



**HAL**  
open science

# Automatic exploration of the natural variability of RNA non-canonical geometric patterns with a parameterized sampling technique

Théo Boury, Yann Ponty, Vladimir Reinharz

► **To cite this version:**

Théo Boury, Yann Ponty, Vladimir Reinharz. Automatic exploration of the natural variability of RNA non-canonical geometric patterns with a parameterized sampling technique. WABI 2023 - 23rd Workshop on Algorithms in Bioinformatics, Texas A&M University, Sep 2023, Houston, United States. hal-04094288v1

**HAL Id: hal-04094288**

**<https://hal.science/hal-04094288v1>**

Submitted on 10 May 2023 (v1), last revised 24 Aug 2023 (v2)

**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.

# Automatic exploration of the natural variability of RNA non-canonical geometric patterns with a parameterized sampling technique

**Théo Boury**

Computer Science Department, Ecole Normale Supérieure de Lyon, France

**Yann Ponty**

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

**Vladimir Reinharz**

Department of Computer Science, Université du Québec à Montréal, Canada

---

## Abstract

**Motivation:** Recurrent substructures in RNA, known as 3D motifs, consist of networks of base pair interactions and are critical to understanding the relationship between structure and function. Their structure is naturally expressed as a graph which has led to many graph-based algorithms to automatically catalog identical motifs found in 3D structures. Yet, due to the complexity of the problem, state-of-the-art methods are often optimized to find exact matches, limiting the search to a subset of potential solutions, or do not allow explicit control over the desired variability.

**Results:** We developed FuzzTree, a method able to efficiently sample subgraphs in an RNA structure that lie in a close neighborhood of a requested motif. It is the first method that allows explicit control over (1) the admissible geometric variability in the interactions, (2) the number of missing edges, and (3) introduction of discontinuities in the backbone given close distances in the 3D structure. Our tool relies on a multidimensional Boltzmann sampling procedure with complexity parameterized by the treewidth of the requested motif. We applied our method to the well-known internal loop Kink-Turn motif, which can be divided into 12 subgroups. Given only the graph representing the main Kink-Turn subgroup, FuzzTree retrieved over 3/4 of all kink-turns. We also highlight two occurrences of new sampled patterns. Our tool is available as free software and can be customized for different parameters and types of graphs.

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

**Keywords and phrases** Subgraph Isomorphism, 3D RNA, Parameterized Complexity, Tree Decomposition, Boltzmann sampling, Neighborhood metrics, Kink-Turn family

**Digital Object Identifier** [10.4230/LIPIcs.WABI.2023.?](https://doi.org/10.4230/LIPIcs.WABI.2023.?.)

**Related Version** full version hosted on arXiv

**Supplementary Material** The source code for the tool and the tests are on [GitHub:FuzzTree](https://github.com/FuzzTree). The RNA structures encoded as python pickle graphs are available at [doi.org/10.5683/SP3/ZR29QE](https://doi.org/10.5683/SP3/ZR29QE)

## 1 Introduction

RNAs' essential regulatory and catalytic roles in cellular processes can largely be attributed to the intriguing and highly versatile nature of their structures [8, 5]. The structure of ncRNAs is inherently modular, with distinct structural domains (loops) divided by stems of rigid canonical bonds, often responsible for their unique functions [19]. This modular architecture has been used for advancements in structure prediction [10] and rational design [11]. Consequently, the characterization of ncRNA structure and identification of structural modules have become critical in the pursuit of understanding their diverse functions and exploiting them for future applications.



© Théo Boury, Yann Ponty and Vladimir Reinharz; licensed under Creative Commons License CC-BY

Workshop on Algorithms in Bioinformatics (WABI) 2023.

Editors: John Q. Open and Joan R. Access; Article No. ?; pp. ?:1-?:25

[Leibniz International Proceedings in Informatics](https://www.lipics.de/)



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Many approaches have been developed to detect and classify conserved modules. These classifications differ in the scale adopted to detect and define a motif: RNA3DMotifsAtlas [25] computes similarity and finds motifs at the atomic level. A drawback of such a method is the computation time, which restrains comparisons between loops, but the granularity allows capturing a certain level of diversity by omitting bulging-out nucleotides. RNA Bricks [6] and RAG3D [32] abstract loops and hairpins as unitary elements. At an intermediate layer, CaRNAval [26, 28] models RNA as graphs where vertices are nucleotides, and edges are the sequence backbone phosphodiester bonds or non-covalent interactions that can be classified following the Leontis-Westhof (LW) annotations in 12 different geometric families [20, 29]. Such an approach allows specific graph algorithms to discover much larger and complex modules than by doing atomic computations while retrieving the known structural modules, but will not be able to identify natural variations since it relies on detecting exact matches.

From the algorithmic point of view the treewidth  $tw$  is a natural parameter. In 1995, Alon *et al* [1] proposed an XP algorithm in  $O(2^{|V_P|} n^{tw(G_P)+1})$  using the color-coding technique. It was shown more generally that only very specific constraints on the input allow us to have algorithms tractable for bounded treewidths [22]. The problem is not fixed-parameter tractable when parameterized only by the treewidth, and it requires other parameters to become tractable. For instance, some approaches are parameterized both by  $tw(G_P)$  and  $|G_P|$ , and conversely, others are parameterized by  $tw(G_T)$  and the maximum degree of  $G_T$  [22].

However there can be an exponential number of variants of a specific pattern so different specialized algorithms allowing missing nodes and edges [24, 12], or requiring only labels to be in a neighborhood [17], have been developed. Such simplifications forget about the precise locations of interactions, which is information that we would like to preserve with RNA structures. A recent approach specific to RNA graph fuzziness uses Relational Graph Convolutional Network to embed the graphs in a vector space, allowing fast computation [23]. Their embedding is based on the nature of base pairs or their isostericity without taking into account gaps or missing edges. In addition, due to the nature of the method, there is no explicit control on the sampled neighborhoods, and thresholds need to be calibrated depending on the context.

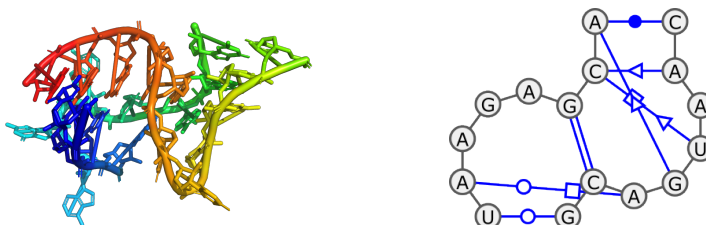
In this paper, we introduce FuzzTree, a multidimensional Boltzmann graph sampling procedure able to sample variants of a motif in a known RNA structure. We allow weighting and control of three key geometric features in the variants: (1) the geometric disruption of mismatched edges, (2) missing edges still constrained by their distance in the 3D structure, and (3) breaks in the backbone also constrained by their distance in the 3D structure. We propose a parameterized bound on the complexity of the algorithm based on the treewidth of the searched motif. We evaluate our method on the well-known interior loop Kink-Turn motif [18] characterized by its sharp bend and clustered into 12 different groups in the RNA3DMotifsAtlas [25]. We show that, from the signature representation of the main subgroup, we sample in 88% of RNAs all their known Kink-Turns. We also retrieve two previously un-annotated loops with a characteristic sharp bend.

## 2 Method

### 2.1 RNA as a graph and problem statement

We define an RNA structure as a graph  $G$  such that its nucleotides are encoded as vertices  $V$ , and the interactions between the nucleotides are encoded as directed edges  $E \subset V \times V$ , with labels  $L(e)$ . Interactions may represent backbone connectivity (phosphodiester bonds),

90 or any of the 12 base-pair types defined by the Leontis-Westhof (LW) nomenclature [29].  
 91 Each type specifies an interacting face (Watson-Crick  $\circ$ , Hoogsteen  $\square$ , Sugar  $\triangleright$ ) for both  
 92 nucleotides, along with an orientation cis (filled) or trans (empty). Note that the geometry  
 93 of the RNA structure is encoded in the edge labels, and our representation does not depend  
 94 on the sequence. We show in Fig. 1 a Kink-Turn motif, represented as a graph with labeled  
 95 edges.



■ **Figure 1 Kink-turn structure** On the left the 3D structure of a Kink-Turn in PDB 3RW6. On the right its representation as a graph of its base pair interactions, the backbone connections are represented as black arrowed edges.

96 We rewrite  $E$ , the set of edges as  $E = B \sqcup \bar{B}$ , composed of two distinct sets:  $B$ , the set  
 97 of edges that are backbone interactions and  $\bar{B}$ , the edges for LW interactions.

98 The classic Subgraph Isomorphism Problem (SIP) identifies all occurrences of a pattern  
 99  $G_P = (V_P, E_P)$  inside a target graph  $G_T = (V_T, E_T)$ . In our case,  $G_P$  is a motif and  $G_T$  is  
 100 an entire RNA structure. Formally, the problem can be defined as:

101 ► **Problem 1. Subgraph Isomorphism Problem (SIP)**

102 **Input:** A pattern graph  $G_P = (V_P, E_P)$ , a target graph  $G_T = (V_T, E_T)$

103 **Output:** A mapping  $M : V_P \rightarrow V_T$  such that

104 ■  $\forall u, v \in V_P^2, M(u) = M(v) \rightarrow u = v$  (Mapping Injectivity)

105 ■  $\forall u, v \in E_P, (M(u), M(v)) \in E_T \Rightarrow L((u, v)) = L((M(u), M(v)))$  (Label Compatibility)

106 ■  $\forall (u, v) \in E_P, (M(u), M(v)) \in E_T$  (No Missing Edge)

107 or  $\emptyset$  if no such mapping exists.

108 We now generalize the problem to embrace the natural diversity of RNA motifs in  
 109 structures. More precisely, we are interested in sampling graph occurrences that are in the  
 110 geometric neighborhood of a core motif. To do so, we allow the motif to be deformed by  
 111 three different biologically relevant edit operations detailed below.

112 Each is additive and has its own **neighborhood threshold** and a corresponding difference  
 113 function as depicted in table 1:

114 ■  $T^L$  represents how much we allow the edge label, the type of the canonical or non-canonical  
 115 bond, to be modified. It measures the geometric difference between two interactions (see  
 116 Sec. 2.3.1). We denote the weights of edits induced by one changing edge as  $D^L$ ;

117 ■  $T^E$  corresponds to the maximum number of edges/base pairs within the pattern structure  
 118 that can be lost (see Sec. 2.3.2);

119 ■  $T^G$  is the maximum allowed distance when introducing a backbone discontinuity, a new  
 120 gap. As insertions alter the distance between bonds,  $T^G$  regulates here the maximum  
 121 sum over these shifts (see Sec. 2.3.3). We denote the weight of edits induced by the  
 122 introduced gap as  $D^G$ .

## ?:4 Exploring the natural fuzziness of RNA non-canonical geometries

Two additional parameters capture the **geometric distance**  $GEO$  between the nucleotides to constrain admissible solutions. First, nucleotides mapped to the nodes of a missing edge have to be at most at a distance of  $D_{edge}$ . Second, we enforce a maximal distance  $D_{gap}$  between the nucleotide on both side of an introduced gap. These values correspond to the phosphodiester atoms distances between the nucleotides. Setting these distances to fixed values ensures a much faster convergence of the algorithm.

We define a **path** as a sequence of more than 3 nucleotides connected through backbone edges. The set  $P$  of paths associated with a target graph  $G_T = (V_T, E_T = B_T \sqcup \bar{B}_T)$  is defined as:

$$P = \bigcup_{k \in \mathbb{N}, k \geq 3} \{(p_0, \dots, p_k) \mid \forall i \in \llbracket 0, k-1 \rrbracket, (p_i, p_{i+1}) \in B_T\}$$

With this definition, gaps are just paths in  $P$  with specific restrictions on length and composition.

A mapping  $M$  lying in a relevant neighborhood of a pattern graph is solution to a problem that we call the **Fuzzy Subgraph Isomorphism Problem (FSIP)**.  $M$  has to respect the five following conditions: The **mapping injectivity** condition enforces each pattern node to map to a different node in the target. The **label compatibility** controls how much the cumulative of geometric differences is allowed between pattern and matched edges (see Sec. 2.3.1). The **missing edge limitation** ensures that missing edges are not backbones and have limitations on their geometric lengths and numbers over the mapping. (see Sec. 2.3.2) The **path size limitation** controls how large the cumulative of gaps geometric lengths can be. (see Sec. 2.3.3) The **no missing backbone path** condition ensures limitation on the geometric length of individual gaps, but also that the gaps are composed only of unmapped nodes in the target graphs to the exclusion of its start and end points that should be mapped nodes (see Sec. 2.3.3)

We can now define formally the FSIP as:

► **Problem 2.** *Fuzzy Subgraph Isomorphism Problem (FSIP)*

**Input:** A pattern graph  $G_P = (V_P, E_P = B_P \sqcup \bar{B}_P)$ , a target graph  $G_T = (V_T, E_T = B_T \sqcup \bar{B}_T)$  and neighborhood thresholds  $(T^L, T^E, T^G, D_{edge}, D_{gap})$

**Output:** A mapping  $M : V_P \rightarrow V_T$  such that

- $\forall u, v \in V_P^2, M(u) = M(v) \rightarrow u = v$  (mapping injectivity)
- $\sum_{(u,v) \in \bar{B}} D^L(L(u,v), L(M(u), M(v))) \leq T^L$  (label compatibility)
- $(|\bar{B}| \leq T^E) \cap (\forall (u,v) \in \bar{B}, GEO(M(u), M(v))) \leq D_{edge}$  (missing edges limitation)
- $\sum_{(p_0, \dots, p_k) \in P} D^G(p_0, p_k) \leq T^G$  (path size limitation)
- $\forall (u,v) \in B, \exists (p_0, p_1, p_2, \dots, p_k) \in P$  such that (no missing backbone path)
  - $p_0 = M(u), p_k = M(v)$
  - $GEO(p_0, p_k) \leq D_{gap}$
  - $\forall i \in \llbracket 1, k-1 \rrbracket, \nexists a \in V_T, p_i = M(a)$

or  $\emptyset$  if no such mapping exists.

Subsequently, we will denote by **neighborhood** $_{G_P}(G_T)$  all the occurrences of the desired motif pattern  $G_P$  (in its geometric neighborhood) in our RNA graph target  $G_T$  as defined by the previous FSIP mapping.

Threshold $T^F$	Difference $d^F$	Fuzzy mapping $M$ of $G_P$ found in $G_T$
$T^L$	Isostericity $ISO$	
$T^E$	Missing edges number	
$T^G$	Geometric $GEO$ from 3D structure	

■ **Table 1 Neighborhood thresholds** Each measure has a threshold over the sum of differences over all edges in the graph pattern.

## 2.2 Sampling as an efficient alternative

Even with fixed neighborhood thresholds, FSIP remains a NP-complete problem. Indeed, FSIP generalizes the Subgraph Isomorphism Problem, yet remains in NP since the various conditions satisfied by a mapping can be verified in polynomial time.

Focusing on neighborhood $_{G_P}(G_T)$  is not an easy task as naive methods would describe both this set and its complementary. In the clique worst case, it consists to explore  $\binom{|G_T|}{|G_P|}$  graphs. Even the simple exploration of neighborhood $_{G_P}(G_T)$  can be tedious, in particular, when neighborhood thresholds are quite large, which is often the case for label and gap thresholds. Furthermore, due to the nature of the neighborhoods, numerous instances a few nucleotides apart will often be found. It is relevant in term of neighborhoods, but, in term of biology, they represent all the same RNA portion and the same underlying geometry and should not be distinguished: a single representative will be enough. It oriented us toward sampling, to identify sets of candidate – ideally diverse – subgraphs inside the target graph  $G_T$  that are at a reasonable “ distance” from the interesting motif  $G_P$ .

This shift in paradigm builds on recent advances in Multidimensional Boltzmann distributions and sampling [2, 14]. Generally, a **Boltzmann distribution** is such that the probability of any possible outcome  $G$  depends on its (pseudo-)energy  $E$

$$\mathbb{P}(G) = \frac{e^{-\beta E(G)}}{\mathcal{Z}} \text{ where } \mathcal{Z} = \sum_{G'} e^{-\beta E(G')} \quad (1)$$

where  $\beta$  is a real number, akin to an inverse temperature. A **Multidimensional Boltzmann distribution** (MBD) is a special type of Boltzmann distribution, where the

energy is a weighted combination over a collection of features  $\{F_i\}$  of interest, such that

$$E(G) = w_1 \times F_1(G) + w_2 \times F_2(G) + \dots$$

178 where  $w_1, w_2 \dots$  are real-valued weights. Weights can be used to steer the sampling towards  
 179 regions of interest. They can also be learned, through convex optimization, to match  
 180 the expectations of  $F_1, F_2 \dots$  to user-specified values. Moreover, sampling with a pseudo-  
 181 temperature  $\beta \rightarrow \infty$  gracefully specializes into a uniform random generation of outcomes  
 182 achieving optimal (i.e. minimal) value for  $E$ .

183 In our case, an outcome is a graph  $G \subset G_T$ , such as  $G$  is the image of mapping  $M$  and  
 184 we have 3 features, one for each neighborhood. Given a specific neighborhood threshold  
 185  $T^F$ , its relative feature  $F$  measures how much the difference  $d^F$  deviate from a given center  
 186  $T^{F*}$ . For instance,  $T^{F*}$  can be chosen equals to 0 if we want to sample mostly  $G$  with  
 187 no fuzziness or equals to  $T^F/2$ , if we want to sample them with average fuzziness. More  
 188 details on this choice of value and more generally about Boltzmann sampling is available at  
 189 part [A.1](#) from supplementary material. MBD is well-suited to the sampling that we want to  
 190 make: the exponential decrease of the probability with the features gives low probabilities  
 191 to the graphs that are far in neighborhood from  $G_P$ , which allows to characterise well  
 192 neighborhood $_{G_P}(G_T)$ . In particular, we can define  $F$  such that it takes a value equals  
 193 to  $+\infty$  when the corresponding neighborhood threshold  $T^F$  is not respected by mapping  
 194  $M$ , forbidding simply this mapping  $M$  to be sampled. Additionally, the Multidimensional  
 195 character of the distribution allows to take into account the 3 neighborhoods on labels, edges  
 196 and gaps at the same time.

197 A general framework called **InfraRed** [31], initially introduced in the context of RNA  
 198 design [14], can be used for generate efficiently in a parameterized the MBD. It automatically  
 199 processes constraints and elements of the scoring into a graph, decomposes it into a Tree  
 200 Decomposition, and generates automatically the bottom-up dynamic programming sampling  
 201 procedures. More details on the Tree Decomposition and the dynamic programming used in  
 202 **InfraRed** can be found in Supplementary Section [A.2](#).

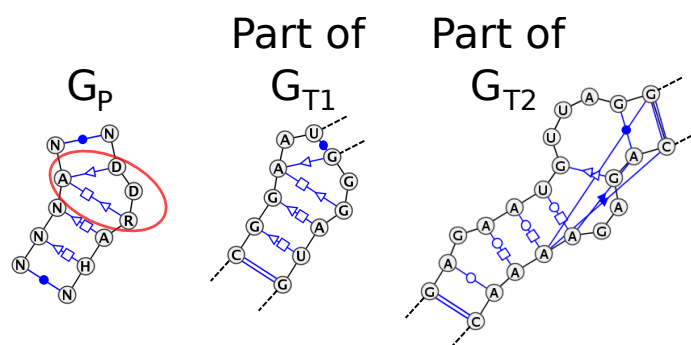
### 203 2.3 Neighborhood difference description

204 Our goal is to be able to retrieve from a general pattern motif all natural occurrences and their  
 205 variability. We can observe in well known motif families that some bases change, some can  
 206 be added or removed. For instance, the graph pattern  $G_P$  on figure 2 is a Kink-Turn whose  
 207 occurrences in the same sub-family can have up to four missing edges. Others sub-family  
 208 of Kink-Turn motifs can have differences in bond types, additional interactions, or even  
 209 gaps induced by additional nucleotides. We will define difference functions that will be the  
 210 features in the MB distribution and will restrain the samples to a “reasonable” neighborhood  
 211 of the pattern  $G_P$  that can be explicitly defined.

212 For any feature  $F$  (here  $F \in \{L, E, G\}$ , where  $L$  are label changes,  $E$  missing edges, and  
 213  $G$  new gaps) the **Neighborhood cumulative difference**  $D^F$  quantifies how distant a  
 214 mapping is relatively to a given neighborhood threshold  $T^F$  that cannot be exceeded.

215 Formally, we define a neighborhood cumulative difference  $D^F$  relatively to a neighborhood  
 216 threshold  $T^F$  as:

► **Definition 1.** *Neighborhood cumulative difference / neighborhood difference* Given a pattern graph  $G_P = (V_P, E_P)$ , a target graph  $G_T = (V_T, E_T)$  and a mapping  $M$ , a neighborhood cumulative difference is a function  $D^F$  relatively to a neighborhood threshold  $T^F$  that act as



■ **Figure 2 Kink-turn signature and targets** On the left, signature graph of the Kink-Turn IL\_29549.9 family and our search pattern. On the middle and on the left, mappings that were missed during the search for the pattern.  $G_{T1}$  due to the same nucleotide merging the end of a cSS and a cWW.  $G_{T2}$  due to its too large difference.

a wrapper around  $d_{G_T}^F$ :

$$D^F(G_P, G_T, M) = \sum_{(u,v) \in E_P} d_{G_T}^F(u, v, M)$$

217 Where  $d_{G_T}^F(u, v, M)$ , the **neighborhood difference** relative to  $G_T$  is a function that measures,  
 218 relatively to  $F$ , how “different” are the edges in the pattern  $((u, v) \in G_P)$  from the edges in  
 219 the mapping  $((M(u), M(v)) \in G_T)$ . How the difference is measured depends on the feature  
 220 as described below.

221 Neighborhood cumulative differences will serve in the Boltzmann distribution to quantify  
 222 each type of edit. Due to the additivity of these deformations, the neighborhood cumulative  
 223 differences are computed over all edges in the pattern and its equivalent in the mapping.  
 224 While our neighborhood cumulative differences are defined relatively to the edges of  $G_P$  here,  
 225 they can be easily defined on nodes should novel sequence-dependant features be included.  
 226 We will now discuss in details the 3 sources of operations and their neighborhood cumulative  
 227 difference. A summary is shown in Table 1.

### 228 2.3.1 The label difference

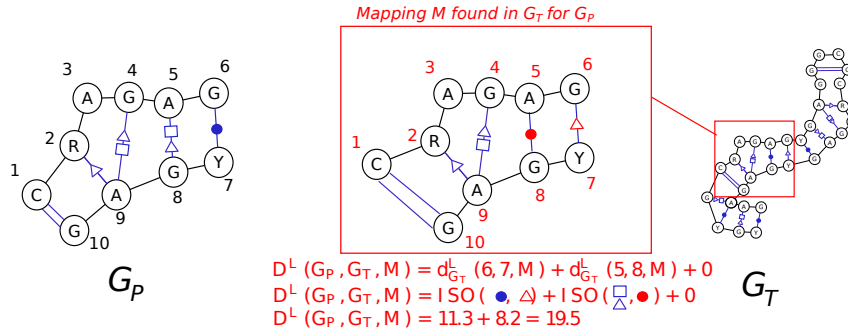
229 The isodiscrepancy index [29] quantifies the geometrically different base pair families and  
 230 provide values measuring three terms: (1) the difference of intra-base pair C1’-C1’ distances,  
 231 (2) after aligning one base, the inter-base pair C1’-C1’ distance between the C1’ atoms of the  
 232 second bases of the base pairs (3) The angle on an axis perpendicular to the base pair plane  
 233 required to superpose the second bases. This isostericity measure is only defined between  
 234 the 12 canonical and non canonical base pairing families (BPFs) which will be named as:  
 235  $BPF(i) \forall i \in \llbracket 1, 12 \rrbracket$ .

236 Inter family variations are frequent and therefore the average isodiscrepancy of a family  
 237 to itself is not 0. Because we use this value to account for the discrepancy between families  
 238 and assume that there should be no cost if there is no change in the interaction type, define  
 239 the *ISO* difference between two families as:

$$\forall (BPF(i), BPF(j)), ISO(BPF(i), BPF(j)) =$$

240 isodiscrepancy( $BPF(i), BPF(j)$ ) – isodiscrepancy( $BPF(i), BPF(i)$ )





■ **Figure 3 Label difference** Computation of the label difference on a mapping between a motif  $G_P$  and an RNA target graph  $G_T$ . Label difference is computed using the isostericity  $ISO$  to account for geometric difference between bounds as described in [29].

We now compute the label difference  $D^L$  relative to the neighborhood threshold  $T^L$  as a neighborhood cumulative difference entirely defined by the sum over each pattern edge of its mapping neighborhood difference  $d_{G_T}^L$  equals to

$$d_{G_T}^L(u, v, M) = ISO(L(u, v), L(M(u), M(v))),$$

241 as shown in Fig. 3.

### 242 2.3.2 The edge difference

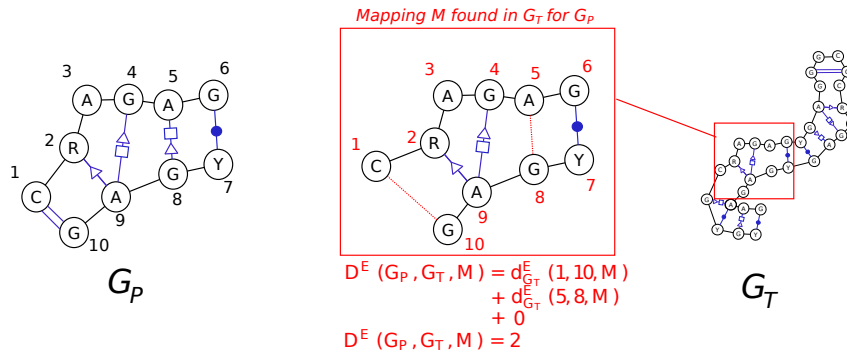
243 While the previous section deals with how to incorporate edges changing their type, that  
 244 is to say their interaction geometry, we must also consider that some of these base pair  
 245 interactions might simply be missing due to the noisiness of the experiments, the accuracy  
 246 of the annotation, or the flexibility of the module. A natural way to account for missing  
 247 edges is to count them and enforce an upper bound on the amount. Doing so would omit  
 248 important geometric information that we have available in the 3D structure. An interaction  
 249 is missing, but we still want to constraint the physical distance between the mapped nodes of  
 250 the missing edge. Indeed, with no limitation on that distance, the partner node of a missing  
 251 edge could be virtually anywhere in the target structure. This is undesirable since we are  
 252 interested in patterns matching the global conformation. It is also highly inefficient in terms  
 253 of computation.

254 Therefore, we will accept mappings of the extremities of an edge in the pattern to nodes  
 255  $u, v$  that are at most at a set threshold distance  $D_{edge}$  computed from the 3D structure  
 256 (i.e.  $GEO(u, v) < D_{edge}$ ). Setting a weight of  $\infty$  to mappings outside the threshold allows  
 257 the sampling to simply reject such instances. The total edge difference  $D^E$  relative to  
 258 neighborhood threshold  $T^E$ , is a neighborhood cumulative difference entirely defined by the  
 259 sum over  $d_{G_T}^E$  with values defined as followed and shown in Fig/ 4:

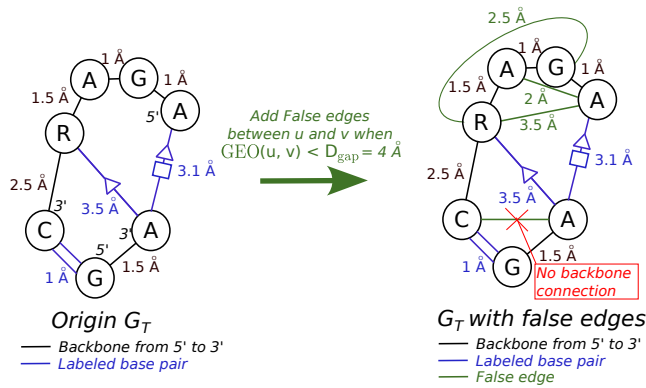
$$260 \quad d_{G_T}^E(u, v, M) = \begin{cases} \infty & \text{if } GEO(M(u), M(v)) > D_{edge} \\ & \text{and } (M(u), M(v)) \notin E_T \\ 1 - \mathbb{1}_{(M(u), M(v)) \in E_T} & \text{otherwise} \end{cases}$$

### 261 2.3.3 The gap difference

262 A frequent type of natural variability in a motif family is the insertion of bulging out  
 263 nucleotides in what would be a continuous sequence in the pattern. These insertions can be



■ **Figure 4 Edge difference** Computation of the edge difference on a mapping between a motif  $G_P$  and an RNA target graph  $G_T$ . We assume here that  $D_{edge} \gg \max(GEO(1, 10), GEO(5, 8))$



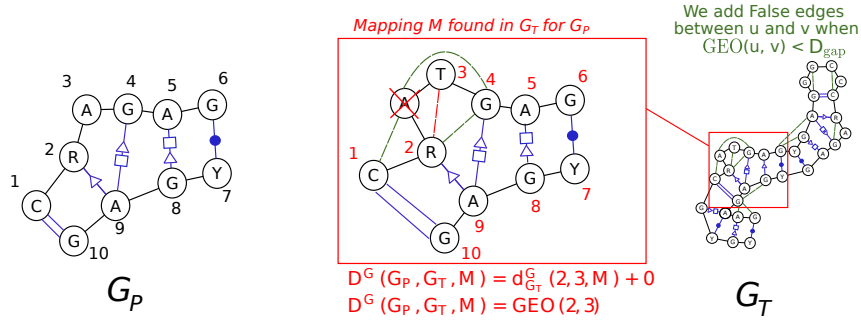
■ **Figure 5 False edges** Addition of False edges to account for gaps. False edges are added only when distance are below  $D_{gap}$  and when both nucleotides are fully connected by backbone edges. For instance here, we add no false edge between C and A at the bottom of  $G_T$  as this two nucleotides are not connected by a full path of backbones.

264 of different size, but we require that they do not modify (too much) the local structure. To  
 265 take arbitrary insertions into account we introduce **false edges** between any two nucleotides  
 266 present on the same backbone that are at a distance below  $D_{gap}$ . An illustration of this  
 267 process is shown in Fig. 5.

268 An additional difference compared to the missing interaction edges of the previous section,  
 269 is how we sum the total neighborhood difference  $D^G$ . We accumulate the total physical  
 270 distance (i.e.  $GEO$ ) between the nodes connected through the false edges. This allows an  
 271 arbitrary large structure to bulge out without the need to verify or specify admissible lengths,  
 272 as long as the nucleotides around this inserted gap are close geometrically as illustrated in  
 273 Fig. 6.

Formally, the gap difference  $D^G$  relative to neighborhood threshold  $T^G$  is a neighborhood cumulative difference over all edges in the matching entirely defined by the sum of the neighborhood differences  $d_{G_T}^G$ :

$$d_{G_T}^G(u, v, M) = \begin{cases} GEO(M(u), M(v)) & \text{if } (M(u), M(v)) \text{ is} \\ & \text{a "False Edge" in } E_T \\ 0 & \text{otherwise} \end{cases}$$



■ **Figure 6 Gap difference** Computation of gap difference on a mapping between a motif  $G_P$  and an RNA target graph  $G_T$ . We remind that nucleotides labels are not taken into account.

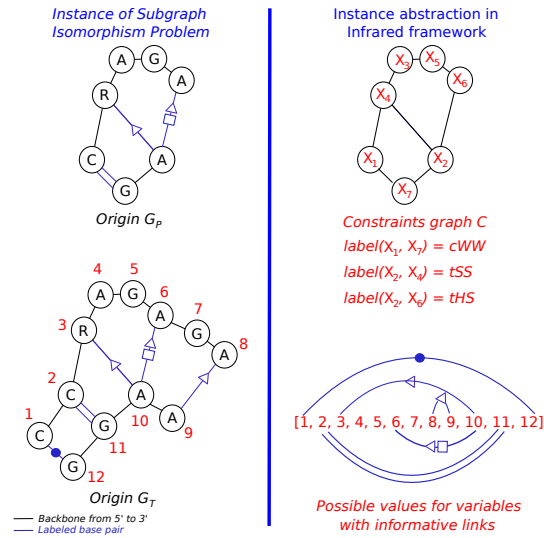
274 A limitation of this approach is that we cannot detect the deletion of nodes from the  
 275 pattern. A workaround is to remove all the nodes in the pattern graph that do directly  
 276 participate in a base pair interaction, and reconnect the disconnected backbones. Using  
 277 the new pattern with a large gap threshold  $G$  would allow to retrieve the original motif  
 278 neighborhood at the cost of performance, and to introduce more spurious matches.

## 279 2.4 Algorithm and complexity

280 Our method is based on *Infrared* [14, 31], a declarative framework which automatically  
 281 generates a dynamic programming procedure for MBD sampling, based on a nice tree  
 282 decomposition (TD). It precomputes the partition function of the MBD through a bottom-  
 283 up recursion, and uses local contributions to perform an exact sampling within the MBD  
 284 distribution. Within this framework, a combinatorial problem is abstracted as a set of  
 285 variables  $\{X_i\}_i$ , each assigned an integer value within a bounded domain. Assignments  
 286 must respect various constraints expressed as functions  $\{C_i\}_i$ , each defined over a subset of  
 287 variables. Similarly, feature functions  $\{F_j\}_j$  associate real-valued contributions to subsets of  
 288 variables, and are summed to represent the pseudo-energy of an assignment.

289 In this setting, we abstract each node  $i$  of the graph pattern  $G_P$  as a variable  $X_i$ ,  
 290 taking value in  $\llbracket 1, n \rrbracket$ . The value of  $X_i$  represents the mapping of node  $i$  in the graph  
 291  $G_T = (V_T, E_T)$  with  $|V_P| = k$  and  $|V_T| = n$ . Remark that all deviations from the pattern  
 292 defined in Sections 2.3.1 through 2.3.3, can be expressed *locally* as sums on the edges of the  
 293 pattern graph. It follows that the dependencies  $dep$  implied by our cumulative differences are  
 294 only binary, and restricted to pairs sharing an edge in  $G_P$ :  $dep = \{(X_i, X_j) \mid (i, j) \in E_P\}$ .  
 295 The graph of constraints is thus reducible to the input pattern graph  $G_P$ , as shown in  
 296 Fig. 7. In mappings sampled with *Infrared*, a neighborhood threshold  $T^F$  act as a global  
 297 property over the mapping, and can only be computed afterward. Sampling is thus followed  
 298 by a simple rejection step, in which samples that exceed a neighborhood threshold are  
 299 rejected. Asymptotically, such rejection will not be impactful with the  $T^F$  that can be  
 300 chosen independantly from  $|G_P|$  and  $|G_T|$ . Due to the neighborhood threshold  $T^F$  being a  
 301 global property over the mapping, the sampling is followed by a rejection step for samples  
 302 that exceed a neighborhood threshold. Asymptotically, such rejection will at worst induce a  
 303 constant factor with  $T^F$  chosen independantly from  $|G_P|$  and  $|G_T|$ .

304 ► **Theorem 1.** *The random generation of  $t$  Boltzmann-distributed (1) solutions before*  
 305 *rejection of FSIP can be done in time  $\mathcal{O}(nkt + kn^{(\phi+2)})$ , where  $\phi$  is the treewidth of the*  
 306 *pattern  $G_P$ .*



■ **Figure 7 Framework abstraction** Interfacing *Infrared* by considering  $G_P$  as the *Infrared* graph of constraints  $C$  and all nodes of  $G_T$  as values that can be taken by the variables in  $C$ .

307 This complexity directly follows from the complexity of the algorithm [14] underlying  
 308 *Infrared* for a graph  $G = (V, E)$ . Restricted to binary constraints/features associated  
 309 with (a subset of)  $E$ , the computation of the partition function can be performed in time  
 310  $\mathcal{O}((|E| + |V|) \times \Delta^{\phi+1})$ , where  $\Delta$  is the size of the assignment domain for individual variables,  
 311 and  $\phi$  is the treewidth of  $G$ . A stochastic backtrack follows, leading to the generation of  
 312  $t$  Boltzmann-distributed assignments in time  $\mathcal{O}(|V| \Delta t)$ . The complexity stated above is  
 313 obtained by observing that  $|E_P| \in \Theta(k^2)$ , that  $\Delta \in \Theta(n)$  and  $k \leq n$ . We conclude by noting  
 314 that preprocessing, including computations of geometrical distances and augmentation of  
 315  $G_T$  graph, can be performed in negligible  $O(n^2)$  time and space, while an optimal tree  
 316 decomposition can be theoretically obtained in time only super polynomial in  $\phi$  [3].

317 A summary of the complexity and capacity of our FuzzTree method is depicted in table 2.  
 318 In term of parameterized complexity [9], the FuzzTree method is XP in the treewidth of the  
 319 pattern graph, both in time and in space. It is a progress compared to VF2 [7], which is  
 320 indeed implemented and efficient in practise due to the profusion of lookahead rules but has  
 321 a worst case time complexity similar to  $O(n^n)$ . In practise, VF2 becomes costly with dense  
 322 graphs, even in its most modern versions. [4, 16] Furthermore, we are able to compete with  
 323 the bound from the Color-Coding [1] technique by improving it in time and space  $2^{O(k)}$  is  
 324 replaced by  $kn \leq n^2$  in our bounds, which allow to get rid of  $k$  as parameter to restrict it to  
 325 simply to the treewidth.

326 In addition, our method, even if tuned for RNAs, supports a more general version of the  
 327 usual Subgraph Isomorphism Problem by handling at the same time multiple labels on edges,  
 328 directed graphs and can integrate node labels. The latter has not been implemented but can  
 329 be added, as with label on edges, without complexity overhead.

Method Name	Color-Coding	VF2	VeRNAL	FuzzTree
Year	1995	2004 (updated up to 2018)	2021	2022
Method	Tree coloring	DFS with search space reduction	Relational Graph Convolution Network	Sampling technique
Time complexity	$2^{O(k)}n^{\phi+1}\log(n)$	$O(\deg(G_T)^n)$	Exponential	$O(knt + kn^{\phi+2})$
Space complexity	$2^{O(k)}n^{\phi+1}$	$O(n)$	Exponential	$O(n^{\phi+2})$
Supported graph	Directed and undirected	Undirected	Directed and undirected	Directed and undirected
Supported labels	One label by edge	One label by node	Any number of labels on edges and nodes	Any number of labels on edges and nodes
Type of found neighborhoods	None	None	Isostericity related	Exact bound on isostericity, missing edge and missing gap.
Implementation?	No	Yes	Yes	Yes

■ **Table 2 Complexity** Comparison of state of the art methods for the Subgraph Isomorphism Problem. With  $\phi = tw(G_P)$ ,  $n = |V_T|$ , and  $t$  the number of samples.

### 3 Results

#### 3.1 Computations

The larger target graphs (of more than 500 nucleotides) were split into overlapping voxels to increase computational efficiency. We extracted  $|G_T|$  graphs centered in each nucleotides  $c$  at a given radius  $R$  from  $c$ . For an extracted graph  $G$ , centered on  $c$ , we have :

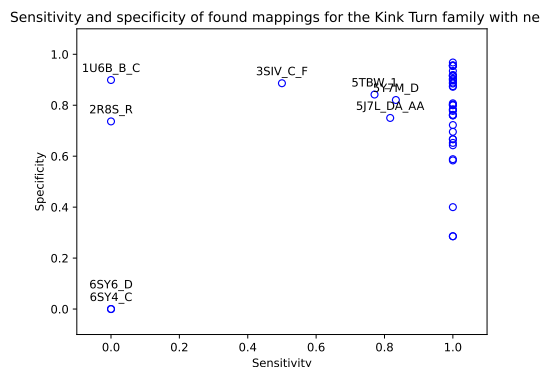
$$\forall j \in G, R(G) = \text{GEO}(j, c) \leq R$$

Choices of technical parameters, such as the value for  $R$ , hardware and computation times are discussed in Supp. Mat. A.4.

#### 3.2 Data: the Kink-Turn group

All interactions in the RNA structures are provided by FR3D [27]. We also use interactions annotated as “near”. The Kink-Turn is an important RNA structural motif common in duplex RNA that creates a sharp axial bend, enabling crucial tertiary interactions and binding [18]. The Kink-Turn has been shown to appear in multitudes of contexts through computational and experimental methods [15, 21]. As of January 2023 there were 72 instances of the Kink-Turn RNA annotated in the RNA3DMotifAtlas [25]. They span 46 different RNAs and are divided in 12 different families with different lengths, between 9 and 23 nucleotides and base pair signature. Members of the same family also differ in term of number of nucleotides and pairing.

The Kink-Turn family IL\_29549.9 in RNA3DMotifsAtlas has the most occurrences (32) and its signature graph shown in Fig. 2 is used as the pattern graph  $G_P$  for the subsequent sampling. We explore more in depth the distances between all instances and this core graph in Supp. Mat. 11.



■ **Figure 8** Sensitivity and Specificity of regions corresponding to sampled graphs in the 46 RNA structures containing Kink-Turns.

### 3.2.1 Results

We use the parameters shown in table 3 with  $G_P$  in Fig. 2 to sample at least 1000 graphs in each of the 46 RNA structures. We also introduce a bias in the Boltzmann distribution in order to favor values of neighborhood thresholds equal to  $\frac{T^F}{2}$  (instead of 0) to favor slightly fuzzy mappings more often than exact mappings or extremely fuzzy ones. This choice is motivated by the focus on the neighborhood more than on the exact mappings for which lots of techniques already exist.

Parameter	$T^L$	$T^E$	$T^G$	$D_{\text{edge}}$	$D_{\text{gap}}$	$R$	nb_samples
Used value	20.0	4	20.0	5.0	10.0	$R(G_P) + \frac{D_{\text{gap}}}{4}$	1000
Relevant range	[0, 50]	[[0, 6]]	[0, 50]	[5, 10]	[5, 20]	$R(G_P) + [\frac{D_{\text{gap}}}{4}, D_{\text{gap}}]$	

■ **Table 3 Parameters** Used parameters and relevant range for FuzzTree computation on the Kink-Turn group.

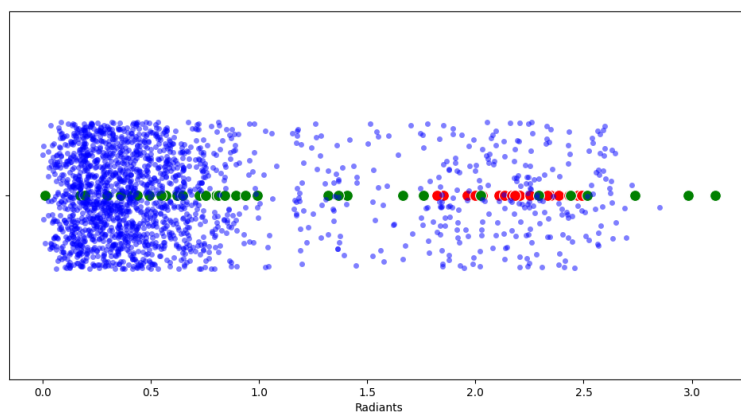
Our sampling returns sub-graphs of the target graphs  $G_T$ . Using a python implementation of VF2 [13, 7] we annotate in the 46 RNAs graphs all nucleotides in any of the mappings. Each of the connected components in the 46 RNAs becomes a hit. The True Positives (TP) are these covering a known Kink-Turn, the False Positives (FP) are the other ones. We show the sensitivity (TP/P) and specificity (1 - FP/(FP + TP)) per RNA structure in Fig. 8.

In 38 out of the 46 RNAs a sensitivity of 1 is achieved, all Kink-Turns are covered in graphs sampled by our method. The missing Kink-Turns fall in two categories. First, too many missing edges : with only 6 Leontis-Westhof interactions in  $G_T$ , allowing more missing edges would match any interaction in the targets. Second, backbone connections replaced by Leontis-Westhof interactions, as seen on the middle of Fig. 2, is not an allowable transformation in our model.

We also obtain in 33 RNAs a specificity over 75%. It indicates that even with relatively lax parameters, not that many other instances in comparison to the amount of known Kink-Turns are close to  $G_T$ .

369 **3.2.2 Other identified regions**

370 An additional 198 locations in the 46 RNAs were identified. The Kink-Turn is essentially  
 371 an internal loop motif. We investigate if other internal loops sharing the same main 3D  
 372 feature, a sharp bend in an interior loop, are found. Using the python library forgi [30] we  
 373 decomposed these regions in their secondary structure elements. The majority, 125, mapped  
 374 to regions forming multiloops. A total of 33 were covering a continuous double stranded  
 375 regions. The angles of surrounding stems for each interior loops in the 46 RNAs (in blue) the  
 376 identified Kink-Turns in these RNAs (red) and the other 33 elements (in green) are shown in  
 377 Fig. 9.



■ **Figure 9 Angles in radians** In blue for stems around every interior loop in the 46 RNAs. In red for the Kink-Turns identified in these RNAs. In green for the additional 33 continuous double stranded regions.

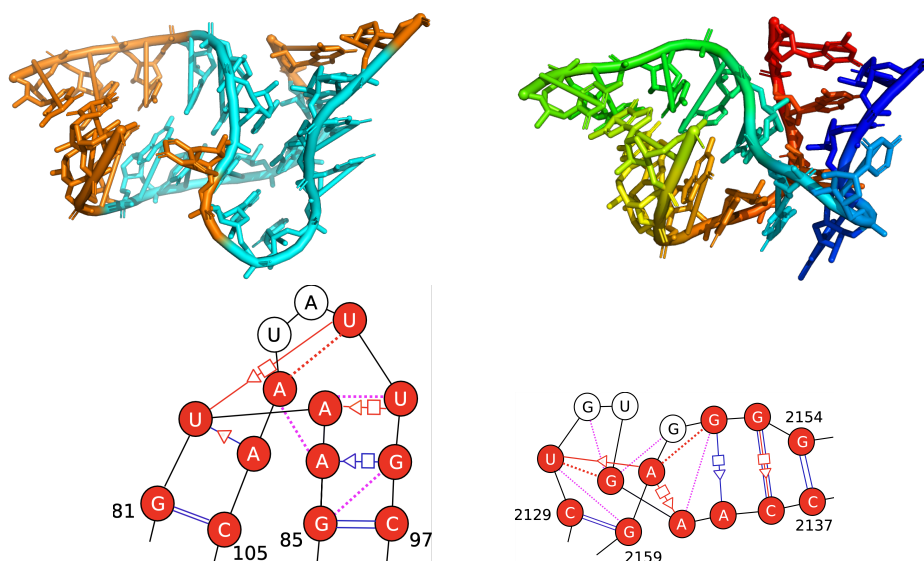
378 There are 10 additional regions with angles above 1.4rad, and two of these had a sharp  
 379 turn in their structure in un-annotated region as seen in Fig. 10. We show below their graph  
 380 of interactions, with the cross strand stackings in orange.

381 The first is in 5J7L chain DA and positions 78–86, 96–108. It overlaps an un-annotated  
 382 motif (IL\_85931.1) that covers positions 81–85, 97–101, and 103–105. The second is located  
 383 in 7RQB, chain 1A, positions 2129–2138, 2153–2160, and is not covered or surrounded by  
 384 any annotated motif.

385 **4 Conclusion**

386 In this paper, we introduce FuzzTree, a multidimensional Boltzmann method for sampling a  
 387 graph pattern neighborhood in a target graph. FuzzTree defines three types of neighborhoods  
 388 based on RNA geometric diversity, LW interaction modifications, missing edges, and breaks  
 389 in the backbone. Each can be explicitly controlled. We show that our sampling method  
 390 complexity is parameterized by the treewidth of the pattern graph.

391 Two main limitations are inherent to our approach. Due to the intrinsic nature of  
 392 sampling, we cannot be assured that all neighboring graphs will be reported. In itself, for  
 393 large patterns, this is a feature since sampling allows uniform exploration of the exponentially  
 394 growing neighborhood. By enabling per-feature biases, FuzzTree can also be calibrated to  
 395 favor the sampling of graphs at a desired location in the neighborhood to favor specific



■ **Figure 10** Other matches 5J7L on the left and 7RQB on the right. The 3D structure on the left has IL\_85931.1 highlighted in cyan, on the right each nucleotide is colored independently. In the graphs, red nodes are matched with the pattern. Blue edges are in the RNA structure and red ones in the pattern, indicating modifications and removal. Red dashed lines are introduced “false edges”. Magenta dashed lines indicate stackings.

396 types of variants (e.g., isosteric distance of modified edges). Letting the sampling run for  
 397 longer will also mitigate the problem. More importantly, some patterns cannot be identified,  
 398 particularly if an LW interaction is replaced by a backbone connection. While such cases are  
 399 rare, they do exist, and additional improvement will be needed to capture them.

400 We evaluate our method on the Kink-Turn group, a well-known interior loop motif that  
 401 induces a sharp bend in the structure and is annotated in 46 different RNA structures.  
 402 The Kink-Turns are grouped in the RNA3DMotifAtlas into 12 different subgroups with  
 403 varying lengths and interactions. Using only the signature graph of one subgroup, FuzzTree  
 404 samples conformations of over 2/3 of all Kink-Turns and identifies all of them in 88% of  
 405 RNA structures. Closer examination of the other sampled patterns reveals two previously un-  
 406 annotated sub-structures, each with a characteristic G-A trans-Hoogsteen-sugar interaction  
 407 and a sharp local bend.

408 Future work to complement this should broaden the evaluation framework by testing  
 409 FuzzTree on diverse RNA modules. There is also a need for new techniques to overcome  
 410 pattern identification limitations and explore adaptive sampling strategies to dynamically  
 411 steer the sampled neighborhood.

412 While FuzzTree was developed and adapted for RNA structure modules, it highlights the  
 413 flexibility of multidimensional Boltzmann sampling and could be applied to other biological  
 414 networks such as protein-protein interaction networks or metabolic pathways. Addressing  
 415 these questions and areas for future work could lead to more comprehensive insights into  
 416 complex RNA structures and other biological networks.

## 417 ——— References ———

- 418 1 Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *J. ACM*, 42(4):844–856, jul 1995.  
 419 [doi:10.1145/210332.210337](https://doi.org/10.1145/210332.210337).



- 420 2 Olivier Bodini and Yann Ponty. Multi-dimensional Boltzmann Sampling of Languages. *Dis-*  
421 *crete Mathematics & Theoretical Computer Science*, DMTCS Proceedings vol. AM, 21st  
422 International Meeting on Probabilistic, Combinatorial, and Asymptotic Methods in the Ana-  
423 lysis of Algorithms (AofA'10), January 2010. URL: <https://dmtdcs.episciences.org/2793>,  
424 [doi:10.46298/dmtcs.2793](https://doi.org/10.46298/dmtcs.2793).
- 425 3 Hans L. Bodlaender and Arie M. C. A. Koster. Combinatorial optimization on graphs of  
426 bounded treewidth. *The Computer Journal*, 51(3):255–269, 2008. [doi:10.1093/comjnl/  
427 bxm037](https://doi.org/10.1093/comjnl/bxm037).
- 428 4 Vincenzo Carletti, Pasquale Foggia, Alessia Saggese, and Mario Vento. Introducing vf3: A  
429 new algorithm for subgraph isomorphism. In Pasquale Foggia, Cheng-Lin Liu, and Mario  
430 Vento, editors, *Graph-Based Representations in Pattern Recognition*, pages 128–139, Cham,  
431 2017. Springer International Publishing.
- 432 5 Thomas R Cech and Joan A Steitz. The noncoding RNA revolution—trashing old rules to  
433 forge new ones. *Cell*, 157(1):77–94, 2014.
- 434 6 G Chojnowski, T Waleń, and JM Bujnicki. RNA Bricks—a database of RNA 3D motifs and their  
435 interactions. *Nucleic Acids Research*, 42, 2013. URL: <https://doi.org/10.1093/nar/gkp011>,  
436 [doi:10.1093/nar/gkt1084](https://doi.org/10.1093/nar/gkt1084).
- 437 7 Luigi Cordella, Pasquale Foggia, Carlo Sansone, and Mario Vento. A (sub)graph isomorphism  
438 algorithm for matching large graphs. *Pattern Analysis and Machine Intelligence, IEEE  
439 Transactions on*, 26:1367 – 1372, 11 2004. [doi:10.1109/TPAMI.2004.75](https://doi.org/10.1109/TPAMI.2004.75).
- 440 8 José Almeida Cruz and Eric Westhof. The dynamic landscapes of RNA architecture. *Cell*,  
441 136(4):604–609, 2009.
- 442 9 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Daniel Marx, Marcin  
443 Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2016.
- 444 10 Rhiju Das, Rachael C Kretsch, Adam J Simpkin, Thomas Mulvaney, Phillip Pham, Ramya  
445 Rangan, Fan Bu, Ronan Keegan, Maya Topf, Daniel Rigden, et al. Assessment of three-  
446 dimensional RNA structure prediction in CASP15. *bioRxiv*, pages 2023–04, 2023.
- 447 11 Sven Findeiß, Christoph Flamm, and Yann Ponty. Rational Design of RiboNucleic Acids  
448 (Dagstuhl Seminar 22381). *Dagstuhl Reports*, 12(9):121–149, 2023. URL: [https://drops.  
449 dagstuhl.de/opus/volltexte/2023/17811](https://drops.dagstuhl.de/opus/volltexte/2023/17811), [doi:10.4230/DagRep.12.9.121](https://doi.org/10.4230/DagRep.12.9.121).
- 450 12 Nagoor Gani. 63. isomorphism on fuzzy graphs. *International Journal of Computational and  
451 Mathematical Sciences*, Vol. 2:200–206, 01 2008. [doi:10.13140/2.1.1873.9847](https://doi.org/10.13140/2.1.1873.9847).
- 452 13 Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and  
453 function using NetworkX. Technical report, Los Alamos National Lab.(LANL), Los Alamos,  
454 NM (United States), 2008.
- 455 14 S. Hammer, W. Wang, S. Will, and Y. Ponty. Fixed-parameter tractable sampling for rna  
456 design with multiple target structures. *BMC Bioinformatics*, 2019.
- 457 15 Lin Huang and David MJ Lilley. The kink turn, a key architectural element in RNA structure.  
458 *Journal of molecular biology*, 428(5):790–801, 2016.
- 459 16 Alpár Jüttner and Péter Madarasi. Vf2++—an improved subgraph isomorphism algorithm. *Dis-*  
460 *crete Applied Mathematics*, 242:69–81, 2018. Computational Advances in Combinatorial Optim-  
461 ization. URL: <https://www.sciencedirect.com/science/article/pii/S0166218X18300829>,  
462 [doi:https://doi.org/10.1016/j.dam.2018.02.018](https://doi.org/10.1016/j.dam.2018.02.018).
- 463 17 Arijit Khan, Nan Li, Xifeng Yan, Ziyu Guan, Supriyo Chakraborty, and Shu Tao. Neighborhood  
464 based fast graph search in large networks. In *Proceedings of the 2011 ACM SIGMOD  
465 International Conference on Management of Data*, SIGMOD '11, page 901–912, New York,  
466 NY, USA, 2011. Association for Computing Machinery. [doi:10.1145/1989323.1989418](https://doi.org/10.1145/1989323.1989418).
- 467 18 Daniel J Klein, T Martin Schmeing, Peter B Moore, and Thomas A Steitz. The kink-turn: a  
468 new RNA secondary structure motif. *The EMBO journal*, 20(15):4214–4221, 2001.
- 469 19 Neocles B Leontis, Aurelie Lescoate, and Eric Westhof. The building blocks and motifs of  
470 RNA architecture. *Current Opinion in Structural Biology*, 16(3):279–287, 2006. Nucleic

- 471 acids/Sequences and topology. URL: [https://www.sciencedirect.com/science/article/](https://www.sciencedirect.com/science/article/pii/S0959440X06000807)  
 472 [pii/S0959440X06000807](https://www.sciencedirect.com/science/article/pii/S0959440X06000807), doi:<https://doi.org/10.1016/j.sbi.2006.05.009>.
- 473 20 Neocles B Leontis and Eric Westhof. Geometric nomenclature and classification of RNA base  
 474 pairs. *Rna*, 7(4):499–512, 2001.
- 475 21 Bin Li, Shurong Liu, Wujian Zheng, Anrui Liu, Peng Yu, Di Wu, Jie Zhou, Ping Zhang, Chang  
 476 Liu, Qiao Lin, et al. RIP-PEN-seq identifies a class of kink-turn RNAs as splicing regulators.  
 477 *Nature Biotechnology*, pages 1–13, 2023.
- 478 22 Dániel Marx and Michal Pilipczuk. Everything you always wanted to know about the  
 479 parameterized complexity of Subgraph Isomorphism (but were afraid to ask). In Ernst W.  
 480 Mayr and Natacha Portier, editors, *31st International Symposium on Theoretical Aspects of*  
 481 *Computer Science (STACS 2014)*, volume 25 of *Leibniz International Proceedings in Informatics*  
 482 *(LIPIcs)*, pages 542–553, Dagstuhl, Germany, 2014. Schloss Dagstuhl–Leibniz-Zentrum fuer  
 483 Informatik. URL: <http://drops.dagstuhl.de/opus/volltexte/2014/4486>, doi:[10.4230/](https://doi.org/10.4230/LIPIcs.STACS.2014.542)  
 484 [LIPIcs.STACS.2014.542](https://doi.org/10.4230/LIPIcs.STACS.2014.542).
- 485 23 Carlos Oliver, Vincent Mallet, Pericles Philippopoulos, William L Hamilton, and Jérôme  
 486 Waldispühl. Vernal: a tool for mining fuzzy network motifs in RNA. *Bioinformatics*, 38(4):970–  
 487 976, 11 2021. arXiv:[https://academic.oup.com/bioinformatics/article-pdf/38/4/970/](https://academic.oup.com/bioinformatics/article-pdf/38/4/970/42319124/btab768.pdf)  
 488 [42319124/btab768.pdf](https://academic.oup.com/bioinformatics/article-pdf/38/4/970/42319124/btab768.pdf), doi:[10.1093/bioinformatics/btab768](https://doi.org/10.1093/bioinformatics/btab768).
- 489 24 Aymeric Perchant and Isabelle Bloch. Fuzzy morphisms between graphs. *Fuzzy Sets and*  
 490 *Systems*, 128(2):149–168, 2002. URL: [https://www.sciencedirect.com/science/article/](https://www.sciencedirect.com/science/article/pii/S0165011401001312)  
 491 [pii/S0165011401001312](https://www.sciencedirect.com/science/article/pii/S0165011401001312), doi:[https://doi.org/10.1016/S0165-0114\(01\)00131-2](https://doi.org/10.1016/S0165-0114(01)00131-2).
- 492 25 Anton I. Petrov, Craig L. Zirbel, and Neocles B. Leontis. Automated classification of rna 3d  
 493 motifs and the rna 3d motif atlas. *RNA*, 2013.
- 494 26 Vladimir Reinharz, Antoine Soulé, Eric Westhof, Jérôme Waldispühl, and Alain Denise. Mining  
 495 for recurrent long-range interactions in RNA structures reveals embedded hierarchies in network  
 496 families. *Nucleic Acids Research*, 46(8):3841–3851, 03 2018. arXiv:[https://academic.oup.](https://academic.oup.com/nar/article-pdf/46/8/3841/24783244/gky197.pdf)  
 497 [com/nar/article-pdf/46/8/3841/24783244/gky197.pdf](https://academic.oup.com/nar/article-pdf/46/8/3841/24783244/gky197.pdf), doi:[10.1093/nar/gky197](https://doi.org/10.1093/nar/gky197).
- 498 27 Michael Sarver, Craig L Zirbel, Jesse Stombaugh, Ali Mokdad, and Neocles B Leontis. FR3D:  
 499 finding local and composite recurrent structural motifs in RNA 3D structures. *Journal of*  
 500 *mathematical biology*, 56:215–252, 2008.
- 501 28 Antoine Soulé, Vladimir Reinharz, Roman Sarrazin-Gendron, Alain Denise, and Jérôme  
 502 Waldispühl. Finding recurrent RNA structural networks with fast maximal common subgraphs  
 503 of edge-colored graphs. *PLoS computational biology*, 17(5):e1008990, 2021.
- 504 29 Jesse Stombaugh, Craig L. Zirbel, Eric Westhof, and Neocles B. Leontis. Frequency and isoster-  
 505 icity of RNA base pairs. *Nucleic Acids Research*, 37(7):2294–2312, 02 2009. arXiv:[https://](https://academic.oup.com/nar/article-pdf/37/7/2294/18783207/gkp011.pdf)  
 506 [academic.oup.com/nar/article-pdf/37/7/2294/18783207/gkp011.pdf](https://academic.oup.com/nar/article-pdf/37/7/2294/18783207/gkp011.pdf), doi:[10.1093/nar/](https://doi.org/10.1093/nar/gkp011)  
 507 [gkp011](https://doi.org/10.1093/nar/gkp011).
- 508 30 Bernhard C Thiel, Irene K Beckmann, Peter Kerpedjiev, and Ivo L Hofacker. 3D based on  
 509 2D: Calculating helix angles and stacking patterns using forgi 2.0, an RNA Python library  
 510 centered on secondary structure elements. *F1000Research*, 8, 2019.
- 511 31 Hua-Ting Yao, Yann Ponty, and Sebastian Will. Developing complex rna design applications  
 512 in the infrared framework. *RNA Folding - Methods and Protocols*, 2022.
- 513 32 M. Zahran, C. Sevim Bayrak, S. Elmetwaly, and T. Schlick. RAG-3D: a search tool for RNA  
 514 3D substructures. *Nucleic acids research*. *Nucleic Acids Research*, 43(19):9474–9488, 2015.  
 515 doi:<https://doi.org/10.1093/nar/gkv823>.

## 516 **A** Supplementary material

### 517 **A.1** About the sampling process

518 Sampling from a Multidimensional distribution in our case can be write formally as below :

► **Definition 2.** *Boltzmann distribution/Partition function* In the **Multidimensional Boltzmann Distribution**, the probability to sample graph  $G$  subgraph of  $G_T$  with features  $F_1, \dots, F_m$  (that embody neighborhoods of  $G_P$  for  $G$ ) of respective weights  $w_1, \dots, w_m$  (that we can write more simply  $w = (w_1, \dots, w_m)$ ) is proportional to its **energy**:

$$\mathbb{P}_{G_P, G_T}(G | w) = \frac{\prod_{i=1}^m e^{-\beta w_i \cdot F_i(G)}}{\mathcal{Z}_w}$$

where  $\beta := (RT)^{-1}$ ,  $R$  is the Gas constant,  $T$  the temperature in Kelvin, and  $\mathcal{Z}_w$  denotes the **partition function**

$$\mathcal{Z}_w = \sum_{G \subseteq G_T} \prod_{i=1}^m e^{-\beta w_i \cdot F_i(G)}$$

519 We can forget about the  $\beta$  contribution as we can rewrite the weight as  $w'_i = \beta w_i$  with  $w_i$  a  
520 value that is chosen or tuned by us.

521 In term of sampling we want to have a sampling center  $\mathcal{T}^{F^*}$  for cumulative difference  $D^F$   
522 in order to obtain matching with feature and so neighborhood cumulative difference close to  
523  $\mathcal{T}^{F^*}$  value.

524 One good choice for  $T^{F^*}$  can be something depending from  $T^F$  such as  $T^F/2$  which is  
525 typically the choice that we used. It favors samplings of graphs that are not too distant in  
526 term of neighborhoods. In addition, we still allows exact match and we can always preprocess  
527 to remove false positive found graph.

By fixing  $T^F$  and so  $T^{F^*}$ , we can tune the weight  $w(F_i)$  in order to give more “importance”  
to the one that are relevant for the sampling. When a feature for a neighborhood vary greatly  
between instances, it means that this neighborhood is strongly relevant to distinguish the  
different matches. It gives us incentive to modify its weight accordingly. To do so, instead of  
choosing weights manually, we solve the following problem:

$$\min_w \sum_{i=1}^m |\mathbb{E}[F_i | w] - F_i^*|$$

528 This problem is know to be convex. We used so convex optimization method, further details  
529 about this problem, including the proof of convexity, are addressed in [14].

## 530 A.2 Computation of the partition function using dynamic programming

### 531 A.2.1 Definitions

532 First, we introduce the formal definition of the treewidth, we also depict what is a nice tree  
533 decomposition as it allows a simpler search for the dynamic programming procedure without  
534 additional cost due to the fact that nice tree decomposition have at most a size  $n = |G_T|$

535 ► **Definition 3.** *Tree Decomposition (TD)*

536 Given a graph  $G = (V, E)$ , a tree decomposition of  $G$  is a tree  $T$ , whose nodes are bags  $Y_1 \dots Y_t$   
537 such that: (definition from Bodlander et al [3])

- 538 1.  $V \subset \bigcup_{i=1}^t Y_i$
- 539 2.  $\forall (u, v) \in E, \exists i \in \llbracket 1, t \rrbracket, (u \in Y_i) \cap (v \in Y_i)$
- 540 3.  $\forall u \in V, \{u | u \in Y_i\}$  is a subtree of  $T$ .

541 ► **Definition 4.** *Nice Tree Decomposition*

542 A tree decomposition  $T$  of  $G = (V, E)$  is said “nice” if each bags  $Y_i$  has one of the tree  
543 following form :

544 ■ *Introduce:* Node  $Y_i$  has exactly one child of index  $c$  in  $T$  and  $Y_i = Y_c \cup \{v\}$

545 ■ *Forget:* Node  $Y_i$  has exactly one child of index  $c$  in  $T$  and  $Y_c = Y_i \cup \{v\}$

546 ■ *Join:* Node  $Y_i$  has exactly two children of indices  $c_1$  and  $c_2$  in  $T$  and  $Y_i = Y_{c_1} = Y_{c_2}$

547 ► **Definition 5.** *treewidth (TW)*

The treewidth  $\phi$  of a graph  $G$  is defined as the biggest bag of the “best” tree decomposition of  
548  $G$  :

$$\phi = \min_{tree\ dec. T\ of\ G} \max_{Y_i \in T} |Y_i| - 1$$

549 The tree decomposition is known to give directly a bottom-up order and a dynamic  
550 programming procedure that we can apply on the pattern graph to search motifs [14]. The  
551 only possible bottleneck appear if the treewidth is too high, but we can have a good hope  
552 about the size of this width when it comes to RNA graphs :

552 ■ In absence of pseudoknots, pattern graph are planar.

553 ■ RNA motifs in databases are known to have mostly treewidth below 4.

## 554 A.2.2 Dynamic programming formula

555 We now address the fact to compute the partition function from 2 through a dynamic  
556 programming procedure on the nice tree decomposition of  $G_T$ .

557 It is a bottom-up dynamic procedure (from leaves to the root) that relies on the following  
558 different equations depending on the type of the node  $Y_i$  in the nice tree decomposition  $T$ .  
559 We denote :

560 ■ The set of neighborhood thresholds:  $F = (T^L, T^E, T^G)$

561 ■  $M_i$ , **partial mapping** at node  $Y_i$  of  $T$ .

562 ■ The **separator node** of  $Y_i$ ,  $sep(Y_i)$  chosen as the first element of the set  $S$ :

$$S = \{x \in Y_i | x \notin Y' \text{ with } Y' \text{ a children of } Y_i\}$$

562 We can point out that with a nice tree decomposition, there exists only a unique choice  
563 for this node and above set is reduced to a singleton.

564 ■ Given partial mapping  $M_i$ , a target graph  $G_T$ ,  $\Delta$  designed all the neighborhood differences  
565 partially assigned yet that can be assigned in current bag  $Y_i$  and relative to a given  
566 neighborhood cumulative difference that we design here by its corresponding threshold  
567  $T^F$  by simplicity:

568 We put the following boolean condition  $C(u_1, u_2, Y_i, M_i) =$

569  $(u_1 = sep(Y_i) \cap M_i(u_2) \neq \emptyset) \cup (u_2 = sep(Y_i) \cap M_i(u_1) \neq \emptyset)$

And:

$$\Delta(M_i, G_T, Y_i, T^F) = \{d_{G_T}^F(u_1, u_2, \cdot) | C(u_1, u_2, Y_i, M_i) \text{ is True}\}$$

570 We fill the dynamic programming table  $P$  that stores the partial computation of the  
571 partition function with equations:

Forget Node  $Y_i$  with child  $Y'$ :

$$P[Y_i; M_i] = P[Y'; M_i]$$

Introduction Node of son  $Y'$ :

$$P[Y_i; M_i] = \sum_{v \in V_T} P[Y'; M_i \cup (\text{sep}(Y_i) \leftarrow v)] \\ \times \prod_{T^F \in F} \prod_{d \in \Delta(M_i, G_T, Y_i, T^F)} e^{-\mu w(T^F) \cdot d(M_i \cup (\text{sep}(Y_i) \leftarrow v))}$$

Join Node :

$$P[Y_i; M_i] = \prod_{Y' \in \text{children}(Y_i)} P[Y'; M_i]$$

It can be synthesised on a single equation on all nodes as :

$$P[Y_i; M_i] = \sum_{v \in V_T} \prod_{Y' \in \text{children}(Y_i)} P[Y'; M_i \cup (\text{sep}(Y_i) \leftarrow v)] \\ \times \prod_{T^F \in F} \prod_{d \in \Delta(M_i, G_T, Y_i, F)} e^{-\mu w(T^F) \cdot d(M_i \cup (\text{sep}(Y_i) \leftarrow v))}$$

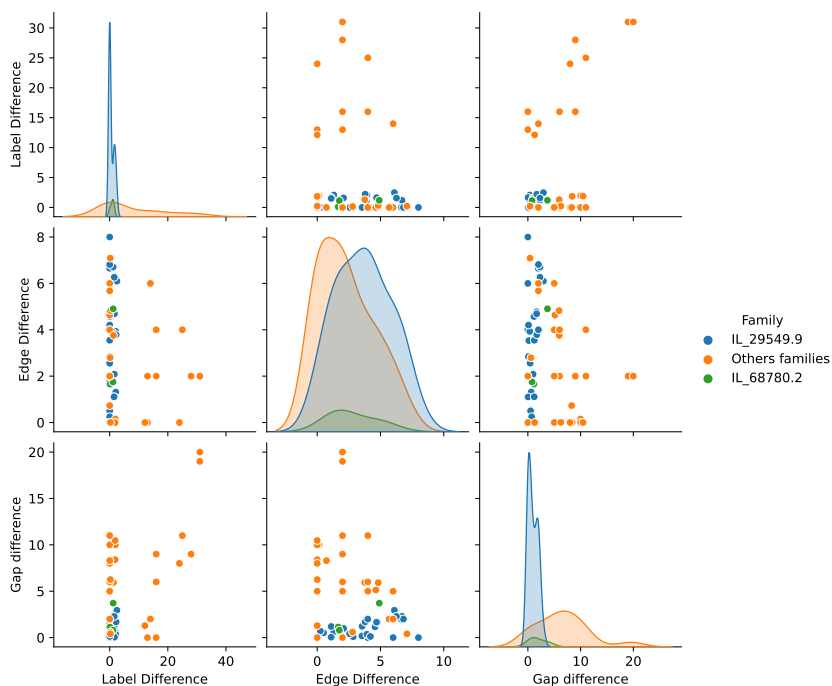
572 The backtracking step to retrieve the value of probability for each graph (and so the  
 573 whole Boltzmann distribution as introduced in 2) use the same type of equations but using  
 574 a procedure that goes from top to bottom: a number is drawn at each node to know if  
 575 we have to add a value for current mapping given the partial partition function that we  
 576 computed at each steps of the forward procedure. Both the forward and backward steps  
 577 are currently known procedures that have been studied and automatised in a framework  
 578 named **Infrared**. [31] We use this framework for our implementation as it allow to be quite  
 579 adjustable with what we named neighborhood cumulative differences and how to define them.  
 580 To ensure the correction of the algorithm, nodes in entry for  $d_{G_T}^F$  functions must be at  
 581 least one time in the same bag in the tree decomposition of  $G_P$ . Here, it is not a problem as,  
 582 by definition of the tree decomposition introduced in part 3, extremities of an edge are at  
 583 least in a same bag.

### 584 A.3 About the cartography and how use it to choose "central" motifs

#### 585 A.3.1 Application in our Kink-Turns case

586 To retrieve the diversity of the Kink-Turn group with our method, we have to request a  
 587 specific motif that will contain all or most of others Kink-Turns in its neighborhood. We will  
 588 call such a motif the **central motif**. In order to choose the best central motif for the given  
 589 family, one can compute from any motif  $O$  the 3 neighborhood cumulative differences between  
 590  $O$  and all the other motifs of the studied RNA group. It gives us a 3D-cartography (one  
 591 dimension for each neighborhood cumulative difference) from an origin  $O$  of all Kink-Turns.  
 592 This cartography allows us to choose more efficiently our neighborhood thresholds. We depict  
 593 first the cartography for the Kink-Turn group from the motif IL\_5TBW\_059 contained in  
 594 the IL\_29549.9 family. The motif can be observed on 2 and the cartography on figure 11.

595 From this cartography, one can observe that a choice of threshold  $T^L = 20$  and  $T^G = 20$   
 596 allows to cover all the Kink-Turns. Nonetheless,  $T^E$  threshold is more difficult to choose.  
 597 Indeed, on the cartography, Edge difference can reach 8. Nonetheless, IL\_5TBW\_059  
 598 contains only 12 edges (2 edges are accounted by bonds as the graph is directed). With  $T^E = 8$ ,  
 599 it means that a motif is recognized even if it contains only 2 bonds from IL\_5TBW\_059,  
 600 which is not acceptable as it will allow to recognize a wild range of patterns that are not Kink-  
 601 Turns. In addition, in terms of performances, incrementing  $T^E$  slows down the algorithm by



■ **Figure 11** Kink-Turn cartography of neighborhood cumulative differences based on IL\_5TBW\_059 origin, member of IL\_29549.9 family.

602 bringing closer the computational and theoretical time complexities. Indeed, allowing one  
 603 additional missing edge means that each point of the algorithm’s current partial mapping  
 604 can be enriched by adding an unmapped edge from  $G_P$  with any couple of nucleotides  
 605 from  $G_T$  ( $i, j$ ) with  $\text{GEO}(i, j) \leq D_{\text{edge}}$ . Thus  $D_{\text{edge}}$  here also acts as a safe guard for good  
 606 performances in practise. As a consequence, we did not allow  $T^E$  to be greater than 4. It is an  
 607 arbitrary choice that can be questioned but allowed until now a good biological significance  
 608 of results and a reasonable time of computation.

609 This cartography also allows us to discuss the way families are clustered in RNA3DAtlas.  
 610 Indeed, on the cartography, motifs of the IL\_29549.9 family reach an Edge difference of 8,  
 611 which is the maximum possible among Kink-Turns, whereas their gap and label differences  
 612 never exceeded 3, way below the maxima. Similarly, if we look at the small IL\_68780.2  
 613 family, we can observe that it has its boundary values in terms of label, edge and gap  
 614 differences included in the boundaries of the IL\_29549.9 family. As a consequence, from  
 615 the point of view of our metrics, there is no way to distinguish between the two families. It  
 616 means that the way we cluster and explore groups of RNA motifs is quite orthogonal to the  
 617 one that was used to build the RNA3DmotifAtlas, due to our focus on base pairs bounds  
 618 themselves instead of atoms in 3D. It gives us good hope to suggest new patterns not easily  
 619 predicted with atomic traditional methods.

### 620 A.3.2 About the creation of the cartography

In order to make the cartography, we have to know how distant the “origin motif” will  
 be from others motifs. To do so our procedure is the following: we extract first all  
 the known motifs from the RNAs and abstract them by relabelling their vertices. It

gave us a set of motifs graphs  $MG$ . We next compute how different every motif is from the origin  $O \in MG$ . Given  $O$  and a motif  $C \in MG$  to compare with, we fixed  $(T^L, T^E, T^G) = (10^{\text{length}(|G_P|^2)}, |G_P|^2, \text{max length } B53 \text{ path})$ . We next solve successively for each neighborhood threshold :

$$\text{argmin}_{T^F} (\text{FuzzTree}(O, C, T^L, T^E, T^G)) \text{ with } F \in \{L, E, G\}$$

621 This optimization step was done using a dichotomy on the value of the neighborhood  
 622 threshold in order to ensure a logarithmic convergence. It is not a huge computation as  
 623 the abstracted motif are of small size in our Kink-Turn case. Order on the neighborhood  
 624 threshold is  $T^E$  then  $T^L$  and then  $T^G$ . Order of  $T^L$  and  $T^G$  is of little importance whereas  
 625  $T^E$  should be proceed first as it include part of others neighborhoods. In particular, too  
 626 small  $T^L$  could force edge to be missing instead of using approximate labels.

627 The cartography can be used on different candidate motifs that can serve as origin  $O$  to  
 628 look which candidate requires the smallest values for  $(T^L, T^E, T^G)$  to delimit the researched  
 629 motifs. At least, if not every of theses thresholds is the smallest at the same time given  
 630 different origins, it permits to select an origin with most of the researched patterns at a  
 631 reasonable neighborhood difference.

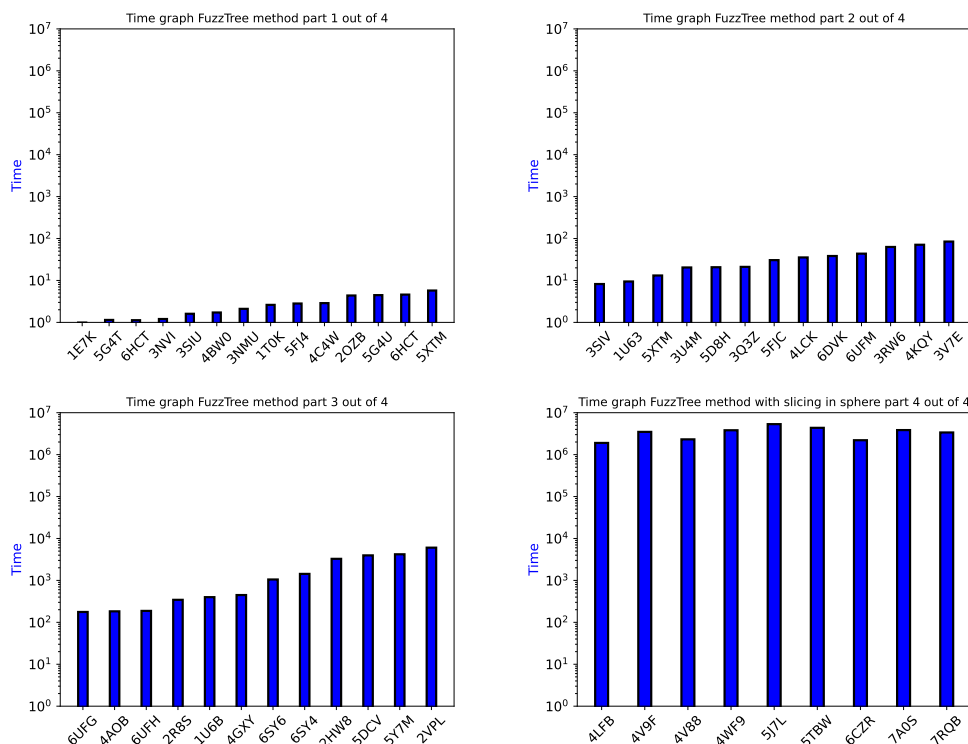
632 In our case, the aim was mainly to reduce the impact of edge difference. However, in  
 633 others RNA or non RNA contexts, missing edge difference can be less relevant and others  
 634 orders for the optimization step of the cartography can be chosen depending on the involved  
 635 metrics. In particular, we propose the cartography as a good way to start the study of a  
 636 dataset different from the Kink-Turn and cartography can be used from our source code.

637 After the cartography is done for different motifs of a dataset, it is not easy task to say  
 638 which one(s) should be the central one(s), here are some guidelines of what we considered:

- 639 ■ We ensure that the motif to choose to be central has no extreme value on some metrics  
 640 in particular concerning the edge difference as it will make impossible to look at these  
 641 motifs.
- 642 ■ We ensure that the motif to choose to be central is a close neighborhood with around  
 643 half of the motifs of the dataset in term of gaps and labels, it was easily the case for  
 644 us as our central motif was a representative of the larger family of Kink-Turns and, on  
 645 RNA3DMotifAtlas, labels do not really vary inside a same family.
- 646 ■ Finally, we should ensure that if a geometry is present in the motif then it is quite  
 647 representative of the dataset, for the Kink-Turn it is the case with a "triangle" of 4  
 648 nucleotides.

#### 649 A.4 Choice on technical parameters

650 For the choice of the radius  $R$  for creating slice of target graph  $G_T$ , given an extracted graph  
 651  $G$  from  $G_T$  centered in nucleotide  $c$ , we first defined  $R(G) = \min_{j \in G} \text{GEO}(j, c)$ . In order to  
 652 be exhaustive with our search, we must ensure that every  $G$  from  $G_T$  is extracted with a  
 653 radius at least equals to  $R(G_P) + D_{\text{gap}}$  as it ensures that we have enough "space" to make  
 654  $G_P$  fit in  $G$  even if some gaps occur. It is due to these gaps that we need to add  $D_{\text{gap}}$  in  $R$ ,  
 655 as it embodies the specific case where the gap would have increased the length of the motif  
 656 to search in  $G_T$  in a single direction by putting gaps one after the others. Due to the rarity  
 657 of this case, we choose, in the tests, to use a smaller radius equals to  $R(G_P) + \frac{D_{\text{gap}}}{4}$ . The only  
 658 taken risk here is to miss some patterns, but it is more convenient to favor time convergence  
 659 as the pathological case on gaps evoked above is not one that we would like to target on.



**Figure 12** Time graph of the FuzzTree method on each group of studied RNA chains. On Narval cluster, computation were done on 1 processor for small RNAs (less than 500 nucleotides, it corresponds to the two first graphs) and on 64 processors for large RNAs (more than 500 nucleotides, it corresponds to the third graphs). In that case, the depicted time is the sum of each time consumed for each processors.

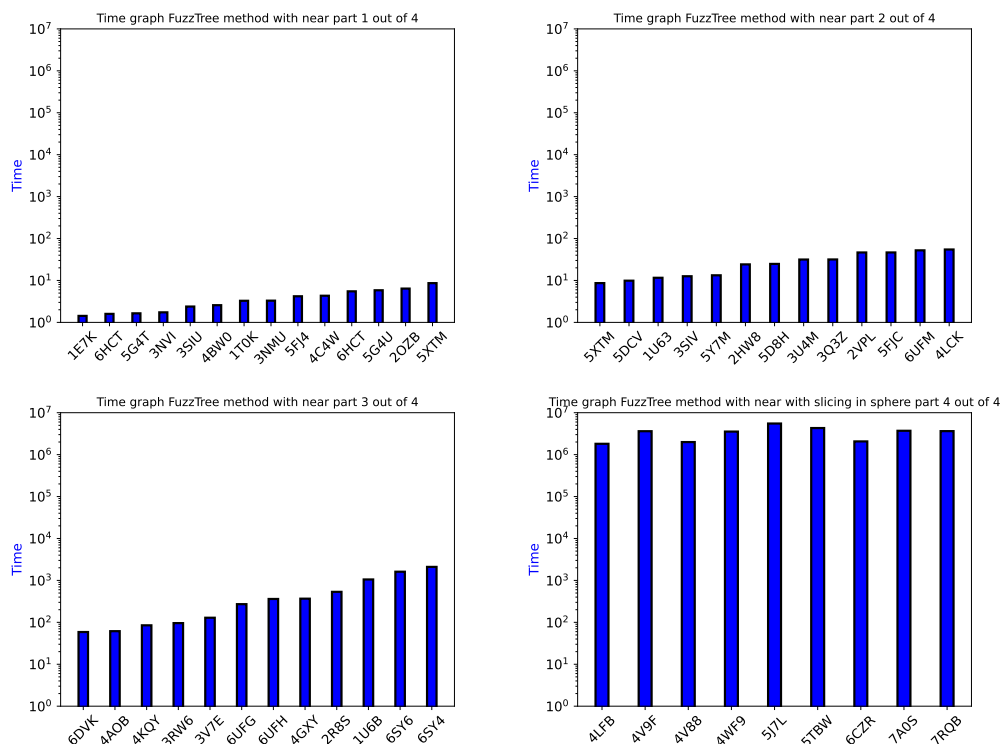
660 We also choose to use a timeout equal to 2000 seconds for the convergence of our algorithm  
 661 on each extracted graphs. Here again, the only risk is to miss some additional patterns.  
 662 Nonetheless, all these limitations only mean that our current results can probably be slightly  
 663 better in term of expressiveness, which means that somebody with more computational  
 664 resources could use this tool and wait for even better performances.

## 665 A.5 Time results on Narval and Beluga clusters for FuzzTree

666 For this paper, computation were done on the Narval cluster and the Beluga cluster of the  
 667 Digital Research Alliance of Canada. Each used node on Narval is made of 64 cores with 2  
 668 CPUs AMD Rome 7532 @ 2.40 GHz. Each used node on Beluga is made of 40 cores with 2  
 669 CPUs Intel Gold 6148 Skylake @ 2.4 GHz. Multiprocessing was used simply by separating  
 670 the computations by chains of a same RNA and next, when relevant, by slices identified in  
 671 these RNA chains.

672 Some time results for computation of the FuzzTree method, by requesting one motif on  
 673 each RNA chains where Kink-Turns are known, are available on figure 13 and 12. Time  
 674 of computation is large but it is something expected with the XP theoretical complexity.  
 675 However, one can notice that in practise the treewidth of the selected pattern is equals to 2  
 676 which allow a complexity in practise in  $O(n^3)$ . No real time discrepancy appear between  
 677 the computation without near edges and the one with. In particular, one should notice that





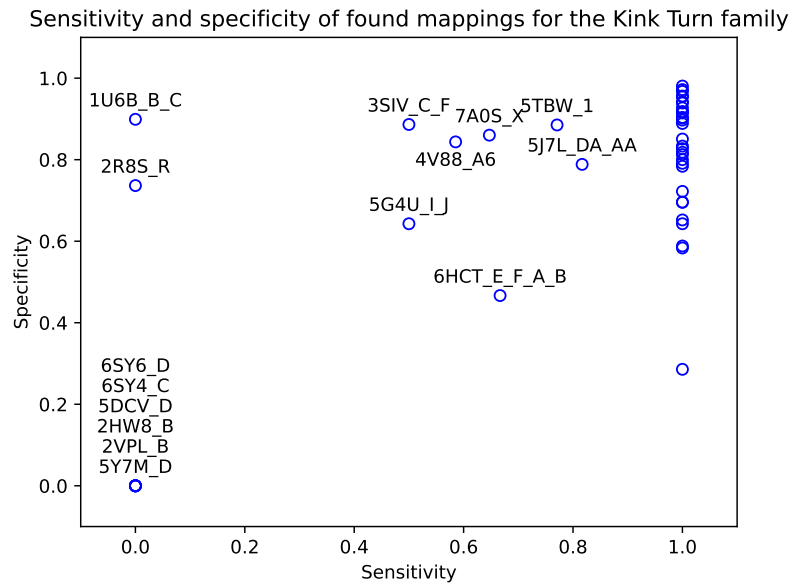
**Figure 13** Time graph of the FuzzTree method on each group of studied RNA chains. On Beluga cluster, computation were done on 1 processor for small RNAs (less than 500 nucleotides, it corresponds to the three first graphs) and on 40 processors for large RNAs (more than 500 nucleotides, it corresponds to the fourth graphs). In that case, the depicted time is the sum of each time consumed for each processors.

678 due to technical restrictions, the used clusters and its underlying hardware changed between  
 679 the two computations. On large graphs, due to the slicing, we are able to reduce the time  
 680 of computation, but not perfectly as slicing computation is still quite redundant: multiple  
 681 graphs cover sometimes the same portion of the Kink-Turn.

## 682 A.6 Results of FuzzTree method on Kink-Turns without the near edges

683 Sensitivity and specificity of our method without near edges on each chains of RNA that  
 684 contains Kink-Turns are depicted on figure 14.

685 With our algorithm without near edges the sensitivity of the nucleotides that was found  
 686 in the Kink-Turn mappings is equals to 1 for 31 RNAs out of 46. This means that from a  
 687 single request, we were able to find two thirds of the Kink-Turns in the RNAs. It is 7 less  
 688 success than the version with near edges. It is important to notice that the specificity for 33  
 689 RNAs out of 46 is also about 0.75 or more like for the benchmark with near. Such results  
 690 confirm the interest of near edges. Indeed, specificity without near edges is not significantly  
 691 better than specificity with near edges. It was not obvious and it emphasizes how relevant  
 692 near edges are from the biological point of view by similarity with others motifs. Near edges  
 693 appear so only in biologically relevant motifs avoiding to generate unwanted noisy motifs.  
 694 As a consequence, there are no specific interest to not take the near into account as it has no  
 695 harmful consequence in term of specificity or time computation. However, as our benchmark



■ **Figure 14** Representation of sensitivity and specificity for each group of chains extracted from a same RNAs without near edges.

696 is only on the Kink-Turn we, cannot have certitude that it is the case for every data set and  
 697 removing near edges can perhaps be a way to gain time on some data sets.