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

Théo Boury, Yann Ponty, Vladimir Reinharz

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.?

Related Version full version hosted on arXiv

Supplementary Material The source code for the tool and the tests are on GitHub:FuzzTree. The RNA structures encoded as python pickle graphs are available at 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.

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 $O\left(2^{|V_P|}n^{tw(G_P)+1}\right)$ 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 $tw\left(G_P\right)$ and $|G_P|$, and conversely, others are parameterized by $tw\left(G_T\right)$ 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 $L\left(e\right)$. Interactions may represent backbone connectivity (phosphodiester bonds),

or any of the 12 base-pair types defined by the Leontis-Westhof (LW) nomenclature [29]. Each type specifies an interacting face (Watson-Crick ∘, Hoogsteen □, Sugar ▷) 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. We show in Fig. 1 a Kink-Turn motif, represented as a graph with labeled edges.

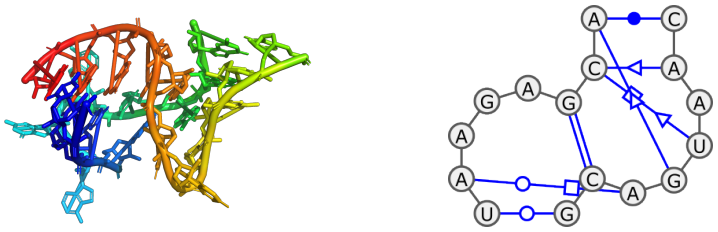


Figure 1 Kink-turn structure On the left the 3D structure of a Kink-Turn in PDB `3RW6`. On the right its representation as a graph of its base pair interactions, the backbone connections are represented as black arrowed edges.

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 for LW interactions.

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

▶ **Problem 1.** *Subgraph Isomorphism Problem (SIP)*
Input: *A pattern graph $G_P = (V_P, E_P)$, a target graph $G_T = (V_T, E_T)$*
Output: *A mapping $M : V_P \to V_T$ such that*

- $\forall u, v \in {V_P}^2, M(u) = M(v) \to u = v$ *(Mapping Injectivity)*
- $\forall u, v \in E_P, (M(u), M(v)) \in E_T \Rightarrow L((u, v)) = L((M(u), M(v)))$ *(Label Compatibility)*
- $\forall (u, v) \in E_P, (M(u), M(v)) \in E_T$ *(No Missing Edge)*

or ∅ if no such mapping exists.

We now generalize the 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 is additive and has its own **neighborhood threshold** and a 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. 2.3.1). We denote the weights of edits induced by one changing edge as D^L;
- T^E corresponds to the maximum number of edges/base pairs within the pattern structure that can be lost (see Sec. 2.3.2);
- 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. 2.3.3). We denote the weight of edits induced by the introduced gap as D^G.

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 $G_T = \left(V_T, E_T = B_T \sqcup \overline{B}_T\right)$ is defined as:

$$P = \bigcup_{k\in\mathbb{N}, k\geqslant 3} \{(p_0, ..., p_k) \mid \forall i \in [\![0, k-1]\!], (p_i, p_{i+1}) \in B_T\}$$

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 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* $G_P = \left(V_P, E_P = B_P \sqcup \overline{B}_P\right)$*, a target graph* $G_T = \left(V_T, E_T = B_T \sqcup \overline{B}_T\right)$ *and neighborhood thresholds* (T^L, T^E, T^G, D_{edge}, D_{gap})
Output: *A mapping* $M : V_P \to V_T$ *such that*

- $\forall u, v \in {V_P}^2, M(u) = M(v) \to u = v$ ***(mapping injectivity)***
- $\sum_{(u,v)\in\overline{B}} D^L\left(L(u,v), L\left(M(u), M(v)\right)\right) \leqslant T^L$ ***(label compatibility)***
- $\left(|\overline{B}| \leqslant T^E\right) \cap \left(\forall (u,v) \in \overline{B}, GEO\left(M(u), M(v)\right)\right) \leqslant D_{edge}$ ***(missing edges limitation)***
- $\sum_{(p_0,...,p_k)\in P} D^G(p_0, p_k) \leqslant T^G$ ***(path size limitation)***
- $\forall (u,v) \in B, \exists (p_0, p_1, p_2, ..., p_k) \in P$ *such that* ***(no missing backbone path)***
  - $p_0 = M(u), p_k = M(v)$
  - $GEO(p_0, p_k) \leqslant D_{gap}$
  - $\forall i \in [\![1, k-1]\!], \nexists a \in V_T, p_i = M(a)$

or $\varnothing$ *if no such mapping exists.*

Subsequently, we will denote by $\textbf{neighborhood}_{G_P}(G_T)$ 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.

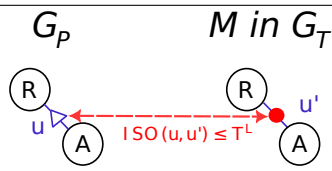
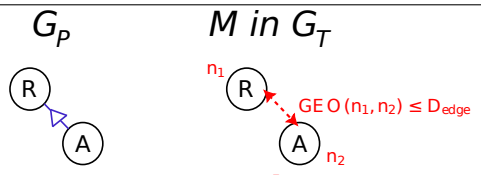
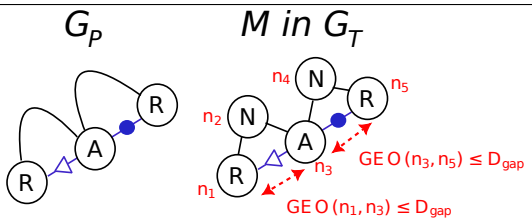
Threshold T^F	Difference d^F	Fuzzy mapping M of G_P found in G_T
T^L	Isostericity ISO	
T^E	Missing edges number	
T^G	Geometric GEO from 3D structure	

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

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

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

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

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

where $w_1, w_2 \dots$ 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 $F_1, F_2 \dots$ to user-specified values. Moreover, sampling with a pseudo-temperature $\beta \to \infty$ gracefully specializes into 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 difference d^F deviate from a given center T^{F*}. For instance, T^F_* can be chosen equals to 0 if we want to sample mostly G with no fuzziness or equals to $T^F/2$, if we want to sample them with average fuzziness. More details on this choice of value and more generally about Boltzmann sampling is available at part A.1 from supplementary material. 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 neighborhood from G_P, which allows to characterise well neighborhood$_{G_P}(G_T)$. In particular, we can define F such that it takes a value equals to $+\infty$ when the corresponding neighborhood threshold T^F is not respected by mapping M, forbidding simply this mapping M to be sampled. Additionally, the Multidimensional character of the distribution allows to take into account the 3 neighborhoods on labels, edges and gaps at the same time.

A general framework called `InfraRed` [31], initially introduced in the context of RNA design [14], can be used for generate efficiently in a parameterized 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 Supplementary Section A.2.

2.3 Neighborhood difference description

Our goal is to be able to retrieve from a general pattern 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 figure 2 is a Kink-Turn whose occurrences in the same sub-family can have up to four missing edges. Others sub-family 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 MB distribution 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 $G_P = (V_P, E_P)$, 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*

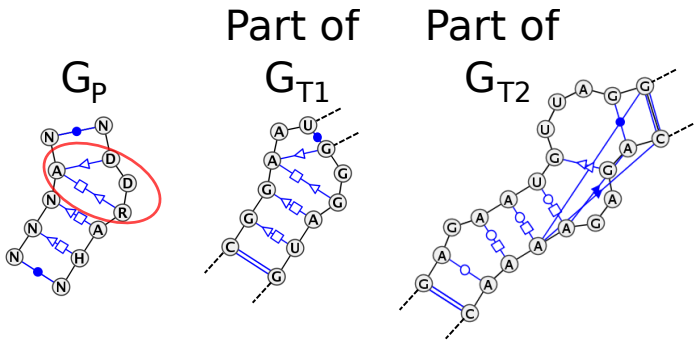


Figure 2 Kink-turn signature and targets On the left, signature graph of the Kink-Turn IL_29549.9 family and our search pattern. On the middle and on the left, mappings that were missed during the search for the pattern. G_{T1} due to the same nucleotide merging the end of a cSS and a cWW. G_{T2} due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

a wrapper around $d^F_{G_T}$*:*

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

Where $d^F_{G_T}(u, v, M)$*, the* ***neighborhood difference*** *relative to* G_T *is 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 will 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 its equivalent in the mapping. While our neighborhood cumulative differences are defined relatively 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 details the 3 sources of operations and their neighborhood cumulative difference. A summary is shown in Table 1.

2.3.1 The label difference

The isodiscrepancy index [29] quantifies the geometrically different 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 only defined between the 12 canonical and non canonical base pairing families (BPFs) which will be named as: $BPF(i)\ \ \forall i \in [\![1, 12]\!]$.

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

$$\forall (BPF(i), BPF(j)), ISO(BPF(i), BPF(j)) =$$
$$\text{isodiscrepancy}(BPF(i), BPF(j)) - \text{isodiscrepancy}(BPF(i), BPF(i))$$

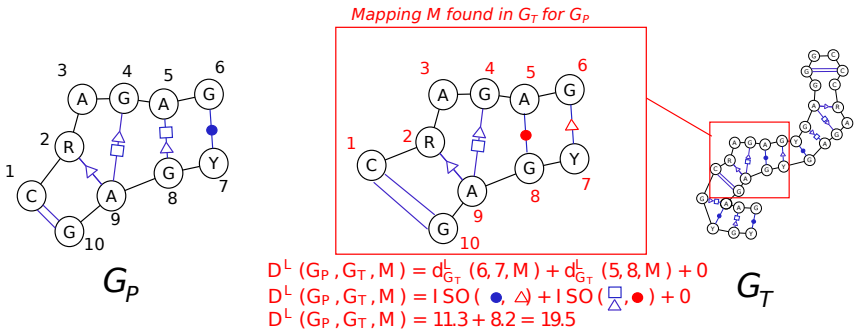


Figure 3 Label difference Computation of the label difference on a mapping between a motif G_P and an RNA target graph G_T. Label difference is computed using the isostericity ISO to account for geometric difference between bounds as described in [29].

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 $d^L_{G_T}$ equals to

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

as shown in Fig. 3.

2.3.2 The edge difference

While the previous section deals with how to incorporate edges changing their type, that is to say 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 constraint 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 global conformation. It is also highly inefficient in terms of computation.

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. $GEO(u, v) < D_{edge}$). Setting a weight of ∞ to mappings outside the threshold allows the sampling to simply reject such instances. 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:

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

2.3.3 The gap difference

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

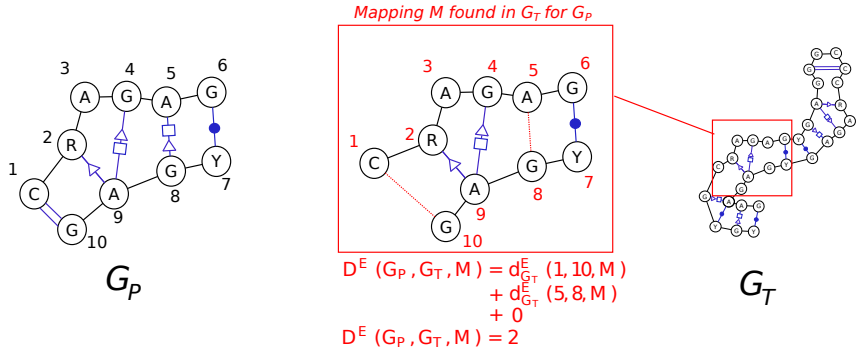


Figure 4 Edge difference Computation of the edge difference on a mapping between a motif G_P and an RNA target graph G_T. We assume here that $D_{\text{edge}} >> max\,(GEO(1,10), GEO(5,8))$

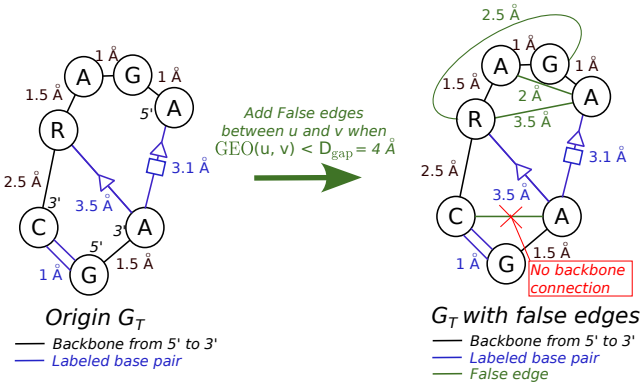


Figure 5 False edges Addition of False edges to account for gaps. False edges are added only when distance are below D_{gap} and when both nucleotides are fully connected by backbone edges. For instance here, we add no false edge between C and A at the bottom of G_T as this two nucleotides are not connected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

of different size, but we require that they do not modify (too much) the local structure. To take arbitrary insertions into account we introduce **false 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.

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 false edges. This allows an arbitrary 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 $d^G_{G_T}$:

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

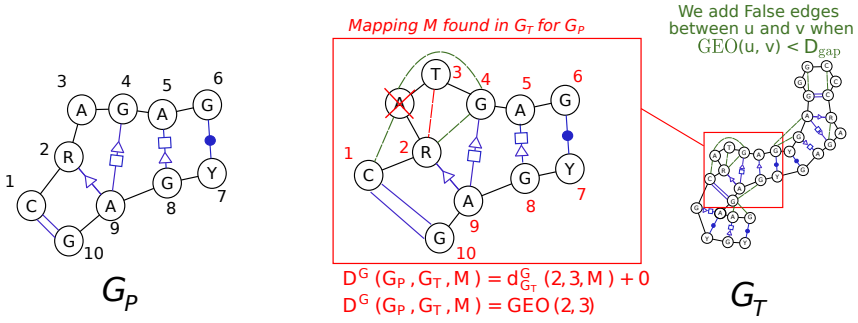


Figure 6 Gap difference Computation of gap difference on a mapping between a motif G_P and an RNA target graph G_T. We remind that nucleotides labels are not taken into account.

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 directly participate in a base pair interaction, and reconnect the disconnected backbones. Using the new pattern with a large gap threshold G would allow to retrieve the original motif neighborhood at the cost of performance, and to introduce more spurious matches.

2.4 Algorithm and complexity

Our method is based on `Infrared` [14, 31], a declarative framework which automatically generates a dynamic programming procedure for MBD sampling, based on a nice tree decomposition (TD). 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 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 $|V_T| = n$. Remark that all deviations from the pattern defined in Sections 2.3.1 through 2.3.3, 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. 7. In mappings sampled with Infrared, a neighborhood threshold T^F act as a global property over the mapping, and can only be computed afterward. Sampling is thus followed by a simple rejection step, in which samples that exceed a neighborhood threshold are rejected. Asymptotically, such rejection will not be impactful with the T^F that can be chosen independantly from $|G_P|$ and $|G_T|$. 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. Asymptotically, such rejection will at worst induce a constant factor with T^F chosen independantly from $|G_P|$ and $|G_T|$.

▶ **Theorem 1.** *The random generation of t Boltzmann-distributed (1) solutions before rejection of FSIP can be done in time $\mathcal{O}\left(n\,k\,t + k\,n^{(\phi+2)}\right)$, where ϕ is the treewidth of the pattern G_P.*

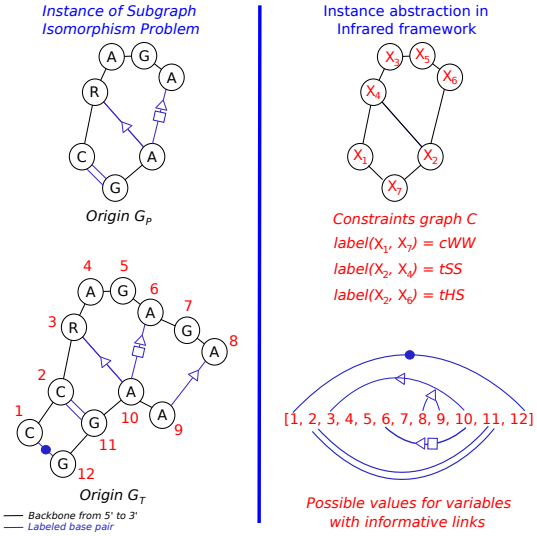


Figure 7 Framework abstraction Interfacing `Infrared` by considering G_P as the `Infrared` graph of constraints C and all nodes of G_T as values that can be taken by the variables in C.

This complexity directly follows from the complexity of the algorithm [14] underlying `Infrared` for a graph $G = (V, E)$. Restricted to binary constraints/features associated with (a subset of) E, the computation of the partition function can be performed in time $\mathcal{O}((|E| + |V|) \times \Delta^{\phi+1})$, where Δ is the size of the assignment domain for individual variables, and ϕ is the treewidth of G. A stochastic backtrack follows, leading to the generation of t Boltzmann-distributed assignments in time $\mathcal{O}(|V|\,\Delta\,t)$. The complexity stated above is obtained by observing that $|E_P| \in \Theta(k^2)$, that $\Delta \in \Theta(n)$ and $k \leqslant n$. We conclude by noting that preprocessing, including computations of geometrical distances and augmentation of G_T graph, can be performed in negligible $O(n^2)$ time and space, while 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. In term of parameterized complexity [9], the FuzzTree method is XP in the treewidth of the pattern graph, both in time and in space. It is a progress compared to VF2 [7], which is indeed implemented and efficient in practise due to the profusion of lookahead rules but has a worst case time complexity similar to $O\,(n^n)$. In practise, VF2 becomes costly with dense graphs, even in its most modern versions. [4, 16] Furthermore, we are able to compete with the bound from the Color-Coding [1] technique by improving it in time and space $2^{O(k)}$ is replaced by $kn \leqslant n^2$ in our bounds, which allow to get rid of k as parameter to restrict it to simply to the treewidth.

In addition, our method, even if tuned for RNAs, supports a more general version of the usual Subgraph Isomorphism Problem by handling 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 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	$O\left(knt + kn^{\phi+2}\right)$
Space complexity	$2^{O(k)}n^{\phi+1}$	$O\,(n)$	Exponential	$O\left(n^{\phi+2}\right)$
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 :

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

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

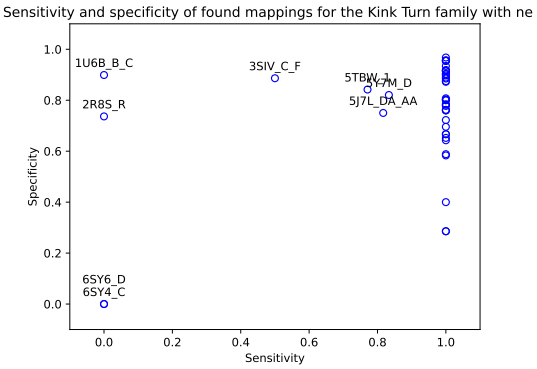


Figure 8 Sensitivity and Specifity of regions corresponding to sampled graphs in the 46 RNA structures containing Kink-Turns.

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_{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	$[0, 50]$	$[\![0, 6]\!]$	$[0, 50]$	$[5, 10]$	$[5, 20]$	$R(G_P) + [\frac{D_{\text{gap}}}{4}, D_{\text{gap}}]$	

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.

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 [30] we decomposed these regions in their secondary structure elements. The majority, 125, mapped to regions forming multiloops. A total of 33 were covering a continuous double stranded regions. The angles of surrounding stems for each interior loops 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. 9.

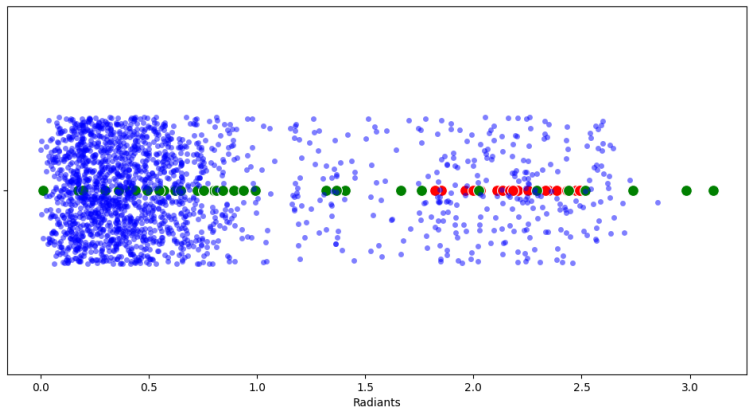


Figure 9 Angles in radiants In blue for stems around every interior loop in the 46 RNAs. In red for the Kink-Turns identified in these RNAs. In green for the additional 33 continuous double stranded regions.

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. 10. 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 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

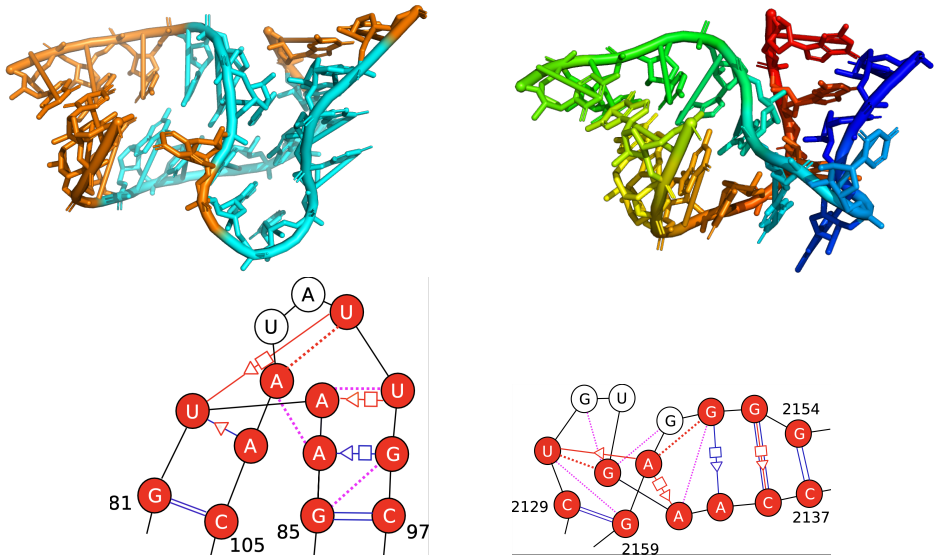


Figure 10 Other matches 5J7L on the left and 7RQB on the right. The 3D structure on the left has IL_85931.1 highlighted in cyan, on the right each nucleotide is colored independently. In the graphs, red nodes are matched with the pattern. Blue edges are in the RNA structure and red ones in the pattern, indicating modifications and removal. Red dashed lines are introduced "false edges". Magenta dashed lines indicate stackings.

types of variants (e.g., isosteric distance 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. 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, jul 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. URL: https://dmtcs.episciences.org/2793, 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. URL: https://doi.org/10.1093/nar/gkp011, 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, 11 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. URL: https://drops.dagstuhl.de/opus/volltexte/2023/17811, 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, 01 2008. doi:10.13140/2.1.1873.9847.

13 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.

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

15 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.

16 Alpár Jüttner and Péter Madarasi. Vf2++—an improved subgraph isomorphism algorithm. *Discrete Applied Mathematics*, 242:69–81, 2018. Computational Advances in Combinatorial Optimization. URL: https://www.sciencedirect.com/science/article/pii/S0166218X18300829, doi:https://doi.org/10.1016/j.dam.2018.02.018.

17 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, page 901–912, New York, NY, USA, 2011. Association for Computing Machinery. doi:10.1145/1989323.1989418.

18 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.

19 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. URL: https://www.sciencedirect.com/science/article/pii/S0959440X06000807, doi:https://doi.org/10.1016/j.sbi.2006.05.009.

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

21 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.

22 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. URL: http://drops.dagstuhl.de/opus/volltexte/2014/4486, doi:10.4230/LIPIcs.STACS.2014.542.

23 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, 11 2021. arXiv:https://academic.oup.com/bioinformatics/article-pdf/38/4/970/42319124/btab768.pdf, doi:10.1093/bioinformatics/btab768.

24 Aymeric Perchant and Isabelle Bloch. Fuzzy morphisms between graphs. *Fuzzy Sets and Systems*, 128(2):149–168, 2002. URL: https://www.sciencedirect.com/science/article/pii/S0165011401001312, doi:https://doi.org/10.1016/S0165-0114(01)00131-2.

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

26 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, 03 2018. arXiv:https://academic.oup.com/nar/article-pdf/46/8/3841/24783244/gky197.pdf, doi:10.1093/nar/gky197.

27 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.

28 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.

29 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, 02 2009. arXiv:https://academic.oup.com/nar/article-pdf/37/7/2294/18783207/gkp011.pdf, doi:10.1093/nar/gkp011.

30 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.

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

32 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:https://doi.org/10.1093/nar/gkv823.

A Supplementary material

A.1 About the sampling process

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, ... F_m (that embody neighborhoods of G_P for G) 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 as $w_i' = \beta w_i$ with w_i a value that is chosen or tuned by us.

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

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

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

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

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 as it allows a simpler search for the dynamic programming procedure without additional cost due to the fact that nice tree decomposition have at most a size $n = |G_T|$

▶ **Definition 3.** *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 4.** *Nice Tree Decomposition*
A tree decomposition T of $G = (V, E)$ is said "nice" if each bags Y_i has one of the tree following form :

- *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 5.** *treewidth (TW)*
The treewidth ϕ of a graph G is defined as the biggest bag of the "best" tree decomposition of G :

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

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

- In absence of pseudoknots, pattern graph are planar.
- RNA motifs in databases are known to have mostly treewidth below 4.

A.2.2 Dynamic programming formula

We now address the fact to compute the partition function from 2 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:

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

  We can point out that with a nice tree decomposition, there exists only a unique choice for this node and above set is reduced to a singleton.
- Given partial mapping M_i, a target graph G_T, Δ designed all the neighborhood differences partially assigned yet that can be assigned in current bag Y_i and relative to a given neighborhood cumulative difference that we design here by its corresponding threshold T^F by simplicity:
  We put the following boolean condition $C\,(u_1, u_2, Y_i, M_i) =$
  $(u_1 = \text{sep}\,(Y_i) \cap M_i\,(u_2) \neq \emptyset) \cup (u_2 = \text{sep}\,(Y_i) \cap M_i\,(u_1) \neq \emptyset)$
  And:

$$\Delta(M_i, G_T, Y_i, T^F) = \left\{d^F_{G_T}(u_1, u_2, .) | C\,(u_1, u_2, Y_i, M_i) \text{ is True}\right\}$$

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 of son Y':

$$P[Y_i; M_i] = \sum_{v \in V_T} P[Y'; M_i \cup (\text{sep}\,(Y_i) \leftarrow v)]$$

$$\times \prod_{T^F \in F} \prod_{d \in \Delta(M_i, G_T, Y_i, T^F)} e^{-\mu w(T^F).d(M_i \cup (\text{sep}(Y_i) \leftarrow v))}$$

Join Node :

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

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

$$P[Y_i; M_i] = \sum_{v \in V_T} \prod_{Y' \in children(Y_i)} P[Y'; M_i \cup (\text{sep}\,(Y_i) \leftarrow v)]$$

$$\times \prod_{T^F \in F} \prod_{d \in \Delta(M_i, G_T, Y_i, F)} e^{-\mu w(T^F).d(M_i \cup (\text{sep}(Y_i) \leftarrow v))}$$

The backtracking step to retrieve the value of probability for each graph (and so the whole Boltzmann distribution as introduced in 2) use the same type of equations but using a procedure that goes 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 that we computed at each steps of the forward procedure. Both the forward and backward steps are currently known procedures that have been studied and automatised in a framework named `Infrared`. [31] We use this framework for our implementation as it allow to be quite adjustable with what we named neighborhood cumulative differences and how to define them.

To ensure the correction of the algorithm, nodes in entry for $d^F_{G_T}$ functions must be at least one time in the same bag in the tree decomposition of G_P. Here, it is not a problem as, by definition of the tree decomposition introduced in part 3, extremities of an edge are at least in a same bag.

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

A.3.1 Application in our Kink-Turns case

To retrieve the diversity of the Kink-Turn group with our method, we have to request a specific motif that will contain all or most of others Kink-Turns in its neighborhood. We will call such a motif the **central motif**. In order to choose the best central motif for the given family, one can compute from any motif O the 3 neighborhood cumulative differences between O and all the other motifs of the studied RNA group. It gives us a 3D-cartography (one dimension for each neighborhood cumulative difference) from an origin O of all Kink-Turns. This cartography allows us to choose more efficiently our neighborhood thresholds. We depict first the cartography for the Kink-Turn group from the motif IL_5TBW_059 contained in the IL_29549.9 family. The motif can be observed on 2 and the cartography on figure 11.

From this cartography, one can observe that a choice of threshold $T^L = 20$ and $T^G = 20$ allows to cover all the Kink-Turns. Nonetheless, T^E threshold is more difficult to choose. Indeed, on the cartography, Edge difference can reach 8. Nonetheless, IL_5TBW_059 contains only 12 edges (2 edges are accounted by bonds as the graph is directed). With $T^E = 8$, it means that a motif is recognized even if it contains only 2 bonds from IL_5TBW_059, which is not acceptable as it will allow to recognize a wild range of patterns that are not Kink-Turns. In addition, in terms of performances, incrementing T^E slows down the algorithm by

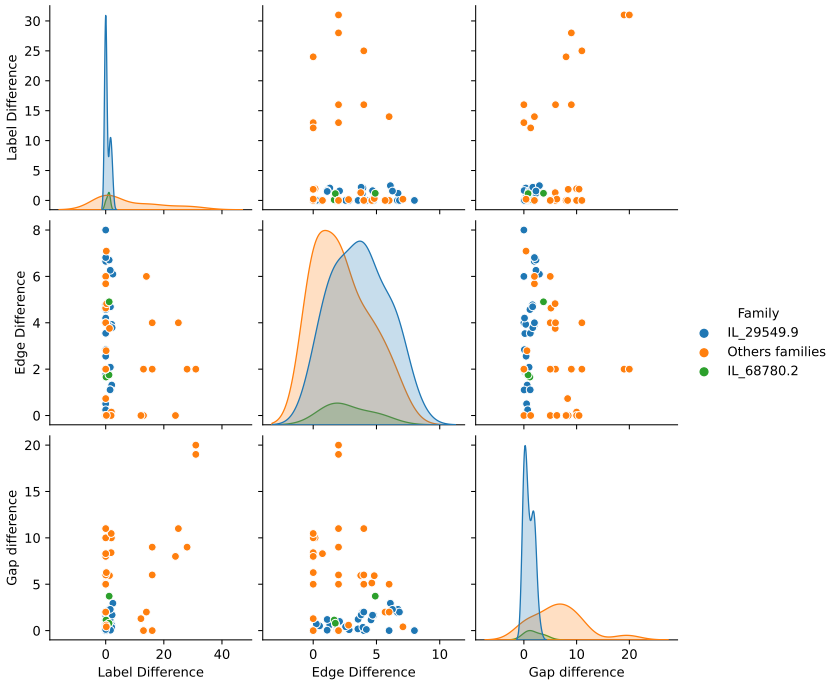


Figure 11 Kink-Turn cartography of neighborhood cumulative differences based on IL_5TBW_059 origin, member of IL_29549.9 family.

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

This cartography also allows us to discuss the way families are clustered in RNA3DAtlas. Indeed, on the cartography, motifs of the IL_29549.9 family reach an Edge difference of 8, which is the maximum possible among Kink-Turns, whereas their gap and label differences never exceeded 3, way below the maxima. Similarly, if we look at the small IL_668780.2 family, we can observe that it has its boundary values in terms of label, edge and gap differences included in the boundaries of the IL_29549.9 family. As a consequence, from the point of view of our metrics, there is no way to distinguish between the two families. It means that the way we cluster and explore groups of RNA motifs is quite orthogonal to the one that was used to build the RNA3DmotifAtlas, due to our focus on base pairs bounds themselves instead of atoms in 3D. It gives us good hope to suggest new patterns not easily predicted with atomic traditional methods.

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 theirs 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 $\left(T^L, TE, T^G\right) = \left(10^{length\left(|G_P|^2\right)}, |G_P|^2, \text{max length } B53 \text{ path}\right)$. We next solve successively for each neighborhood threshold :

$$argmin_{T^F}\left(FuzzTree\left(O, C, T^L, T^E, T^G\right) \text{ with } F \in \{L, E, G\}\right)$$

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

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

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

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

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

A.4 Choice on technical parameters

For the choice of the radius R for creating slice of target graph G_T, given an extracted graph G from G_T centered in nucleotide c, we first defined $R\left(G\right) = \min_{j \in G} \text{GEO}\left(j, c\right)$. In order to be exhaustive with our search, we must ensure that every G from G_T is extracted with a radius at least equals to $R\left(G_P\right) + D_{\text{gap}}$ 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, as 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 others. Due to the rarity of this case, we choose, in the tests, to use a smaller radius equals to $R\left(G_P\right) + \frac{D_{\text{gap}}}{4}$.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 on.

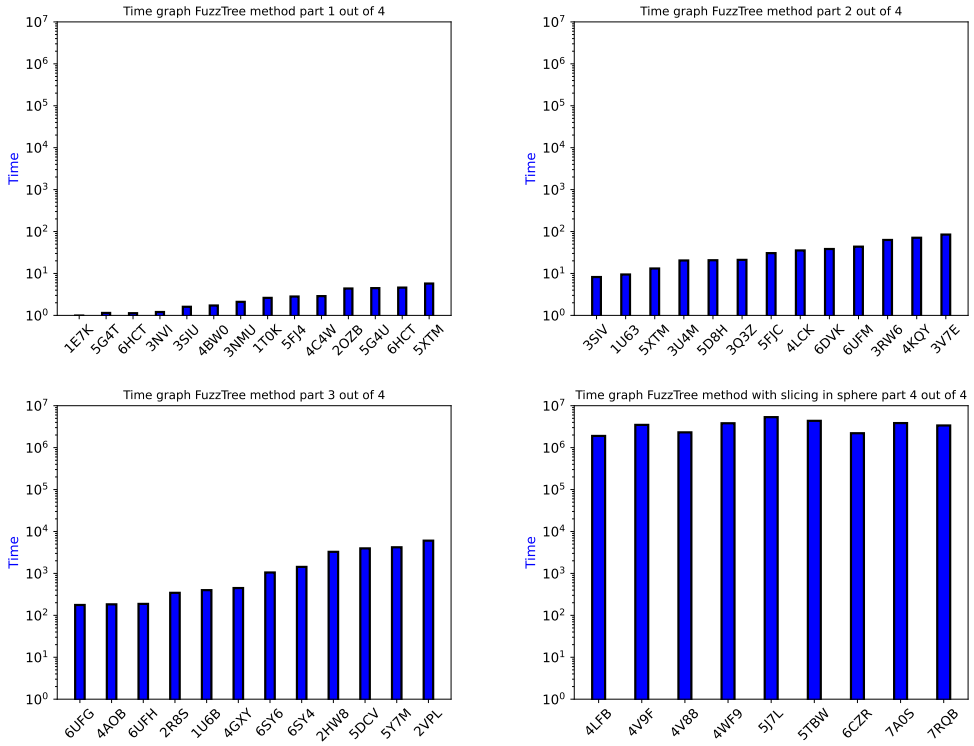


Figure 12 Time graph of the FuzzTree method on each group of studied RNA chains. On Narval cluster, computation were done on 1 processor for small RNAs (less than 500 nucleotides, it corresponds to the two first graphs) and on 64 processors for large RNAs (more than 500 nucleotides, it corresponds to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

We also choose to use a timeout equal to 2000 seconds for the convergence of our algorithm on each extracted graphs. 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 in term of expressiveness, which means that somebody with more computational resources could use this tool and wait for even better performances.

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

For this paper, computation 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 a 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 chains where Kink-Turns are known, are available on figure 13 and 12. Time of computation is large but it is something expected with the XP theoretical complexity. However, one can notice that in practise the treewidth of the selected pattern is equals to 2 which allow a complexity in practise in $O\left(n^3\right)$. No real time discrepancy appear between the computation without near edges and the one with. In particular, one should notice that

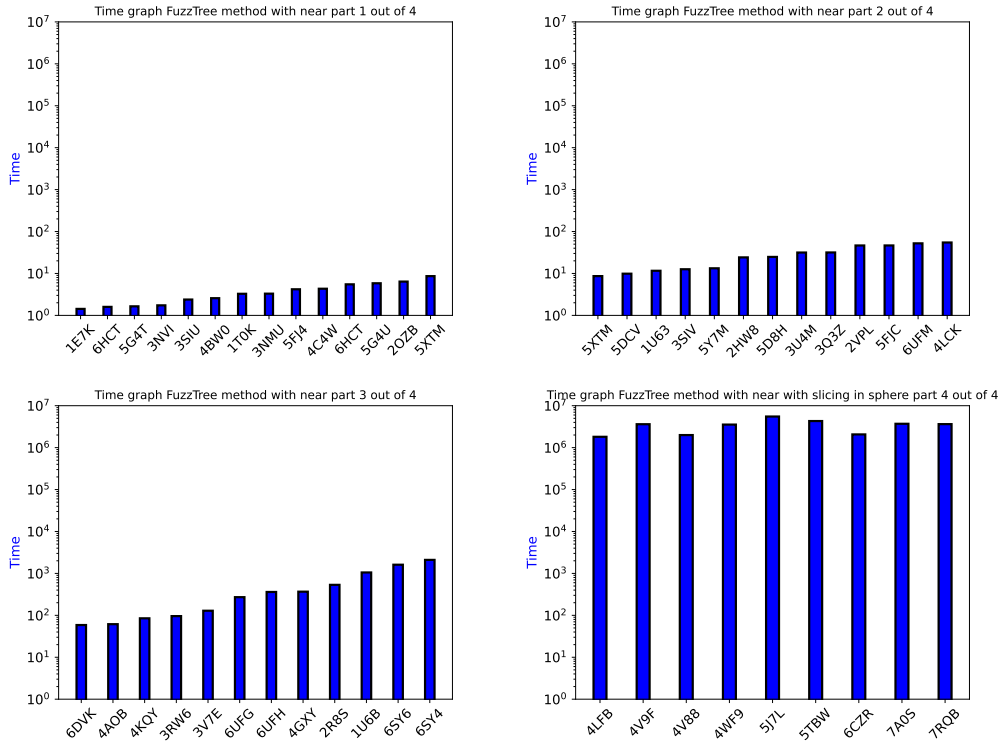


Figure 13 Time graph of the FuzzTree method on each group of studied RNA chains. On Beluga cluster, computation were done on 1 processor for small RNAs (less than 500 nucleotides, it corresponds to the three first graphs) and on 40 processors for large RNAs (more than 500 nucleotides, it corresponds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

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

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

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

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

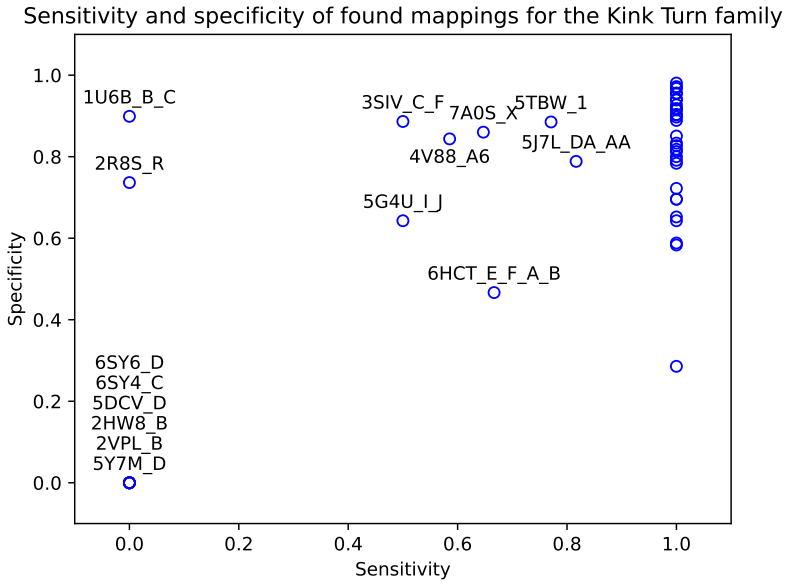


Figure 14 Representation of sensitivity and specificity for each group of chains extracted from a same RNAs without near edges.

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