



HAL
open science

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

Théo Boury, Laurent Bulteau, Yann Ponty

► To cite this version:

Théo Boury, Laurent Bulteau, Yann Ponty. RNA inverse folding can be solved in linear time for structures without isolated stacks or base pairs. 2024. hal-04589901v2

HAL Id: hal-04589901

<https://hal.science/hal-04589901v2>

Preprint submitted on 5 Jun 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

1 RNA inverse folding can be solved in linear time 2 for structures without isolated stacks or base pairs

3 Théo Boury 

4 Laboratoire d'Informatique de l'Ecole Polytechnique (LIX; UMR 7161), Institut Polytechnique de
5 Paris, France

6 Laurent Bulteau  

7 LIGM, CNRS, Université Gustave Eiffel, France

8 Yann Ponty¹  

9 Laboratoire d'Informatique de l'Ecole Polytechnique (LIX; UMR 7161), Institut Polytechnique de
10 Paris, France

11 — Abstract —

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

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

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

27 **2012 ACM Subject Classification** Applied computing → Molecular structural biology

28 **Keywords and phrases** RNA structure, String Design, Parameterized Complexity, Uniform Sampling

29 **Digital Object Identifier** [10.4230/LIPIcs.WABI.2024.XXX](https://doi.org/10.4230/LIPIcs.WABI.2024.XXX)

30 **Supplementary Material** The source code for the project are on [GitLab:Linear-BP-Design](https://gitlab.com/Linear-BP-Design).

¹ Both second and third authors should be considered as corresponding authors



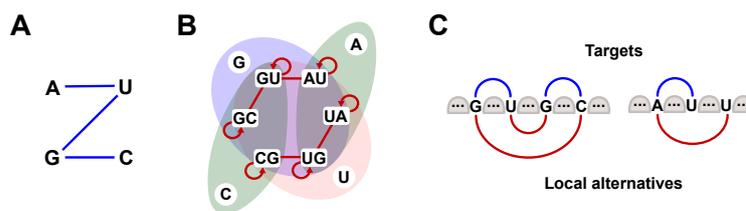
1 Introduction

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 $\Phi : \{A, C, G, U\}^* \rightarrow \mathcal{S} \cup \{\perp\}$ 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 [8], 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* [18]. 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 2018, when Bonnet *et al* [2] 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 [8, 3, 1, 22, 16], bio-inspired metaheuristics [11, 4, 9, 12], global sampling [15, 21], constraint programming [5, 7] and, more recently, neural networks-inspired generative models [17].

In parallel to complexity studies, Hales *et al* [6] 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* [2]) and practical facility for inverse folding.

In this work, we further those studies and show that, while conceptually simpler, the existence of a separated design for a given structure remains NP-hard. Conversely, any structure with helices of length greater than 3 base pairs is either trivially undesignable (*i.e.* contains $\{m_{3\bullet}, m_5\}$), or separable and can 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. This result hinges on the introduction of a modulo m version of separability, coinciding with general separability whenever $m \geq n/2$, for which we give a Fixed-Parameter Tractable (FPT) algorithm running in time $\mathcal{O}(n \cdot 2^m)$. We proved that this algorithm solves all instances with minimal helix lengths of 3 BPs when invoked with $m = 2$ and, even in this restricted setting, solves many instances with shorter



■ **Figure 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).

78 helices in practice. Based on an unambiguous dynamic programming, our algorithm can be
 79 adapted into a random generator of separated designs. Finally, we show through empirical
 80 studies that separated sequences, despite being only guaranteed to constitute designs with
 81 respect to base pair maximization, are also likely to represent designs in the more realistic
 82 Turner energy model, and are far superior in this setting than compatible sequences.

2 Problem statement, definitions, and prior work

84 Algorithmically, RNA can be abstracted as a nucleotide sequence, *i.e.* a string $w \in$
 85 $\{A, C, G, U\}^n$ where n denotes the length of w . Given a length n , a (non crossing/pseudoknot-
 86 free) secondary structure is a set $T \subset [1, n]^2$ consisting of base pairs such that:

- 87 ■ Each position in $[1, n]$ is involved in at most one base pair;
- 88 ■ Base pairs in T are pairwise non-crossing: $\forall (i, j) \neq (k, l) \in T, i < k$, either $i < k < l < j$
 89 or $i < j < k < l$.

90 The set \mathcal{S}_w of secondary structures compatible with an RNA sequence w is defined as:
 91 $\mathcal{S}_w := \{\text{Secondary structure } T \mid \forall (i, j) \in T, \{w_i, w_j\} \in \{\{G, C\}, \{A, U\}, \{G, U\}\}\}$.

92 Without loss of generality, a secondary structure can be represented as a tree $T =$
 93 $(V(T), E(T))$, whose nodes $V(T)$ are in bijection with base pairs (internal nodes²) and
 94 unpaired regions (leaves), and whose edges represent the inclusion of base pairs. Given a
 95 node $v \in V(T)$, we denote by $\text{parent}(v)$ the parent of v in T , and by $\text{children}(v)$ the list of
 96 children of v in T . A *loop* is the subtree restricted to node and its (direct) children. The tree
 97 is rooted in a special **Root** node, associated with the whole sequence interval. An *helix* of
 98 length ℓ of the tree is a maximal path v_1, \dots, v_ℓ of base pair nodes such that each v_i with
 99 $i < \ell$ has a single child v_{i+1} (no leaf attached). A helix of length 1 is an *isolated base pair*. A
 100 helix of length 2 is an *isolated stack*. We define h_{\min} as the minimum length over all helices
 101 of T . As the target tree is always explicit and unmodified through proofs and algorithms we
 102 do not specify it explicitly in the notations.

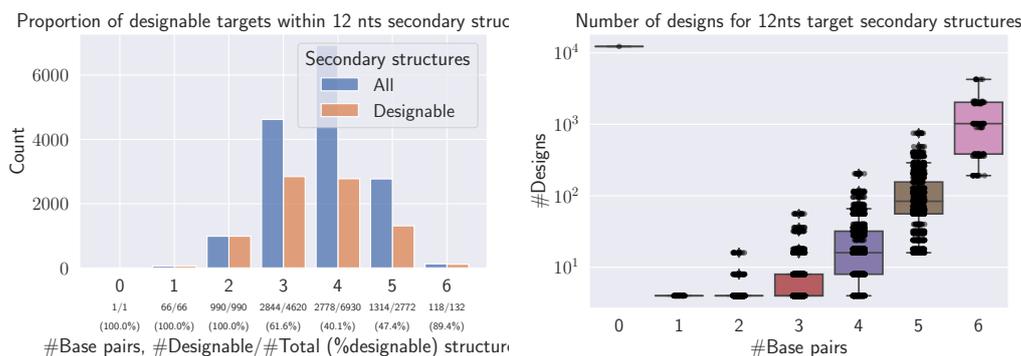
103 RNA inverse folding starts from a target secondary structure T , and attempts to construct
 104 a sequence $\omega \in \{A, C, G, U\}^n$ whose only base-pair maximizing secondary structure is T .

105 ► **Problem 1.** INVERSE-FOLDING_{BP}

106 **Input:** Target secondary structure T , sequence length n

107 **Output:** Sequence $w \in \{A, C, G, U\}^n$ satisfying both:

² 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.



■ **Figure 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.

- 108 ■ *Compatibility with target structure:* $T \in \mathcal{S}_w$;
 109 ■ *Uniqueness of the target as the optimal fold for the sequence:* $\forall T' \in \mathcal{S}_w, T' \neq T, |T'| < |T|$.
 110 or \perp if no such sequence exists.

111 Nevertheless, INVERSE-FOLDING_{BP}, mildly extended to allow further restrictions on individual
 112 sequence positions, was shown to be NP-hard by Bonnet *et al* [2].

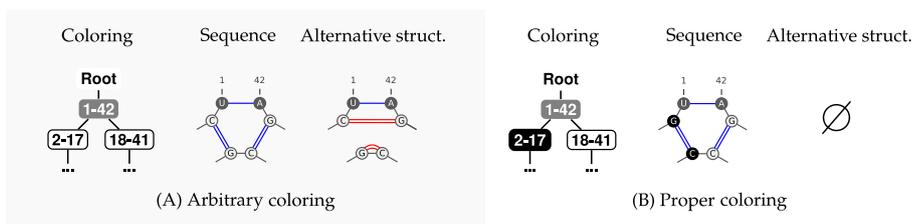
113 A sequence is called a design for a structure T if it represents a solution to the inverse
 114 folding problem for the input T . Note that the uniqueness condition can be tested in
 115 polynomial time using a variant of the Nussinov algorithm [13, 6]. In addition to showing
 116 that INVERSE-FOLDING_{BP} is in P, such an algorithm enables, for moderate sequence lengths,
 117 a systematic folding of all sequences in order to characterize the set of structures admitting
 118 a solution. For instance, Figure 2 shows that, while only 2.4% of RNA sequences of length
 119 12 represent a design for some target, roughly half of the secondary structure admits at least
 120 one solution sequence, and ≈ 49 on average, for the inverse folding problem.

121 We remind that, as noted by Halès *et al* [6], two key motifs are not designable in a *base*
 122 *pair maximization* setting:

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

133 Any occurrence of these structures (or of any other undesignable structure, *cf* [20]) as a
 134 subgraph of an instance makes the instance undesignable.

XXX:4 Exact linear-time RNA design for min Helix length 3



■ **Figure 3 A proper coloring is necessary towards design.** In (A), having two \circ 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.

135 2.1 Inverse folding as a tree coloring problem

136 We start by reminding the coloring framework introduced by Halès *et al* [6].

137 ► **Definition 1 (Coloring).** A coloring of a (secondary structure) tree T is a function $\chi : V(T) \rightarrow \{\bullet, \circ, \emptyset\}$ associating a color to each node (except the root and the leaves which always get \emptyset).

140 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 $(i, j) \in T$:

- 143 ■ If $\chi((i, j)) = \bullet \rightarrow (w_i, w_j) = (G, C)$;
- 144 ■ If $\chi((i, j)) = \circ \rightarrow (w_i, w_j) = (C, G)$;
- 145 ■ If $\chi((i, j)) = \bullet \rightarrow (w_i, w_j) \in \{(A, U), (U, A)\}$.

146 For \bullet 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_{BP}, since the problem asks for the production of a single sequence.

152 Denote by \bar{c} the inverse of a color c , defined as $\bar{\circ} = \bullet$, $\bar{\bullet} = \circ$ and $\bar{\bullet} = \bullet$.

► **Definition 2 (Proper Coloring).** A coloring χ is proper when, for each node $v \in V(T)$, the vector of colors C , assigned to the node and its children, respects the following constraints:

$$|C|_{\bullet} \leq 1, |C|_{\circ} \leq 1 \text{ and } |C|_{\bullet} \leq 2 \text{ with } C := [\bar{\chi(v)}].[\chi(v') \mid v' \in \text{children}(v)].$$

153 These conditions must also hold for the colorless Root, but with C being restricted to the colors of children(Root).

155 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 will be associated with sequences that can be locally reconfigured, this without losing any base pair (see Figure 3 for an example).

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

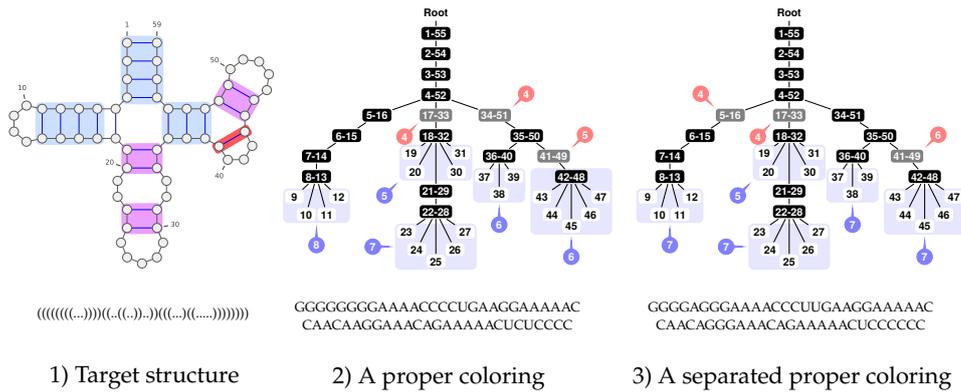


Figure 4 1) 2D and dot-bracket representations of a secondary structure. Helices of sizes respectively 1, 2 and 3 are represented in light red, purple and blue. 2) Same secondary structure as a tree. The tree is colored with a proper non-separated coloring as the level of the leaf 19 is the same as the level of the \bullet 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 \bullet nodes). Notably, this coloring is 2-separated, as leaves and \bullet nodes end up at odd and even levels respectively.

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

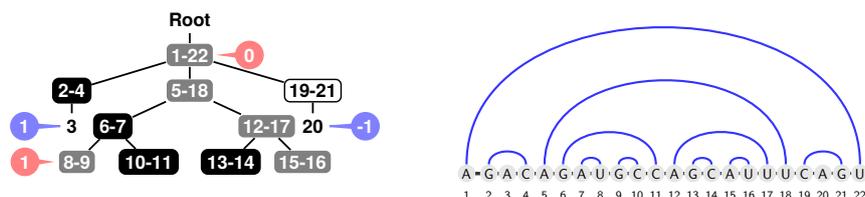
► **Definition 4** (Separated coloring). A coloring χ is separated for a target T if and only if it is proper and the levels of \bullet -colored nodes and leaves do not overlap:

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

171 This immediately suggests a design strategy that associates A to unpaired positions and
 172 assigns \bullet and \circ colors such that \bullet nodes end up as different levels as the leaves. Indeed,
 173 in this setting, Hales *et al* [6] showed that the proper coloring of a saturated structure
 174 (without unpaired position) yields a sequence that uniquely folds with respect to base pair
 175 maximization. It follows that a competitive/alternative structure may only result from a base
 176 pair $(i, j) \notin T$, a position of which is a \bullet node while the other is a leaf. Ensuring that all \bullet
 177 nodes and leaves are found at different levels is thus sufficient to guarantee the designability
 178 of T , *i.e.* the existence of a solution to this instance of the inverse folding problem.

179 More generally, we say that a target secondary structure T is *separable* if there exists a
 180 coloring χ such that χ is separated for T . We recall the main results of Halès *et al* [6] here.

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



■ **Figure 5 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).

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

185 ► **Remark 2.** Note that any design sequence w , generated through a separated coloring,
 186 avoids any alternative structure featuring GU base pair(s). Indeed, every G and C need to be
 187 paired to achieve the number of base pairs featured in the MFE. Meanwhile, the formation
 188 of any GU base pair, leaves one C and one A unpaired, resulting in the overall loss of at least
 189 one base pair. Structures featuring GU base pairs can thus be safely ignored.

190 3 Separability: Intrinsic and computational limits

191 Despite utilizing separability to explore a design of approximative instances, the work of Halès
 192 *et al* [6] left open the complexity of searching for a separated coloring, as well as the existence
 193 of designable, yet non-separable, structures. An exhaustive search for all structures with
 194 up to 12 bases, summarized in Figure 2, shows that for such small instances, all designable
 195 instances are separable.

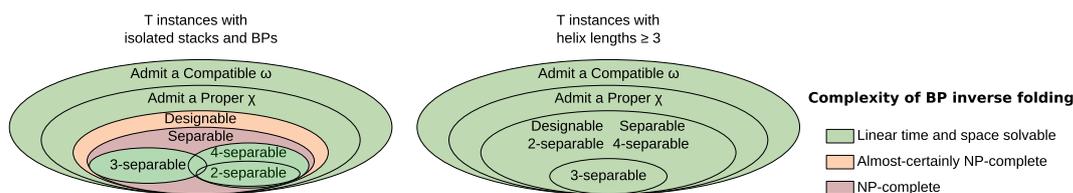
196 However, we show that non-separable designable instances can be constructed.

197 ► **Proposition 1** (Designable $\not\Rightarrow$ Separable). *There exists a target structure which: i) does*
 198 *not admit a separated coloring; and ii) admits a solution to the inverse folding problem.*

199 **Proof.** We use the tree T of Figure 5 as a counterexample to the notion that separability
 200 fully captures designability. First, note that a separated coloring χ of T would be extremely
 201 constrained. Node 5 – 18 should be ● and the nodes 2 – 4 and 19 – 21 are ● and ○
 202 respectively, or vice-versa due to their respective leaf. Thus, we have two leaves at levels 1
 203 and –1. At least, one of the two children of 5 – 18, w.l.o.g 6 – 7 is ● or ○. One child of
 204 6 – 7 is then necessarily ●, leading to a ● child of level +1 or –1. With two leaves at level
 205 +1 and –1, a direct consequence is that T is non-separable.

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

213 Notice that, despite not being separated, the coloring shown in Figure 5 is compatible with
 214 a sequence that is a design for its target. This illustrates the fact that, while not being



■ **Figure 6 Instances of $\text{INVERSE-FOLDING}_{\text{BP}}$.** For unconstrained instances (Left), $\text{INVERSE-FOLDING}_{\text{BP}}$ is likely NP-hard, as suggested by the hardness of a constrained version [2]. 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), Thm 6 applies and the hierarchy collapses: any instance becomes 2-separable (\implies separable and designable) and $\text{INVERSE-FOLDING}_{\text{BP}}$ can be solved in $\Theta(n)$ time.

215 guaranteed to uniquely fold as their intended target, sequences produced from non-separated
 216 colorings may still represent solutions for the inverse folding problem.

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

220 ► **Problem 2. SEPARABILITY**

221 **Input:** Target tree T (without any occurrence of m_3 or m_5 motif)

222 **Output:** Coloring χ of the tree T such that χ is separated

223 ► **Theorem 3. SEPARABILITY is NP-complete.**

224 The proof can be found in the appendix. It is obtained by reduction from BIN PACKING,
 225 with a tree using one branch per item. Leaves and \bullet nodes enforce that items must be
 226 packed in consecutive ranges of levels (with \bullet levels at transitions between successive items
 227 and other levels saturated with leaves). Then, separating \bullet nodes are placed to enforce
 228 that series of consecutive items sum up to the target bin size, thus enforcing that items are
 229 ordered according to a correct bin packing.

230 **4 Modulo separability as a parameterized tractable alternative**

231 Then, we introduce a stratified version of separability, called modulo m -separability, or
 232 m -separability in short, which prescribes different modular values for the levels of \bullet and
 233 leaves nodes. Figure 6 describes the relative positioning of classes of instances and associated
 234 complexity results.

► **Definition 5 ((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 and only if χ is proper
 and

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

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

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

237 Clearly, modulo separability implies classic separability: if a coloring χ is m -separated for
 238 a target structure T , then χ is separated for T . Conversely, if a target structure admits a
 239 separated coloring, assigning levels in $[-a, b]$ to \bullet and leaf nodes, then the same coloring
 240 is provably m' -separated for $m' := (b + a + 1)$ (since, for $l, l' \in [-a, b]$, $l \neq l'$ implies that

XXX:8 Exact linear-time RNA design for min Helix length 3

241 $l \bmod m' \neq l' \bmod m'$). Note that, since there are at most $n/2$ base pairs/internal nodes in
 242 a target tree, then $0 \leq a, b \leq n/2$, and we have $m' \leq n$.

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

245 ► **Problem 3. MODULO SEPARABILITY**

246 **Input:** A tree T (with no $m_{3\bullet}$ or m_5 motif), a modulus $m \in \mathbb{N}$

247 **Output:** A coloring χ of T that is m -separated, or \perp if no such coloring exists.

248 As noted above, the problem specializes in the SEPARABILITY problem when $m = n$, implying
 249 that MODULO SEPARABILITY remains NP-complete. However, it can be efficiently solved for
 250 moderate values of m , as shown below. Practically, one may focus on small values of m since
 251 99% of instances without isolated base pairs are separable with modulus $m \leq 6$ (cf Table 9).

252 4.1 Fixed parameter tractable algorithm for modulo-separability

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

255 Towards that goal, we consider a constrained version of MODULO SEPARABILITY, where
 256 the modular values of levels are prescribed. Formally, we enforce that leaves only occur at
 257 modular levels in $\xi_L \subseteq [0, m[$, and \bullet nodes only occur at levels $[0, m[\setminus \xi_L$. In this constrained
 258 version of MODULO SEPARABILITY, the existence of a valid solution can be solved in linear
 259 time using dynamic programming.

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

$$\begin{aligned}
 265 \quad d_{v \rightarrow c, \ell}^{\xi_L} = & \begin{cases} \text{False} & \text{if } \ell \in \xi_L \wedge c = \bullet \\ & \text{or } \ell' \notin \xi_L, \text{ and } \exists \text{ leaf in children}(v) \\ \text{True} & \text{if children}(v) = \emptyset \\ & \bigvee_{\substack{c' \text{ proper} \\ \text{coloring of} \\ \text{children}(v) \\ \text{given } v \rightarrow c}} \bigwedge_{v' \in \text{children}(v)} d_{v' \rightarrow c'(v'), \ell'}^{\xi_L} & \text{otherwise.} \end{cases} \\
 266 \quad & \text{with } \ell' := \ell + \delta(c) \bmod m
 \end{aligned}$$

267 where δ denotes the level increment induced by a color c , defined as $\delta(\bullet) = +1$, $\delta(\circ) = -1$
 268 and $\delta(\bullet) = 0$. Moreover, in the outermost loop, the color assignment explored for children is
 269 meant to be locally proper: the colors $c(v')$ of the children, in conjunction with the color
 270 c of v must obey the conditions of Definition 2. Note that, in the absence of $m_{3\bullet}$ and m_5 ,
 271 the number of (proper) assignments is bounded by a constant, so this conjunctive loop
 272 does not impact the complexity. The existence of a ξ_L coloring for the full tree is then
 273 $\text{Separable}_{\xi_L} := d_{\text{Root} \rightarrow \emptyset, 0}^{\xi_L}$.

274 The decision version of the problem can thus be solved in $\Theta(m.n)$ time. Indeed, the
 275 number of left-hand side terms scales in $\Theta(m.n)$, the number of proper coloring for children
 276 is bounded by a constant (since avoiding $m_{3\bullet}$ and $m_5 \implies |\text{child}(v)| < 5$), and the total

277 number of executions of the conjunctive loops is in overall $\Theta(n)$. A backtracking procedure
 278 could also be defined to reconstruct a solution coloring in $\Theta(n)$ if such a solution exists
 279 ($\text{Separable}_{\xi_L} = \text{True}$) or return \perp otherwise ($\text{Separable}_{\xi_L} = \text{False}$).

280 An algorithm for MODULO SEPARABILITY can then be obtained by explicitly considering
 281 all the possible subsets of admissible modular levels for leaves:

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

286 The algorithm is correct since any ξ_L solution is also m -separated, and any m -separated
 287 coloring implies a partition of the leaves and \bullet nodes into disjoint levels ξ_L and $\chi_{\bullet} \subseteq [0, m[\setminus \xi_L$
 288 respectively. A m -separated coloring is thus always found by invoking the DP algorithm over
 289 the 2^m subsets $\xi_L \in [0, m[$. The overall complexity of the algorithm is in $\Theta(n.m.2^m)$ time
 290 and $\Theta(m.n)$ memory, and we conclude with the parameterized complexity of the problem
 291 with respect to m .

292 ► **Theorem 4.** MODULO SEPARABILITY is Fixed Parameter Tractable for the modulus m

293 4.2 Random generation of m -separated RNA sequences

294 We then turn to the uniform random generation of m -separated sequences, defined as a
 295 design w for T , featuring A on unpaired positions, and such that the coloring χ_w , obtained by
 296 replacing base pairs with suitable color ($(G, C) \rightarrow \bullet$, $(C, G) \rightarrow \circ$ and (A, U) or $(U, A) \rightarrow \bullet$),
 297 is m -separated.

► **Problem 4.** 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

$$\mathbb{P}(w \mid \chi_w \text{ is } m\text{-separated}) = \frac{1}{|\{w' \mid \chi_{w'} \text{ is } m\text{-separated}\}|}$$

298 Again, we approach this problem by first solving a more constrained version where the
 299 modular levels of leaves are explicitly given as a set ξ_L . Then, in the spirit of Reinharz *et*
 300 *al* [15], we adapt the above recurrence, through a simple algebra change, to count the number
 301 $p_{v \rightarrow \mu, l}^{\xi_L}$ of RNA sequences, associated with a ξ_L separated coloring (for a subtree of T rooted
 302 at v , with v occurring at level l , and being assigned a nucleotide assignment μ).

$$303 \quad p_{v \rightarrow \mu, l}^{\xi_L} = \begin{cases} 0 & \text{if } l \in \xi_L \text{ and } \mu \in \{(A, U), (U, A)\} \\ 0 & \text{if } l' \notin \xi_L \text{ and } v \text{ has a leaf attached} \\ 1 & \text{if } \text{children}(v) = \emptyset \\ \sum_{\substack{\mu' \text{ proper assignment} \\ \text{children}(v) \rightarrow \Sigma^2 \cup \{\emptyset\}}} \prod_{v' \in \text{children}(v)} p_{v' \rightarrow \mu'(v'), l'}^{\xi_L} & \text{otherwise } (l' := l + \delta(\mu) \bmod m). \end{cases}$$

304 where μ' is a nucleotide assignment to the children of v , consistent with a proper coloring
 305 and additionally respecting natural constraints on the content $((A, U)$ or $(U, A))$ of pairs of
 306 \bullet nodes (same for both if one parent of other, different content if siblings). Once again, the
 307 colorless Root node needs to be distinguished, and the overall number of designs is given by
 308 $\#\text{Designs}_{\xi_L} := p_{\text{Root} \rightarrow \emptyset, 0}^{\xi_L}$.

XXX:10 Exact linear-time RNA design for min Helix length 3

309 The following backtrack procedure then produces a uniform random RNA sequence that
 310 corresponds to a m -separated coloring for a fixed set ξ_L . In that case, by abuse of language,
 311 we say that the sequence is ξ_L *separated*. More precisely, $\text{backtrack}(v, c, \ell)$ produces a random
 312 sequence, associated with a ξ_L separated coloring, for the subtree anchored in v , reached at
 313 height ℓ , where the root is assigned a pair of bases $\mu \in \Sigma^2$. It first picks a random proper
 314 assignment μ' for the children, weighted by the corresponding number of solutions (namely,
 315 $\prod_{v' \in \text{children}(v)} p_{v' \rightarrow \mu'(v'), \ell'}^{\xi_L}$, with $\ell' := \ell + \delta(\mu) \pmod m$). The resulting sequence is then

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

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

319 To leverage the uniform generation for a fixed ξ_L into a uniform generation of m -separated
 320 designs, we implement a strategy (see [14, pp 77] for details), proven in Appendix C, which
 321 start by generating some ξ_L , and then uses a suitable rejection to correct the emissions
 322 probabilities of sequences compatible with several ξ_L .

323 ► **Theorem 5.** UNIFORM MODULO SEPARATED GENERATION *can be performed in an*
 324 *average-case complexity that is Fixed Parameter Tractable for the modulus parameter m .*

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

326 Although separability does not give a full characterization of designability in general (cf
 327 Prop. 1), we obtain a much stronger result for structures without small helices, as hinted by
 328 the fact that all counter-examples and hardness gadgets heavily use isolated base pairs in
 329 their construction. Indeed, we show that a 2-separated coloring can be constructed for *all*
 330 structures without forbidden motifs $(m_{3\bullet}, m_5)$ and $h_{\min} \geq 3$, so indeed all such structures
 331 are designable. Since avoiding $(m_{3\bullet}, m_5)$ is a necessary condition for designability, we obtain
 332 the stronger characterization stated in Corollary 9.

333 ► **Theorem 6.** *Every $(m_{3\bullet}, m_5)$ -avoiding target T , having $h_{\min} \geq 3$, admits a 2-separated*
 334 *coloring*

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

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

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

- 343 ■ Any proper coloring of a multiloop features at least one \bullet node, so the levels of chil-
 344 dren/nodes need to be set to a level $\bar{\xi}_L := \xi_L + 1 \pmod 2$;
- 345 ■ Any IBH loop features at least one leaf within its children, which needs to be set to a
 346 modular level ξ_L .

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

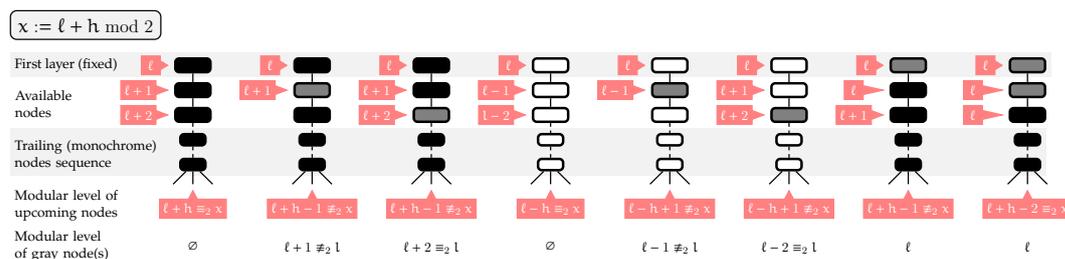


Figure 7 Alternative colorings for helices consisting of 3+ base pairs ($h_{\min} \geq 3$), such that the modular level of the following nodes is offset as needed. Such colorings can be chosen to respect a prescribed level for \bullet nodes and, a predetermined color for the first node/base pair of the helix.

349 ▶ **Lemma 7.** Let $\bar{\xi}_L$ denote the prescribed modular level for \bullet nodes. Consider an helix H
 350 consisting of 3 BPs or more ($h_{\min} \geq 3$), whose first BPs is assigned some color $c \in \{\bullet, \circ, \circ\}$.

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

- 353 ■ The modular level of the upcoming nodes, i.e. those immediately following H , is l' ;
- 354 ■ Base pairs can only be \bullet -colored at modular level $\bar{\xi}_L$.

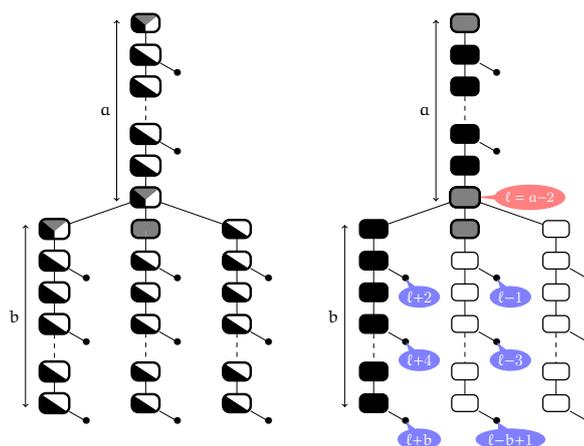
355 **Proof.** The proof is essentially based on case decomposition, and summarized in Figure 7.
 356 We show that, for any l and $h_{\min} \geq 3$, there exists a color assignment to the first 3 nodes
 357 of the helix, such that the modular level of upcoming nodes is either 0 or 1, so l' can be
 358 reached. Moreover, if such a coloring starts with \bullet or \circ , and uses a single \bullet node, then
 359 there exists an alternative coloring placing this \bullet node at the opposite modular level, so
 360 one of them places their \bullet node at the intended level $\bar{\xi}_L$. Finally, if the first node is set
 361 to \bullet , then the consistency condition above implies that $l \bmod 2 = \bar{\xi}_L$, so that \bullet nodes are
 362 naturally found at an admissible modular level. ◀

363 It follows that any helix tree starting with an initial helix H can be colored into a 2-separated
 364 coloring. Starting at initial level $l = 0$ and having initial BP color c ($\neq \bullet$ if $\xi_L = 0$), color
 365 the rest of H as shown in the proof of Lemma 7, depending on $\bar{\xi}_L$ and the type of upcoming
 366 loop (target $l' = \bar{\xi}_L$ for Multiloops; $l' = \xi_L$ for IBH loops), while ensuring that \bullet nodes end
 367 up at $\bar{\xi}_L$ modular level (which can always be done from Lemma 7). The remaining nodes of
 368 the loop are then colored in a proper/greedy manner, and we iterate the process recursively
 369 on the children helices of the loop (if any) until the full tree is colored.

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

377 ▶ **Corollary 8.** INVERSE FOLDING, restricted to instances with $h_{\min} \geq 3$ (containing no
 378 isolated base pair and no isolated stacks) is solvable in linear time and space.

379 It is a direct consequence of Theorem 6 and of the DP scheme introduced in Section 4.1.
 380 Indeed, for $m = 2$, the DP algorithm only needs to be run twice ($\xi_L = 0$ and $\xi_L = 1$) in linear
 381 time/space, to produce a 2-separated coloring whenever such a coloring exists (guaranteed
 382 by Theorem 6). The coloring can then be transformed into a design, i.e. a solution to the



■ **Figure 8** 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 \bullet nodes. Note that along with the \bullet 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 .

383 INVERSE FOLDING problem. Similarly, UNIFORM MODULO SEPARATED GENERATION can
 384 also be performed in linear expected time and space as long as input instances contain only
 385 helices of size 3 or more.

386 ► **Corollary 9.** *Let T be a target structure with $h_{\min} \geq 3$, then the following are equivalent:*
 387 *i) T is designable; ii) T is 2-separable; and iii) T avoids $(m_{3\bullet}, m_5)$.*

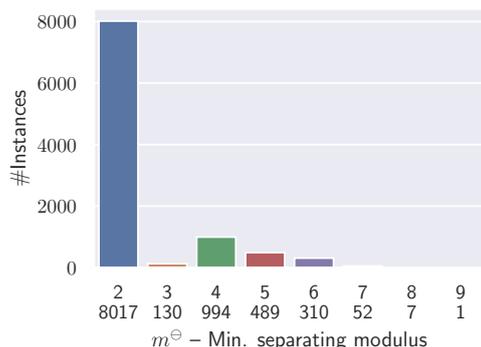
388 With this result, the hierarchy of instances collapses as depicted on the left of Figure 6 A
 389 natural follow-up question is whether the bound 3 on the helix length is tight. Indeed, there
 390 are non-separable and designable instances with $h_{\min} = 1$ (Proposition 1), but the question
 391 remains for $h_{\min} = 2$. In Proposition 10 we give a non-separable instance without isolated
 392 base pairs, so $h_{\min} = 3$ is indeed tight to ensure separability.

393 ► **Proposition 10.** *There exist non-separable structures with $h_{\min} = 2$.*

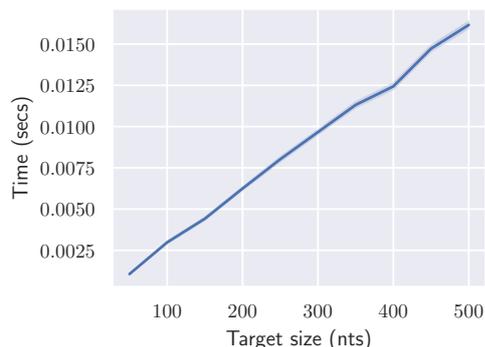
394 The full proof relies on a counterexample built from the gadget in Figure 8. Intuitively,
 395 $T(a, b)$ saturates all levels modulo b with leaves, so that none remains available for \bullet nodes.
 396 Meanwhile, the presence of multiloops forces proper colorings to use \bullet nodes, so a collision
 397 occurs and the gadget is not m -separable for any $m \leq b$. By assembling 5 copies of $T(a, b)$
 398 with large b and increasing values of a , we obtain a target that is not separable for any m .

399 6 On the relevance of separated sequences towards realistic designs

400 While the existence of a linear-time algorithm for a reasonable restriction of the inverse folding
 401 problem is already notable, its practical relevance may be perceived as hindered by several
 402 limitations: our algorithms are only guaranteed to produce design solutions for helices beyond
 403 3 base pairs; proper colorings only allows the design of highly-constrained (multi)loops; and
 404 solutions to the base pair inverse folding are not guaranteed to represent good solutions in more
 405 realistic energy models, such as the Turner nearest-neighbor model. To assess the promises
 406 of separated designs in realistic settings, we performed computational experiments, using a
 407 Python implementation available at <https://gitlab.inria.fr/amibio/linearbpdesign>,
 408 to assess the potential of separated colorings to inform future RNA design methods.



■ **Figure 9** Minimal modulus \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 $\ominus \leq 9$.



■ **Figure 10** 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).

409 6.1 Targets with isolated BPs/stacks are frequently separable

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

420 To further measure the proportion of separable structures within larger targets featuring
 421 isolated stacks, we implemented a uniform random generation algorithm [14]. We produced
 422 random target secondary structures of length 100 with a min base pair span of $\theta = 3$. We
 423 used rejection to produce a synthetic dataset consisting of 10 000 targets having at least
 424 one helix of size 2 while avoiding $m_{3\bullet}$ and m_5 . For each target T , we ran an in-house
 425 implementation of the algorithm in Section 4.1 with increasing modulus, to find the minimal
 426 modulus m^\ominus such that T admits a m^\ominus separated coloring. Table 9 summarizes our results,
 427 which we discuss below.

428 Remarkably, all of the 10k targets in the datasets could be designed using our algorithm,
 429 and thus admit a separable coloring. Moreover, roughly three-quarters (80%) of the targets
 430 were found to be 2-separable, and less than 1% of the targets required the consideration of
 431 values for m^\ominus beyond 6. The max value for m^\ominus in this dataset was 9, an order of magnitude
 432 lower than the sequence length. Clearly, since we have shown the existence of non-separable
 433 instances with isolated stacks and no isolated base pair, this observation does not generalize
 434 to arbitrary sequence lengths. However, the large size of these counterexamples suggests that
 435 the proportion of separable structures, despite ultimately decaying exponentially [20], may
 436 remain non-negligible for relevant RNA target sizes.

437 **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 [19]. 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 :

$$\Delta\Delta G(w, T) := \Delta G(w, \alpha_{d^-}(w, T)) - \Delta G(w, T), \alpha(w, T) := \min\{\Delta G(w, T') \mid |T', T| \geq d^-\}$$

438 where $\Delta G(w, T)$ is the Turner free-energy, $|T, T'| := |T \triangle T'|$ denotes the base-pair distance,
 439 and d^- represents the minimum base pair distance to T . Both ΔG and $\alpha_{d^-}(w, T)$ can be
 440 obtained by appropriate calls to the ViennaRNA package [8], namely RNAeval and RNAsubopts,
 441 using max energy distance parameter $E = 5$ (so our estimation of $\Delta\Delta G(w, T)$ is bounded by
 442 5). A positive energy distance confirms that w is a solution to the Turner version of inverse
 443 folding, and dominates its competitors by $\Delta\Delta G(w, T)$ kcal.mol⁻¹. Meanwhile, a negative
 444 energy distance indicates that the target T is dominated by some alternative structure,
 445 having $\Delta\Delta G(w, T)$ kcal.mol⁻¹ lower free-energy than the target.

446 We consider three strategies for sampling sequences: i) The *compatible* model uniformly
 447 generates random sequences compatible with the target (A for unpaired positions; AU, UA,
 448 GC or CG for base pairs); ii) The *separated* model uses the sampler described in Section 4.2
 449 to generate sequences that are 2-separated and proper; iii) The *relaxed* model generates
 450 sequences that are 2-separated, but not necessarily proper by assigning uniform random
 451 pairs to the base pairs of a multiloop. The *relaxed* model enables a heuristic extension of
 452 our algorithms supporting multiloops of arbitrary degrees, noting that the local refolding
 453 (see Figure 3) occurring in the BP model for non-proper sequences are either unrealistic or
 454 outright impossible, in the Turner energy model.

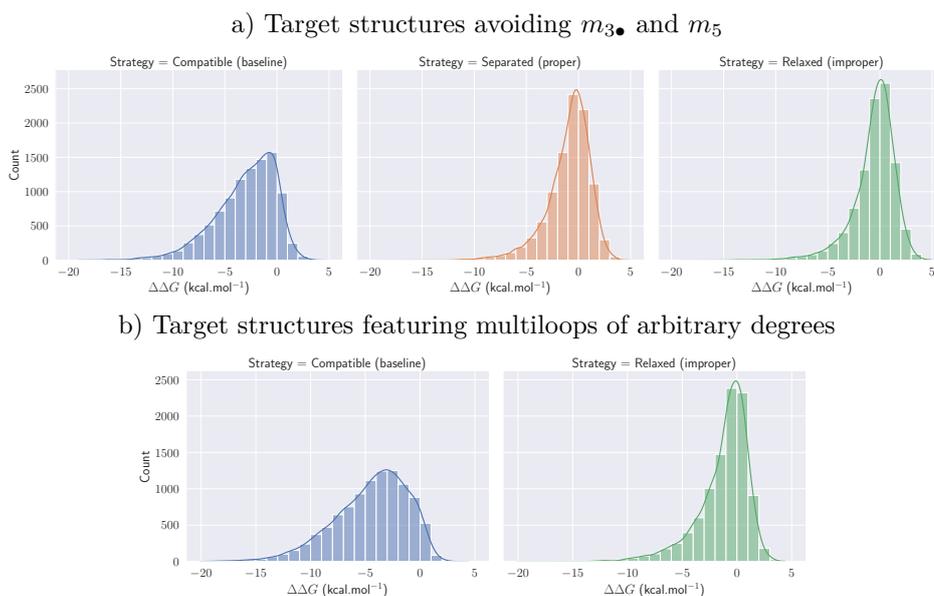
455 **Separated sequences substantially improve over compatible random sequences.**

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

462 The results, summarized in Figure 11.top suggest that separated sequences represent a
 463 substantial improvement over merely compatible sequences. Indeed, while 10% of compatible
 464 sequences ended up being good design candidates ($\Delta\Delta G > 0$), the proportion of successful
 465 designs increases to approximately one-third (35%) for separated sequences, and further to
 466 43% for relaxed design. A similar trend can be observed for the average $\Delta\Delta G$ (distance to
 467 the first alternative/competitor) among successful designs, being of 0.79/0.98/1.06 kcal.mol⁻¹
 468 in the compatible, separated and relaxed models respectively. The surprisingly good behavior
 469 of the relaxed model, which was mostly introduced to overcome unrealistic limitations on
 470 multiloops, remains to be explained.

471 **Relaxed sequences enable designs for multiloops having higher degrees.**

472 We also tested the capacity of the relaxed model to generate solutions for multiloops of higher degrees,
 473 noting that the avoidance of $m_{3\bullet}$ and m_5 restricts the maximum degree of a multiloop to
 474 4. We used the above-mentioned generation algorithm to generate uniform design targets
 475 of size $n = 100$, featuring at least one (but frequently many) occurrence of $m_{3\bullet}$ and m_5 .
 476 As shown in Figure 11.bottom, compatible sequences are again substantially outperformed
 477 by the relaxed separated model in this setting, with 31.5% of the separated/non-proper



■ **Figure 11 Comparison of compatible (baseline), separated, and relaxed models for targets having $\theta = 3$, $h_{\min} = 3$.** For energy distance parameters, we took $d^- = 3$ and $E = 5$.

478 sequences (as opposed to only 5.1% of compatible sequences) representing successful designs
 479 ($\Delta\Delta G > 0$), on average $0.86 \text{ kcal.mol}^{-1}$ more stable than their best competitor.

480 7 Conclusion

481 Adapting a coloring perspective initially introduced by Halès *et al* [6], we have shown that
 482 the inverse folding problem can be solved in linear time for all target secondary structures
 483 having minimum helix length equal to 3. Towards that main result, we have established the
 484 existence of designable, yet non-separable, instances of inverse folding, and the NP-hardness
 485 of finding a separable design in the initial sense of Halès *et al*. We have also introduced
 486 concrete algorithms for the problem of finding a m modulo-separated coloring, which we
 487 have shown to be NP-hard yet FPT-solvable for m . Already for $m = 2$, the scope of our
 488 algorithms encompasses all targets without isolated base pairs and stacks, but also extends
 489 much beyond, in a way that remains to be fully characterized. Beyond base pair maximization,
 490 modulo-separated sequences may also represent a solid foundation towards concrete design
 491 methodologies. Namely, we empirically showed that, for the Turner energy model, separated
 492 sequences tend to represent better design candidates than merely compatible sequences,
 493 and that the limitations on loop degrees (intrinsic to the BP maximization model) can be
 494 overcome by relaxing our design model while retaining substantial performances.

495 Future work should focus on how much of designable sequences are covered by sequences
 496 obtained with (modulo)-separated colorings. More importantly, does the space of (modulo)-
 497 separated colorings always/often contain a design with respect to the nearest-neighborhood
 498 Turner energy model? Even if it unlikely to hold unconditionally, it is plausible that some
 499 extensions of separability and m -separability will achieve theoretical and practical solutions
 500 for inverse folding in more general energy models. As a first step, separability in a stacking
 501 energy model seems a relevant goal, even if less ambitious than the Turner model. It would
 502 probably require to go beyond the current coloring formalism, and motivate the introduction
 503 of more general notions of defect to capture imbalance at the dinucleotide level.

504 — References —

- 505 1 Mirela Andronescu, Anthony P. Fejes, Frank Hutter, Holger H. Hoos, and Anne Condon. A new
506 algorithm for rna secondary structure design. *Journal of Molecular Biology*, 336(3):607–624,
507 2004. URL: <https://www.sciencedirect.com/science/article/pii/S0022283603015596>,
508 [doi:10.1016/j.jmb.2003.12.041](https://doi.org/10.1016/j.jmb.2003.12.041).
- 509 2 Édouard Bonnet, Paweł Rzażewski, and Florian Sikora. Designing rna secondary structures is
510 hard. *Journal of Computational Biology*, 27(3):302–316, 2020. PMID:32160034. [arXiv:https://arxiv.org/abs/2009.0420](https://arxiv.org/abs/2009.0420),
511 [doi:10.1089/cmb.2019.0420](https://doi.org/10.1089/cmb.2019.0420), [doi:10.1089/cmb.2019.0420](https://doi.org/10.1089/cmb.2019.0420).
- 512 3 Anke Busch and Rolf Backofen. INFO-RNA—a fast approach to inverse RNA folding. *Bioin-*
513 *formatics*, 22(15):1823–31, 2006.
- 514 4 Ali Esmaili-Taheri and Mohammad Ganjtabesh. ERD: a fast and reliable tool for RNA design
515 including constraints. *BMC Bioinform.*, 16:20:1–20:11, 2015.
- 516 5 Juan Antonio Garcia-Martin, Ivan Dotu, and Peter Clote. RNAiFold 2.0: a web server
517 and software to design custom and Rfam-based RNA molecules. *Nucleic Acids Research*,
518 43(W1):W513–W521, 05 2015. [arXiv:https://academic.oup.com/nar/article-pdf/43/W1/](https://academic.oup.com/nar/article-pdf/43/W1/W513/7476300/gkv460.pdf)
519 [W513/7476300/gkv460.pdf](https://academic.oup.com/nar/article-pdf/43/W1/W513/7476300/gkv460.pdf), [doi:10.1093/nar/gkv460](https://doi.org/10.1093/nar/gkv460).
- 520 6 Jozef Hales, Alice Héliou, Ján Manuch, Yann Ponty, and Ladislav Stacho. Combinatorial RNA
521 design: Designability and structure-approximating algorithm in watson-crick and nussinov-
522 jacobson energy models. *Algorithmica*, 79(3):835–856, 2017.
- 523 7 Stefan Hammer, Wei Wang, Sebastian Will, and Yann Ponty. Fixed-parameter tractable
524 sampling for RNA design with multiple target structures. *BMC bioinformatics*, 20:209, April
525 2019. [doi:10.1186/s12859-019-2784-7](https://doi.org/10.1186/s12859-019-2784-7).
- 526 8 Ivo L Hofacker, Walter Fontana, Peter F Stadler, L Sebastian Bonhoeffer, Manfred Tacker,
527 and Peter Schuster. Fast folding and comparison of RNA secondary structures. *Monatshefte*
528 *für Chemie/Chemical Monthly*, 125(2):167–188, 1994.
- 529 9 Robert Kleinkauf, Martin Mann, and Rolf Backofen. antaRNA: ant colony-based RNA sequence
530 design. *Bioinformatics*, 31(19):3114–3121, 05 2015. [doi:10.1093/bioinformatics/btv319](https://doi.org/10.1093/bioinformatics/btv319).
- 531 10 William Andrew Lorenz and Yann Ponty. Non-redundant random generation algorithms for
532 weighted context-free grammars. *Theoretical Computer Science*, 502:177–194, 2013. Generation
533 of Combinatorial Structures. URL: [https://www.sciencedirect.com/science/article/pii/](https://www.sciencedirect.com/science/article/pii/S0304397513000443)
534 [S0304397513000443](https://www.sciencedirect.com/science/article/pii/S0304397513000443), [doi:10.1016/j.tcs.2013.01.006](https://doi.org/10.1016/j.tcs.2013.01.006).
- 535 11 Rune B. Lyngsø, James W. J. Anderson, Elena Sizikova, Amarendra Badugu, Tomas Hyland,
536 and Jotun Hein. Frnakenstein: multiple target inverse RNA folding. *BMC Bioinform.*, 13:260,
537 2012.
- 538 12 Nono S. C. Merleau and Matteo Smerlak. arnaque: an evolutionary algorithm for inverse
539 pseudoknotted RNA folding inspired by lévy flights. *BMC Bioinform.*, 23(1):335, 2022.
- 540 13 R Nussinov and A B Jacobson. Fast algorithm for predicting the secondary structure of
541 single-stranded rna. *Proceedings of the National Academy of Sciences*, 77(11):6309–6313,
542 1980. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.77.11.6309>, [arXiv:https://arxiv.org/abs/1908.09211](https://arxiv.org/abs/1908.09211),
543 www.pnas.org/doi/pdf/10.1073/pnas.77.11.6309, [doi:10.1073/pnas.77.11.6309](https://doi.org/10.1073/pnas.77.11.6309).
- 544 14 Yann Ponty. *Ensemble Algorithms and Analytic Combinatorics in RNA Bioinformatics and*
545 *Beyond*. Habilitation à diriger des recherches, Université Paris-Saclay, May 2020. URL:
546 <https://theses.hal.science/tel-03219977>.
- 547 15 Vladimir Reinharz, Yann Ponty, and Jérôme Waldispühl. A weighted sampling algorithm
548 for the design of RNA sequences with targeted secondary structure and nucleotide dis-
549 tribution. *Bioinformatics*, 29(13):i308–i315, 06 2013. [arXiv:https://arxiv.org/abs/1308.5070](https://arxiv.org/abs/1308.5070),
550 [https://academic.oup.com/](https://academic.oup.com/bioinformatics/article-pdf/29/13/i308/50704314/bioinformatics_29_13_i308.pdf)
551 [bioinformatics/article-pdf/29/13/i308/50704314/bioinformatics_29_13_i308.pdf](https://academic.oup.com/bioinformatics/article-pdf/29/13/i308/50704314/bioinformatics_29_13_i308.pdf),
[doi:10.1093/bioinformatics/btt217](https://doi.org/10.1093/bioinformatics/btt217).
- 552 16 Matan Drory Retwitzer, Vladimir Reinharz, Alexander Churkin, Yann Ponty, Jérôme
553 Waldispühl, and Danny Barash. incaRNAfbinv 2.0: a webserver and software with
554 motif control for fragment-based design of RNAs. *Bioinformatics*, 36(9):2920–2922,

- 555 01 2020. [arXiv:https://academic.oup.com/bioinformatics/article-pdf/36/9/2920/](https://academic.oup.com/bioinformatics/article-pdf/36/9/2920/48986446/bioinformatics_36_9_2920.pdf)
556 [48986446/bioinformatics_36_9_2920.pdf](https://academic.oup.com/bioinformatics/article-pdf/36/9/2920/48986446/bioinformatics_36_9_2920.pdf), doi:10.1093/bioinformatics/btaa039.
- 557 17 Frederic Runge, Danny Stoll, Stefan Falkner, and Frank Hutter. Learning to design RNA. In
558 *Proceedings of ICLR 2019*, 2019.
- 559 18 Michael Schnall-Levin, Leonid Chindelevitch, and Bonnie Berger. Inverting the viterbi
560 algorithm: an abstract framework for structure design. In *ICML*, volume 307 of *ACM*
561 *International Conference Proceeding Series*, pages 904–911. ACM, 2008.
- 562 19 Douglas H. Turner and David H. Mathews. NNDB: the nearest neighbor parameter data-
563 base for predicting stability of nucleic acid secondary structure. *Nucleic Acids Research*,
564 38(suppl_1):D280–D282, 10 2009. [arXiv:https://academic.oup.com/nar/article-pdf/38/](https://academic.oup.com/nar/article-pdf/38/suppl_1/D280/11217894/gkp892.pdf)
565 [suppl_1/D280/11217894/gkp892.pdf](https://academic.oup.com/nar/article-pdf/38/suppl_1/D280/11217894/gkp892.pdf), doi:10.1093/nar/gkp892.
- 566 20 Hua-Ting Yao, Cedric Chauve, Mireille Regnier, and Yann Ponty. Exponentially few RNA
567 structures are designable. In *ACM-BCB 2019 - 10th ACM Conference on Bioinformatics,*
568 *Computational Biology, and Health Informatics*, pages 289–298, Niagara-Falls, United States,
569 September 2019. ACM Press. URL: <https://inria.hal.science/hal-02141853>, doi:10.
570 [1145/3307339.3342163](https://inria.hal.science/hal-02141853).
- 571 21 Hua-Ting Yao, Jérôme Waldispühl, Yann Ponty, and Sebastian Will. Taming Disruptive Base
572 Pairs to Reconcile Positive and Negative Structural Design of RNA. In *Proc. of the 25th*
573 *Annual International Conferences on Computational Molecular Biology (RECOMB'21)*, 2021.
574 URL: <https://inria.hal.science/hal-02987566>.
- 575 22 Joseph N. Zadeh, Brian R. Wolfe, and Niles A. Pierce. Nucleic acid sequence design via
576 efficient ensemble defect optimization. *Journal of Computational Chemistry*, 32(3):439–452,
577 2011. doi:10.1002/jcc.21633.

578 **A NP-completeness of general separability (Proof of Theorem 3)**

579 SEPARABILITY is clearly in NP, since any coloring (certificate) can be checked in linear
 580 time. We prove hardness by reduction from BIN PACKING which we formulate as an interval
 581 packing problem.

582 ► **Problem 5.** INTERVAL PACKING

583 **Input:** set of pairwise distinct integers $A = \{a_1, \dots, a_n\}$, integers k and B

584 **Output:** function x from A to intervals of $[0, kB - 1[$ such that:

- 585 ■ $x(a_i)$ is an interval of size a_i
- 586 ■ $x(a_i)$ and $x(a_j)$ are disjoint for $i \neq j$
- 587 ■ $x(a_i)$ does not contain both $jB - 1$ and jB for any i, j .

588 This is a reformulation of BIN PACKING: fitting items for a total size of B is equivalent to
 589 finding a partition of a size- B interval into smaller intervals. The problem remains NP-hard
 590 even when input integers are encoded in unary (which corresponds to the fact that BIN
 591 PACKING is strongly NP-hard). We further require that all items have size $a_i \geq 5$

592 **Object and border gadgets.** We first give the main gadgets for our reduction, see figure 12
 593 for more details.

594 ► **Definition 6.** An object gadget of size $q \geq 3$ is a chain of $q + 3$ nodes c_0, \dots, c_{q+2} with a
 595 child attached to c_1 and c_{q+1} and leaves attached to all other nodes c_i .

596 A period- p border gadget of size q is a chain of q nodes c_0, \dots, c_{q-1} with a child attached
 597 to c_i for all $i \equiv 0 \pmod p$ and leaves attached to all other nodes c_i .

598 ► **Proposition 11.** If an object gadget of size q appears in a tree with a separated coloring χ ,
 599 with $\ell = \min\{L(c_i) \mid 1 \leq i \leq q\}$ such that

- 600 ■ there are ● nodes at levels $\ell + 2$ and $\ell + (q + 2)$
- 601 ■ there are leaves at levels $\ell + i$ for all $1 \leq i \leq q + 3$, $i \neq 2, q + 2$.

602 If a period- p gadget of size q appears in a tree with a separated coloring χ , with the root
 603 at level ℓ , then there exists some direction $d \in \{-1, 1\}$ such that

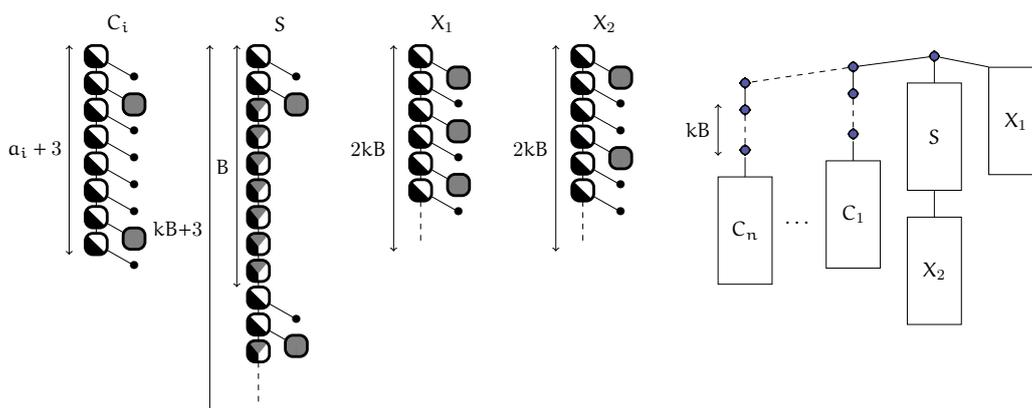
- 604 ■ there are ● nodes at levels $\ell + d \cdot i + 1$ for all $1 \leq i \leq q$, $i \equiv 0 \pmod p$;
- 605 ■ there are leaves at levels $\ell + d \cdot i$ for all $1 \leq i \leq q + 3$, $i \not\equiv 0 \pmod p$.

606 **Proof.** First note that in either gadget, all nodes c_i have the same non-● color. Indeed,
 607 nodes with a leaf attached or a leaf sibling cannot be ●, so all c_i are ● or ○. Furthermore,
 608 by the proper coloring constraints, consecutive nodes must be of the same color, so all c_i
 609 have the same color. Thus, writing ℓ_r for the root level, we have that the level below each
 610 node c_i is $\ell_r + di$, with $d = 1$ if the whole chain is ● and $d = -1$ otherwise.

611 Furthermore, all nodes attached to the chain must be ● by the proper coloring constraints.
 612 This directly gives the desired property for border gadgets. For object gadgets, the minimum
 613 level ℓ along the chain is either ℓ_r (if $d = 1$) or $\ell_r - q - 3$ (if $d = -1$), and in both cases, for
 614 each level $\ell + i$ with $1 \leq i \leq q + 3$, there is either a ● node ($i = 2$ or $i = q + 2$) or a leaf
 615 (otherwise). ◀

616 **Reduction.** Given an instance A, k, B of INTERVAL PACKING, we build a tree T as follows:

- 617 ■ We start with a chain P of $n + 1$ nodes denoted p_0, \dots, p_n .
- 618 ■ For each $i \geq 1$ we attach a chain (denoted P_i) of Bk nodes to p_i , and an object gadget
 619 C_i of size a_i to the end of the chain.
- 620 ■ We attach a period-2 border gadget of size $2kB$ to p_0 , denoted X_1 .



■ **Figure 12** Left: details of the four main parts of the reduction, i.e. an object gadget C_i of size a_i (in this example with $a_i = 5$), border gadgets X_1 and X_2 with respective periods 2 and 3, and the separator chain S). Right: general layout of the tree built in the reduction.

- 621 ■ We attach a chain S of $kB + 3$ nodes to p_0 with:
- 622 ■ a leaf to the $(iB + 1)$ st node of S for each $0 \leq i \leq k$,
 - 623 ■ a second child, called *separator*, to the $(iB + 2)$ nd node of S for each $0 \leq i \leq k$,
 - 624 ■ a period-3 border gadget of size $2kB$ at the end of S , denoted X_2 .

625 We will now show that there exists a solution for unary bin packing if and only one can find
626 a separated coloring for T .

627 From interval packing to separated coloring.

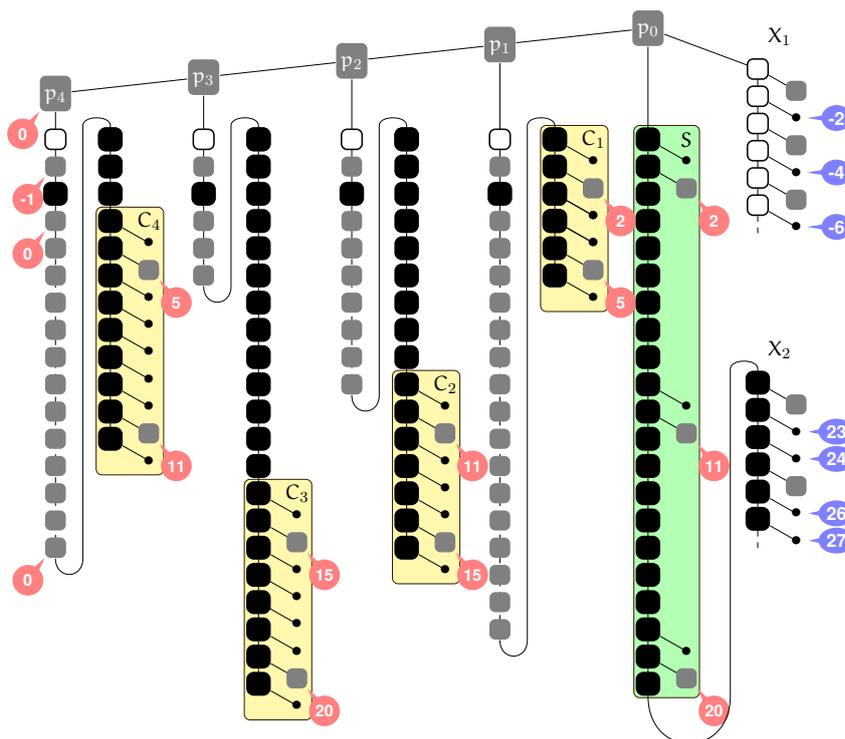
628 In this section, we consider an interval packing x assigning an interval of $[0, kB - 1[$ to
629 each item a_i . We write x_i such that $x(a_i) = [x_i, x_i + a_i - 1[$, and we color the tree T as
630 follows (see Figure 13):

- 631 ■ All nodes c_i in object gadgets, all non-separator nodes in S and all nodes c_i in X_1 are
632 colored ●,
- 633 ■ All nodes c_i in X_2 are colored ○.
- 634 ■ The first three nodes of P_i are colored ○ ● ●, and the last x_i nodes of P_i are colored ●
635 (note that P_i has length $kB \geq x_i + 3$ since $x_i + a_i < kB$ and $a_i \geq 5$).
- 636 ■ All remaining nodes are colored ●.

637 We show that this coloring is separated, in particular, we show that the level of each ●
638 node is of one of the following types, and that leaves are *not* of these types:

- 639 a. 0, 2 and $kB + 2$
- 640 b. $x_i + 2$ for each $1 \leq i \leq n$
- 641 c. $j - 1$ for $j \leq 0$, $j \equiv 0 \pmod{2}$
- 642 d. $kB + j + 4$ for $j \geq 0$, $j \equiv 0 \pmod{3}$

643 For the chain P , all nodes are ● and have level 0 (type a). For each P_i , there are ● nodes
644 at levels -1 and 0 (types a and c), and the chain ends at level x_i . For each object gadget
645 C_i , there are ● nodes at levels $x_i + 2$ (type b), and $x_i + a_i + 2$ (type b or a, since this
646 corresponds to the start of the next interval or to $kB + 2$). There are also leaves in C_i at
647 each level $x_i + j$ for $j = 1, 3, 4, \dots, a_i, a_i + 1, a_i + 3$ which are all values between 1 and $kB + 3$
648 and indeed do not correspond to any of the four types above. For gadget X_1 , there are ●
649 nodes at odd levels from -1 down to $-2kB + 1$ (type c), and leaves at even negative levels.
650 For the chain S , there are ● nodes attached at levels $iB + 2$ for each $0 \leq i \leq k$, which are
651 necessarily of the form $x_i + 2$ (type b) for some i (since each iB must be the start of some



■ **Figure 13** Example of the reduction with $n = 4$ items with sizes $\{3, 4, 5, 6\}$ to be sorted into $k = 2$ size-9 bins. A separated coloring is shown, corresponding to the solution $\{3, 6\}, \{4, 5\}$ (a selection of leaf and \bullet levels are depicted). Each item is mapped into a branch P_i followed by an object gadget C_i , containing 2 \bullet nodes separated by the size of the item. Leaves in object gadget enforce that any two gadgets may overlap only if the \bullet nodes are aligned. The bins are implemented using the separator sequence S , with \bullet nodes at every B th position, enforcing that series of consecutive items are packed into size- B bins. Finally, border gadgets X_1 and X_2 may not overlap with any other gadget, and enforce that all object gadgets and separators are packed together in a size- kB range of levels.

652 interval of x). Leaves in S are at level 1 and $kB + 1$, which are not of any type (in particular
 653 for type **b**, this is true since $a_i \geq 5$). Finally, for gadget X_2 , the \bullet nodes are of type **d**, and
 654 the leaves occupy remaining levels beyond $kB + 4$.

655 **From separated coloring to interval packing**

656 Suppose now that T admits a separated coloring χ , and consider the gadget X_1 . Its root is
 657 at level $\ell_{X_1} \in \{-1, 0, 1\}$, and by Proposition 11, there exists some $d_{X_1} \in \{-1, 1\}$ such that,
 658 for each level $\ell_{X_1} + d_{X_1}j$, there is a leaf (for even j) or a \bullet node (odd j). Without loss
 659 of generality, we assume that $d_{X_1} = -1$ (i.e., the chain in X_1 is \circ): if this is not the case
 660 we swap \circ and \bullet colors overall. Thus, there are leaves and \bullet nodes at alternating levels
 661 between -2 and $-2kB + 1$ (at least).

662 Consider the chain S . For any $0 \leq i \leq k$, the $(iB + 2)$ nd node of the chain cannot be \bullet
 663 (since it has a leaf sibling) so one of its two children must be \bullet . We write $s_0 \leq s_1 \leq \dots \leq s_k$
 664 for the levels of such \bullet nodes in ascending order: from the position of the nodes we have
 665 $s_{j+1} \leq s_j + B$. Furthermore, $s_0 \leq 3$ and $s_k \leq kB + 3$ (using the distances to the root).

666 Consider now X_2 . Its root is at most one level away from a separator, so at level ℓ_{X_2}

667 with $s_0 - 1 \leq \ell_{X_2} \leq s_k + 1$. By Proposition 11, there exists some $d_{X_2} \in \{-1, 1\}$ such that,
 668 for each level $\ell_{X_2} + d_{X_2}j$ with $1 \leq j \leq 2kB$, there is a \bullet node ($i \equiv 0 \pmod{3}$) or a leaf
 669 (otherwise). In particular, we necessarily have $d_{X_2} = 1$, since otherwise there would be two
 670 consecutive \bullet levels among levels $\{-2, -3, -4\}$, which would raise a conflict with X_1 .

671 For any $i \in [1, n]$, consider object gadget C_i . Its minimum level is ℓ_i with $-kB - n - a_i - 3 \leq$
 672 $\ell_i \leq kB + a_i + n + 3$, and by Proposition 11, for each level $\ell_i + j$ with $1 \leq j \leq a_i + 3$, there
 673 is a \bullet node ($j = 2, a_i + 2$) or a leaf (otherwise). In particular, $\ell_i \geq s_0 - 5$ (as otherwise
 674 there would be consecutive leaves at consecutive levels under $s_0 - 2$, in conflict with X_1)
 675 and $\ell_i + a_j \leq s_k + 5$ (otherwise there would be leaves at consecutive levels higher than
 676 $s_k + 3$, in conflict with X_2). Finally, since levels s_0 and s_k have \bullet nodes and $a_i \geq 5$, then
 677 for i such that $\ell_i \leq s_0 - 2$, we have $\ell_i = s_0 - 2$. Similarly, for i such that $\ell_i + a_i + 2 \geq s_k$,
 678 we have $\ell_i + a_i + 2 = s_k$. And for any i and j , if $\ell_i + 2 \leq s_j \leq \ell_i + a_i + 2$, we have
 679 $s_j \in \{\ell_i + 2, \ell_i + a_i + 2\}$.

680 Pick any two object gadgets $C_i, C_{i'}$ with $\ell_i \leq \ell_{i'}$. Then $\ell_i \neq \ell_{i'}$ (otherwise, since $a_i \neq a_{i'}$,
 681 there would be a conflict at level $\ell_i + \min\{a_i, a_{i'}\} + 2$), and $\ell_{i'} \geq \ell_i + a_i$ (otherwise, there
 682 would be a conflict at level $\ell_{i'} + 2$).

683 We now have all the tools to build an interval packing. We write $x_i = \ell_i - s_0 + 2$ and
 684 $\sigma_j = s_j - s_0$. By the remarks above, we have that intervals $[x_i, x_i + a_i - 1[$ are pairwise
 685 disjoint. Furthermore, they are all included in interval $[0, \sigma_k - 1[$. Since they have total
 686 size $\sum_{i=1}^n a_i = kB$ and $\sigma_k = s_k - s_0 \leq kB$, we have $\sigma_k = kB$, which is only possible with a
 687 fully \bullet chain S : so we get $\sigma_j = jB$ for all $0 \leq j \leq k$. And finally, if $\sigma_j \in [x_i, x_i + a_i - 1[$,
 688 then $\ell_i + 2 \leq s_j \leq \ell_i + a_i + 2$ which yields $s_j \in \{\ell_i + 2, \ell_i + a_i + 2\}$. This translates into
 689 $\sigma_j \in \{x_i, x_i + a_i\}$, so necessarily $\sigma_j = x_i$ and $\sigma_j - 1 \notin [x_i, x_i + a_i[$. Overall gadget levels
 690 relative to the first separator s_0 give a valid partition of $[0, kB - 1[$ into pairwise disjoint
 691 size- a_i intervals non-overlapping block border positions jB , so they give a valid INTERVAL
 692 PACKING solution.

693 **B** Non-separable target w/o isolated BPs (Proof of Proposition 10)

694 We start with the following remark:

695 **► Proposition 2.** *If u_0, \dots, u_k is a path in T and each u_i for even i has a leaf attached to it*
 696 *then, for any coloring χ of the path, we have $\chi(u_0) \in \{\bullet, \circ\}$ and $\chi(u_i) = \chi(u_0)$ for all i .*

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

701 We now build a non-separable instance I without size-1 helix nor $(m_{3\bullet}, m_5)$ motif. Let
 702 $a \geq 2$ and $b \geq 2$ be even numbers. Let $T(a, b)$ be the gadget from Fig 8, containing a length- a
 703 path from the root to an internal node denoted t , and three length- b branches attached to t .
 704 Further attach a leaf to every node at an even distance from the root (except t itself). Note
 705 that all helices in $T(a, b)$ have length 2. The *level* of a copy of some $T(a, b)$ gadget is the
 706 level reached under node t of this gadget.

707 We build the instance I as a tree containing 5 copies of the gadget $T(a, b)$, precisely
 708 $I = (((T[10, 100], T[20, 100])), ((T[30, 100], T[40, 100])), T[50, 100])$.

709 First note that for a copy of gadget $T(a, b)$ at level ℓ in any separable coloring, there is
 710 a \bullet node at level ℓ , since the node t has three children and at least one must be \bullet . Also,
 711 there exist two integers u, v such that, for every $x \in [1, b]$, there is a leaf at level $\ell + ux$ if x

XXX:22 Exact linear-time RNA design for min Helix length 3

712 is odd, and level $\ell + vx$ if x is even. Indeed, pick one gray child U of t , and one non-gray
 713 child V . All vertices under U form an all-white or all-black branch by Proposition 2 (we let
 714 respectively $u = -1$ and $u = 1$), and vertices at levels $\ell + u, \ell + 3u, \dots, \ell + bu$ (or $\ell + (b-1)u$)
 715 have a pending leaf. We similarly define $v = 1$ if V is black and $v = -1$ if V is white, and
 716 vertices at levels $\ell + 2v, \ell + 4v, \dots, \ell + bv$ (or $\ell + (b-1)v$) have a pending leaf. From the
 717 above, if there are \bullet nodes at levels ℓ_1 and ℓ_2 with $\ell - b \leq \ell_1 < \ell < \ell_2 \leq \ell + b$, then $\ell_1 \not\equiv \ell_2$
 718 mod 2 (since otherwise, one of ℓ_1, ℓ_2 could be written as $\ell + ux$ with even x , so that level
 719 would be a leaf level).

720 Aiming at a contradiction, assume that I admits a separable coloring. Let $\ell_1 \leq \ell_2 \leq$
 721 $\ell_3 \leq \ell_4 \leq \ell_5$ be the levels of all five copies of the $T[a, b]$ gadgets of I , in ascending order.
 722 Then from the length of the branches from the root, we have $\ell_i \in [-50, 50]$ and $\ell_i \neq \ell_j$.
 723 Then by the remark above applied to the gadget with level ℓ_2 , we have $\ell_1 \not\equiv \ell_3 \pmod{2}$, and
 724 similarly using gadgets with level ℓ_4 we have $\ell_3 \not\equiv \ell_5 \pmod{2}$ and $\ell_1 \not\equiv \ell_5 \pmod{2}$, leading
 725 to a contradiction (any three integers such as ℓ_1, ℓ_3 and ℓ_5 may not have pairwise distinct
 726 parities).

727 **C** Leveraging random generators at fixed modular levels into a 728 uniform random generation of separated sequences

729 **► Theorem 12.** UNIFORM MODULO SEPARATED GENERATION *can be performed in an*
 730 *average-case complexity that is Fixed Parameter Tractable for the modulus parameter m .*

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

- 734 1. Choose some $\xi_L \subseteq [0, m[$ with probability $\mathbb{P}(\xi_L) = \#\text{Designs}_{\xi_L} / \mathcal{Z}_m$
- 735 2. Generate a ξ_L separated sequence w
- 736 3. Compute the number Ξ_w of $\xi'_L \subseteq [0, m[$ such that w is ξ'_L separated
- 737 4. Accept/return w with probability $1/\Xi_w$; 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 [14, pp 77]), and we have:

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

738 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
 739 that $p_w \propto 1/\mathcal{Z}_m$, a term that no longer depends on w , from which we conclude that the
 740 generation is uniform.

741 Complexity-wise, a prior accumulation of the 2^m terms $\#\text{Designs}_{\xi_L}$, each smaller than
 742 4^m , into a suitable data structure (see Lorenz and Ponty [10] for details) enables a random
 743 choice of ξ_L (Step 1.) in $\Theta(n.m)$. Once ξ_L is chosen, the above DP algorithm uniformly
 744 generates w in time $\Theta(m.n)$ (Step 2). The computation of Ξ_w (Step 3) is trivial and consists
 745 in identifying, in time $\Theta(n+m)$, the subset $\Phi_w \subseteq [0, m[$ of modular levels that are populated
 746 by neither leaves nor \bullet nodes in χ_w . Indeed, those levels represent the only degrees of
 747 freedom available while choosing a compatible ξ_L , the others modular values being forced
 748 to either \bullet or leaves. Since such modular values can be independently chosen to be in
 749 or out of ξ_L , then we have $\Xi_w = 2^{|\Phi_w|}$. Clearly, we have $\Xi_w \leq 2^m$, so the expectation
 750 of the number of (independent) rejections admits an upper bound in 2^m , and the overall
 751 average-case complexity is in $\Theta(n.m.2^m)$.