RNA inverse folding can be solved in linear time for structures without isolated stacks or base pairs

Théo Boury, Laurent Bulteau, Yann Ponty

RNA inverse folding can be solved in linear time for structures without isolated stacks or base pairs

Théo Boury[1], Laurent Bulteau[2*], Yann Ponty[1*]

[1*]Laboratoire d'Informatique de l'Ecole Polytechnique (LIX CNRS UMR 7161), France, Institut Polytechnique de Paris, 1 Rue Honoré d'Estienne d'Orves, Palaiseau, 91120, France.

[2*]Laboratoire d'Informatique Gaspard Monge (LIGM CNRS UMR 8049), France, Université Gustave Eiffel, 5 Boulevard Descartes – Champs sur Marne, Marne La Vallée, 77454, France.

*Corresponding author(s). E-mail(s): laurent.bulteau@univ-eiffel.fr; yann.ponty@lix.polytechnique.fr;

Abstract

Inverse folding is a classic instance of negative RNA design which consists in finding a sequence that uniquely folds into a target secondary structure with respect to energy minimization. A breakthrough result of Bonnet *et al.* shows that, even in simple base pairs-based (BP) models, the decision version of a mildly constrained version of inverse folding is NP-hard.

In this work, we show that inverse folding can be solved in linear time for a large collection of targets, including every structure that contains no isolated BP and no isolated stack (or, equivalently, when all helices consist of 3^+ base pairs). For structures featuring shorter helices, our linear algorithm is no longer guaranteed to produce a solution, but still does so for a large proportion of instances.

Our approach introduces a notion of modulo $\boldsymbol{m}$-separability, generalizing a property pioneered by Hales *et al.* Separability is a sufficient condition for the existence of a solution to the inverse folding problem. We show that, for any input secondary structure of length $\boldsymbol{n}$, a modulo $\boldsymbol{m}$-separated sequence can be produced in time $\boldsymbol{\mathcal{O}(n\, m\, 2^m)}$ anytime such a sequence exists. Meanwhile, we show that any structure consisting of 3^+ base pairs is either trivially non-designable, or always admits a modulo-**2** separated solution. Solution sequences can thus be produced in linear time, and even be uniformly generated within the set of modulo-**2** separable sequences.

Keywords: RNA structure, String Design, Parameterized Complexity, Uniform Sampling

1 Introduction

RNA inverse folding is a fascinating algorithmic problem which, given a target secondary structure T, consists of designing one or several sequences, all of which should uniquely fold into the target T according to a reference folding prediction algorithm. Considering a folding prediction algorithm as a mathematical function $\Phi : \{A, C, G, U\}^\star \to \mathcal{S} \cup \{\perp\}$ mapping an RNA sequence to a unique predicted structure (or $\perp$ if equally likely alternatives exist), inverse folding can be abstracted as the search for a preimage $w \in \Phi^{-1}(T)$ of the target structure T. This naturally generalizes into a variety of design tasks which, given a predictive algorithm implementing a function Φ, aim to create one or multiple instances predicted to behave in a certain way. Such a formulation is, in general, overly broad (*e.g.* it encompasses the concept of one-way functions in cryptography) to inspire reasonable hopes for a general solution. Still, a restriction of the inverse problem to certain types of computable functions/algorithms (*e.g.* amenable to dynamic programming) appears realistic and generally relevant to (synthetic) biology, yet poorly studied to this day.

In the specific case of RNA, despite being the object of substantial attention since its formal introduction in the early 1990s [1], the complexity of RNA inverse folding has remained elusive for almost three decades. A generalization of RNA inverse folding, including the energy model as part of the input, was shown to be NP-hard by Schnall-Levin *et al.* [2]. However, their reductions critically relied on (ab)using the energy model to encode a 3SAT instance, leaving the hardness of the problem largely open for a fixed energy model. The classic complexity of inverse folding was only settled, in 2020, when Bonnet *et al.* [3] finally showed the NP-hardness of RNA folding in a classic base pairs maximization setting. Such computational intractability (retrospectively) legitimizes a very large quantity of heuristic or exponential-time methods, based on local search [1, 4–7], bio-inspired metaheuristics [8–11], global sampling [12, 13], constraint programming [14, 15] and, more recently, neural networks-inspired generative models [16].

In parallel to complexity studies, Hales *et al.* [17] revisited the problem from a structural angle, attempting to characterize designable or undesignable families of secondary structures. The authors showed that saturated structures, having all positions paired, are designable if and only if their multiloop degrees do not exceed 4. They also introduced a notion of separability, a sufficient, yet not necessary in general, condition for a sequence to be a design for a given target. This notion allowed them to show that any target structure either features an occurrence of a locally-undesignable motif $\{m_{3\bullet}, m_5\}$, or can always be transformed into a separable structure by adding at most one base pair per helix. More strikingly, they proposed linear-time algorithms for producing a single solution for each characterized class of designable structures, painting a – puzzling – contrasted picture of general hardness (as per Bonnet *et al.* [3]) and practical facility for inverse folding.

In this work, we further those studies and show that:

- While conceptually elegant, we show that separability unfortunately remains challenging: Finding a separated design for a given structure is NP-hard, even when restricted to structures avoiding isolated base pairs;

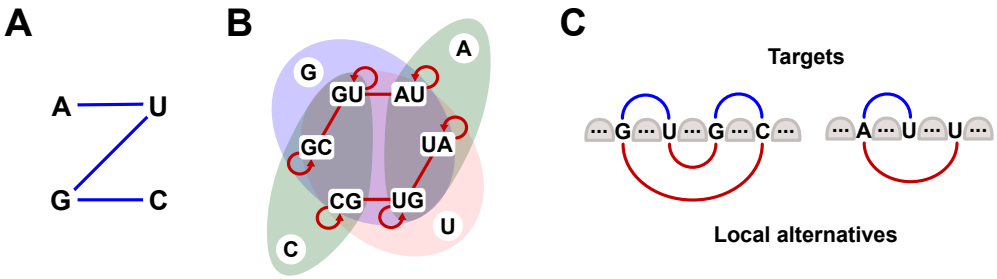


Fig. 1 Local design rules. Base pair compatibility graph (A) and incompatibility graph for base pairs and unpaired nucleotides occurring within a loop (B): Connected base pairs, when jointly occurring within a loop of the target structure, can refold to form a local, an alternative structure having same number of base pairs as the target (C, left). Unpaired nucleotides may also interfere with some (A or C) or every (G or U) base pairs, leading to local alternatives (C, right).

- Conversely, we prove that any structure with helices of length greater than 3 base pairs is either trivially not designable (*i.e.* contains $\{m_{3\bullet}, m_5\}$), or separable. Moreover, if designable, a solution sequence can then be designed in linear-time. This constraint is relevant to the objectives of RNA design, as targeted secondary structures are typically stable and tend to avoid shorter – unstable – helices;
- To establish this result, we introduce of the concept of modulo m-separability, a refined version of separability, which coincides with general separability upon setting $m \geq n/2$. Deciding m-separability clearly remains NP-hard in general, but it can be solved (+ a solution sequence be produced) in time $\mathcal{O}(n\,2^m)$ by a Fixed-Parameter Tractable (FPT) algorithm for m;
- We prove that this algorithm solves all instances of inverse folding with minimal helix lengths of 3 BPs when invoked with $m = 2$ and, even in this restricted setting, many instances with shorter helices;
- We adapt our algorithm into a uniform random generator of separated designs, combining a mildly unambiguous dynamic programming scheme with a rejection strategy that achieves an average-case $\mathcal{O}(n\,m\,2^m)$ time complexity;
- Finally, we empirically observe that m-separated sequences often represent solutions for instances featuring isolated base pairs or stacks. Moreover, despite being only guaranteed to represent designs with respect to base pair maximization, are also likely to represent designs in the more realistic Turner energy model and, in a relaxed setting, are also superior than mere compatible sequences for multiloops of larger cardinalities. Finally, we observe that m-separated sequences seem to often sufficient diversity to enable a control of the GC%.

2 Problem statement, definitions, and prior work

Algorithmically, RNA can be abstracted as a nucleotide sequence, *i.e.* a string $w \in \Sigma^n$, $\Sigma := \{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{U}\}$, where n denotes the length of w. Given a length n, a (non crossing/pseudoknot-free) secondary structure is a set $T \subset [1, n]^2$ consisting of base pairs such that:

- Each position in $[1, n]$ is involved in at most one base pair;

- Base pairs in T are pairwise non-crossing: $\forall (i,j) \neq (k,l) \in T$, $i < k$, either $i < k < l < j$ or $i < j < k < l$.
- Minimal distance in nucleotide number is parameterized by θ (default θ equals 0).

The set $\mathcal{S}_w$ of secondary structures compatible with an RNA sequence w is defined as: $\mathcal{S}_w := \{\text{Secondary structure } T \mid \forall (i,j) \in T, \{w_i, w_j\} \in \{\{\mathsf{G},\mathsf{C}\},\{\mathsf{A},\mathsf{U}\},\{\mathsf{G},\mathsf{U}\}\}\}$.

Without loss of generality, a secondary structure can be represented as a tree $T = (V(T), E(T))$, whose nodes $V(T)$ are in bijection with base pairs (internal nodes[1]) and unpaired regions (leaves), and whose edges represent the inclusion of base pairs. Given a node $v \in V(T)$, we denote by parent(v) the parent of v in T, and by children(v) the list of children of v in T. A *loop* is the subtree restricted to node and its (direct) children. The tree is rooted in a special Root node, associated with the whole sequence interval. An *helix* of length ℓ of the tree is a maximal path $v_1, \ldots, v_\ell$ of base pair nodes such that each v_i with $i < \ell$ has a single child v_{i+1} (no leaf attached). A helix of length 1 is an *isolated base pair*. A helix of length 2 is an *isolated stack*. We define $h_{\min}$ as the minimum length over all helices of T. As the target tree is always explicit and unmodified through proofs and algorithms we do not specify it explicitly in the notations.

RNA inverse folding considers a target secondary structure T, and constructs a sequence $\omega \in \Sigma^n$ whose unique base-pair maximizing secondary structure is T.

Problem 1. INVERSE-FOLDING$_{BP}$
Input: *Target secondary structure T, sequence length n*
Output: *Sequence $w \in \Sigma^n$ satisfying both:*

- *Compatibility with target structure: $T \in \mathcal{S}_w$;*
- *Uniqueness of the target as the optimal fold for the sequence:*

$$\forall T' \in \mathcal{S}_w, T' \neq T, |T'| < |T|.$$

or $\perp$ if no such sequence exists.

Nevertheless, INVERSE-FOLDING$_{\text{BP}}$, mildly extended to allow further restrictions on individual sequence positions, was shown to be NP-hard by Bonnet *et al.* [3]. (The used restriction requires the inclusion of some constraints of the form "nucleotide i must be labeled by the base letter b")

A sequence is called a design for a structure T if it represents a solution to the inverse folding problem for the input T. Note that the uniqueness condition can be tested in polynomial time using a variant of the Nussinov algorithm [17, 18]. In addition to showing that INVERSE-FOLDING$_{\text{BP}}$ is in NP, such an algorithm enables, for moderate sequence lengths, a systematic folding of all sequences in order to characterize the set of structures admitting a solution. For instance, Figure 2 shows that, while only 2.4% of RNA sequences of length 12 represent a design for some target, roughly half of the secondary structure admits at least one solution sequence, and ≈ 49 on average, for the inverse folding problem.

[1] Base pairs may also be leaves of the tree when involving consecutive positions, which happens rarely in practice. We thus qualify as *internal node* any node in bijection with a base pair.

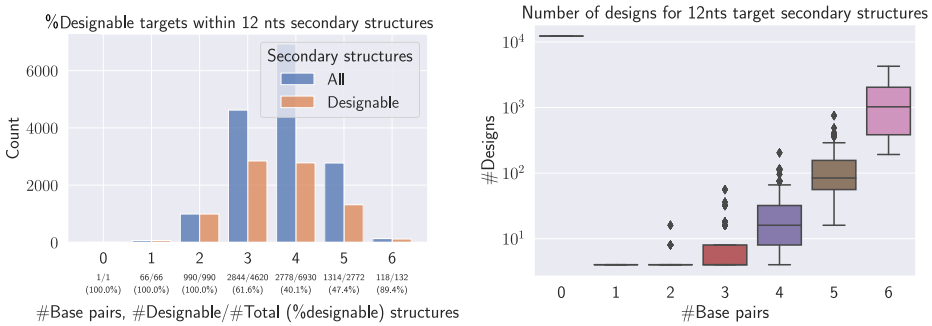


Fig. 2 Exhaustive designability analysis of 12nts RNA sequences/structures. (Left) For a minimum base pair span of $\theta = 0$, there exists 15 511 secondary structures over 12 nucleotides, of which little over half (8 111) admits at least a solution to the inverse folding problem. (Right) The number of valid solutions varies substantially between targets and appears to depend on the number of base pairs. Overall, out of the 16 777 216 RNA sequences of length 12, only 399 348 ($\approx 2.4\%$) represent a valid design for some structure.

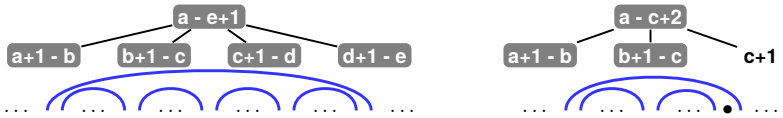


Fig. 3 Forbidden motifs. Motifs m_5 (left) and $m_{3\bullet}$ (right), both shown as a tree (with a, b, c d, e arbitrary integers) and as nested base-pairs. Note that the relative order of the children base-pairs and the leaf in the $m_{3\bullet}$ pattern is irrelevant. Any assignment of base pair letters (either matching a proper coloring of the tree or not) leads to a possible local rerooting of at least two base pairs yielding an alternative thus making the structure undesignable. [17].

We remind that, as noted by Halès *et al.* [17], two key motifs are not designable in a *base pair maximization* setting, see Figure 3:

- The m_5 motif consists of 5 base pairs occurring on the same loop (not counting the Root). No sequence can be designed for such a motif, since exposing 5 base pairs on a loop always allows for local refolding to have the same number of base pairs. This follows from the inspection of Figure 1, where the largest set of mutually compatible base pairs clearly has cardinality 4;
- The $m_{3\bullet}$ motif consists of 3 base pairs (excluding the Root) and at least one unpaired position. Indeed, as shown in Figure 1, the presence of an unpaired nucleotide either forbids the co-occurrence of any adjacent base pair (G or U), or only allows three (C or A). Since at most two of those base pairs can co-occur in a successful loop design, $m_{3\bullet}$ is not designable.

Any occurrence of these structures (or of any other undesignable structure, *cf* [19]) as a subgraph of an instance makes the instance undesignable.

2.1 Inverse folding as a tree coloring problem

We start by reminding the coloring framework introduced by Halès *et al.* [17].

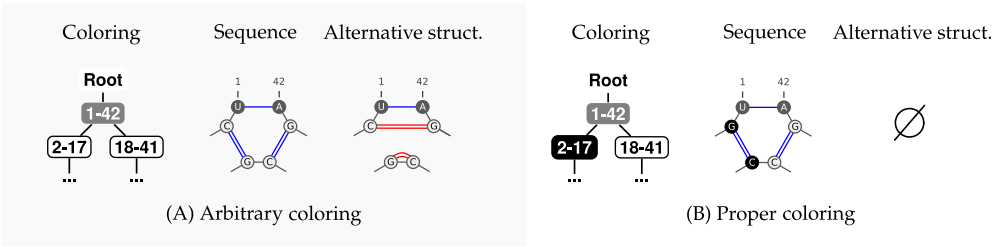


Fig. 4 A proper coloring is necessary towards design. In (A), having two ○ children implies that the sequence derived from this coloring features a motif where G and C can reconfigure locally. In that case, they form an alternative structure that contains the same number of base pairs. Conversely, in (B), the proper coloring ensures that locally no alternative of equal (or better) energy exists by forcing some consecutive incompatibilities.

Definition 1 (Coloring)**.** *A coloring of a (secondary structure) tree T is a function $\chi : V(T) \to \{●, ◍, ○, \varnothing\}$ associating a color to each node (except the root and the leaves which always get $\varnothing$).*

A coloring of a tree T typically induces multiple RNA sequences that are compatible with, but not guaranteed to fold into, the given secondary structure through letters assignment rules. Namely, in any sequence w derived from a coloring χ, we have for each $(i, j) \in T$:

- If $\chi((i,j)) = ● \to (w_i, w_j) = (\mathsf{G}, \mathsf{C})$;
- If $\chi((i,j)) = ○ \to (w_i, w_j) = (\mathsf{C}, \mathsf{G})$;
- If $\chi((i,j)) = ◍ \to (w_i, w_j) \in \{(\mathsf{A}, \mathsf{U}), (\mathsf{U}, \mathsf{A})\}$.

For ◍ nodes, the freedom in choosing (A, U) or (U, A) depends on the context: the choice may be unconstrained (*e.g.* when isolated within a helix), or forced (*e.g.* when two gray nodes are involved in a multiloop or stack). However, this property will only impact the number of sequences associated with the coloring, but bears no consequence on the existence of a solution to Inverse-Folding$_{\text{BP}}$, since the problem asks for the production of a single sequence.

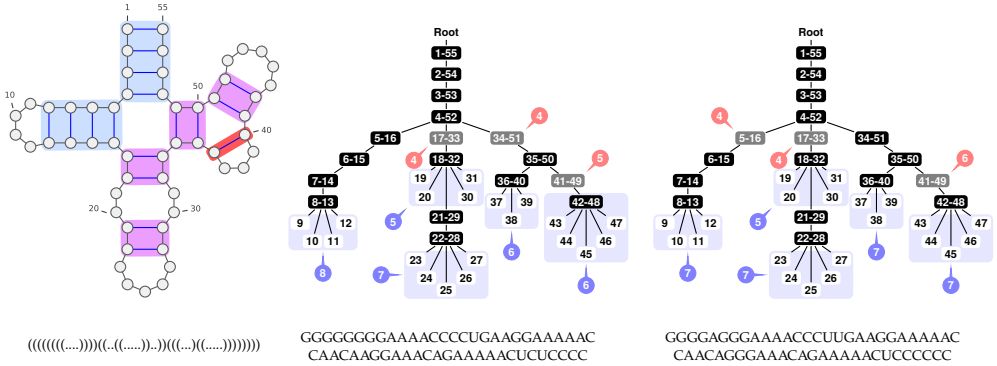
Denote by $\overline{c}$ the inverse of a color c, defined as $\overline{○} = ●$, $\overline{●} = ○$ and $\overline{◍} = ◍$. Denote by $|C|_c$ the number of occurrences of color c in vector C.

Definition 2 (Proper Coloring)**.** *A coloring χ is proper when, for each node $v \in V(T)$, the vector of colors C, composed of the complementary color of the node concatenated with the colors of its children, respects the following constraints:*

$$|C|_● \leq 1, |C|_○ \leq 1 \text{ and } |C|_◍ \leq 2 \text{ with } C := \left[\overline{\chi(v)}\right].\left[\chi(v') \mid v' \in \textit{children}(v)\right].$$

The use of the complementary color of v in C enables a compact definition: it forbids ● and ○ to have respectively ○ and ● children which would result in an alternative rerooting of the pairs. These conditions must also hold for the colorless Root, but with C being restricted to the colors of children(Root).

In terms of RNA design, the proper condition is necessary for an associated sequence to be a solution to inverse folding. Indeed, any coloring that is not proper



1) Target structure 2) A proper coloring 3) A separated proper coloring

Fig. 5 1) 2D and dot-bracket representations of a secondary structure. Helices of sizes respectively 1 (isolated base pairs), 2 (isolated stacks) and more than 3 are represented in light red, purple and blue. 2) Same secondary structure as a tree. The tree is colored and levels are represented in red and blue bubbles. The coloring is proper and non-separated as the level of the leaf 19 is the same as the level of the ● node 34-51. A non-separated coloring is not guaranteed to induce a design for its target, but may still do so, as is the case here. 3) Same secondary structure, colored in a separated (necessarily proper) manner. This coloring yields one or multiple designs (depending on the choice of AU or UA for ● nodes). Notably, this coloring is 2-separated, as leaves and ● nodes end up at odd and even levels respectively.

will be associated with sequences that can be locally reconfigured, this without losing any base pair (see Figure 4 for an example).

Definition 3 (Levels)**.** *Given a coloring χ of a tree T, the level $L : V(T) \to \mathbb{Z}$ of a node v is $L(v) := |p|_{\bullet} - |p|_{\circ}$ where p denotes the color vector associated with the shortest node sequence from parent(v) to Root.*

On an RNA level, the concept of level helps categorize, and possibly control, the set of alternative structures to the target. Indeed, consider a sequence w generated from a coloring χ. First remark that, in order for an alternative structure to be competitive, every occurrence of C must be paired. Whenever two positions i and j interact to form a base pair, it can be shown that the inner interval $]i, j[$ interval contains $L(i) - L(j)$ more G than C. Meanwhile the outermost interval $[1, i[\cup]j, n]$ features the opposite imbalance ($L(i) - L(j)$ more C than G). In other words, any structure that contains a base pair $(i, j) \notin T$ already has $2 \times |L(i) - L(j)|$ fewer base pairs than the target structure. Thus only structures made of pairs (i, j) such that $L(i) = L(j)$ need to be considered as viable alternatives to T. This property can be exploited as a design principle, as formalized by the following property.

Definition 4 (Separated coloring)**.** *A coloring χ is separated for a target T if and only if it is proper and the levels of ●-colored nodes and leaves do not overlap:*

$$\{L(v) \mid \chi(v) = \bullet\} \cap \{L(v) \mid v \text{ is a leaf}\} = \varnothing$$

This immediately suggests a design strategy that associates A to unpaired positions and assigns ● and ○ colors such that ● nodes end up as different levels as the

leaves. Indeed, in this setting, Hales *et al.* [17] showed that the proper coloring of a saturated structure (without unpaired position) yields a sequence that uniquely folds with respect to base pair maximization. It follows that a competitive/alternative structure may only result from a base pair $(i, j) \notin T$, a position of which is a ● node while the other is a leaf. Ensuring that all ● nodes and leaves are found at different levels is thus sufficient to guarantee the designability of T, *i.e.* the existence of a solution to this instance of the inverse folding problem.

More generally, we say that a target secondary structure T is *separable* if there exists a coloring χ such that 'χ is separated for T. We recall the main results of Halès et al. [17] here.

Theorem 1 (Separable $\implies$ Designable (Halès et al., 2017))**.** *If a tree/secondary structure T is separable, then T is designable.*

Moreover, given a separated coloring, an RNA sequence that uniquely folds into T, *i.e.* a solution to the inverse folding problem, can be found in linear time.

Remark 1. *Note that any design sequence w, generated through a separated coloring, avoids any alternative structure featuring* GU *base pair(s). Indeed, every* G *and* C *need to be paired to achieve the number of base pairs featured in the MFE. Meanwhile, the formation of any* GU *base pair, leaves one* C *and one* A *unpaired, resulting in the overall loss of at least one base pair. Structures featuring* GU *base pairs can thus be safely ignored.*

Remark 2. *Note that an alternative assignment of letters would be* C *to the unpaired positions,* UA *for* ● *nodes,* AU *for* ○ *nodes . It has no impact thanks to the symmetry of the base pair compatibility graph as depicted on Figure 1. In practice, it gives access if desired to double the number of sequences with the ones with the unpaired position at* C *that could have a slightly different content in terms of* G *and* C *even if not studied in this manuscript.*

3 Separability: Intrinsic and computational limits

3.1 Structures containing small helices can escape the scope of separability

Despite utilizing separability to explore a design of approximative instances, the work of Halès *et al.* [17] left open the complexity of searching for a separated coloring, as well as the existence of designable, yet non-separable, structures. An exhaustive search for all structures with up to 12 bases, summarized in Figure 2, shows that for such small instances, all designable instances are separable.

However, we show that non-separable designable instances can be constructed.

Proposition 2 (Designable $\not\implies$ Separable)**.** *There exists a target structure which: i) does not admit a separated coloring; and ii) admits a solution to the inverse folding problem.*

Proof. We use the tree T of Figure 6 as a counterexample to the notion that separability fully captures designability. First, note that a separated coloring χ of T would

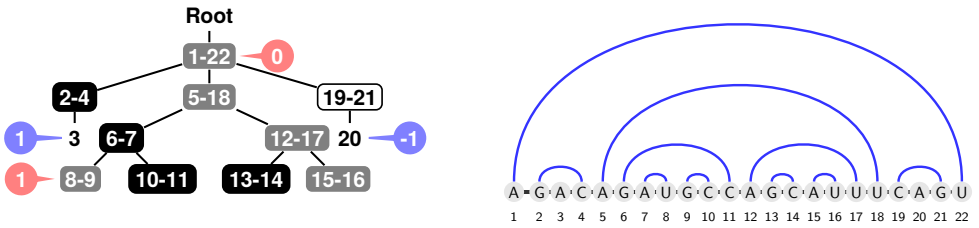


Fig. 6 Designability does not imply separability. Left: A target structure that does not admit any separated coloring instance. Note that the coloring χ shown here puts the ● node 8-9 and the leaf 3 both at level 1. Right: Sequence w compatible with the coloring χ, which provably admits T as its single base pair-maximization structure (i.e. w is a design for T).

be extremely constrained. Node $5-18$ should be ● and the nodes $2-4$ and $19-21$ are ● and ○ respectively, or vice-versa due to their respective leaf. Thus, we have two leaves at levels 1 and -1. At least, one of the two children of $5-18$, w.l.o.g $6-7$ is ● or ○. One child of $6-7$ is then necessarily ●, leading to a ● child of level $+1$ or -1. With two leaves at level $+1$ and -1, a direct consequence is that T is non-separable.

Now, we show that T is designable. We propose the sequence w of Figure 6. Using a simple dynamic programming algorithm, it is possible to check that the best folding for w is unique and corresponds to the secondary structure encoded as the tree T. Intuitively, the only competitive alternative base pair is the one corresponding to the overlap of the levels. It consists of joining the U from $8-9$ with the A at position 3. By doing so, note that the base pair $5-18$ will be disconnected with no way to pair A with another U due to the connection between 5 and 7. □

Notice that, despite not being separated, the coloring shown in Figure 6 is compatible with a sequence that is a design for its target. This illustrates the fact that, while not being guaranteed to uniquely fold as their intended target, sequences produced from non-separated colorings may still represent solutions for the inverse folding problem.

Proposition 3. *There exist non-separable structures with* $h_{\min} = 2$.

The full proof relies on a counterexample built from the gadget in Figure 7 and is given in the next paragraph. Intuitively, $T(a, b)$ saturates all levels modulo b with leaves, so that none remains available for ● nodes. Meanwhile, the presence of multiloops forces proper colorings to use ● nodes, so a collision occurs and the gadget is not m-separable for any $m \leq b$. By assembling 5 copies of $T(a, b)$ with large b and increasing values of a, we obtain a target that is not separable for any m.

Non-separable target without isolated base pairs

We start with the following remark:

Proposition 4. *If* $u_0, \ldots, u_k$ *is a path in* T *and each* u_i *for even* i *has a leaf attached to it then, for any coloring* χ *of the path, we have* $\chi(u_0) \in \{●, ○\}$ *and* $\chi(u_i) = \chi(u_0)$ *for all* i.

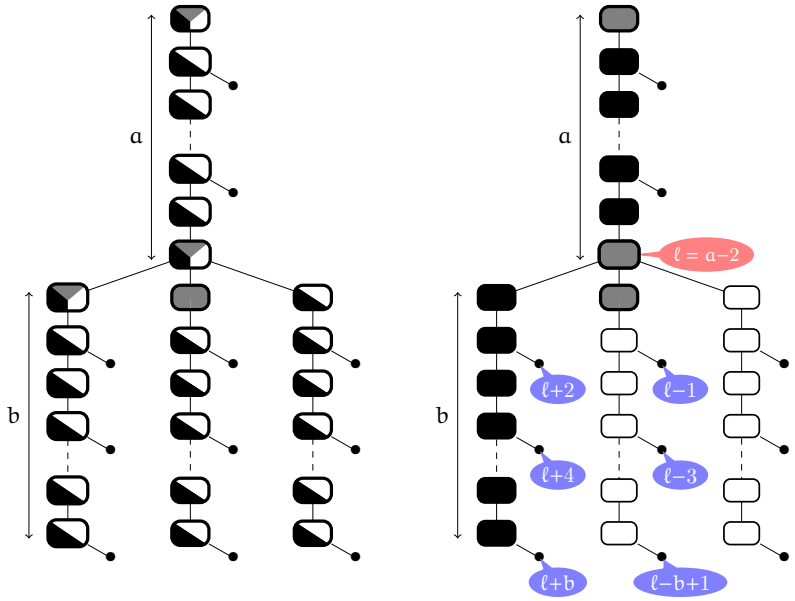


Fig. 7 Main gadget used to build non-separable instances with $h_{\min} = 2$. Left: Admissible colors for each node (up to branch symmetries). Right: Example coloring and levels of a selection of leaves and ◍ nodes. Note that along with the ◍ node at level ℓ, there always exists a leaf at level $\ell + m$ or $\ell - m$ for $2 \leq m \leq b$, ruling out modulo separability for small m.

Proof. Indeed, by the proper coloring constraint, every node with an attached leaf or with a leaf sibling may not be ◍, so all $\chi(u_i) \in \{●, ○\}$ for all i. Moreover, there can be no direct edge between ○ and ● nodes, so $\chi(u_i) = \chi(u_{i-1})$ for all i which gives the desired property by induction. □

We now build a non-separable instance I without size-1 helix nor $(m_{3\bullet}, m_5)$ motif. Let $a \geq 2$ and $b \geq 2$ be even numbers. Let $T(a, b)$ be the gadget from Fig 7, containing a length-a path from the to an internal node denoted t, and three length-b branches attached to t. Further attach a leaf to every node at an even distance from the root (except t itself). Note that all helices in $T(a, b)$ have length 2. The *level* of a copy of some $T(a, b)$ gadget is the level reached under node t of this gadget.

We build the instance I as a tree containing 5 copies of the gadget $T(a, b)$, precisely $I = (((T[10, 100], T[20, 100])), ((T[30, 100], T[40, 100])), T[50, 100])$.

First note that for a copy of gadget $T(a, b)$ at level ℓ in any separable coloring, there is a ◍ node at level ℓ, since the node t has three children and at least one must be ◍. Also, there exist two integers u, v such that, for every $x \in [1, b[$, there is a leaf at level $\ell + ux$ if x is odd, and level $\ell + vx$ if x is even. Indeed, pick one gray child U of t, and one non-gray child V. All vertices under U form an all-white or all-black branch by Proposition 4 (we let respectively $u = -1$ and $u = 1$), and vertices at levels $l + u, l + 3u, \ldots, l + bu$ (or $l + (b-1)u$) have a pending leaf. We similarly define $v = 1$ if V is black and $v = -1$ if V is white, and vertices at levels $l + 2v, l + 4v, \ldots, l + bv$ (or $l + (b-1)v$) have a pending leaf. From the above, if there are ◍ nodes at levels ℓ_1

and ℓ_2 with $\ell - b \leq \ell_1 < \ell < \ell_2 \leq \ell + b$, then $\ell_1 \neq \ell_2 \mod 2$ (since otherwise, one of ℓ_1, ℓ_2 could be written as $\ell + ux$ with even x, so that level would be a leaf level).

Aiming at a contradiction, assume that I admits a separable coloring. Let $\ell_1 \leq \ell_2 \leq \ell_3 \leq \ell_4 \leq \ell_5$ be the levels of all five copies of the $T[a, b]$ gadgets of I, in ascending order. Then from the length of the branches from the root, we have $\ell_i \in [-50, 50]$ and $\ell_i \neq \ell_j$. Then by the remark above applied to the gadget with level ℓ_2, we have $\ell_1 \not\equiv \ell_3 \mod 2$, and similarly using gadgets with level ℓ_4 we have $l_3 \not\equiv l_5 \mod 2$ and $l_1 \not\equiv l_5 \mod 2$, leading to a contradiction (any three integers such as ℓ_1, ℓ_3 and ℓ_5 may not have pairwise distinct parities).

3.2 Computational hardness of deciding separability

Regarding computational complexity, although looking for a separable coloring is not directly equivalent to finding a design for a structure, we show that this decision problem (formalized below) is also NP-complete.

Problem 2. Separability
Input: *Target tree T (without any occurrence of $m_{3\bullet}$ or m_5 motif)*
Output: *Coloring χ of the tree T such that χ is separated*

Theorem 5. Separability *is NP-complete.*

We further that even when isolated base pairs are forbidden in the input structure (e.g. helices are all of size 2 or more), the separability problem is still NP-hard. Thus, unless $P = NP$, the hope to find a polynomial algorithm for separability holds only when helices are of size 3 or more:

Problem 3. 2-helix Separability
Input: *Target tree T (without any occurrence of $m_{3\bullet}$ or m_5 motif) whose corresponding target structure contains no isolated base pair ($h_{\min} = 2$)*
Output: *Coloring χ of the tree T such that χ is separated*

Theorem 6. 2-Helix Separability *is NP-complete.*

Clearly, Theorem 5 follows from Theorem 6, since the latter relates to a strictly more general problem. We first give an outline of the reduction below, then provide the full proof in the following subsections.

Although proper colorability is a local constraint, separability implies a form a synchronization between different branches of the tree, since a conflict can appear between a leaf and a ● node even if they are in remote sections of the tree. However, we do not have a direct way to enforce that a specific level has ● nodes or leaves, since there is a lot of freedom in proper coloring constraints, especially in trees with $h_{\min} = 2$. The first building block for our reduction is a *blocking* gadget that saturates one parity (either odd or even levels) in some interval with leaves. This is matched with a constant-size *synchronization* gadget where two levels of different parities necessarily have ● nodes. So, even if both gadgets are present in different branches, they must be placed at different levels.

Using these two gadgets, we build our reduction from Bin Packing with a tree using one branch per item. Each item has a blocking gadget having the size of the item

surrounded by two synchronization gadgets. This enforces that items must be packed in non-overlapping ranges of levels. Additional synchronization gadgets further enforce that series of consecutive items sum up to the target bin size, thus enforcing that items are ordered according to a correct bin packing. However, the synchronization gadget induces some margin of freedom on the specific position of the ● levels, so consecutive items may be misaligned by some constant margin. This leads to a formulation of Bin Packing as an interval packing problem, with *blurred* endpoints with some constant margin L.

3.3 Formulation of Bin Packing as L-blurred interval packing

2-helix Separability is clearly in NP, since any coloring (certificate) can be checked in linear time. We prove hardness by reduction from Bin Packing which we formulate as an interval packing problem using unary encoding and *blurred* endpoints.

Definition 5. *Given a set of pairwise distinct even integers $A = \{a_1, \cdots, a_n\}$, integers k and B with $kB = \sum_{i=1}^{n} a_i$ and a constant L, an* L-blurred interval packing *of (A, k, B) is a set of integers u_i, v_i for each $1 \leq i \leq n$ and x_j for $0 \leq j \leq k$ such that:*

- $-L \leq x_j \leq kB + L$ *for all* j *and* $x_0 - L \leq u_i, v_j \leq x_k + L$
- $|x_{j+1} - x_j| \leq B$
- $v_i \in [u_i + a_i - L, u_i + a_i + L]$
- *for* $i \neq i'$, *intervals* $[u_i, v_i[$ *and* $[u_{i'}, v_{i'}[$ *have an intersection of size at most* L
- *there is no* i, j *such that* $x_j \in]u_i + L, v_i - L[$.

Let $A_0 = \{\alpha_1, \ldots, \alpha_n\}, k, B_0$ be an instance of Bin Packing with $kB_0 = \sum_{i=1}^{n} \alpha_i$ and L be any constant. Let M be the smallest even integer with $M > (n+4)L$. Write $a_i = M\alpha_i$ and $B = MB_0$.

Lemma 1. *The following are equivalent:*

1. (A_0, k, B_0) *is a yes-instance of Bin Packing*
2. (A, k, B) *admits an* L*-blurred interval packing*
3. (A, k, B) *admits a* 0*-blurred interval packing*

Proof. We show 1. $\Rightarrow$ 3. $\Rightarrow$ 2. $\Rightarrow$ 1..

1. $\Rightarrow$ 3. Set $x_j = jB$ for each $0 \leq j \leq k$. Let $(p_1, \ldots, p_n)$ be a permutation of $[1, n]$ such that bin 1 contains elements $\alpha_{p_1}, \alpha_{p_2}, \ldots, \alpha_{p_m}$ for some m, then bin 2 contains elements $\alpha_{p_{m+1}}, \alpha_{p_{m+2}}, \ldots, \alpha_{p_{m'}}$ for some m', etc. Define u_i with $u_{p_1} = 0$, $u_{p_{i+1}} = u_{p_i} + a_{p_i}$, and $v_i = u_i + a_i$. Then the first four conditions are trivially verified. For the final condition, for each j, the items in the first $j-1$ bins have sizes summing to exactly $(j-1)B$, so there is some i such that $u_i = x_j$, and by the fourth condition there is no i' with $u_{i'} + L < x_j < v_{i'} - L$.

3. $\Rightarrow$ 2.. Trivial, all conditions are weaker for L-blurred interval packing than 0-blurred interval packing.

2. $\Rightarrow$ 1. We start with the following observation: by the constraints 1. and 2., one can have $x_j < x_{j-1}$. However, considering only indices such that $x_j \geq x_{j-1}$, the union of intervals $[x_{j-1}, x_j[$ contains at least $[x_0, x_k[$. Let I_j be the set of indices i such that $x_{j-1} - L \leq u_i < x_j - L$. Sets I_j form a partition of $[1, n]$ (it is clear that they are

disjoint, and each $i \in [1, n]$ is in some I_j since otherwise $u_i < x_0 - L$ or $u_i \geq x_k - L$ so $v_i > u_i + L \geq x_k$). For each j, and $i \in I_j$, we have $v_i \leq x_j + L$, so interval $[u_i, v_i[$ is included in $[x_{j-1} - L, x_j + L[$. Each interval has size between $a_i - L$ and $a_i + L$, and these intervals overlap on at most L positions, so $\sum_{i \in I_j} a_i \leq x_j - x_{j-1} + |I_j|L + 2L \leq B + (n+2)L$. Since each a_i and B is a multiple of $M > (n+2)L$, we have $\sum_{i \in I_j} a_i \leq B$, and $\sum_{i \in I_j} \alpha_i \leq B_0$: sets I_j form a solution of Bin Packing(A_0, k, B_0). □

3.4 Reduction from Interval Packing to Separability

The reduction is based on two gadgets called *blocking* and *synchronization* gadgets. The first gives a long chain of nodes with a leaf attached to every other node; we show that this enforces a long interval of levels with leaves at all odd or even level. The second has a fixed size, and is incompatible with a blocking gadget since any separated coloring has both an odd and an even ◍ level. Both gadgets are defined in the following paragraphs, as well as their main properties. These properties are formulated in terms of *synchronized* and *blocked* levels, defined now.

We set $H = 12$ (chosen as the height of the synchronization gadget defined below), and use a blur value $L = 3H + 2$.

A level u is *H-synchronized* if there are two ◍ levels with different parity in $[u - H, u + H]$.

A level u is *H-blocked* if either all odd or all even levels in $[u - H, u + H]$ are leaf levels.

Observation 1. *In any separated coloring, no level can be both H-synchronized and H-blocked.*

Blocking gadget

A *blocking gadget* of size q in a tree is a chain of q node $s_1, \ldots, s_q$ with a leaf attached to s_i for each odd i.

Proposition 7. *In any proper coloring of a size-q blocking gadget, all nodes have the same ● or ○ color. Furthermore, let ℓ_1, ℓ_2, be the levels above the root and below the last node of the chain, such that $\ell_1 \leq \ell_2$. Then $\ell_2 = \ell_1 + q$ and all levels in interval $[\ell_1 + H, \ell_2 - H]$ are H-blocked.*

Proof. By the proper coloring condition, no parent or sibling of a leaf can be ◍, and 2 adjacent non-◍ nodes must have the same color, so all nodes s_i are given the same non-◍ color. Furthermore, if the gadget is colored ●, then ℓ_1 is the level above the root, and $\ell_2 = \ell_1 + q$. Plus, there are leaves at levels $\ell_1 + 2i + 1$ for all i such that $1 \leq 2i + 1 \leq q$, so all levels j with $\ell_1 + H \leq j \leq \ell_2 - H$ are H-blocked. Similarly if the gadget is colored ○, then ℓ_2 is the level of the root, $\ell_1 = \ell_2 - q$, and again levels $\ell_1 + H \leq j \leq \ell_2 - H$ are H-blocked. □

Synchronization gadget

The main gadget for our reduction is a fixed-sized tree for which any separated coloring uses two ◍ levels with distinct parity (see Lemma 2 below).

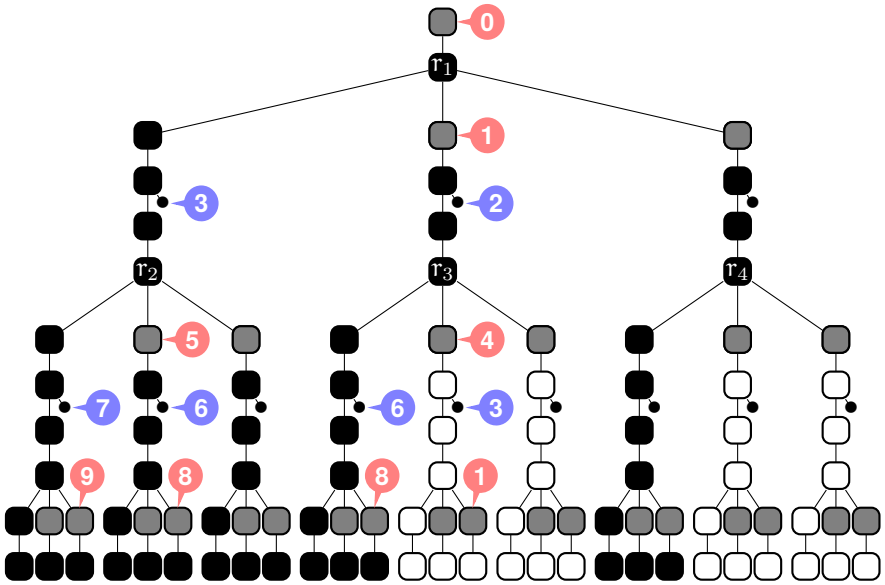


Fig. 8 The synchronization gadget with a proper coloring of its nodes using leaf levels $\{2, 3, 6, 7\}$ and ● levels $\{0, 1, 4, 5, 8, 9\}$. By Lemma 2, any proper coloring has two ● levels with distinct parity so the root level is H-blocked.

Lemma 2. *The* synchronization gadget *shown in Figure 8 admits a separated coloring with ● root and using only leaf and ● levels in $[r+0, r+9]$ (where r is the level of its root). Moreover, for any separated coloring with the level of the root is H-synchronized.*

Proof. Let r be the level of the root. The coloring with gray levels in $[r+0, r+9]$ is given in Figure 8.

For the main part of the proposition, assume that the gadget admits a coloring χ such that all ● nodes have the same level parity. Since all distances to the root are at most H, we need to ensure that there are two ● nodes with levels at different parity anywhere in the gadget.

Suppose first that some node r_i ($i \in \{1, 2, 3, 4\}$) of the gadget is colored ●, then among the 3 chains below r_i, one starts with a ● node, one starts with a ● node, and the last with a ○ node. We denote the nodes of the chain starting by ● c_1, c_2, c_3 and c_4 with c_1 a ● node and the chain starting by ● b_1, b_2, b_3, b_4. Note that b_1 and b_2 are ●. c_2 can only be ● or ○ as it has a leaf child (w.l.o.g. we assume it is ●). c_3 should also be ● due to the leaf of c_2 and to the proper condition. c_4 should also be ● to avoid conflict with the leaf child of b_2. Thus, c_4 has necessarily a ● child and it has a parity different from the ● node c_1 as there are 3 ● nodes between them.

Suppose now that each r_i, $i \in \{1, 2, 3, 4\}$ is non-●. Consider r_1, we again denote the four vertices starting on chain starting on a ● node c_1, c_2, c_3 and c_4. Then c_1 is ●, c_2 and c_3 have the same non-● color (again because of the leaf attached to c_2), and $c_4 = r_i$ for some $i \in \{2, 3, 4\}$ also has the same non-● color. Let c'_1 be a ● children of c_4: the level difference between c_1 and c'_1 is 3, so they have different parity, which concludes the proof.

□

Object gadget

An *object gadget* of size a (with a even and $a \geq H$) is a chain of $a+1$ nodes, with two synchronization gadgets attached respectively to the first and last nodes in the chain, and a leaf attached to the ith node for each even $i > H$.

Proposition 8. *If an object gadget of size a appears in a tree with a separated coloring χ, there exist levels $u \leq v$ such that:*

- *levels u and v are H-synchronized*
- *$a - L \leq v - u \leq a + L$ (recall that $L \geq 3H + 2$)*
- *all levels in $[u + L, v - L]$ are H-blocked.*

Proof. We define u and v as the levels of the roots of both synchronization gadgets (with $u \leq v$). Both u and v are H-synchronized by Lemma 2. Write b_1, b_2 respectively for the $(H+1)$th and ath node of the object gadget. The chain from b_1 to b_2 form a blocking gadget of size $a-H$. Let u', v' be the levels above b_1 and below b_2 respectively, with $u' \leq v'$. By the distance in the tree, $|u' - u| \leq H + 1$ and $|v' - v| \leq H + 1$. Moreover, by Proposition 7, $|v' - u'| = a - H$ (so $a - 3H - 2 \leq |v - u| \leq a + 3H + 2$) and all levels in $[u' + H, v' - H]$ (that contains $[u + L, v - L]$) are L-blocked. □

Given an instance A, k, B of L-Blur Interval Packing, we build a tree T as follows:

- We start with a chain P of length $2n$, with vertices denoted $q_0, p_0, q_1, p_1, \ldots, q_n, p_n$. (In order to avoid isolated base pairs, we only attach subtrees to nodes p_i, not q_i).
- For each $i \geq 1$ we attach a chain (denoted P_i) of kB nodes to p_i followed by an object gadget C_i of size a_i.
- We attach a blocking gadget X_1 of size $4kB$ to p_0.
- We attach a long chain to p_0 composed successively of:
  - a chain S of $kB+2$ nodes with a synchronization gadget attached to the $(iB+2)$th node for each $0 \leq i \leq k$
  - a subtree X_2 composed of a blocking gadget of size $2kB$ with a synchronization gadget attached to the last node.

3.5 Correctness proof

We now complete the correctness proof of the reduction with the following Lemma.

Lemma 3. *We have the following two implications*

$$(A, k, B) \text{ admits a 0-blurred interval packing} \Rightarrow T \text{ is separable}$$
$$T \text{ is separable} \Rightarrow (A, k, B) \text{ admits an } L\text{-blurred interval packing}$$

The proof is given in the following two sections. This lemma completes the proof of Theorem 5, since together with Lemma 1 we have that (A_0, k, B_0) admits a Bin Packing if and only if T is separable. Moreover, using the strong NP-hardness of

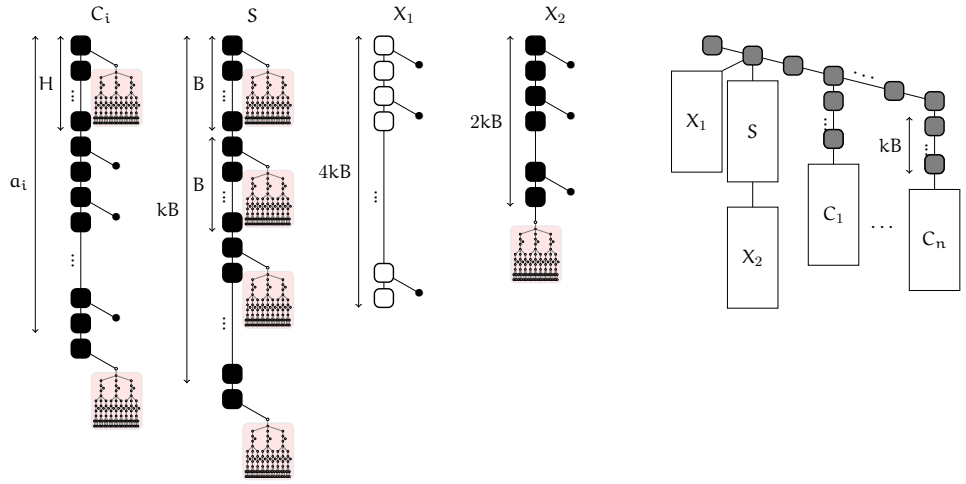


Fig. 9 Left: details of the four main parts of the reduction, i.e. an object gadget C_i of size a_i, the chain S, and blocking gadgets X_1 and X_2). Right: general layout of the tree built in the reduction.

Bin Packing, we can assume that all integers α_i are bounded by a polynomial in $|A_0|$ (corresponding to a unary encoding), so T can be built in polynomial time from (A_0, k, B_0). Finally, it can easily be checked that T does not have isolated base pairs (however, T does contain isolated stacks, so $h_{\min} = 2$).

From 0-blurred interval packing to separated coloring

We consider a 0-blurred interval packing assigning integers u_i, v_i to each item a_i (and $x_j = jB$) as defined in Figure 10. In words, chain P is colored ◍. Each chain P_i ($i \geq 1$) starts with a ● node, ends with u_i ● nodes, and all remaining nodes are ◍. All synchronization gadgets are colored as in Figure 8. The chain X_1 is ○, and all other nodes are colored ●.

We show that this coloring is separated. Note that ◍ nodes either have level 0 or 1, or are part of a synchronization gadget. Let $X = \{u_i \mid 1 \leq i \leq n\} \cup \{kB\}$, all synchronization gadgets have level $x \in X$, so ◍ nodes have levels in $\{0, 1\} \cup \bigcup_{x \in X} [x, x + 9]$. Since synchronization gadgets are separated (locally), it remains to verify that no leaf in the rest of the tree has a level in this set. Indeed, leaves in C_i have levels between $u_i + L$ and $v_i - 2$, while X_1 and X_2 have leaf levels < 0 or $> kB$.

From separated coloring to L-blurred interval packing

Suppose now that T admits a separated coloring χ. Assume that the level below node p_0 is 0 (otherwise, apply an offset to all level values below). Consider first blocking gadget X_1. By Proposition 7, it is either colored ● or ○. Without loss of generality, assume it is colored ○. Then since it has size $4kB$, all levels in $[-4kB + H, -H]$ are H-blocked. In particular, since there is no path of length $\geq 4kB - H$ in the rest of the tree, all synchronization gadgets must have a level $\geq -H$.

Consider now the blocking gadget X_2, let N be the level of its first node. We have $|N| \leq kB + 1$. If X_2 is colored ○, then the synchronization gadget below it

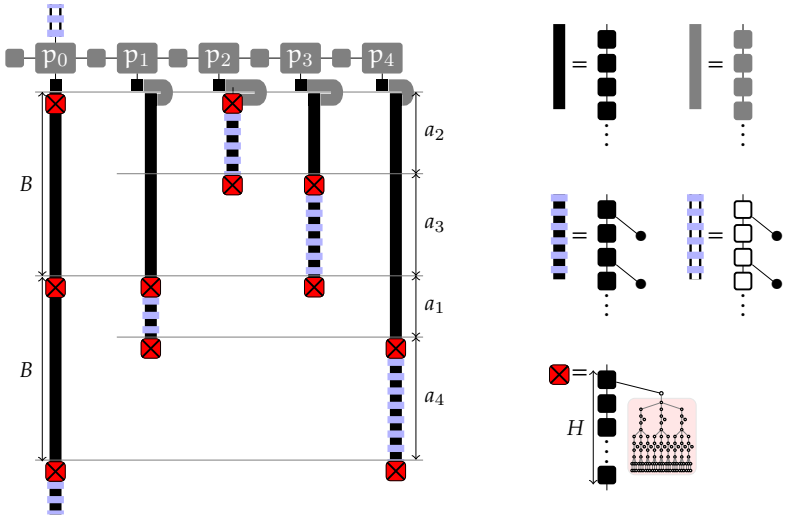


Fig. 10 Example of the reduction with $n = 4$ items with sizes $\{a_1, a_2, a_3, a_4\}$ to be sorted into $k = 2$ size-B bins. Each item is mapped into a branch P_i followed by an object gadget C_i, containing 2 synchronization gadgets (shown as red nodes with crosses) separated by the size of the item. Leaves in object gadget enforce that any two gadgets may overlap only if the synchronization gadgets are aligned (within a margin of L levels). The bins are implemented using the chain S, with synchronization gadgets at every Bth position, enforcing that series of consecutive items are packed into size-B bins. Finally, blocking gadgets X_1 and X_2 may not overlap with any synchronization gadget, and enforce that all object gadgets as well as the chain S are packed together in a size-kB range of levels.

would be at level $N - 2kB \in [-4kB + H, -H]$, which is not possible (this level is blocked), so X_2 is colored ●, and all levels in $[N + H, N + 2kB - H]$ are H-blocked. Since all synchronization gadgets in the S or C_i chains are at distance at most $2n+kB+\max a_i+2$ from the root, and this distance is upper bounded by $N+2kB-H$ (with the reasonable assumption that B is large enough with respect to n, precisely $kB \geq 2n + \max a_i + 2$), they all have level at most $N + H - 1 \leq kB + H \leq kB + L$.

Write x_j for the level of the jth synchronization gadget in S: x_j is a synchronized level. By the size of path between successive gadgets, $|x_{j+1} - x_j| \leq B$, and by the remark above, $-L \leq x_j \leq kB + L$.

For each object gadget C_i, by Proposition 8, there exist H-synchronized levels u_i, v_i such that $v_i \in [u_i - L, u_i + L]$, and such that all levels in $[u_i + L, v_i - L]$ are H-blocked. Overall, we have integers x_i, u_i, v_i satisfying the conditions for an L-blurred interval packing.

4 Modulo separability as a parameterized tractable alternative

Then, we introduce a stratified version of separability, called modulo m-separability, or m-separability in short, which prescribes different modular values for the levels of ● and leaves nodes. Figure 11 describes the relative positioning of classes of instances and associated complexity results.

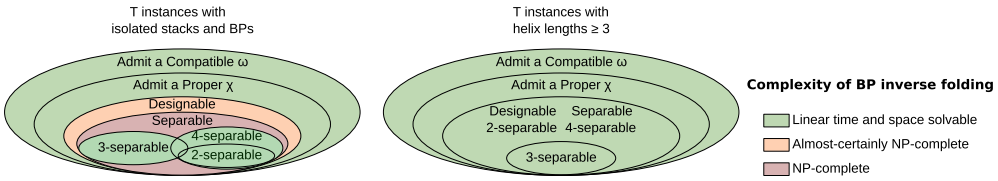


Fig. 11 Instances of INVERSE-FOLDING$_{\mathbf{BP}}$**.** For unconstrained instances (Left), INVERSE-FOLDING$_{\mathrm{BP}}$ is likely NP-hard, as suggested by the hardness of a constrained version [3]. Finding a design for a separable target is also NP-hard but, for any fixed modular level m, m-separable targets can be designed in $\Theta(n)$ time. This suggests an algorithm, FPT on m, for all separable structures. When $h_{\min} \geq 3$ (Right), Theorem 11 applies and the hierarchy collapses: any instance becomes 2-separable ($\implies$ separable and designable) and INVERSE-FOLDING$_{\mathrm{BP}}$ can be solved in $\Theta(n)$ time.

Definition 6 ((Modulo) m-separability)**.** *Let m be an integer. A coloring χ is m-*separated *(or* separated with modulus m*) for a target secondary structure T, if an only if χ is proper and*

$$\{L(v) \bmod m \mid \chi(v) = \bullet\} \cap \{L(v) \bmod m \mid v \text{ is a leaf}\} = \varnothing$$

using for negative levels $l < 0$ the classic $l \bmod m := (l + \lceil -x/m \rceil \times m) \mod m$.

*Structure T is m-*separable *if it admits an m-separated coloring.*

Clearly, modulo separability implies classic separability: if a coloring χ is m-separated for a target structure T, then χ is separated for T. Conversely, if a target structure admits a separated coloring, assigning levels in $[-a, b]$ to $\bullet$ and leaf nodes, then the same coloring is provably m'-separated for $m' := (b + a + 1)$ (since, for $l, l' \in [-a, b]$, $l \neq l'$ implies that $l \bmod m' \neq l' \bmod m'$). Note that, since there are at most $n/2$ base pairs/internal nodes in a target tree, then $0 \leq a, b \leq n/2$, and we have $m' \leq n$.

The concept of m-separability thus provides an angle to address the generation of separated colorings, so we introduce below the associated formalized algorithmic problem.

Problem 4. MODULO SEPARABILITY
Input: *A tree T (with no $m_{3\bullet}$ or m_5 motif), a modulus $m \in \mathbb{N}$*
Output: *A coloring χ of T that is m-separated, or $\perp$ if no such coloring exists.*

As noted above, the problem specializes in the SEPARABILITY problem when $m = n$, implying that MODULO SEPARABILITY remains NP-complete. However, it can be efficiently solved for moderate values of m, as shown below. Practically, one may focus on small values of m since 99% of instances without isolated base pairs are separable with modulus $m \leq 6$ (*cf* Table 13).

4.1 Fixed parameter tractable algorithm for modulo-separability

We now show that, for any fixed modulus m, MODULO SEPARABILITY can be solved in linear time. In particular, the problem is Fixed Parameter Tractable (FPT) for the parameter m.

Towards that goal, we consider a constrained version of Modulo Separability, where the modular values of levels are prescribed. Formally, we enforce that leaves only occur at modular levels in $\xi_L \subseteq [0, m[$, and ● nodes only occur at levels $[0, m[\backslash \xi_L$. In this constrained version of Modulo Separability, the existence of a valid solution can be solved in linear time using dynamic programming.

Namely, let us denote by $\mathrm{d}^{\xi_L}_{v \to c, \ell}$ the existence of a valid assignment (*i.e. solution*) for a subtree of T rooted at internal node v, with v occurring at level ℓ, and being assigned a prior color c. Provably, $\mathrm{d}^{\xi_L}_{v \to c, \ell}$ can be computed recursively by progressing along the tree, keeping track of the current level and checking that leaves and ● end up being assigned at modular levels ξ_L and $[0, m[\backslash \xi_L$ respectively. This leads to the following formula:

$$\mathrm{d}^{\xi_L}_{v \to c, l} = \begin{cases} \mathsf{False} & \text{if } \ell \in \xi_L \wedge c = \bullet \\ & \text{or } \ell' \notin \xi_L, \text{ and } \exists \text{ leaf in children}(v) \\ \mathsf{True} & \text{if children}(v) = \emptyset \\ \displaystyle\bigvee_{\substack{c' \text{ proper coloring of} \\ \text{children}(v) \text{ given } v \to c}} \bigwedge_{v' \in \text{children}(v)} \mathrm{d}^{\xi_L}_{v' \to c'(v'), \ell'} & \text{otherwise.} \end{cases}$$

with $\ell' := \ell + \delta(c) \bmod m$

where δ denotes the level increment induced by a color c, defined as $\delta(\bullet) = +1$, $\delta(\circ) = -1$ and $\delta(\bullet) = 0$. Moreover, in the outermost loop, the color assignment explored for children is meant to be locally proper: the colors $c(v')$ of the children, in conjunction with the color c of v must obey the conditions of Definition 2. Note that, in the absence of $m_{3\bullet}$ and m_5, the number of (proper) assignments is bounded by a constant, so this conjunctive loop does not impact the complexity. The existence of a ξ_L coloring for the full tree is then $\text{Separable}_{\xi_L} := \mathrm{d}^{\xi_L}_{\mathsf{Root} \to \varnothing, 0}$.

The decision version of the problem can thus be solved in $\Theta(m.n)$ time. Indeed, the number of left-hand side terms scales in $\Theta(m.n)$, the number of proper coloring for children is bounded by a constant (since avoiding $m_{3\bullet}$ and $m_5 \implies |\text{child}(v)| < 5$), and the total number of executions of the conjunctive loops is in overall $\Theta(n)$. A backtracking procedure could also be defined to reconstruct a solution coloring in $\Theta(n)$ if such a solution exists ($\text{Separable}_{\xi_L} = \mathsf{True}$) or return $\perp$ otherwise ($\text{Separable}_{\xi_L} = \mathsf{False}$).

An algorithm for Modulo Separability can then be obtained by explicitly considering all the possible subsets of admissible modular levels for leaves:

- If T contains $m_{3\bullet}$ or m_5, return $\perp$
- For each $\xi_L \subseteq [0, m[$:
  - If $\#\text{Designs}_{\xi_L} > 0$, then backtrack to produce ξ_L-separated design
- Return $\perp$

The algorithm is correct: for any m-separated coloring χ, there exists at least one $\xi_L \subseteq [0, m[$ corresponding to the leaves of χ solution and any the m-separated property implies a partition of the leaves and ● nodes into disjoint levels ξ_L and $\chi_{\bullet} \subseteq [0, m[\backslash \xi_L$ respectively. A m-separated coloring is thus always found by invoking the DP algorithm over the 2^m subsets $\xi_L \in [0, m[$. The overall complexity of the algorithm is in $\Theta(n.m.2^m)$ time and $\Theta(m.n)$ memory, and we conclude with the parameterized complexity of the problem with respect to m.

Theorem 9. Modulo Separability *is Fixed Parameter Tractable for the modulus parameter* m

4.2 Random generation of m-separated sequences

We then turn to the uniform random generation of m-separated sequences, defined as a design w for T, featuring A on unpaired positions, and such that the coloring χ_w, obtained by replacing base pairs with suitable color $((\mathsf{G}, \mathsf{C}) \to$ ●, $(\mathsf{C}, \mathsf{G}) \to$ ○ and (A, U) or $(\mathsf{U}, \mathsf{A}) \to$ ◍), is m-separated.

Problem 5. Uniform Modulo Separated Generation
Input: *Target tree* T *(with no* $m_{3\bullet}$ *or* m_5 *motif)*
Output: *RNA sequence* w*, associated with* m*-separated coloring* χ_w*, such that*

$$\mathbb{P}(w \mid \chi_w \textit{ is } m\textit{-separated}) = \frac{1}{|\{w' \in \Sigma^n \textit{ such that } \chi_{w'} \textit{ is } m\textit{-separated}\}|}$$

4.2.1 Linear-time uniform sampling for fixed modular assignments

Once again, we approach this problem by first solving a more constrained version where the modular levels of leaves are explicitly given as a set ξ_L, denoted as *modular assignment* in the following. Then, in the spirit of Reinharz *et al.* [12], we adapt the above recurrence, through a simple algebra change, to count the number $\mathrm{p}^{\xi_L}_{v\to\mu,l}$ of RNA sequences, associated with what we call a ξ_L *separated coloring*, that is to say a m-separated coloring such are all leaves levels are ξ_L. (for a subtree of T rooted at v, with v occurring at level l, and being assigned a nucleotide assignment μ).

$$\mathrm{p}^{\xi_L}_{v\to\mu,\ell} = \begin{cases} 0 & \text{if } \ell \in \xi_L \text{ and } \mu \in \{(A,U),(U,A)\} \\ 0 & \text{if } \ell' \notin \xi_L \text{ and } v \text{ has a leaf attached} \\ 1 & \text{if children}(v) = \varnothing \\ \displaystyle\sum_{\substack{\mu' \text{ proper assignment} \\ \text{children}(v)\to\Sigma^2\cup\{\emptyset\}}} \prod_{v'\in\text{children}(v)} \mathrm{p}^{\xi_L}_{v'\to\mu'(v'),\ell'} & \text{otherwise.} \end{cases}$$

with $\ell' := \ell + \delta(\mu) \bmod m$

where μ' is function assigning nucleotides to the children of v, consistent with a proper coloring and additionally respecting natural constraints on the content ((A, U) or (U, A)) of pairs of ◍ nodes (same for both if one parent of other, different content if siblings). Once again, the colorless Root node needs to be distinguished, and the overall

number of designs is given by $\#\text{Designs}_{\xi_L} := \mathrm{p}^{\xi_L}_{\mathsf{Root}\to\varnothing,0}$. We next propose a backtrack procedure $\mathsf{backtrack}^{\xi_L}_{v'\to\mu'(v'),\ell'}$ with exactly the same parameters than $\mathrm{p}^{\xi_L}_{v\to\mu,\ell}$ and that process exactly the same cases and then produces a uniform random RNA sequence that corresponds to a m-separated coloring for a fixed set ξ_L. In that case, by abuse of language, we say that the sequence is ξ_L *separated*. More precisely, $\mathsf{backtrack}^{\xi_L}_{(\to v,,}c,\ell)$ produces a random sequence, associated with a ξ_L separated coloring, for the subtree anchored in v, reached at height ℓ, where the root is assigned a pair of bases $\mu \in \Sigma^2$. It first picks a random proper assignment μ' for the children, weighted by the corresponding number of solutions (namely, $\prod_{v'\in\text{children}(v)} \mathrm{p}^{\xi_L}_{v'\to\mu'(v'),\ell'}$, with $\ell' := \ell + \delta(\mu) \mod m$). The resulting sequence is then

$$\prod_{v\in\text{children and leaves}(v)} \begin{cases} \mathsf{A} & \text{If } v' \text{ is a leaf} \\ b.(\mathsf{backtrack}^{\xi_L}_{v'\to\mu'(v'),\ell'}).b' & \text{otherwise, with } \mu'(v') = b.b' \end{cases}$$

The resulting algorithm, consisting of precomputing all $\mathrm{p}^{\xi_L}_{v\to\mu,\ell}$, followed by a sequence of k backtracks, provably returns k random, uniformly-distributed and independent designs that are ξ_L separated in time $\Theta(n.m + k.n)$.

4.3 Integrating over all modular assignments and correcting for uniformity through rejection

Clearly, a random generation algorithm could be obtained by generating a random modular assignment ξ_L uniformly, and then use the above algorithm to produce a design. However, if naively implemented, such a strategy would suffer from multiple shortcomings:

1. It may not always produce a valid design, even when such a design exists. Indeed, a naive choice of ξ_L may lead to zero ξ_L separated design;
2. The overall generation scheme would not be uniform over the set of m-separated sequences (at a fixed m). Indeed, the emission probability of a m-separated sequence w that is only compatible with a single assignment ξ_L (*i.e.* populating all levels in ξ_L), is then strictly inversely proportional (for a fixed m) to the number of designs compatible with ξ_L. Such a probability will thus typically differ across modular assignments, inducing a bias. Even if corrected by a suitable correction upon choosing ξ_L, this scheme will favor sequences that are compatible with multiple modular assignments, thus motivating further countermeasures.

To correct those issues, and leverage the uniform generation for a fixed ξ_L into a uniform generation of m-separated designs, we implement a classic rejection strategy (see [20, pp 77] for a general exposition). It start by generating some ξ_L according to a suitable distribution, and then uses a suitable rejection to correct the emissions probabilities of sequences compatible with several ξ_L.

Theorem 10. Uniform Modulo Separated Generation *can be performed in* $\Theta(n.m.2^m)$ *average-case complexity,* i.e. *Fixed Parameter Tractable on the modulus parameter* m*.*

We consider a rejection-based approach, which starts by precomputing all $\#\text{Designs}_{\xi_L}$ in time $\Theta(n.m.2^m)$ (see Section 4.2), and accumulates them into $\mathcal{Z}_m := \sum_{\xi'_L \subseteq [0,m[} \#\text{Designs}_{\xi'_L}$. It then iterates the following steps until a suitable sequence is returned:

1. Choose some $\xi_L \subset [0, m[$ with probability $\mathbb{P}(\xi_L) = \#\text{Designs}_{\xi_L}/\mathcal{Z}_m$
2. Generate a ξ_L separated sequence w
3. Compute the number Ξ_w of $\xi'_L \subset [0, m[$ such that w is ξ'_L separated
4. With probability $1/\Xi_w$, accept/return w ; Reject/restart from **1.** otherwise.

Due to the full reset on each rejection, the emission probability p_w of any suitable w does not depend on the prior sequence of rejections (folklore, proven in [20, pp 77]), and we have:

$$\begin{aligned} p_w &\propto \sum_{\substack{\xi_L \text{ such that } w \\ \text{is } \xi_L \text{ separated}}} \mathbb{P}(\xi_L) \times \mathbb{P}(w \mid \xi_L) \times \frac{1}{\Xi_L} \\ &= \sum_{\substack{\xi_L \text{such that } w \\ \text{is } \xi_L \text{ separated}}} \frac{\#\text{Designs}_{\xi_L}}{\mathcal{Z}_m} \times \frac{1}{\#\text{Designs}_{\xi_L}} \times \frac{1}{\Xi_w}. \end{aligned}$$

Some terms directly cancel out and, by definition, we have

$$\sum_{\substack{\xi_L \text{such that } w \\ \text{is } \xi_w \text{ separated}}} 1 = \Xi_w.$$

It follows that $p_w \propto 1/\mathcal{Z}_m$, a term that no longer depends on w, from which we conclude that the overall generation is uniform.

Complexity-wise, a prior accumulation of the 2^m terms $\#\text{Designs}_{\xi_L}$, each smaller than 4^m, into a suitable data structure (see Lorenz and Ponty [21] for details) enables a random choice of ξ_L (Step 1.) in $\Theta(n.m)$. Once ξ_L is chosen, the above DP algorithm uniformly generates w in time $\Theta(m.n)$ (Step 2). The computation of Ξ_w (Step 3) is trivial and consists in identifying, in time $\Theta(n+m)$, the subset $\Phi_w \subseteq [0, m[$ of modular levels that are populated by neither leaves nor ● nodes in χ_w. Indeed, those levels represent the only degrees of freedom available while choosing a compatible ξ_L, the others modular values being forced to either ● or leaves. Since such modular values can be independently chosen to be in or out of ξ_L, then we have $\Xi_w = 2^{|\Phi_w|}$. Clearly, we have $\Xi_w \leq 2^m$, so the expectation of the number of (independent) rejections admits an upper bound in 2^m, and the overall average-case complexity is in $\Theta(n.m.2^m)$.

5 Structures without isolated stacks and base pairs are 2-separable

Although separability does not give a full characterization of designability in general (cf Prop. 2 and Prop. 3), we obtain a much stronger result for structures without small helices, as hinted by the fact that all counter-examples and hardness gadgets heavily use isolated base pairs or isolated stacks in their construction. Indeed, we show that a 2-separated coloring can be constructed for *all* structures without forbidden motifs $(m_{3\bullet}, m_5)$ and $h_{\min} \geq 3$, so indeed all such structures are designable. Since avoiding $(m_{3\bullet}, m_5)$ is a necessary condition for designability, we obtain the stronger characterization stated in Corollary 2.

Theorem 11. *Every $(m_{3\bullet}, m_5)$-avoiding target T, having $h_{\min} \geq 3$, admits a 2-separated coloring*

Proof. First, let us remark that helices can be treated as atomic objects, and compacted into the edges of a *helix tree*, whose edges are helices (sequence of consecutive BP nodes), and whose internal nodes are either:

- Multiloops, consisting of 2 or 3 children/BPs/Helices, and no leaf (so $m_{3\bullet}$ does not occur);
- Internal/Bulges/Hairpin (IBH) loops, consisting of at most 1 BP/Helix and featuring at least one leaf/unpaired node.

Remark that, while constructing a separated coloring assigning a modular level ξ_L to leaves, those two motifs are the only sources of immutable constraints:

- Any proper coloring of a multiloop features at least one ◍ node, so the levels of children/nodes need to be set to a level $\overline{\xi_L} := \xi_L + 1 \bmod 2$;
- Any IBH loop features at least one leaf within its children, which needs to be set to a modular level ξ_L.

Conversely, beyond their first BP, helices may be colored with very limited constraints and can be used to *offset* multiloops and IBH loops.

Lemma 4. *Let $\overline{\xi_L}$ denote the prescribed modular level for ◍ nodes. Consider an helix H consisting of 3 BPs or more ($h_{\min} \geq 3$), whose first BPs is assigned some color $c \in \{●, ○, ◍\}$.*

Then for each modular level $l \in [0,1]$ for the first BP of H ($c =$ ◍ only if $l = \overline{\xi_L}$), and targeted exit modular level $l' \in [0,1]$, there exists a coloring for the rest of H such that:

- *The modular level of the upcoming nodes,* i.e. *those immediately following H, is l';*
- *Base pairs can only be ◍-colored at modular level $\overline{\xi_L}$.*

Proof. The proof is essentially based on case decomposition, and summarized in Figure 12. We show that, for any l and $h_{\min} \geq 3$, there exists a color assignment to the first 3 nodes of the helix, such that the modular level of upcoming nodes is either 0 or 1, so l' can be reached. Moreover, if such a coloring starts with ● or ○, and uses a single ◍ node, then there exists an alternative coloring placing this ◍ node at the

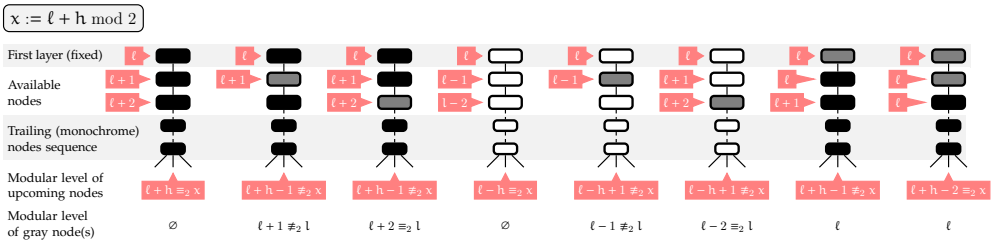


Fig. 12 Alternative colorings for helices consisting of 3+ base pairs ($h_{\min} \geq 3$), such that the modular level of the following nodes is offset as needed. Such colorings can be chosen to respect a prescribed level for ● nodes and, a predetermined color for the first node/base pair of the helix.

opposite modular level, so one of them places their ● node at the intended level $\overline{\xi_L}$. Finally, if the first node is set to ●, then the consistency condition above implies that $l \bmod 2 = \overline{\xi_L}$, so that ● nodes are naturally found at an admissible modular level. □

It follows that any helix tree starting with an initial helix H can be colored into a 2-separated coloring. Starting at initial level $l = 0$ and having initial BP color c ($\neq$ ● if $\xi_L = 0$), color the rest of H as shown in the proof of Lemma 4, depending on $\overline{\xi_L}$ and the type of upcoming loop (target $l' = \overline{\xi_L}$ for Multiloops; $l' = \xi_L$ for IBH loops), while ensuring that ● nodes end up at $\overline{\xi_L}$ modular level (which can always be done from Lemma 4). The remaining nodes of the loop are then colored in a proper/greedy manner, and we iterate the process recursively on the children helices of the loop (if any) until the full tree is colored.

Since its level cannot be offset, the Root node must be treated as a special case. Indeed, if the Root has at least one leaf/unpaired position, then the modular value 0 is taken by the leaf, so we must have $\xi_L = 0$. Conversely, if the Root supports at least 3 helices, then at least one needs to start with a ● node, so we must have $\xi_L = 1$. Regardless of this restriction on ξ_L, in both cases the first base pair of each helix (if any) supported by the Root can be properly colored, and helices can be independently colored using the above strategy, ultimately yielding a 2-separated coloring. □

Corollary 1. Inverse Folding, *restricted to instances with* $h_{\min} \geq 3$ *(containing no isolated base pair and no isolated stacks) is solvable in linear time and space.*

It is a direct consequence of Theorem 11 and of the DP scheme introduced in Section 4.1. Indeed, for $m = 2$, the DP algorithm only needs to be run twice ($\xi_L = 0$ and $\xi_L = 1$) in linear time/space, to produce a 2-separated coloring whenever such a coloring exists (guaranteed by Theorem 11). The coloring can then be transformed into a design, *i.e.* a solution to the Inverse Folding problem. Similarly, Uniform Modulo Separated Generation can also be performed in linear expected time and space as long as input instances contain only helices of size 3 or more.

Corollary 2. *Let T be a target structure with $h_{\min} \geq 3$, then the following are equivalent: i) T is designable; ii) T is 2-separable; and iii) T avoids $(m_{3\bullet}, m_5)$.*

With this result, the hierarchy of instances collapses as depicted on the left of Figure 11 A natural follow-up question is whether the bound 3 on the helix length is

tight. Indeed, there are non-separable and designable instances with $h_{\min} = 1$ (Proposition 2), but the question remains for $h_{\min} = 2$. In Proposition 3 we give a non-separable instance without isolated base pairs, so $h_{\min} = 3$ is indeed tight to ensure separability.

6 Assessing the relevance of separated sequences towards realistic designs

While the existence of a linear-time algorithm for a reasonable restriction of the inverse folding problem is already notable, its practical relevance may be perceived as hindered by several limitations: our algorithms are only guaranteed to produce design solutions for helices beyond 3 base pairs; proper colorings only allows the design of highly-constrained (multi)loops; and solutions to the base pair inverse folding are not guaranteed to represent good solutions in more realistic energy models, such as the Turner nearest-neighbor model. To assess the potential of separated designs to inform future RNA design methods, we performed computational experiments, using a `Python` implementation available at:

https://gitlab.inria.fr/amibio/linearbpdesign

6.1 Targets with isolated BPs/stacks are frequently separable

While our algorithm is only guaranteed to produce a design when $h_{\min} \geq 3$, it also produces (guaranteed correct) solutions for input with smaller helices, as long as a separated coloring exists for them. For very small targets, an exhaustive analysis is feasible, consisting of folding/testing the unicity of the MFE folding for all sequences of length $n = 12$ (see Figure 2). Moreover, once a design w is found for a target T, it is easy to test if the associated coloring χ_w is separated, and to compute minimal modulus value $m^{\ominus}$ such that χ_w is $m^{\ominus}$ separated. We found that *all of the 8 111 designable targets are also separable*, despite a very large proportion of them featuring isolated stacks and base pairs. Moreover, all designable targets admit separated solutions associated with very small values of the modulus m (7 690 for $m = 2$, 420 for $m = 3$ and $m = 1$ only for the empty structure).

To further measure the proportion of separable structures within larger targets featuring isolated stacks, we implemented a uniform random generation algorithm [20]. We produced random target secondary structures of length 100 with a min base pair span of $\theta = 3$. Note that our dynamic programming algorithm does not make use of this property as we forbid every alternative base pair with no regard to the distance between the extremities of these base pairs. However, it is realistic to focus our attention on the target structures with $\theta = 3$ relevant in the Turner energy model. We used rejection to produce a synthetic dataset consisting of 10 000 targets having at least one helix of size 2 while avoiding $m_{3\bullet}$ and m_5. For each target T, we ran an in-house implementation of the algorithm in Section 4.1 with increasing modulus, to find the minimal modulus $m^{\ominus}$ such that T admits a $m^{\ominus}$ separated coloring. Table 13 summarizes our results, which we discuss below.

Remarkably, all of the 10k targets in the datasets could be designed using our algorithm, and thus admit a separable coloring. Moreover, roughly three-quarters (80%) of the targets were found to be 2-separable, and less than 1% of the targets required

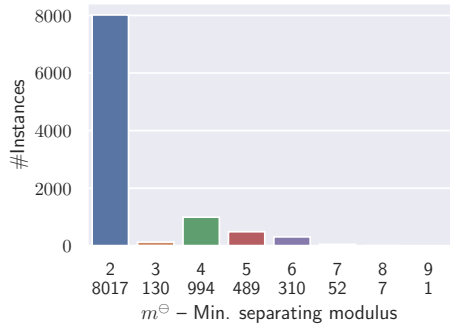


Fig. 13 Minimal modulus $m^{\ominus}$ required to separate 10 000 random targets ($n = 100$; $\theta = 3$) featuring 1^+ isolated stack(s). All targets were found to be separable, with $m^{\ominus} \leq 9$.

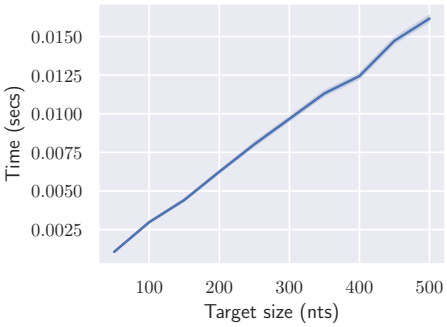


Fig. 14 Average runtime of our algorithm (preprocessing + sampling of single instance) for separable instances ($h_{\min}$=3; no $m_{3\bullet}/m_5$) on a domestic laptop (AMD Ryzen 7 3700U).

the consideration of values for $m^{\ominus}$ beyond 6. The max value for $m^{\ominus}$ in this dataset was 9, an order of magnitude lower than the sequence length. Clearly, since we have shown the existence of non-separable instances with isolated stacks and no isolated base pair, this observation does not generalize to arbitrary sequence lengths. However, the large size of these counterexamples suggests that the proportion of separable structures, despite ultimately decaying exponentially [19], may remain non-negligible for relevant RNA target sizes.

6.2 Separated designs are promising candidates in the Turner model

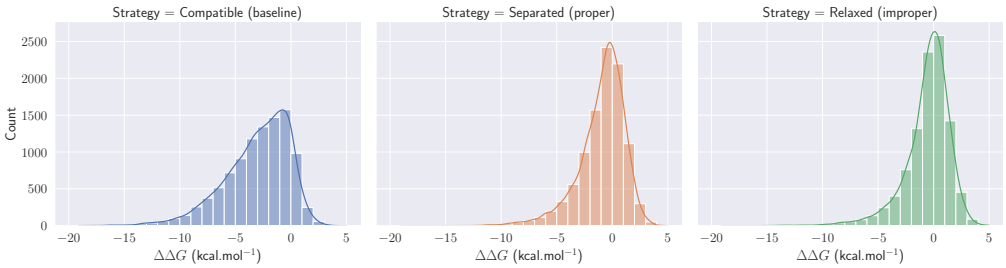
We now consider a more realistic setting, where the inverse folding problem is now considered with respect to the Turner nearest-neighbor energy model [22]. To assess the value of a sequence in the Turner model, we introduce a metrics which we call the (signed) *energy distance* $\Delta\Delta G(w, T)$ of a target T to its *most stable distant alternative* for the sequence w:

$$\Delta\Delta G(w, T) := \Delta G(w, \alpha_{d^-}(w, T)) - \Delta G(w, T),$$

where $\alpha(w, T) := \min\{\Delta G(w, T') \mid |T', T| \geq d^-\}$ with $\Delta G(w, T)$ the Turner free-energy, $|T, T'| := |T \triangle T'|$ denotes the base-pair distance, and d^- represents the minimum base pair distance to T. Both ΔG and $\alpha_{d^-}(w, T)$ can be obtained by appropriate calls to the `ViennaRNA` package [1], namely `RNAeval` and `RNAsubopts`, using max energy distance parameter $E = 5$ (so our estimation of $\Delta\Delta G(w, T)$ is bounded by 5). A positive energy distance confirms that w is a solution to the Turner version of inverse folding, and dominates its competitors by $\Delta\Delta G(w, T)\,\text{kcal.mol}^{-1}$. Meanwhile, a negative energy distance indicates that the target T is dominated by some alternative structure, having $\Delta\Delta G(w, T)\,\text{kcal.mol}^{-1}$ lower free-energy than the target.

We consider three strategies for sampling sequences: i) The *compatible* model uniformly generates random sequences compatible with the target (A for unpaired

a) Target structures avoiding $m_{3\bullet}$ and m_5



b) Target structures featuring multiloops of arbitrary degrees

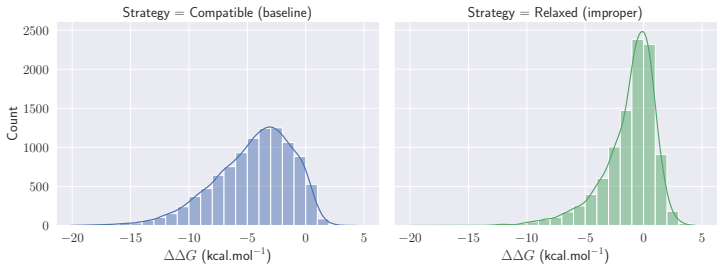


Fig. 15 Comparison of compatible (baseline), separated, and relaxed models for targets having $n = 100, \theta = 3, h_{\min} = 3$. For energy distance parameters, we took $d^- = 3$ and $E = 5$.

positions; AU, UA, GC or CG for base pairs); ii) The *separated* model uses the sampler described in Section 4.2 to generate sequences that are 2-separated and proper; iii) The *relaxed*, sometimes also called the *unproper* model, generates sequences that are 2-separated, but not necessarily proper by assigning uniform random pairs to the base pairs of a multiloop. The *relaxed* model enables a heuristic extension of our algorithms supporting multiloops of arbitrary degrees, noting that the local refolding (see Figure 4) occurring in the BP model for non-proper sequences are either unrealistic or outright impossible, in the Turner energy model.

Separated sequences substantially improve over compatible random sequences. We first asked a basic question: *Are separated sequences better candidates for design in the Turner model than sequences compatible with the target?* The answer is not obvious since separated sequences are only guaranteed to represent designs for the BP max. model. We considered instances of size $n = 100$ admitting a solution to INVERSE-FOLDING$_{\text{BP}}$ ($\theta = 3$; no $m_{3\bullet}/m_5$; $h_{\min} \geq 3$). We generated 10 000 random targets and, for each target, sampled a single sequence using each of the 3 strategies above and computed the energy distance.

The results, summarized in Figure 15.top suggest that separated sequences represent a substantial improvement over merely compatible sequences. Indeed, while 10% of compatible sequences ended up being good design candidates ($\Delta\Delta G > 0$), the proportion of successful designs increases to approximately one-third (35%) for separated sequences, and further to 43% for relaxed design. A similar trend can be observed for the average $\Delta\Delta G$ (distance to the first alternative/competitor) among successful designs, being of 0.79/0.98/1.06 kcal.mol^{-1} in the compatible, separated and relaxed

models respectively. The surprisingly good behavior of the relaxed model, which was mostly introduced to overcome unrealistic limitations on multiloops, remains to be explained. As a small hint why the character unproper of the design does not seem to matter, note first that if we have an unproper m-separated design ω then if it has an unpaired position in a high-degree multiloop M, pairs AU will still be forbidden in M due to the separability condition. Furthermore, most of the local rearrangements will mainly produce some rerooting unrealistic thanks to $\theta = 3$ or that would represent some base pairs that have high chances to worsen the Turner energy of the structure over ω.

Relaxed sequences enable designs for higher degrees multiloops. We also tested the capacity of the relaxed model to generate solutions for multiloops of higher degrees, noting that the avoidance of $m_{3\bullet}$ and m_5 restricts the maximum degree of a multiloop to 4. We used the above-mentioned generation algorithm to generate uniform design targets of size $n = 100$, featuring at least one (but frequently many) occurrence of $m_{3\bullet}$ and m_5. As shown in Figure 15.bottom, compatible sequences are again substantially outperformed by the relaxed separated model in this setting, with 31.5% of the separated/non-proper sequences (as opposed to only 5.1% of compatible sequences) representing successful designs ($\Delta\Delta G > 0$), on average 0.86 kcal.mol^{-1} more stable than their best competitor.

6.3 Using multidimensional Boltzmann sampling to control GC content

Targeting a realistic G+C content (GC%) is a traditional secondary objective of inverse folding [12]. In particular, it is generally believed that solution sequences featuring artificially high GC% ($\gg 50\%$) are somewhat easier to find computationally yet may suffer from slow kinetics due to the transient formation of alternative stable helices which, delaying convergence to the thermodynamic equilibrium.

To control the GC% of m-separated designs produced by our random generation algorithm (cf Section 4.2), we use multidimensional Boltzmann sampling, a technique introduced in the context of enumerative combinatorics [23, 24], and more recently adapted to efficiently constrain stochastic sampling within classified dynamic programming [12, 15, 25, 26]. Its core idea is to induce a Boltzmann distribution for the emission probabilities, such that:

$$\mathbb{P}(w \mid T, \pi_{\mathsf{GC}}) = \frac{e^{\pi_{\mathsf{GC}} \cdot \#\mathsf{GC}(w)}}{\mathcal{Z}_{\pi_{\mathsf{GC}}}}, \quad \mathcal{Z}_{\pi_{\mathsf{GC}}} := \sum_{w' \ m\text{-sep. for } T} e^{\pi_{\mathsf{GC}} \cdot \#\mathsf{GC}(w')},$$

using a weight $\pi_{\mathsf{GC}} \in \mathbb{R}$ whose value can be used to control the expected GC% of generated sequences. Generating in such a distribution can be achieved through a simple adaptation of the random generation algorithm from Section 4.2, to incorporate a multiplicative weight $e^{2\pi_{\mathsf{GC}}}$ anytime a G·C or C·G content is chosen for a given base pair. Then, a rejection strategy keeps only those sequences having desired GC%, resulting in a random generation algorithm with guaranteed uniformity at fixed GC%. Using a

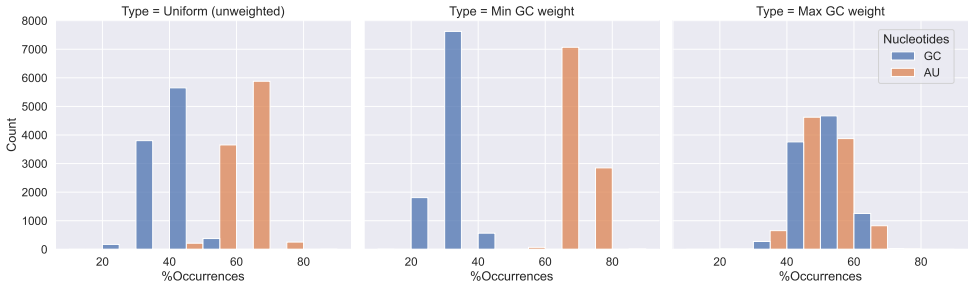


Fig. 16 Modulating the GC% of random 2 separated designs. *Natural* GC% distribution (Left) observed among random 2-separated, uniformly-distributed, sequences for random targets (1 structure per target). Using a Boltzmann distribution $\mathbb{P}(w) \propto e^{\pi_{\mathsf{GC}} \cdot \#\mathsf{GC}(w)}$ instead allows a fine control over the GC%. Moreover, using extremes values of π_{GC}, the GC% can either be minimized (Center), or maximized (Right).

binary search for a suitable value of π_{GC}, the associated average-case time complexity then increases as $\Theta(n\sqrt{n})$ under mild concentration assumptions (*i.e.* asymptotic convergence of GC% to a Normal distribution).

We generated 10 000 uniform random separable structures ($m_{3\bullet}/m_5$-free; $h_{\min} = 3$) of length 100nts. For each structure, we produced a single m-separated design (mod 2-separated sequences), initially in the uniform distribution $\pi_{\mathsf{GC}} = 0$ to determine the typical GC% distribution. The results, summarized in Figure 16 (Left), show that a low average GC% of 40% (40% median) can be consistently reached (5.1% std). Unsurprisingly, extreme GC% values are difficult to reach, both due to the assignment of A to unpaired positions (37.5% of total positions), and the necessity to alternate G·C/C·G and A·U/U·A within multiple loops.

We then reprocessed the same structure dataset, this time using a numerical iteration to determine values of π_{GC} which minimize GC% while avoiding numerical underflows ($-59 \leq \pi_{\mathsf{GC}} \leq -34$). Figure 16 (Center) showcases the resulting GC% distribution, which is tightly concentrated around 32.7% (3.8 std).

Finally, the GC% can be pushed by setting π_{GC} to its maximum while avoiding numerical overflows ($18 \leq \pi_{\mathsf{GC}} \leq 39$). Again, we observe a relatively tight concentration around the mean value of 51% (6.8 std), approximately equating the GC% and AU%.

Overall, this study confirms that modulo 2 separated sequences, despite being a strict subset of all designable sequences, represent a sufficiently rich family to imprint further constraints, as demonstrated here by our modulation of the GC%. Future work may consider the utilization of such sequences as reasonable starting points (*aka* seeds) for design heuristics targeting the Turner energy model [1].

7 Conclusion

Adapting a coloring perspective initially introduced by Halès *et al.* [17], we have shown that the inverse folding problem can be solved in linear time for all target secondary structures having minimum helix length equal to 3. Towards that main result, we have

established the existence of designable, yet non-separable, instances of inverse folding, and the NP-hardness of finding a separable design in the initial sense of Halès *et al.* We have also introduced concrete algorithms for the problem of finding a m modulo-separated coloring, which we have shown to be NP-hard yet FPT-solvable for m. Already for $m = 2$, the scope of our algorithms encompasses all targets without isolated base pairs and stacks, but also extends much beyond, in a way that remains to be fully characterized. Beyond base pair maximization, modulo-separated sequences may also represent a solid foundation towards concrete design methodologies. Namely, we have empirically observed that, for the Turner energy model, separated sequences tend to represent better design candidates than merely compatible sequences, and that the limitations on loop degrees (intrinsic to the BP maximization model) can be overcome by relaxing our design model while retaining substantial performances. Moreover, we have showed that m-separated sequences offer sufficient diversity to modulate the GC content of produced sequences.

Future work should focus on how much of designable sequences are covered by sequences obtained with (modulo)-separated colorings. More importantly, does the space of (modulo)-separated colorings always/often contain a design with respect to the nearest-neighborhood Turner energy model? Even if it unlikely to hold unconditionally, it is plausible that some extensions of separability and m-separability will achieve theoretical and practical solutions for inverse folding in more general energy models. As a first step, separability in a stacking energy model seems a relevant goal, even if less ambitious than the Turner nearest-neighbor model. It would probably require to go beyond the current coloring formalism, and motivate the introduction of more general notions of defects to capture imbalance at the level of dinucleotides compositions. Finally, extensions of this work may explore generalizations of the notion of (m-)separability, possibly in combination with further constraints-based filters, to directly address *real world* design scenarios. Towards that goal, we have introduced the concept of biseparability to enable a joint presence of As and Cs in unpaired positions, and used the produced sequence as a starting point (*aka* seeds) in the context of various popular heuristics, leading to improved performances [27].

References

[1] Hofacker, I.L., Fontana, W., Stadler, P.F., Bonhoeffer, L.S., Tacker, M., Schuster, P.: Fast folding and comparison of RNA secondary structures. Monatshefte für Chemie/Chemical Monthly **125**(2), 167–188 (1994)

[2] Schnall-Levin, M., Chindelevitch, L., Berger, B.: Inverting the viterbi algorithm: an abstract framework for structure design. In: ICML. ACM International Conference Proceeding Series, vol. 307, pp. 904–911. ACM, ??? (2008)

[3] Bonnet, E., Rzazewski, P., Sikora, F.: Designing rna secondary structures is hard. Journal of Computational Biology **27**(3), 302–316 (2020) https://doi.org/10.1089/cmb.2019.0420 https://doi.org/10.1089/cmb.2019.0420. PMID:32160034

[4] Busch, A., Backofen, R.: INFO-RNA–a fast approach to inverse RNA folding. Bioinformatics **22**(15), 1823–31 (2006)

[5] Andronescu, M., Fejes, A.P., Hutter, F., Hoos, H.H., Condon, A.: A new algorithm for rna secondary structure design. Journal of Molecular Biology **336**(3), 607–624 (2004) https://doi.org/10.1016/j.jmb.2003.12.041

[6] Zadeh, J.N., Wolfe, B.R., Pierce, N.A.: Nucleic acid sequence design via efficient ensemble defect optimization. Journal of Computational Chemistry **32**(3), 439–452 (2011) https://doi.org/10.1002/jcc.21633

[7] Retwitzer, M.D., Reinharz, V., Churkin, A., Ponty, Y., Waldispühl, J., Barash, D.: incaRNAfbinv 2.0: a webserver and software with motif control for fragment-based design of RNAs. Bioinformatics **36**(9), 2920–2922 (2020) https://doi.org/10.1093/bioinformatics/btaa039 https://academic.oup.com/bioinformatics/article-pdf/36/9/2920/48986446/bioinformatics_36_9_2920.pdf

[8] Lyngsø, R.B., Anderson, J.W.J., Sizikova, E., Badugu, A., Hyland, T., Hein, J.: Frnakenstein: multiple target inverse RNA folding. BMC Bioinform. **13**, 260 (2012)

[9] Esmaili-Taheri, A., Ganjtabesh, M.: ERD: a fast and reliable tool for RNA design including constraints. BMC Bioinform. **16**, 20–12011 (2015)

[10] Kleinkauf, R., Mann, M., Backofen, R.: antaRNA: ant colony-based RNA sequence design. Bioinformatics **31**(19), 3114–3121 (2015) https://doi.org/10.1093/bioinformatics/btv319

[11] Merleau, N.S.C., Smerlak, M.: arnaque: an evolutionary algorithm for inverse pseudoknotted RNA folding inspired by lévy flights. BMC Bioinform. **23**(1), 335 (2022)

[12] Reinharz, V., Ponty, Y., Waldispühl, J.: A weighted sampling algorithm for the design of RNA sequences with targeted secondary structure and nucleotide

distribution. Bioinformatics **29**(13), 308–315 (2013) https://doi.org/10.1093/bioinformatics/btt217

[13] Yao, H.-T., Waldispühl, J., Ponty, Y., Will, S.: Taming Disruptive Base Pairs to Reconcile Positive and Negative Structural Design of RNA. In: Proc. of the 25th Annual International Conferences on Computational Molecular Biology (RECOMB'21) (2021). https://inria.hal.science/hal-02987566

[14] Garcia-Martin, J.A., Dotu, I., Clote, P.: RNAiFold 2.0: a web server and software to design custom and Rfam-based RNA molecules. Nucleic Acids Research **43**(W1), 513–521 (2015) https://doi.org/10.1093/nar/gkv460 https://academic.oup.com/nar/article-pdf/43/W1/W513/7476300/gkv460.pdf

[15] Hammer, S., Wang, W., Will, S., Ponty, Y.: Fixed-parameter tractable sampling for rna design with multiple target structures. BMC Bioinformatics **20**(1) (2019) https://doi.org/10.1186/s12859-019-2784-7

[16] Runge, F., Stoll, D., Falkner, S., Hutter, F.: Learning to design RNA. In: Proceedings of ICLR 2019 (2019)

[17] Hales, J., Héliou, A., Manuch, J., Ponty, Y., Stacho, L.: Combinatorial RNA design: Designability and structure-approximating algorithm in watson-crick and nussinov-jacobson energy models. Algorithmica **79**(3), 835–856 (2017)

[18] Nussinov, R., Jacobson, A.B.: Fast algorithm for predicting the secondary structure of single-stranded rna. Proceedings of the National Academy of Sciences **77**(11), 6309–6313 (1980) https://doi.org/10.1073/pnas.77.11.6309 https://www.pnas.org/doi/pdf/10.1073/pnas.77.11.6309

[19] Yao, H.-T., Chauve, C., Regnier, M., Ponty, Y.: Exponentially few RNA structures are designable. In: Conference on Bioinformatics, Computational Biology, and Health Informatics ACM-BCB, pp. 289–298. ACM Press, Niagara-Falls, United States (2019). https://doi.org/10.1145/3307339.3342163 . https://inria.hal.science/hal-02141853

[20] Ponty, Y.: Ensemble Algorithms and Analytic Combinatorics in RNA Bioinformatics and Beyond. Habilitation à diriger des recherches, Université Paris-Saclay (May 2020). https://theses.hal.science/tel-03219977

[21] Lorenz, W.A., Ponty, Y.: Non-redundant random generation algorithms for weighted context-free grammars. Theoretical Computer Science **502**, 177–194 (2013) https://doi.org/10.1016/j.tcs.2013.01.006 . Generation of Combinatorial Structures

[22] Turner, D.H., Mathews, D.H.: NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. Nucleic Acids Research **38**(suppl_1), 280–282 (2009) https://doi.org/10.1093/nar/gkp892

[23] DUCHON, P., FLAJOLET, P., LOUCHARD, G., SCHAEFFER, G.: Boltzmann samplers for the random generation of combinatorial structures. Combinatorics, Probability and Computing **13**(4–5), 577–625 (2004) https://doi.org/10.1017/S0963548304006315

[24] Bodini, O., Ponty, Y.: Multi-dimensional Boltzmann Sampling of Languages. Discrete Mathematics & Theoretical Computer Science **DMTCS Proceedings vol. AM 21st International Meeting on Probabilistic Combinatorial and Asymptotic Methods in the Analysis of Algorithms (AofA'10)** (2010) https://doi.org/10.46298/dmtcs.2793

[25] Waldispühl, J., Ponty, Y.: An unbiased adaptive sampling algorithm for the exploration of rna mutational landscapes under evolutionary pressure. Journal of Computational Biology **18**(11), 1465–1479 (2011) https://doi.org/10.1089/cmb.2011.0181 . PMID: 22035326

[26] Yao, H.-T., Marchand, B., Berkemer, S.J., Ponty, Y., Will, S.: Infrared: a declarative tree decomposition-powered framework for bioinformatics. Algorithms for Molecular Biology **19**(1) (2024) https://doi.org/10.1186/s13015-024-00258-2

[27] Boury, T., Sidl, L., Hofacker, I.L., Ponty, Y., Yao, H.-T.: Old dog, new tricks: Exact seeding strategy improves RNA design performances. Submitted to recomb 2025 (2024). https://hal.science/hal-04756160