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 approximate instances of an RNA motif, abstracted as a subgraph within a target RNA structure. 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) the introduction of discontinuities in the backbone given close distances in the 3D structure. Our tool relies on a multidimensional Boltzmann sampling, having 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 highlighted 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.2023.20

Related Version *Full Version*: https://hal.science/hal-04094288/

Supplementary Material *Dataset*: https://doi.org/10.5683/SP3/ZR29QE
Software: https://github.com/theoboury/FuzzTree
archived at swh:1:dir:3a4c9bfc1b1ee5432433713486308ce401a97192

Funding *Yann Ponty*: PaRNAssum project from the French Agence Nationale de la Recherche (ANR-19-CE45-0023)
Vladimir Reinharz: NSERC RGPIN-2020-05795, FRQS CBJ1

23rd International Workshop on Algorithms in Bioinformatics (WABI 2023).
Editors: Djamal Belazzougui and Aïda Ouangraoua; Article No. 20; pp. 20:1–20:22
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

The essential regulatory and catalytic roles played by RNAs 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 [20]. 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.

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 [26] computes similarity and finds motifs at the atomic level. It can capture local similarities omitting bulged nucleotides. A drawback of such a method is the computation time, which restrains comparisons between loops. RNA Bricks [6] and RAG3D [34] abstract loops and hairpins as unitary elements. At an intermediate layer, CaRNAval [27, 30] models RNA as graphs where vertices are nucleotides, and edges are the sequence backbone phosphodiester bonds or non-covalent interactions. These interactions can be classified following the Leontis-Westhof (LW) annotations in 12 different geometric families [21, 31]. Such an approach allows specific graph algorithms to discover much larger and more complex modules than by doing atomic computations while retrieving the known structural modules. However, this approach is not 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 to find a match of a pattern graph G_P inside a target graph G_T. In 1995, Alon *et al* [1] proposed an XP [9] algorithm in $O\big(2^{|V_P|}n^{tw(G_P)+1}\big)$ using the color-coding technique. It was shown more generally that only very specific constraints on the input allow having algorithms tractable for bounded treewidths [23]. 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 [23].

However, there can be an exponential number of variants of a specific pattern so different specialized algorithms allowing missing nodes and edges [25, 12], or requiring only labels to be in a neighborhood [18], 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 [24]. Their embedding is based on the nature of base pairs or their isostericity without taking into account gaps or missing edges. By its nature, such a method gives no explicit control over 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 [19] characterized by its sharp bend and clustered into 12 different groups in the RNA3DMotifsAtlas [26]. We show that, from the signature representation of the main subgroup, we sample all their known Kink-Turns in 88% of RNAs. We also retrieve two previously un-annotated loops with a characteristic sharp bend.

2 Method

2.1 RNA as a graph and fundamental problems

We define an RNA structure as a graph G such that its nucleotides are encoded as vertices V, and nucleotide interactions (canonical/non-canonical base pairs, stacking...) are encoded as directed edges $E \subset V \times V$, with labels $L(e)$. Interactions may represent backbone connectivity (phosphodiester bonds), or any of the 12 base-pair types defined by the Leontis-Westhof (LW) nomenclature [31]. Each type specifies an interacting face (Watson-Crick $\circ$, Hoogsteen $\square$, Sugar $\triangleright$) for both nucleotides, along with an orientation cis (filled) or trans (empty). Note that the geometry of the RNA structure is encoded in the edge labels, and our representation does not depend on the sequence. In this work, we are interested in RNA 3D **motifs**, which we abstract as RNA **pattern graphs** as depicted above. We show in Fig. 1 a Kink-Turn motif, represented as a graph with labeled edges.

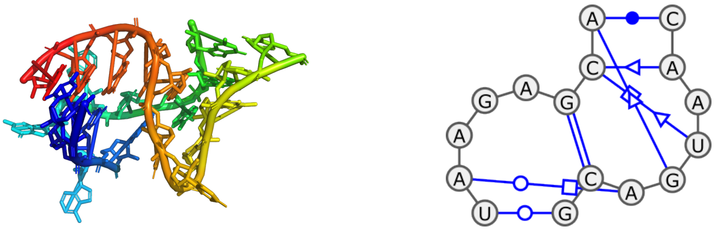


Figure 1 Kink-turn structure. On the left, the 3D structure of a Kink-Turn motif in PDB 3RW6. On the right, its representation as a pattern graph of its base pair interactions. The backbone connections are represented as black arrowed edges.

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

Moreover, since vertices in both pattern and target graphs are indexed by their sequence position, we introduce a precedence relation $\prec$, inducing a strict total order within the pattern and target graphs. A valid occurrence of a pattern within a target must be monotonous, *i.e.* remain consistent with the strict precedence relation $\prec$.

The Monotonous Subgraph Isomorphism (MSI) problem identifies an occurrence of a pattern $G_P = (V_P, E_P)$ inside a target graph $G_T = (V_T, E_T)$. In the context of RNA, G_P is a (closed) motif and $\prec -$Hamiltonian, *i.e.* the total order over V_P induced by the relation $\prec$ represents a (Hamiltonian) path in G_P, while G_T represents an entire RNA structure. Formally, the problem of searching for G_P within G_T can be defined as:

▶ **Problem 1.** *Monotonous Subgraph Isomorphism Problem (MSI)*
Input: *Pattern graph ($\prec -$Hamiltonian) $G_P = (V_P, E_P)$; Target graph $G_T = (V_T, E_T)$*
Output: *Mapping $M : V_P \to V_T$ such that*

- $\forall (u, v) \in {V_P}^2, u \prec v \Rightarrow M(u) \prec M(v)$ ***(monotonicity)***
- $\forall (u, v) \in E_P, (M(u), M(v)) \in E_T \Rightarrow L((u, v)) = L((M(u), M(v)))$ ***(label comp.)***
- $\forall (u, v) \in E_P, (M(u), M(v)) \in E_T$ ***(no missing edge)***

or $\varnothing$ if no such mapping exists.

The MSI problem represents a constrained version of Subgraph Isomorphism, a well-studied NP-complete problem [13, 23] with mildly-depressing prospects regarding parameterized complexity. Indeed, Subgraph Isomorphism does not admit Fixed-Parameter Tractable (FPT) or slicewise polynomial (XP) solutions for various graph parameters, including the treewidths

$tw(G_P)$ and $tw(G_T)$ of the pattern and target graphs. Namely, the problem was shown [23] to be NP-hard even when $\max(tw(G_P), tw(G_T)) \leq 2$ (Para NP-hardness), ruling out the existence of FPT or XP algorithms under standard hypotheses.

The MSI problem retains the classical NP-hardness of Subgraph Isomorphism since it can be shown to generalize the NP-hard structure-sequence alignment in RNA [28]. However, MSI can be solved in time $\mathcal{O}\left(|E_P|.|V_T|^{tw(V_P)+1}\right)$ (XP algorithm) using classic dynamic programming based on a tree decomposition of G_P (see Section 2.5 and Supp. Mat. A.2.2). Such an algorithm has polynomial complexity for any fixed value of the treewidth $tw(G_P)$, a parameter that remains bounded in practice (typically 2 or 3) for RNA motifs.

2.2 Capturing geometric and chemical similarities

We now extend our problem to embrace the natural diversity of RNA motifs in structures. More precisely, we are interested in sampling graph occurrences that are in the geometric neighborhood of a core motif. To do so, we allow the motif to be deformed by three different biologically relevant edit operations detailed below. Each contributes additively and has its own **neighborhood threshold**, and corresponding difference function, as depicted in Table 1:

- T^L represents how much we allow the edge label, the type of the canonical or non-canonical bond, to be modified. It measures the geometric difference between two interactions (see Sec. 2.4.1).
- T^E corresponds to the maximum number of edges/base pairs within the pattern structure that can be omitted (see Sec. 2.4.2).
- T^G is the maximum allowed distance when introducing a backbone discontinuity, a new gap. As insertions alter the distance between bonds, T^G regulates here the maximum sum over these shifts (see Sec. 2.4.3).

We denote by GEO the **geometric distance** between two nucleotides u_1 and u_2 as

$$\mathsf{GEO}(u_1, u_2) = \min_{a_i \in \text{atoms}(u_i) | i \in \{1,2\}} ||a_1 - a_2||_2,$$

and use it to define two additional criteria to constrain admissible solutions:

- First, nucleotides mapped to the nodes of a missing edge must be closer than D_{edge} Å;
- Second, we enforce a maximal distance D_{gap} between the nucleotide on both sides of an introduced gap. These values correspond to the phosphodiester atoms' distances between the nucleotides. Capping these distances beyond a fixed value not only yields more realistic outputs but also greatly improves the runtime of our algorithm.

We use the **isodiscrepancy index** [31] to quantify geometrically the difference between base pair families and provide values measuring three terms: (1) the difference of intra-base pair C1'–C1' distances; (2) after aligning one base, the inter-base pair C1'–C1' distance between the C1' atoms of the second bases of the base pairs; (3) The angle on an axis perpendicular to the base pair plane required to superpose the second bases. This isostericity measure is defined for pairs of base pairing families (BPF), each representing one of the 12 canonical and non-canonical conformations and named as $\mathsf{BPF}_i, \forall i \in [\![1, 12]\!]$. Inter-family variations are frequent and therefore the average isodiscrepancy of a family to itself is not 0. To correct for this phenomenon, we define the ISO difference between two families as:

$$\mathsf{ISO}(\mathsf{BPF}_i, \mathsf{BPF}_j) = \mathsf{isodiscrepancy}(\mathsf{BPF}_i, \mathsf{BPF}_j) - \mathsf{isodiscrepancy}(\mathsf{BPF}_i, \mathsf{BPF}_i).$$

Moreover, we set the value of ISO to 0 involving undefined labels, backbones or phantom interactions.

We define a **backbone path** as a sequence of at least 2 nucleotides connected through backbone edges.

The set P of paths associated with a target graph $G_T = \big(V_T, E_T = B_T \sqcup \overline{B}_T\big)$ is defined as:

$$P = \bigcup_{k\in\mathbb{N}, k\geqslant 2} \{(p_0, ..., p_k) \mid \forall i \in [\![0, k-1]\!], (p_i, p_{i+1}) \in B_T\}.$$

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 a solution to a problem that we call the **Fuzzy Monotonous Subgraph Isomorphism problem (FMSI)**, which can be defined as:

▶ **Problem 2.** *Fuzzy Monotonous Subgraph Isomorphism problem (FMSI)*
Input: *Pattern graph* $G_P = \big(V_P, E_P = B_P \sqcup \overline{B}_P\big)$ *($\prec$ −Hamiltonian) , target graph* $G_T = \big(V_T, E_T = B_T \sqcup \overline{B}_T\big)$ *and neighborhood thresholds (T^L, T^E, T^G, D_{edge}, D_{gap})*
Output: *Mapping* $M : V_P \to V_T$ *such that:*

1. $\forall (u, v) \in {V_P}^2, u \prec v \Rightarrow M(u) \prec M(v)$ ***(monotonicity)***
2. $\sum_{(u,v)\in\overline{B}_P} \mathit{ISO}(L(u, v), L(M(u), M(v))) \leqslant T^L$ ***(label compatibility)***
3. $\sum_{(u,v)\in\overline{B}_P} 1 - \mathbb{1}_{(M(u),M(v))\in\overline{B}_T} \leqslant T^E$ ***(few missing edges)***
4. $\forall (u, v) \in \overline{B}_P, (M(u), M(v)) \notin \overline{B}_T, \mathit{GEO}(M(u), M(v)) \leqslant D_{edge}$ ***(edge distance limit)***
5. $\sum_{(p_0,...,p_k)\in P, k\geqslant 3} \mathit{GEO}(p_0, p_k) \leqslant T^G$ ***(path size limitation)***
6. $\forall (u, v) \in B_P, \exists (p_0, p_1, p_2, ..., p_k) \in P$ *such that* ***(no missing backbone path)***
   - $p_0 = M(u), p_k = M(v)$ ***(*)***
   - $\mathit{GEO}(p_0, p_k) \leqslant D_{gap}$ ***(**)***

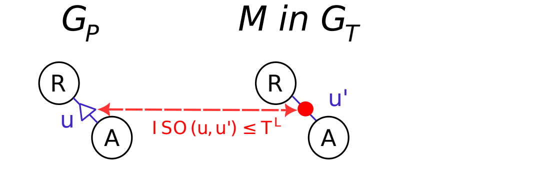
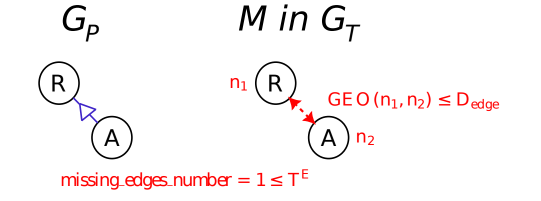
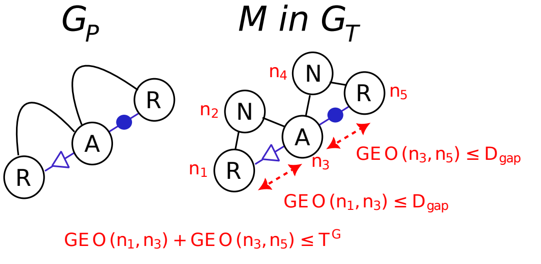
or ∅ if no such mapping exists.

Intuitively, a valid mapping M has to respect the six following conditions: The **monotonicity** condition enforces pattern nodes to map successive nodes in the target. The **label compatibility** controls how much the geometric differences cumulative is allowed between pattern and matched edges (see Sec. 2.4.1). The **few missing edges** constraint ensures that pattern edges that are not mapped to an edge in the target are not numerous. (see Sec. 2.4.2) The **edge missing limit** forces each couple of mapped nodes with no edges to have a bounded geometric distance between each other. (see Sec. 2.4.2) The **path size limitation** controls how large the cumulative of gaps geometric lengths can be. (see Sec. 2.4.3) The **no missing backbone path** condition (as unfolded in Prob. 2) ensures that the start and end points of a path are mapped nodes **(*)**. It also restrained allowed geometric length of individual path **(**)**. (see Sec. 2.4.3) We note that due to the monotonicity condition, it implies that no target node in $p_1, \ldots, p_{k-1}$ can belong to the mapping.

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

In practice, RNA graphs are fully ordered but do not necessarily contain a Hamiltonian path due to backbone disconnections, leading to a graph composed of multiple strands. We can reconstitute a Hamiltonian path (with no complexity overhead) in the pattern graph by adding some "phantom edges" (with a specific label) when the backbone is missing which correspond to the set of edges $\{(i, i+1) \mid i \in G_P, (i, i+1) \notin E_P \cup L(i, j) \neq \text{"B53"}\}$. Additionally, to ensure that such edge can be mapped in the target G_T in a way that will conserve the monotonicity of the mapping, we add in G_T the set of edges $\{(i, j) \mid (i, j) \in G_T, i \prec j \cap L(i, j) \neq \text{"B53"}\}$.

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

Threshold T^F	Difference d^F	Fuzzy mapping M of G_P found in G_T
T^L	Isostericity ISO	 G_P M in G_T R A u R A u' ISO(u, u') ≤ T^L
T^E	Missing edges number	 G_P M in G_T R A n_1 R A n_2 GEO(n_1, n_2) ≤ D_{edge} missing_edges_number = 1 ≤ T^E
T^G	Geometric GEO from 3D structure	 G_P M in G_T R A R n_4 N R n_5 n_2 N A n_3 n_1 R GEO(n_3, n_5) ≤ D_{gap} GEO(n_1, n_3) ≤ D_{gap} GEO(n_1, n_3) + GEO(n_3, n_5) ≤ T^G

2.3 Locating alternative occurrences through sampling

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 of a few nucleotides apart will often be found. It is relevant in terms of neighborhoods, but, from the biological standpoint, 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, 15].

Generally, a **Boltzmann distribution** is such that the probability of any possible outcome G depends on its (pseudo-)**energy** E:

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

where β 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

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

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

In our case, an outcome is a graph $G \subset G_T$, such as G is the image of mapping M and we have 3 features, one for each neighborhood. Given a specific neighborhood threshold T^F, its relative feature F measures how much the weight of edits D^F relative to neighborhoods, further introduced as a difference in 2.4, deviate from a given center T^{F*}. For instance, T^{F*} can be chosen as equal to 0 if we want to sample mostly G with no fuzziness or as equal to $T^F/2$ if we want to sample them with average fuzziness. More details on this choice and about Boltzmann sampling are available at Supp. Mat. A.1. MBD is well-suited to the sampling that we want to make: the exponential decrease of the probability with the features gives low probabilities to the graphs that are far in terms of neighborhoods from G_P, which allows us to characterize well neighborhood$_{G_P}(G_T)$. In particular, we can define F such that it takes a value equal to $+\infty$ when the corresponding neighborhood threshold T^F, for a mapping M, is not respected, forbidding simply M to be sampled. Additionally, the Multidimensional character of the distribution allows us to take into account the 3 neighborhoods on labels, edges and gaps at the same time.

A general framework called `InfraRed` [33], initially introduced in the context of RNA design [15], can be used to generate efficiently, in a parameterized manner, the MBD. It automatically processes constraints and elements of the scoring into a graph, decomposes it into a Tree Decomposition, and generates automatically the bottom-up dynamic programming sampling procedures. More details on the Tree Decomposition and the dynamic programming used in `InfraRed` can be found in Supp. Mat. A.2.

2.4 Neighborhood difference description

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

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

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

▶ **Definition 1** (Neighborhood cumulative difference / neighborhood difference). *Given a pattern graph $G_P = \big(V_P, E_P = B_P \sqcup \overline{B}_P\}\big)$, a target graph $G_T = \big(V_T, E_T = B_T \sqcup \overline{B}_T\big)$ and a mapping M, a neighborhood cumulative difference is a function D^F relatively to a neighborhood threshold T^F that act as a wrapper around $d^F_{G_T}$:*

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

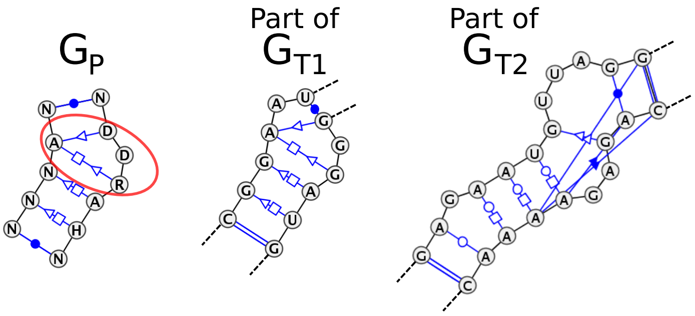


Figure 2 Kink-turn signature and targets. On the left, signature graph of the Kink-Turn IL_29549.9 family and our search pattern. In the middle and on the left, mappings that were missed during the search for the pattern. G_{T1} due to the same nucleotide merging the end of a cSS and a cWW. G_{T2} due to its too large difference.

where $d^F_{G_T}(u, v, M)$ is the ***neighborhood difference*** *relative to G_T, a function that measures, relatively to F, how "different" are the edges in the pattern ($(u, v) \in G_P$) from the edges in the mapping ($(M(u), M(v)) \in G_T$).*

How the difference is measured depends on the feature as described below.

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

2.4.1 The label difference

The label difference, as represented in Fig. 3, accounts for the difference between base pairs families and we use for that the isodiscrepancy [31] as introduced in part 2.2. 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 $d^L_{G_T}$ equals to:

$$d^L_{G_T}(u, v, M) = \mathsf{ISO}(L(u, v), L(M(u), M(v))).$$

2.4.2 The edge difference

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

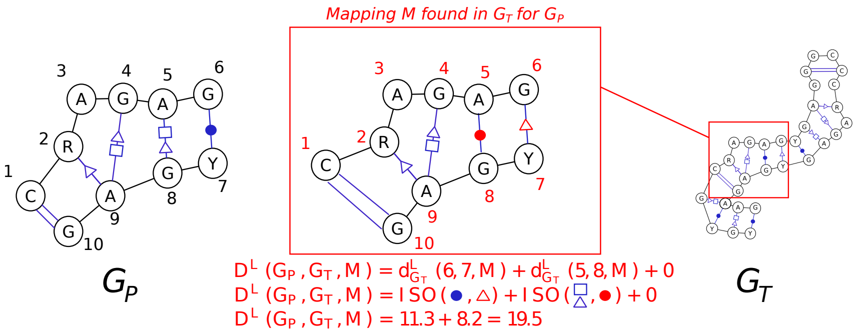

Mapping M found in G_T for G_P
G_P
G_T
D^L (G_P, G_T, M) = $d^L_{G_T}$ (6, 7, M) + $d^L_{G_T}$ (5, 8, M) + 0
D^L (G_P, G_T, M) = ISO(●, △) + ISO(□△, ●) + 0
D^L (G_P, G_T, M) = 11.3 + 8.2 = 19.5

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 the geometric difference between bounds as described in Stombaugh et al [31].

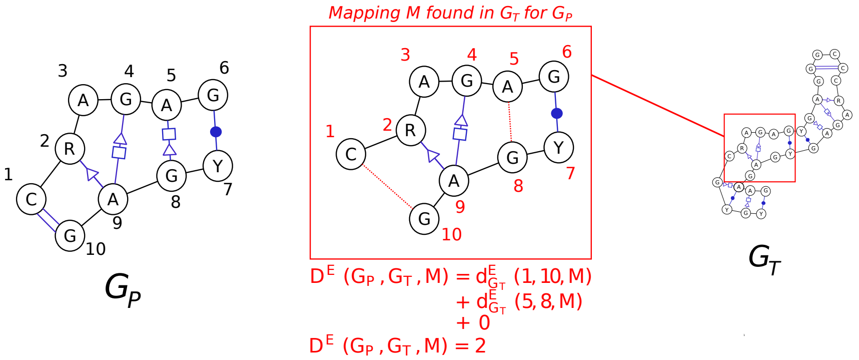

Mapping M found in G_T for G_P
G_P
G_T
D^E (G_P, G_T, M) = $d^E_{G_T}$ (1, 10, M)
+ $d^E_{G_T}$ (5, 8, M)
+ 0
D^E (G_P, G_T, M) = 2

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 $D_{\text{edge}} >> max(\mathsf{GEO}(1,10), \mathsf{GEO}(5,8))$.

Therefore, we will accept mappings of the extremities of an edge in the pattern to nodes u, v that are at most at a set threshold distance D_{edge} computed from the 3D structure (*i.e.* $\mathsf{GEO}(u, v) < D_{edge}$). Setting a weight of ∞ to mappings outside the threshold allows the sampling to simply reject such instances. We additionally use the edge difference to reject cases where backbones are mapped to couples of nodes that are not backbones by putting a weight ∞ in that case. The total edge difference D^E relative to neighborhood threshold T^E, is a neighborhood cumulative difference entirely defined by the sum over $d^E_{G_T}$ with values defined as followed and shown in Fig. 4:

$$d^E_{G_T}(u, v, M) = \begin{cases} 0 & \text{if } (u, v) \in B_P \cap (M(u), M(v)) \in B_T \\ & \text{or } (u, v) \in \overline{B}_P \cap (M(u), M(v)) \in \overline{B}_T \\ 1 & \text{if } (u, v) \in \overline{B}_P \cap (M(u), M(v)) \notin \overline{B}_T \\ & \text{and } \mathsf{GEO}(M(u), M(v)) \leqslant D_{\text{edge}} \\ \infty & \text{otherwise.} \end{cases}$$

2.4.3 The gap difference

A frequent type of natural variability in a motif family is the occurrence of bulging out nucleotides in what would be a continuous sequence in the pattern. These insertions can be of different sizes, but we require that they do not modify (too much) the local structure. To

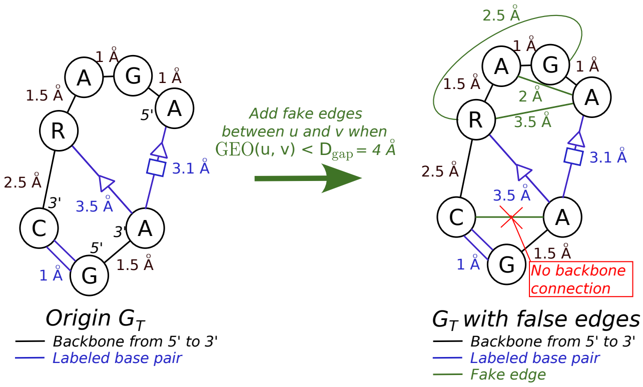


Figure 5 Fake edges. Addition of fake edges to account for gaps. Fake edges are added only when distance is below D_{gap} and when both nucleotides are fully connected by backbone edges. For instance here, we add no fake edge between C and A at the bottom of G_T as these two nucleotides are not connected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

take arbitrary insertions into account, we introduce **fake edges** between any two nucleotides present on the same backbone that are at a distance below D_{gap}. An illustration of this process is shown in Fig. 5. For convenience, these edges are added in B_T to keep valid the cases of the edge difference where backbones are wrongly mapped.

An additional difference compared to the missing interaction edges of the previous section is how we sum the total neighborhood difference D^G. We accumulate the total physical distance (*i.e.* GEO) between the nodes connected through the fake edges. This allows an arbitrarily large structure to bulge out without the need to verify or specify admissible lengths, as long as the nucleotides around this inserted gap are close geometrically as illustrated in 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 $d^G_{G_T}$:

$$d^G_{G_T}(u, v, M) = \begin{cases} \mathsf{GEO}(M(u), M(v)) & \text{if } (M(u), M(v)) \text{ is} \\ & \text{a “Fake Edge” in } E_T \\ 0 & \text{otherwise.} \end{cases}$$

A limitation of this approach is that we cannot detect the deletion of nodes from the pattern. A workaround is to remove all the nodes in the pattern graph that do not directly participate in a base pair interaction, and reconnect the disconnected backbones. Using the new pattern with a large gap threshold T^G would allow us to retrieve the original motif neighborhood efficiently, but introduce more spurious matches.

2.5 Algorithm and complexity

Our method is based on `Infrared` [15, 33], a declarative framework that automatically generates a dynamic programming procedure for MBD sampling, based on a nice tree decomposition (TD). The dynamic programming procedure used in Infrared is described in Supp. Mat. A.2. It precomputes the partition function of the MBD through a bottom-up recursion and uses local contributions to perform an exact sampling within the MBD distribution. Within this framework, a combinatorial problem is abstracted as a set of

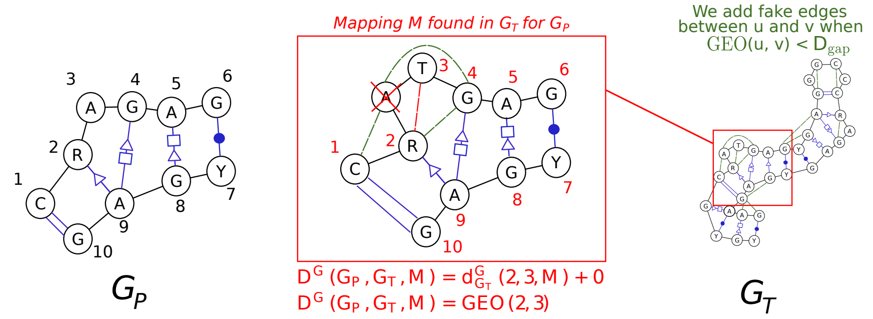

Mapping M found in G_T for G_P
We add fake edges between u and v when GEO(u, v) < D_gap
3
4
5
6
2
1
7
8
9
10
A
G
A
G
R
C
A
G
Y
G
T
G_P
$D^G(G_P, G_T, M) = d^G_{G_T}(2, 3, M) + 0$
$D^G(G_P, G_T, M) = GEO(2, 3)$
G_T

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

variables $\{X_i\}_i$, each assigned an integer value within a bounded domain. Assignments must respect various constraints expressed as functions $\{C_i\}_i$, each defined over a subset of variables. Similarly, feature functions $\{F_j\}_j$ associate real-valued contributions to subsets of variables, and are summed to represent the pseudo-energy of an assignment.

In this setting, we abstract each node i of the graph pattern G_P as a variable X_i, taking value in $[\![1, n]\!]$. The value of X_i represents the mapping of node i in the graph $G_T = (V_T, E_T)$ with $|V_P| = k$ and $|V_T| = n$. Within RNA motifs, the number of partners of a position is bounded, so we have $|E_P| \in \mathcal{O}(k)$. Remark that all deviations from the pattern defined in Sections 2.4.1 through 2.4.3, can be expressed *locally* as sums on the edges of the pattern graph. It follows that the dependencies *dep* implied by our cumulative differences are only binary, and restricted to pairs sharing an edge in G_P: $dep = \{(X_i, X_j) \mid (i, j) \in E_P\}$. The graph of constraints is thus reducible to the input pattern graph G_P, as shown in Fig. 8.

Due to the neighborhood threshold T^F being a global property over the mapping, the sampling is followed by a rejection step for samples that exceed a neighborhood threshold. An example of such rejection is depicted in Fig. 7. Asymptotically, such rejection will at worst induce a constant overhead with T^F chosen independently from $|G_P|$ and $|G_T|$.

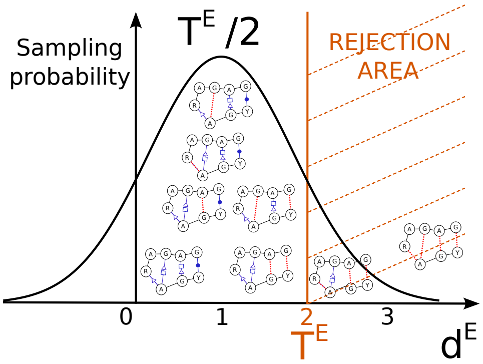


Figure 7 Rejection step. In the above example, rejection is depicted only for the edge neighborhood for the sake of simplicity. Found motifs above T^E thresholds are rejected afterward. Found motifs with an edge difference close to $\frac{T^E}{2} = 1$ here have more chance to be sampled.

▶ **Proposition 2.** *A generation of t Boltzmann-distributed* (1) *putative solutions to FMSI can be performed in time $\mathcal{O}\big(n\,k\,t + k\,n^{(\phi+1)}\big)$ where ϕ is the treewidth of the pattern G_P.*

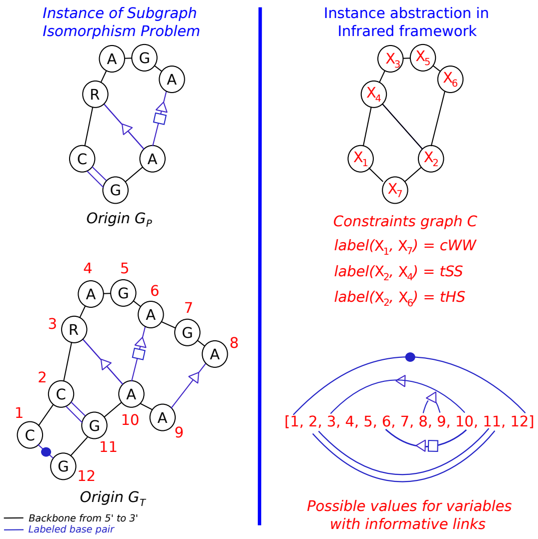

Instance of Subgraph Isomorphism Problem
A
G
A
R
C
A
G
Origin G_P
Instance abstraction in Infrared framework
X_3
X_5
X_6
X_4
X_1
X_2
X_7
Constraints graph C
label(X_1, X_7) = cWW
label(X_2, X_4) = tSS
label(X_2, X_6) = tHS
4
5
6
7
8
3
2
1
10
9
11
12
Origin G_T
Backbone from 5' to 3'
Labeled base pair
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
Possible values for variables with informative links

Figure 8 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.

This complexity directly follows from the complexity of the algorithm [15] underlying `Infrared` for a graph $G_P = (V_P, E_P)$ (with $|V_P| = k$). Restricted to binary constraints/features associated with (a subset of) E, the computation of the partition function can be performed in time $\mathcal{O}((|E_P| + |V_P|) \times \Delta^{\phi+1})$, where Δ is the size of the assignment domain for individual variables, and ϕ is the treewidth of G_P. A stochastic backtrack follows, leading to the generation of t Boltzmann-distributed assignments in time $\mathcal{O}(|V_P|\,\Delta\,t)$. The complexity stated above is obtained by observing that $|E_P| \in \Theta(k)$, and that $\Delta \in \Theta(n)$.

We conclude by noting that preprocessing, including computations of geometrical distances and augmentation of G_T graph, can be performed once, in $O(n^2)$ time and space, leading to a negligible overhead in comparison to the computation of the partition function. Meanwhile, an optimal tree decomposition can be theoretically obtained in time only super polynomial in ϕ [3].

A summary of the complexity and capacity of our FuzzTree method is depicted in Table 2. Regarding the parameterized complexity [9], the FuzzTree method is XP in the treewidth of the pattern graph, both in time and in space. It represents progress compared to VF2 [7], which is indeed implemented and efficient in practice due to the profusion of lookahead rules but has a worst-case time complexity similar to $O(n^n)$. In practice, VF2 becomes costly with dense graphs, even in its most modern versions [4, 17]. Furthermore, we compete with the bound from the Color-Coding [1] technique by improving it in time and space. $2^{O(k)}$ is replaced by $k \leqslant n$ in our bounds, which allows us to get rid of k as a parameter to restrict it simply to the treewidth in our RNA case.

In addition, our method handles at the same time multiple labels on edges, directed graphs and can integrate node labels. The latter has not been implemented but can be added, as with labels on edges, without complexity overhead.

Table 2 Complexities for RNA motif search. Comparison of state-of-the-art methods for RNA motif search. With $\phi = tw(G_P)$, $n = |V_T|$, $k = |V_P|$ and t the number of samples.

Method Name	Color-Coding [1]	VF2 [7]	VeRNAl [24]	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	$O\big(knt + kn^{\phi+1}\big)$
Space complexity	$2^{O(k)} n^{\phi+1}$	$O(n)$	Exponential	$O\big(n^{\phi+1}\big)$
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

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 nucleotide c at a given radius R from c. For an extracted graph G, centered on c, we have:

$$\forall j \in G, R(G) = \mathsf{GEO}(j, c) \leqslant R.$$

Choices of technical parameters, such as the value for R, hardware and computation times are discussed in Supp. Mat. A.3. For the sake of efficiency, we refrained from adding “phantom edges” described in Section 2.2. Doing so enables possible violations of the monotonicity, leading to the detection of motif occurrences in the context of a more remote homology, but necessitated a further round of rejection (whose impact on performances remained negligible).

3.2 Data: the Kink-Turn motifs family

All interactions in the RNA structures are provided by FR3D [29]. 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 [19]. The Kink-Turn has been shown to appear in multitudes of contexts through computational and experimental methods [16, 22]. As of January 2023, there were 72 instances of the Kink-Turn RNA annotated in the RNA3DMotifAtlas [26]. One was omitted because it was not annotated on the main structure but one of its symmetric alternatives. The others span 46 different RNAs and are divided into 12 different families with different lengths, between 9 and 23 nucleotides and base pair signature. Members of the same family also differ in terms 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.

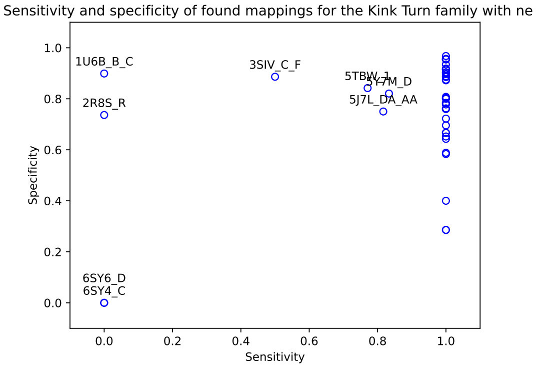


Figure 9 Sensitivity and Specificity of regions corresponding to sampled graphs in the 46 RNA structures containing Kink-Turns. Each dot represents an RNA chain, where one or multiple Kink-Turns can be found. To keep track of them, nodes whose sensitivity is not equal to one, are named of the graph “RNAname“_“chain”.

Empirically, RNA 3D motifs are small motifs that, despite not being tree-like, have relatively small treewidth. It is especially the case for the Kink-Turn family, where 50 Kink-Turns pattern graphs have treewidth equal to 2 and 21 have treewidth equal to 3, which makes our parameterization in treewidth practically quite relevant.

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 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.

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

Parameter	T^L	T^E	T^G	D_{edge}	D_{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	$[0, 50]$	$[\![0, 6]\!]$	$[0, 50]$	$[5, 10]$	$[5, 20]$	$R(G_P) + [\frac{D_{\text{gap}}}{4}, D_{\text{gap}}]$	

Our sampling returns sub-graphs of the target graphs G_T. Using a python implementation of VF2 [14, 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 found by our method. The True Negative (TN) are those that do not cover a Kink-Turn, rightly not found by our method. P designs the set of all Kink-Turn motifs and N the set of all other motifs. We show the sensitivity (TP/P) and specificity (TN/N) per RNA structure in Fig. 9.

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 into 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 in 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.

3.2.2 Other identified regions

An additional 198 locations in the 46 RNAs were identified. The Kink-Turn is essentially an internal loop motif. We investigate if other internal loops sharing the same main 3D feature, a sharp bend in an interior loop, are found. Using the python library forgi [32] we decomposed these regions in their secondary structure elements. The majority, 125, mapped to regions forming multiloops. A total of 33 were covering continuous double-stranded regions. The angles of surrounding stems for each interior loop in the 46 RNAs (in blue) the identified Kink-Turns in these RNAs (red) and the other 33 elements (in green) are shown in Fig. 10.

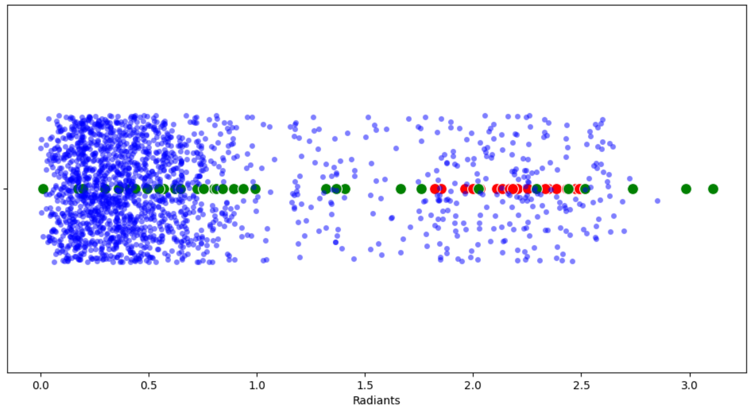


Figure 10 Angles in radiants. In blue for stems around every interior loop in the 46 RNAs. In red for the Kink-Turns identified in these RNAs. In green for the additional 33 continuous double-stranded regions.

There are 10 additional regions with angles above 1.4rad, and two of these had a sharp turn in their structure in un-annotated region as seen in Fig. 11. We show below their graph of interactions, with the cross-strand stackings in orange.

The first is in 5J7L chain DA and positions 78–86, 96–108. It overlaps an un-annotated motif (IL_85931.1) that covers positions 81–85, 97–101, and 103–105. The second is located in 7RQB, chain 1A, positions 2129–2138, 2153–2160, and is not covered or surrounded by any annotated motif.

4 Conclusion

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

Two main limitations are inherent to our approach. Due to the intrinsic nature of sampling, we cannot be assured that all neighboring graphs will be reported. In itself, for large patterns, this is a feature since sampling allows uniform exploration of the exponentially

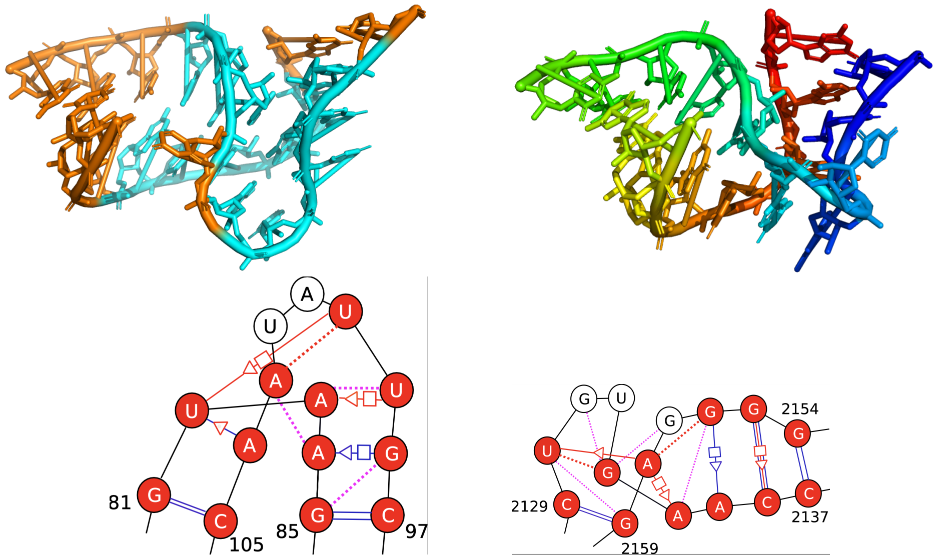


Figure 11 Other matches. 5J7L on the left and 7RQB on the right. The 3D structure on the left has IL_85931.1 highlighted in cyan, on the right each nucleotide is colored independently. In the graphs, red nodes are matched with the pattern. Blue edges are in the RNA structure and red ones are in the pattern, indicating modifications and removal. Red dashed lines are introduced “Fake edges”. Magenta dashed lines indicate stackings.

growing neighborhood. By enabling per-feature biases, FuzzTree can also be calibrated to favor the sampling of graphs at a desired location in the neighborhood to favor specific types of variants (e.g., isostericity of modified edges). Letting the sampling run for longer will also mitigate the problem. More importantly, some patterns cannot be identified, particularly if an LW interaction is replaced by a backbone connection. While such cases are rare, they do exist, and additional improvement will be needed to capture them.

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

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

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

References

1 Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *J. ACM*, 42(4):844–856, July 1995. doi:10.1145/210332.210337.

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

3 Hans L. Bodlaender and Arie M. C. A. Koster. Combinatorial optimization on graphs of bounded treewidth. *The Computer Journal*, 51(3):255–269, 2008. doi:10.1093/comjnl/bxm037.

4 Vincenzo Carletti, Pasquale Foggia, Alessia Saggese, and Mario Vento. Introducing vf3: A new algorithm for subgraph isomorphism. In Pasquale Foggia, Cheng-Lin Liu, and Mario Vento, editors, *Graph-Based Representations in Pattern Recognition*, pages 128–139, Cham, 2017. Springer International Publishing.

5 Thomas R Cech and Joan A Steitz. The noncoding RNA revolution – Trashing old rules to forge new ones. *Cell*, 157(1):77–94, 2014.

6 G Chojnowski, T Waleń, and JM Bujnicki. RNA Bricks – A database of RNA 3D motifs and their interactions. *Nucleic Acids Research*, 42, 2013. doi:10.1093/nar/gkt1084.

7 Luigi Cordella, Pasquale Foggia, Carlo Sansone, and Mario Vento. A (sub)graph isomorphism algorithm for matching large graphs. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26:1367–1372, November 2004. doi:10.1109/TPAMI.2004.75.

8 José Almeida Cruz and Eric Westhof. The dynamic landscapes of RNA architecture. *Cell*, 136(4):604–609, 2009.

9 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Daniel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2016.

10 Rhiju Das, Rachael C Kretsch, Adam J Simpkin, Thomas Mulvaney, Phillip Pham, Ramya Rangan, Fan Bu, Ronan Keegan, Maya Topf, Daniel Rigden, et al. Assessment of three-dimensional RNA structure prediction in CASP15. *bioRxiv*, pages 2023–04, 2023.

11 Sven Findeiß, Christoph Flamm, and Yann Ponty. Rational Design of RiboNucleic Acids (Dagstuhl Seminar 22381). *Dagstuhl Reports*, 12(9):121–149, 2023. doi:10.4230/DagRep.12.9.121.

12 Nagoor Gani. 63. isomorphism on fuzzy graphs. *International Journal of Computational and Mathematical Sciences*, Vol. 2:200–206, January 2008. doi:10.13140/2.1.1873.9847.

13 M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

14 Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using NetworkX. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

15 S. Hammer, W. Wang, S. Will, and Y. Ponty. Fixed-parameter tractable sampling for rna design with multiple target structures. *BMC Bioinformatics*, 2019.

16 Lin Huang and David MJ Lilley. The kink turn, a key architectural element in RNA structure. *Journal of molecular biology*, 428(5):790–801, 2016.

17 Alpár Jüttner and Péter Madarasi. Vf2++ – An improved subgraph isomorphism algorithm. *Discrete Applied Mathematics*, 242:69–81, 2018. Computational Advances in Combinatorial Optimization. doi:10.1016/j.dam.2018.02.018.

18 Arijit Khan, Nan Li, Xifeng Yan, Ziyu Guan, Supriyo Chakraborty, and Shu Tao. Neighborhood based fast graph search in large networks. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, SIGMOD '11, pages 901–912, New York, NY, USA, 2011. Association for Computing Machinery. doi:10.1145/1989323.1989418.

19 Daniel J Klein, T Martin Schmeing, Peter B Moore, and Thomas A Steitz. The kink-turn: a new RNA secondary structure motif. *The EMBO journal*, 20(15):4214–4221, 2001.

20 Neocles B Leontis, Aurelie Lescoute, and Eric Westhof. The building blocks and motifs of RNA architecture. *Current Opinion in Structural Biology*, 16(3):279–287, 2006. Nucleic acids/Sequences and topology. doi:10.1016/j.sbi.2006.05.009.

21 Neocles B Leontis and Eric Westhof. Geometric nomenclature and classification of RNA base pairs. *Rna*, 7(4):499–512, 2001.

22 Bin Li, Shurong Liu, Wujian Zheng, Anrui Liu, Peng Yu, Di Wu, Jie Zhou, Ping Zhang, Chang Liu, Qiao Lin, et al. RIP-PEN-seq identifies a class of kink-turn RNAs as splicing regulators. *Nature Biotechnology*, pages 1–13, 2023.

23 Dániel Marx and Michal Pilipczuk. Everything you always wanted to know about the parameterized complexity of Subgraph Isomorphism (but were afraid to ask). In Ernst W. Mayr and Natacha Portier, editors, *31st International Symposium on Theoretical Aspects of Computer Science (STACS 2014)*, volume 25 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 542–553, Dagstuhl, Germany, 2014. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.STACS.2014.542.

24 Carlos Oliver, Vincent Mallet, Pericles Philippopoulos, William L Hamilton, and Jérôme Waldispühl. Vernal: a tool for mining fuzzy network motifs in RNA. *Bioinformatics*, 38(4):970–976, November 2021. doi:10.1093/bioinformatics/btab768.

25 Aymeric Perchant and Isabelle Bloch. Fuzzy morphisms between graphs. *Fuzzy Sets and Systems*, 128(2):149–168, 2002. doi:10.1016/S0165-0114(01)00131-2.

26 Anton I. Petrov, Craig L. Zirbel, and Neocles B. Leontis. Automated classification of rna 3d motifs and the rna 3d motif atlas. *RNA*, 2013.

27 Vladimir Reinharz, Antoine Soulé, Eric Westhof, Jérôme Waldispühl, and Alain Denise. Mining for recurrent long-range interactions in RNA structures reveals embedded hierarchies in network families. *Nucleic Acids Research*, 46(8):3841–3851, March 2018. doi:10.1093/nar/gky197.

28 Philippe Rinaudo, Yann Ponty, Dominique Barth, and Alain Denise. Tree decomposition and parameterized algorithms for rna structure-sequence alignment including tertiary interactions and pseudoknots. In Ben Raphael and Jijun Tang, editors, *Algorithms in Bioinformatics*, pages 149–164, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

29 Michael Sarver, Craig L Zirbel, Jesse Stombaugh, Ali Mokdad, and Neocles B Leontis. FR3D: finding local and composite recurrent structural motifs in RNA 3D structures. *Journal of mathematical biology*, 56:215–252, 2008.

30 Antoine Soulé, Vladimir Reinharz, Roman Sarrazin-Gendron, Alain Denise, and Jérôme Waldispühl. Finding recurrent RNA structural networks with fast maximal common subgraphs of edge-colored graphs. *PLoS computational biology*, 17(5):e1008990, 2021.

31 Jesse Stombaugh, Craig L. Zirbel, Eric Westhof, and Neocles B. Leontis. Frequency and isostericity of RNA base pairs. *Nucleic Acids Research*, 37(7):2294–2312, February 2009. doi:10.1093/nar/gkp011.

32 Bernhard C Thiel, Irene K Beckmann, Peter Kerpedjiev, and Ivo L Hofacker. 3D based on 2D: Calculating helix angles and stacking patterns using forgi 2.0, an RNA Python library centered on secondary structure elements. *F1000Research*, 8, 2019.

33 Hua-Ting Yao, Yann Ponty, and Sebastian Will. Developing complex rna design applications in the infrared framework. *RNA Folding – Methods and Protocols*, 2022.

34 M. Zahran, C. Sevim Bayrak, S. Elmetwaly, and T. Schlick. RAG-3D: a search tool for RNA 3D substructures. Nucleic acids research. *Nucleic Acids Research*, 43(19):9474–9488, 2015. doi:10.1093/nar/gkv823.

A Supplementary material

A.1 About the sampling process

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

▶ **Definition 3** (Boltzmann distribution/Partition function). *In the **Multidimensional Boltzmann Distribution**, the probability to sample graph G, subgraph of G_T with features F_1, ... F_m (that embody neighborhoods differences of G_P for mapped graph G in G_T) of respective weights w_1, ... w_m (that we can write more simply $w = (w_1, ...w_m)$) is proportional to its **energy**:*

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

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

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

We can forget about the β contribution as we can rewrite the weight $w'_i = \beta w_i$. The weights w_i are values chosen or tuned by us.

Tuning the weights is done by fixing a mean T^{F*} (and T^F threshold) for each type of neighborhood. We can then tune the weight $w(F_i)$ to give more "importance" that will favor value around T^{F*}. In practice, when a feature for a neighborhood varies greatly between instances, it means that this neighborhood is strongly relevant to distinguish the different matches. It gives us an incentive to modify its weight accordingly. To do so, instead of choosing weights manually, we solve the following problem:

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

This problem is known to be convex. We used so convex optimization method. Further details about this problem, including the proof of convexity, are addressed in [15].

A.2 Computation of the partition function using dynamic programming

A.2.1 Definitions

First, we introduce the formal definition of the treewidth, we also depict what is a nice tree decomposition (NTD) as it allows a simpler search during the dynamic programming procedure. NTD implies no additional cost because an NTD has at most a size $n = |G_T|$.

▶ **Definition 4** (Tree Decomposition (TD)). *Given a graph $G = (V, E)$, a tree decomposition of G is a tree T, whose nodes are bags $Y_1 ... Y_t$ such that: (definition from Bodlander et al [3])*

1. $V \subset \bigcup_{i=1}^{t} Y_i$
2. $\forall (u, v) \in E, \exists i \in [\![1, t]\!], (u \in Y_i) \cap (v \in Y_i)$
3. $\forall u \in V, \{u | u \in Y_i\}$ *is a subtree of* T.

▶ **Definition 5** (Nice Tree Decomposition). *A tree decomposition T of $G = (V, E)$ is said "nice" if each bags Y_i has one of the three following forms:*

- *Introduce: Node Y_i has exactly one child of index c in T and $Y_i = Y_c \cup \{v\}$*
- *Forget: Node Y_i has exactly one child of index c in T and $Y_c = Y_i \cup \{v\}$*
- *Join: Node Y_i has exactly two children of indices c_1 and c_2 in T and $Y_i = Y_{c_1} = Y_{c_2}$*

▶ **Definition 6** (Treewidth). *The treewidth ϕ of a graph G is defined as the biggest bag of the "best" tree decomposition of G:*

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

A.2.2 Dynamic programming solution

We now address the computation of the partition function [15] from 3 through a dynamic programming procedure on the nice tree decomposition of G_T.

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

- The set of neighborhood thresholds: $F = \left(T^L, T^E, T^G\right)$.
- M_i, **partial mapping** at node Y_i of T.
- The **separator node** of Y_i, $\text{sep}(Y_i)$ chosen as the first element of the set S:

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

  We can point out that, with a nice tree decomposition, there exists only a unique choice for this node and the set S is reduced to a singleton.
- Given a partial mapping M_i, we introduce the following Boolean condition to map each contribution to a single bag and avoid multiple computations of it:

  $$C(u_1, u_2, Y_i, M_i) = (u_1 = \text{sep}(Y_i) \cap M_i(u_2) \neq \emptyset) \cup (u_2 = \text{sep}(Y_i) \cap M_i(u_1) \neq \emptyset).$$

  From this we introduce $\Delta(\cdot)$ to denote the global contribution

  $$\Delta(M_i', G_T, Y_i, T^F) = \left\{d^F_{G_T}(u_1, u_2, M_i') \mid C(u_1, u_2, Y_i, M_i') \text{ is True}\right\}.$$

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

- Forget Node Y_i with child Y':

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

- Introduction Node, creating vertex $s := \text{sep}(Y_i) \in V_P$ having child Y':

  $$P[Y_i; M_i] = \sum_{v \in D(s|M_i)} P[Y'; M_i \cup (s \leftarrow v)] \times \prod_{\substack{T^F \in F \\ \delta \in \Delta(M_i \cup (s \leftarrow v), G_T, Y_i, T^F)}} e^{-\mu . w(T^F) . \delta}$$

  where $D(v \mid M)$ denotes the set of admissible mappings for $v \in V_P$, consistent with prior assignment M, such that:

  $$D(v \mid M) := \begin{cases} V_T & \text{if } M = \varnothing \\ \bigcap\limits_{\substack{u \in M \\ \text{s.t. } u \prec v}} \{x \in V_T \mid M(u) \prec x\} \bigcap\limits_{\substack{u \in M \\ \text{s.t. } v \prec u}} \{x \in V_T \mid x \prec M(u)\} & \text{otherwise.} \end{cases}$$

- Join Node:

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

The backtracking step to retrieve the value of probability for each graph (and so the whole Boltzmann distribution as introduced in 3) uses the same type of equations but going from top to bottom: a number is drawn at each node to know if we have to add a value for current mapping, given the partial partition function computed at each step of the forward procedure. Both the forward and backward steps are currently known procedures that have been studied and automatized in a framework named `Infrared`. [33], which has the advantage to be quite permissive about the definition of the neighborhood cumulative differences.

A.3 Choice on technical parameters

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

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

A.4 Time results on Narval and Beluga clusters for FuzzTree

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

Some time results for computation of the FuzzTree method, by requesting one motif on each RNA chain where Kink-Turns are known, are available in Fig. 12. The time of computation is large but it is something expected with the XP theoretical complexity. However, one can notice that in practice the treewidth of the selected pattern is equal to 2 which allows a complexity in $O\left(n^3\right)$. No true time discrepancy appears between the computation without near edges and the one with. On large graphs, due to the slicing, the time of computation is reduced, but such reduction is not perfect as slicing computation is still quite redundant: multiple graphs cover sometimes the same portion of the Kink-Turn.

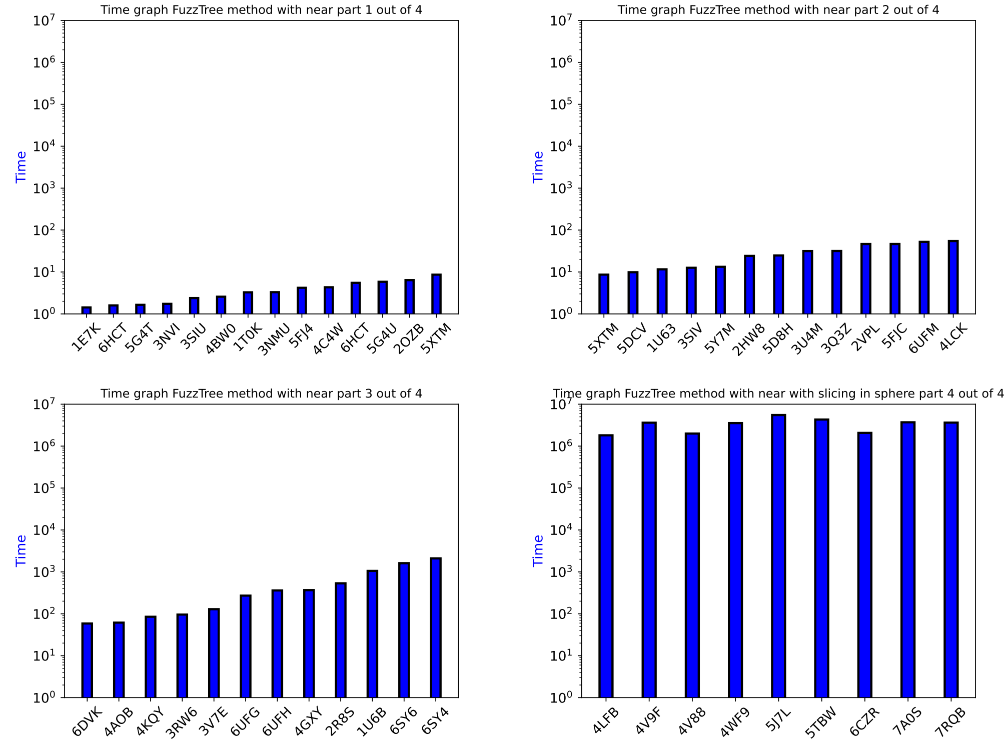


Figure 12 Time graph of the FuzzTree method on each group of studied RNA chains. On the Beluga cluster, computations were done on 1 processor for small RNAs (less than 500 nucleotides, which corresponds to the three first graphs) and on 40 processors for large RNAs (more than 500 nucleotides, which corresponds to the fourth graph). In that case, the depicted time is the sum of each time consumed for each processor.