomes considerably faster practice than its competitors. Of course, one needs to

	Samples% avg (std.dev)	$\Delta\Delta G$ avg (std.dev)	Base pair dist. avg (std.dev)
Within search space	59.57% (21.00)	0.071 (0.309)	0.129 (0.289)
Outside search space	40.42% (21.00)	1.248 (0.925)	1.550 (0.619)
Global average	100.00% (–)	<b>0.547</b> (0.817)	<b>0.703</b> (0.757)

Table 1. **Discrepancy between Nussinov and Turner LOSSes.** For our RNASTrand dataset, 1 000 Nussinov LOSSes were generated. For each structure, one of the closest Turner LOSS was determined using a gradient descent. On average, a Nussinov LOSS is distant by  $\leq 0.547\text{kcal.mol}^{-1}$ , and by 0.7 base pairs, from its closest Turner LOSS.

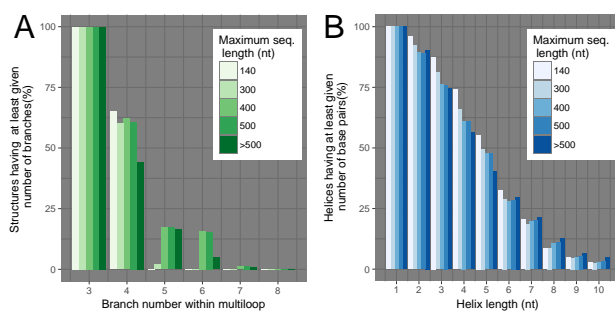


Fig. 6. **Rationale for our restricted search space.** Effect of structural limitations on structure counts obtained from RNASTrand database (Andronescu et al., 2008). A) Proportion of structures having maximal multi loop branches below a given threshold. A branch number within multi loop is the number of stems sorting from a multiloop and is equal to  $\gamma + 1$ . B) Proportion of helices having at least given length.

exercise caution before setting restrictions that would lead to the omission of important conformations, and it would probably not be wise to set  $\gamma = 3$  for RNAs beyond 80nts. However, setting  $\gamma = 4$  makes RNANR faster than any other tested software, while only missing a negligible proportion of existing conformations (cf Fig. 6, for multiloop branch number equal to 6) for sequences beyond 300nts.

A second basic time benchmark, described in Section 3.4c and in Fig. 1, measures the rate of production of distinct Turner LOSSes generated using different software. We observe that RNANR returns more unique LOSSes in a given time, even when including the time for precomputations and gradient descents, than both versions of RNALocmin and RNALocopt (note that RNASLOpt does not perform sampling). This is mainly due to the redundancy within the returned samples, as indicated by the diminishing production speeds for both methods. The second version of asearch uses an alternative strategy for  $\xi$ -scheduling which increases its number of samples linearly between iterations, and seems to eventually perform better than its initial version. However it starts slower, proving that the redundancy is still an issue.

#### 4.3 Non-redundant sampling allows a deeper exploration of kinetic landscapes

The main new contribution of RNANR is its non-redundant sampling algorithm, which allows to obtain a set of locally optimal secondary structures, each of whom appears at most once. This method of sampling has its advantages which will be discussed in this and next subsection. The more obvious one, discussed here, is the fact it allows to obtain higher number of different samples faster. Instead of sampling same structures with free energy close to the minimal free-energy over and over, it consecutively picks up the structures with higher energies which were not sampled previously.

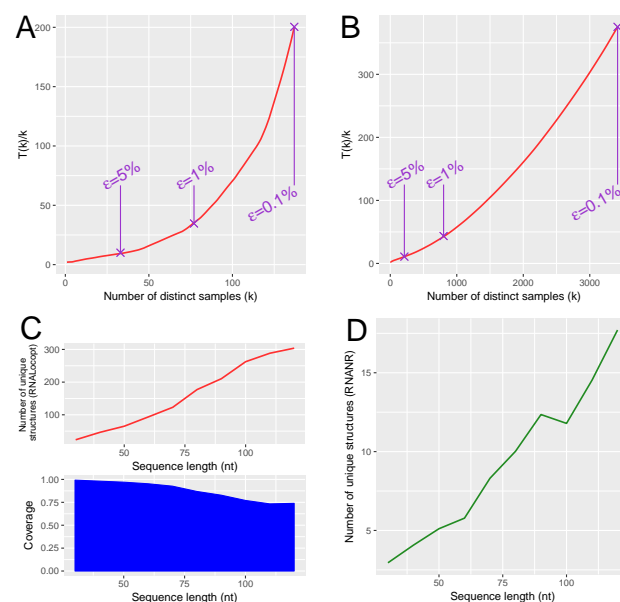


Fig. 7. **Comparison of classical and non-redundant sampling.** A) Theoretical speedup  $T(k)/k$  using non-redundant sampling when compared to redundant sampling for a 5S ribosomal RNA of *Thermoplasma acidophilum* (123 nt). B) Same test for a telomerase RNA of *Tetrahymena silvana* (154 nt). Purple points indicate the coverage of  $1 - \epsilon$ , for  $\epsilon = 5\%$ ,  $1\%$  and  $0.1\%$  respectively. C) Number of unique LOSSes and coverage  $c$  from 1 000 samples generated by RNALocopt on our uniform artificial dataset. The number of unique LOSSes, generated using RNANR in order to achieve a coverage  $c$ , is plotted in D.

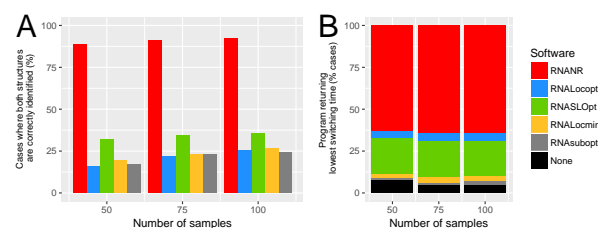
To demonstrate this point, we first computed the value of speedup  $T(k)/k$  as defined in Section 3.3 on two sequences: a 123 nts 5S ribosomal RNA of *Thermoplasma acidophilum*, and a 154 nts telomerase RNA of *Tetrahymena silvana*. A non-redundant sampling was performed, until a coverage of 0.99% was achieved, and the evolution of the speedup was calculated. The results are shown on Fig. 7 A and B. The purple dots mark the points with  $c = 1 - \epsilon$  for  $\epsilon = 5\%$ ,  $1\%$  and  $0.1\%$  for the final coverage.

The value of  $T(k)/k$  increases with  $k$ , which is expected since the probability of generating a novel structure decreases with each iteration. The speedup thus becomes more important for higher values of  $c$  (or lower values of  $\epsilon$ ), where the probability of a new unique structure almost vanishes. This means that the structures with higher free energies are considerably easier to generate by non-redundant sampling than by their redundant counterpart, since their initial probability increases along with the sampling. Moreover, different sequences exhibit different evolutions for  $c$ , as shown on Fig. 7 A and B, depending on the concentration of the Boltzmann distribution around a few conformations. This means that higher values of  $c$  will be more difficult to attain for some sequences than for others, and for these cases the usage of non-redundant sampling might prove more advantageous.

Our second test consisted in creating the samples using redundant generation, and comparing these results with the non-redundant sampler of RNANR. For this purpose, we used RNALocopt to generate 1 000 LOSSes for each sequence of our uniform dataset. For each set of structures, the number of unique samples and the coverage  $c$  were computed. RNANR was then used to generate a set of structures achieving the same coverage  $c$ . The averaged number of structures for each length was reported, leading to the values shown in Fig. 7 C and D.

While the number of unique structures returned by RNALocopt increases with the sequence length, the coverage value  $c$  diminishes. This is caused by the increasing number of local minima for longer sequences, which results in lower Boltzmann probabilities for individual structures.





**Fig. 8. Efficiency of analysis performed by various software on bistable structures.** A. Proportion of artificial bistable sequences for which both structures are found. B. Frequency of lowest switch time predicted using different values of  $n_{sam}$ .

This means that, to attain a given coverage  $c$ , a sharply increasing number of LOSSES must be generated for longer sequences, leading to a higher number of repeats. Fig. 7 D shows that the number of samples necessary to achieve identical  $c$  by RNANR as the one achieved by RNALocopt is considerably lower. This partially stems from the fact that while RNALocopt encompasses the entirety of Turner LOSSES, RNANR explores a much more drastically reduced search space, and thus requires less samples to achieve a given coverage. On the other hand, this also means that a target coverage is achieved faster while generating most of interesting structures with the correct parameter settings, meaning that RNANR can considerably speed up and simplify the analysis of the folding landscape.

#### 4.4 Non-redundant sampling of Turner LOSS enables a faster, more accurate analysis of RNA kinetics

The main interest of non-redundant sampling is, besides its inherent speedup, its capacity to dig deeper within the space of suboptimal structures when approximating the folding landscape of a given RNA. In this final validation, we tested whether this increased diversity translates into sampled sets of higher quality, leading to more accurate kinetics analysis.

More specifically, we evaluated the capacity of a simple, *real-life*, analysis pipeline to estimate the switching time of bistable artificial RNAs from samples of small size, generated using RNANR, RNALocmin, RNALocopt and RNASubopt. Details are described in Section 3.5, its results are shown in Table 8, and it is illustrated in Fig. 9.

First, we observe that RNANR detects structures that are similar to both the MFE and metastable LOSSES more consistently than its competitors. This is not overly surprising, since these two structures were initially identified from an independent execution of RNANR (albeit from a much larger sampled set, see Section 3.1). This however means that RNANR generates sets of structures that, while not strictly overlapping, represent the main dominant conformations even for small sampled sets, and may be used for reproducible further analysis. RNALocmin and RNALocopt both suffered from redundancy, and generally failed to identify the two dominant conformation for about  $\sim 80\%$  of the bistable RNAs. RNASLOpt exhibit the same tend, probably due to an aggressive filtering of LOSSES.

Then we compared the switching time, defined in Section 3.5, as predicted by our pipeline from the different sampled sets. We reasoned that, since both undersampled landscapes and single-path heuristics lead to an overestimation of energy barriers, a good sampled set, by populating important energy basins, would be associated to a fast perceived kinetics, *i.e.* a fast switching time. On the other hand, a set of scattered, or highly similar, structures would practically disconnect the landscape, leading to slow predicted kinetics. Faster predicted switching times therefore indicate better approximate folding landscapes.

Our results, summarized in Fig. 8, show that using RNANR leads to the shortest switching time  $\sim 70\%$  of the time, irrespectively of the number of samples. RNASLOpt is a clear second, and dominates other tools with respect to the switching time in about 20% of the sequences. While high

computational demands, on both the design and analysis tasks, disallowed us to repeat this on larger bistable sequences, we expect this trend to carry for larger sequences and sets.

## Acknowledgements

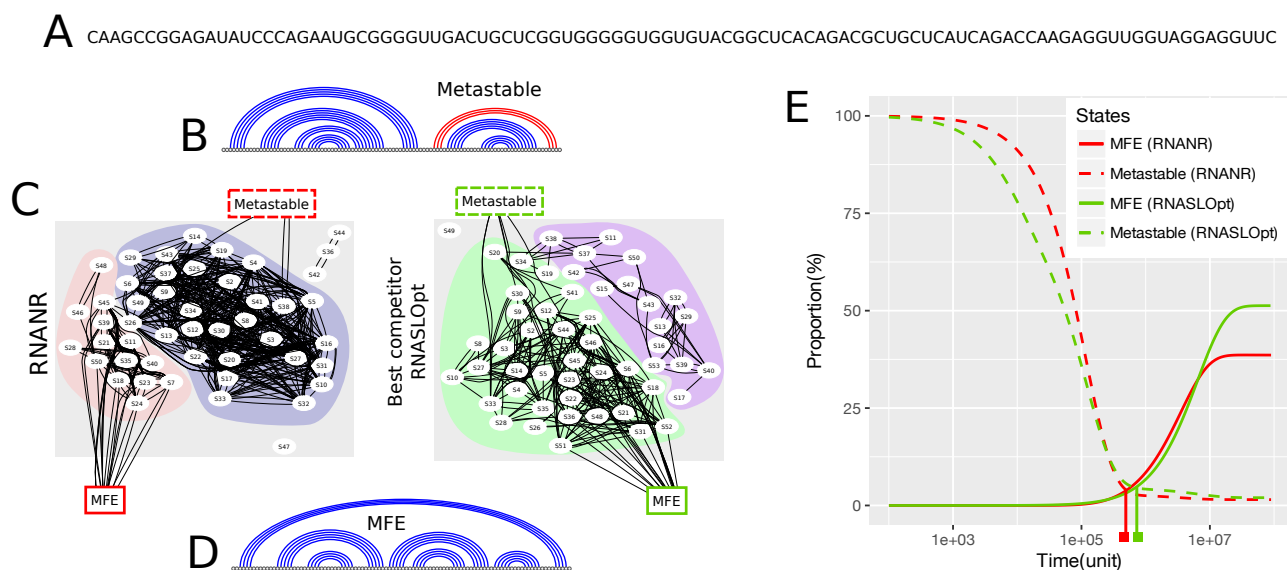
The authors wish to thank Ronny Lorenz and Gregor Entzian for their invaluable help in interfacing the versatile Vienna RNAlib, and Ivo Hofacker for his feedback and suggestions at an earlier stage of this work.

## Funding

This work has been supported by the RNALands project, jointly funded by the French *Agence Nationale de la Recherche* (ANR-14-CE34-0011) and the Austrian *Fonds zur Förderung der wissenschaftlichen Forschung* (FWF-I-1804-N28).

## References

- Akutsu, T. (2000). Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots. *Discrete Appl Math*, **104**(1-3), 45–62.
- Andronescu, M., Bereg, V., Hoos, H. H., and Condon, A. (2008). RNA strand: the RNA secondary structure and statistical analysis database. *BMC Bioinf*, **9**, 340.
- Badelt, S., Hammer, S., Flamm, C., and Hofacker, I. L. (2015). Chapter eight - thermodynamic and kinetic folding of riboswitches. In S.-J. Chen and D. H. Burke-Aguero, editors, *Computational Methods for Understanding Riboswitches*, volume 553 of *Methods in Enzymology*, pages 193 – 213. Academic Press.
- Baumstark, T., Schröder, A. R., and Riesner, D. (1997). Viroid processing: switch from cleavage to ligation is driven by a change from a tetraloop to a loop e conformation. *The EMBO journal*, **16**, 599–610.
- Cech, T. R. (2015). RNA world research-still evolving. *RNA (New York, N.Y.)*, **21**, 474–475.
- Cruz, J. A. and Westhof, E. (2011). Sequence-based identification of 3d structural modules in RNA with rmdetect. *Nat Methods*, **8**, 513–521.
- Danilova, L. V., Pervouchine, D. D., Favorov, A. V., and Mironov, A. A. (2006). RNA-kinetics: a web server that models secondary structure kinetics of an elongating RNA. *J Bioinform Comput Biol*, **4**, 589–596.
- Denise, A., Ponty, Y., and Termier, M. (2010). Controlled non-uniform random generation of decomposable structures. *Theoretical Computer Science*, **411**(40-42), 3527–3552.
- Ding, Y. and Lawrence, E. (2003). A statistical sampling algorithm for RNA secondary structure prediction. *Nucleic Acids Res*, **31**(24), 7280–7301.
- Flajolet, P., Zimmermann, P., and Van Cutsem, B. (1994). Calculus for the random generation of labelled combinatorial structures. *Theoretical Computer Science*, **132**, 1–35. A preliminary version is available in INRIA Research Report RR-1830.
- Flamm, C. and Hofacker, I. L. (2008). Beyond energy minimization: approaches to the kinetic folding of RNA. *Monatshefte für Chemie - Chemical Monthly*, **139**(4), 447–457.
- Flamm, C., Fontana, W., Hofacker, I. L., and Schuster, P. (2000). RNA folding at elementary step resolution. *RNA (New York, N.Y.)*, **6**, 325–338.
- Flamm, C., Hofacker, I. L., Maurer-Stroh, S., Stadler, P. F., and Zehl, M. (2001). Design of multistable RNA molecules. *RNA (New York, N.Y.)*, **7**, 254–265.
- Flamm, C., Hofacker, I. L., Stadler, P. F., and Wolfinger, M. T. (2002). Barrier trees of degenerate landscapes. *Zeitschrift für Physikalische Chemie*, **216**(2), 155.
- Isambert, H. (2009). The jerky and knotty dynamics of RNA. *Methods (San Diego, Calif.)*, **49**, 189–196.
- Kucharik, M., Hofacker, I. L., Stadler, P. F., and Qin, J. (2014). Basin hopping graph: a computational framework to characterize RNA folding landscapes. *Bioinformatics (Oxford, England)*, **30**, 2009–2017.
- Kushwaha, M., Rostain, W., Prakash, S., Duncan, J. N., and Jaramillo, A. (2016). Using RNA as molecular code for programming cellular function. *ACS synthetic biology*, **5**, 795–809.
- Li, H., Labeau, T. H., and Leong, K. W. (2011). Nucleic acid-based nanoengineering: novel structures for biomedical applications. *Interface focus*, **1**, 702–724.
- Li, Y. and Zhang, S. (2011). Finding stable local optimal RNA secondary structures. *Bioinformatics (Oxford, England)*, **27**, 2994–3001.
- Lorenz, R., Flamm, C., and Hofacker, I. L. (2009). 2d projections of RNA folding landscapes. In I. Grosse, S. Neumann, S. Posch, F. Schreiber, and P. F. Stadler,



**Fig. 9. Illustration of bistable RNA analysis.** Starting from an artificial 100nts bistable RNA (A), metastable (B) and MFE (D) states are identified, along with key landmarks of the kinetic landscapes, using the non-redundant sampling of RNANR (C, left) and RNASLOpt (C, right). Due to the stochastic nature of the landscape reconstruction methods, the metastable structure identified by RNANR has 3 extra base pairs (B, colored in red). Numerical integration of the master equation using *Treekin* (D) predicts a shorter switching time for RNANR kinetics landscape than its competitors, suggesting a better coverage of important kinetic intermediate structures.

- editors, *German Conference on Bioinformatics 2009*, volume 157 of *Lecture Notes in Informatics*, pages 11–20, Bonn. Gesellschaft f. Informatik.
- Lorenz, R., Bernhart, S. H., Höner Zu Siederdisen, C., Tafer, H., Flamm, C., Stadler, P. F., and Hofacker, I. L. (2011). ViennaRNA package 2.0. *Algorithms for molecular biology : AMB*, **6**, 26.
- Lorenz, W. A. and Clote, P. (2011). Computing the partition function for kinetically trapped RNA secondary structures. *PLoS One*, **6**, e16178.
- Lorenz, W. A. and Ponty, Y. (2013). Non-redundant random generation algorithms for weighted context-free grammars. *Theoretical Computer Science*, **502**, 177–194.
- Mañuch, J., Thachuk, C., Stacho, L., and Condon, A. (2011). Np-completeness of the energy barrier problem without pseudoknots and temporary arcs. *Natural Computing*, **10**(1), 391–405.
- Mathews, D. H. (2004). Using an RNA secondary structure partition function to determine confidence in base pairs predicted by free energy minimization. *RNA (New York, N.Y.)*, **10**, 1178–1190.
- McCaskill, J. S. (1990). The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, **29**, 1105–1119.
- Miao, Z., Adamiak, R. W., Blanchet, M.-F., Boniecki, M., Bujnicki, J. M., Chen, S.-J., Cheng, C., Chojnowski, G., Chou, F.-C., Cordero, P., Cruz, J. A., Ferré-D'Amaré, A. R., Das, R., Ding, F., Dokholyan, N. V., Dunin-Horkawicz, S., Kladwang, W., Krokhotin, A., Lach, G., Magnus, M., Major, F., Mann, T. H., Masquida, B., Matelska, D., Meyer, M., Peselis, A., Popena, M., Purzycka, K. J., Serganov, A., Stasiewicz, J., Szachniuk, M., Tandon, A., Tian, S., Wang, J., Xiao, Y., Xu, X., Zhang, J., Zhao, P., Zok, T., and Westhof, E. (2015). RNA-puzzles round ii: assessment of RNA structure prediction programs applied to three large RNA structures. *RNA (New York, N.Y.)*, **21**, 1066–1084.
- Morgan, S. R. and Higgs, P. G. (1998). Barrier heights between ground states in a model of RNA secondary structure. *J Phys A: Math Gen*, **31**(14), 3153.
- Nawrocki, E. P., Burge, S. W., Bateman, A., Daub, J., Eberhardt, R. Y., Eddy, S. R., Floden, E. W., Gardner, P. P., Jones, T. A., Tate, J., and Finn, R. D. (2015). Rfam 12.0: updates to the RNA families database. *Nucleic Acids Res*, **43**, D130–D137.
- Nussinov, R. and Jacobson, A. B. (1980). Fast algorithm for predicting the secondary structure of single-stranded RNA. *Proc Natl Acad Sci U S A*, **77**, 6309–6313.
- Saffarian, A., Giraud, M., de Monte, A., and Touzet, H. (2012). RNA locally optimal secondary structures. *Journal of computational molecular cell biology*, **19**, 1120–1133.
- Schultes, E. A. and Bartel, D. P. (2000). One sequence, two ribozymes: implications for the emergence of new ribozyme folds. *Science (New York, N.Y.)*, **289**, 448–452.
- Senter, E., Dotu, I., and Clote, P. (2015). RNA folding pathways and kinetics using 2d energy landscapes. *J Math Biol*, **70**, 173–196.
- Sharova, L. V., Sharov, A. A., Nedorezov, T., Piao, Y., Shaik, N., and Ko, M. S. H. (2009). Database for mRNA half-life of 19 977 genes obtained by dna microarray analysis of pluripotent and differentiating mouse embryonic stem cells. *DNA research : an international journal for rapid publication of reports on genes and genomes*, **16**, 45–58.
- Sheikh, S., Backofen, R., and Ponty, Y. (2012). Impact of the energy model on the complexity of RNA folding with pseudoknots. In J. Kärkkäinen and J. Stoye, editors, *Combinatorial Pattern Matching*, volume 7354 of *Lecture Notes in Computer Science*, pages 321–333, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Smola, M. J., Rice, G. M., Busan, S., Siegfried, N. A., and Weeks, K. M. (2015). Selective 2'-hydroxyl acylation analyzed by primer extension and mutational profiling (shape-map) for direct, versatile and accurate RNA structure analysis. *Nat Protoc*, **10**, 1643–1669.
- Tinoco, I. and Bustamante, C. (1999). How RNA folds. *J Mol Biol*, **293**(2), 271 – 281.
- 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.
- Turner, D. H., Sugimoto, N., and Freier, S. M. (1988). RNA structure prediction. *Annu Rev Biophys Chem*, **17**, 167–192.
- Waldispühl, J. and Clote, P. (2007). Computing the partition function and sampling for saturated secondary structures of RNA, with respect to the turner energy model. *Journal of computational biology : a journal of computational molecular cell biology*, **14**, 190–215.
- Watters, K. E., Strobel, E. J., Yu, A. M., Lis, J. T., and Lucks, J. B. (2016). Cotranscriptional folding of a riboswitch at nucleotide resolution. *Nature structural & molecular biology*, **23**, 1124–1131.
- Wilkinson, K. A., Gorelick, R. J., Vasa, S. M., Guex, N., Rein, A., Mathews, D. H., Giddings, M. C., and Weeks, K. M. (2008). High-throughput shape analysis reveals structures in hiv-1 genomic RNA strongly conserved across distinct biological states. *PLoS Biol*, **6**, e96.
- Wolfinger, M. T., Svrcek-Seiler, A., Flamm, C., Hofacker, I. L., and Stadler, P. F. (2004). Efficient computation of RNA folding dynamics. *J. Phys. A: Math*, **37**.
- Wuchty, S., Fontana, W., Hofacker, I., and Schuster, P. (1999). Complete suboptimal folding of RNA and the stability of secondary structures. *Biopol.*, **49**, 145–164.
- Xayaphoummine, A., Viasnoff, V., Harlepp, S., and Isambert, H. (2007). Encoding folding paths of RNA switches. *Nucleic Acids Res*, **35**, 614–622.