Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: 30 Jan 2020
Applications Note

Subject Section

Finding recurrent RNA structural networks with fast maximal common subgraphs of edge-colored graphs

Antoine Soulé [1,2], Vladimir Reinharz [3], Roman Sarrazin-Gendron [1], Alain Denise[4, 5] and Jérôme Waldispühl [1,*]

[1]School of Computer Science, McGill University, Montréal, Canada [2]LiX, École Polytechnique, Paris, France [3]Department of Computer Science, Université du Québec à Montréal, Montréal, Canada [4]Université Paris-Saclay, CNRS, Laboratoire de recherche en informatique, 91405, Orsay, France and [5] Université Paris-Saclay, CEA, CNRS, Institute for Integrative Biology of the Cell (I2BC), 91198, Gif-sur-Yvette, France

[*]Corresponding author.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

RNA tertiary structure is crucial to its many non-coding molecular functions. RNA architecture is shaped by its secondary structure composed of stems, stacked canonical base pairs, enclosing loops. While stems are captured by free-energy models, loops composed of non-canonical base pairs are not. Nor are distant interactions linking together those secondary structure elements (SSEs). Databases of conserved 3D geometries not captured by energetic models are leveraged for structure prediction and design, but the computational complexity has limited their study to local elements, loops, and recently to those covering pairs of SSEs. Systematically capturing recurrent patterns on a large scale is a main challenge in the study of RNA structures.

In this paper, to automatically capture this topological information, we present a new general and efficient algorithm that leverages the fact that we can assign a *proper edge coloring* to graphs representing such structures. This allows to generalize previous approaches and systematically find for the first time modules over more than 2 SSEs, while improving speed a hundredfold. We then proceed to extract all recurrent base pairs networks between any RNA tertiary structures in our non-redundant dataset. We observed occurrences that are over 36 different SSEs, between the 23S ribosomes of *E. Coli* and of *Thermus thermophilus.* In addition to detecting them, our method organizes them into a network according to the similarities of their structures. Relaxing constraints, as not differentiating between local and distant interactions, reduces the number of isolated component in the network of structures. This behaviour can be leveraged to study the emergence of those intricate structures.

1 Introduction

RNA tertiary structures are mainly stabilized by canonical Watson-Crick base pairs and base pairs stacking that constitute the secondary structure. The secondary structure is composed of helices of stacked canonical base pairs (stems) separated by loops (terminal loops, bulges, interior loops, multiloops, etc,...) which are mainly structured by non-canonical base pairs. Compared to loops, stems are relatively easier to predict since both the contributions of canonical base pairs and stacking are relatively well known and modelled. Non-canonical base pairs pose a greater challenge, due to their relative rarity and lesser stability. However, the position of the stems depends on the structure of loops and interactions between distant secondary structure elements (SSEs), which may also involve non-canonical base pairs. A way to circumvent this issue lies in recurrent base pairs patterns often observed in those motifs, called RNA modules.

RNA modules are small and (generaly) densely connected base pairs patterns that can be observed in a variety of different molecules, sometimes

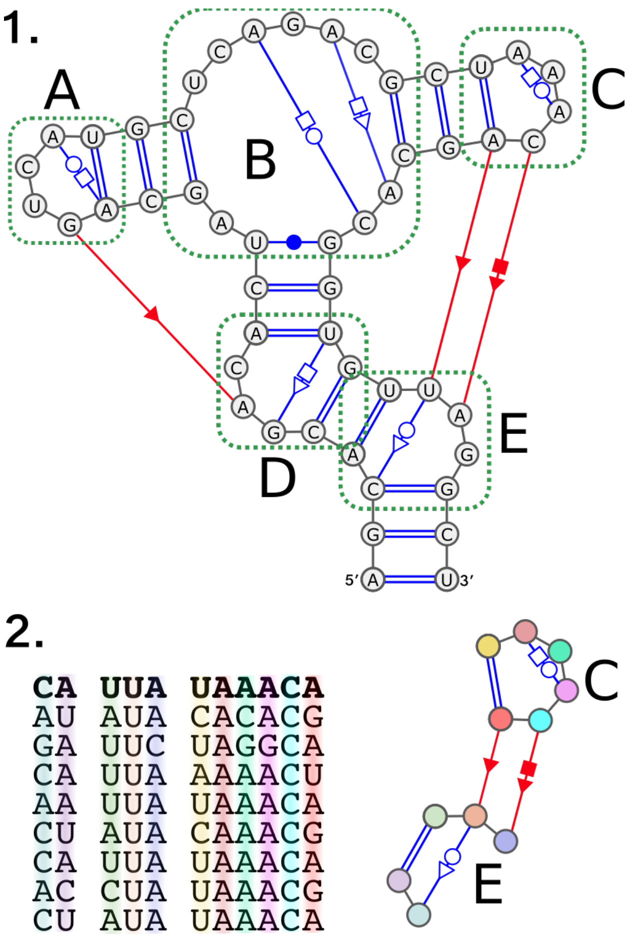

Fig. 1. Secondary structure and module In (1) we show an RNA secondary structure augmented with non-canonical interactions. Double lines indicated canonical base pairs forming the secondary structure. Interactions blue are local while the ones in red are between two distant elements. Each loop is surround by green dotted lines, A and C are hairpins, D and E are interior loops, while B is a multi-loop. In (2) we show the two parts comprising a module. On the right there is the base pair pattern of the module and on the left sequences that have been observed in that configuration. The first sequence is the one in the structure (1).

in multiple locations. We show in Fig. 1 an RNA secondary structure with its SSEs (A, B, C, D, . . .) and, below, a module from the same structure. The conservation of RNA modules suggests an evolutionary pressure to preserve specific interaction patterns, constraining the possible set of sequences adopting those interactions. As such, identifying RNA modules in a sequence provides information about base pairs that can be used to infer the 3D structure of the whole molecule [11, 15, 16, 14, 20, 13].

Some RNA modules have received a specific attention such as *GNRA loops*, *Kink-turns*, *G-bulges*, and the various types of *A-minor*s. However, in this work, we rather consider the whole landscape of RNA modules rather than an RNA module in particular. Furthermore, we aim at extracting recurrent patterns in the secondary structure rather than in the sequence or in the tertiary structure. However, those patterns capture topological information that implies a similar tertiary structure and a consensus RNA sequence can be derived from it. As such they constitute interesting RNA modules candidates. Our goal is to automatically capture this topological information. We thus introduce *RNA structural elements* (RSEs) as a medium for this topological information. *RNA structural elements* are subgraphs of RNA tertiary structures represented as graphs.

Several works have been presented, proposing computational methods to detect RNA modules in tertiary structures using either geometry or graph-based approaches [1, 5, 6, 7, 8, 9, 19, 23, 25, 27, 28, 4, 2]. However, the purpose of the majority of those methods is to search known modules in new structures. A couple of methods have been proposed that search local modules without any prior knowledge of their geometry or topology [5, 9]. In addition to those methods, databases of RNA modules found in experimentally determined RNA tertiary structures have been proposed such as RNA 3D Motif Atlas [20] and RNA Bricks [3]

In previous work, we presented an algorithm to find between two RNAs all identical *interaction networks* [22], which capture the topological information of interaction modules (i.e. RNA modules over two, non-adjacent, secondary structure elements or SSEs) but not the sequences. We also presented an extensive catalogue, named CaRNAval, of the *Recurrent Interaction Networks* (RINs) computed on the non-redundant structures in RNA3DHub [18]. RINs are common subgraphs of RNA structure graphs with additional constraints. Those constraints aim at ensuring the soundness of the RINs but some also lower the execution time. For instance, RINs do not capture pure stems extensions (*i.e.* stems without non-canonical interactions) since they are of little interest in this context. This spares us the need to extend matches through pure stems past a certain point.

The method developed for CaRNAval is automated and does not use any prior knowledge of neither the topology nor the geometry of the structures it detects. Doing so allows to underline the universality and fundamental nature of these recurrent architectures. The collection of RINs is organized and made available on a dedicated website that provides additional information such as the RNA sequences associated with each RINs and the structural contexts they have been found in as well as different search tools.

In this paper, we present a novel algorithm to find similar topologies between RNAs structures. Leveraging the *proper edge coloring* of a structure graph allows to improve execution time a hundredfold on the method in CaRNAval. The time gains allows to extend the idea of RINs to an arbitrary large number of SSEs. As discussed in Sec. 3.2.5, the largest generalized RIN spans 36 SSEs.

2 Method

From a set of *mmCIF* files describing 3D structures of RNA chains, we first annotate the interactions with FR3D. The method presented analysis these annotations in four steps.

1. We first build for each chain a directed edge-labelled graph such that the edges represent the phosphodiester bonds as well as the canonical and non-canonical interactions. The labels on the edges corresponding to the interaction types plus the indication of the interaction being either local (inside one SSE) or long-range (between two SSEs)
2. For each pair of RNA graphs, we extract all the Maximal Common Subgraphs such that edges are matched to edges with the same labels
3. Each Maximal Common Subgraph is then filtered to obtain the Recurrent Structural Elements (constrained common subgraphs) it contains
4. Finally we gather the Recurrent Structural Elements found together into a non-redundant collection and create a network of direct inclusions.

2.1 RNA 2D Structure Graphs

We rely on RNA 2D structure graphs to represent the structures of RNA chains. RNA 2D structure graphs are directed edge-labelled graphs. Each node represents a nucleotide, each edge represents an interaction (basepair or backbone). The edge are labelled according to the annotation of the interaction they correspond to. Annotations of basepairs interactions follows the Leontis-Westhof geometric classification [12]. They are any combination of the orientation cis (c) (resp. trans (t)) with the name of the side which interacts for each of the two nucleotides: Watson-Crick (W) cis ● (or ○ for trans), Hoogsteen (H) ■ (or □) or Sugar-Edge (S) ▶ (resp. ▷). Thus, each base pair is annotated by a string from the set: {c,t}×{W,S,H}^{2} or by combining previous symbols. To represent a canonical cWW interaction, a double line is generally used instead of (● ●). Moreover, each basepairs interaction can also be annotated as either *local* or *long range*, depending on the secondary structure elements the nucleotides involved are found in (our method to generate the secondary structure is described in section 3.1). The backbone is represented with directed edges, labelled $b35$.

As a consequence, an annotation (and thus a label) is composed of three characters $xYZ \in [c \mid t][W \mid S \mid H]^2$ plus a parameter $C \in$ [local | long-range]. Interactions are either symmetric (xYY) or not symmetric (xYZ). Each non symmetric interaction between nucleobases xYZ is complemented by an interaction xZY between the same nucleobases and the same value of C but in the opposite direction. We introduce an abstract type/label $b35$ to complement the $b53$ label. We can thus define a bijection ι as follow:

- $\iota(xYZ, C) = xZY, C$
- $\iota(xYY, C) = xYY, C$
- $\iota(b53, \text{local}) = b35, \text{local}$
- $\iota(b35, \text{local}) = b53, \text{local}$

An interactions of type t between nucleotides a,b (represented by nodes v_a,v_b), is represented by two directed edges $\{v_a, v_b\}$ and $\{v_b, v_a\}$ whose respective labels are t and $\iota(t)$.

We represent each RNA chain in the dataset as a RNA 2D structure graph, the annotations of the RNA basepairs interactions corresponding to the labels of the edges of the graph (cf. Fig. 2).

2.2 Graph Matching & Proper Edge-Coloring

As we transpose RNA structures into edge-labelled graphs, finding common substructures in the RNA structures comes down to finding common subgraphs in the RNA 2D structure graphs.

Problems that consist in matching graphs or parts of graphs are called *Graph Matching* problems. We are especially interested in finding common subraphs, a NP-hard problem in general. However, RNA 2D structure graphs inherit some of the constraints of the RNA structures they represent, constraints that translate into a graph property useful for graph matching.

The chemical constraints of nucleotides interactions are such that each edge of a nucleotide should be involved in at most one interaction. This translates in terms of graphs as follows: for all RNA 2D structure graphs $G = \{V, E\}$ and for all a node $v \in V$, there are no two edges $e_1, e_2 \in E$ that originate from v with the same label. To put it differently, the set of labels on the edges of any RNA 2D structure graphs naturally forms a *Proper Edge-Coloring* (PEC). We designed three graph matching algorithms designed to take advantage of the proper edge-coloring the RNA 2D structure graphs come equipped with.

2.3 Exceptions

We observed a few nucleotides annotated with two interactions involving the same Leontis-Westhof edges in some RNA structures (0.02% of the nucleotides of our reference dataset cf. section 3.1). Those interactions could either be annotation errors or biologically relevant. Given the rarity of those exceptions, we chose to duplicate the graphs concerned into different proper edge-colored versions, each covering a different interpretations.

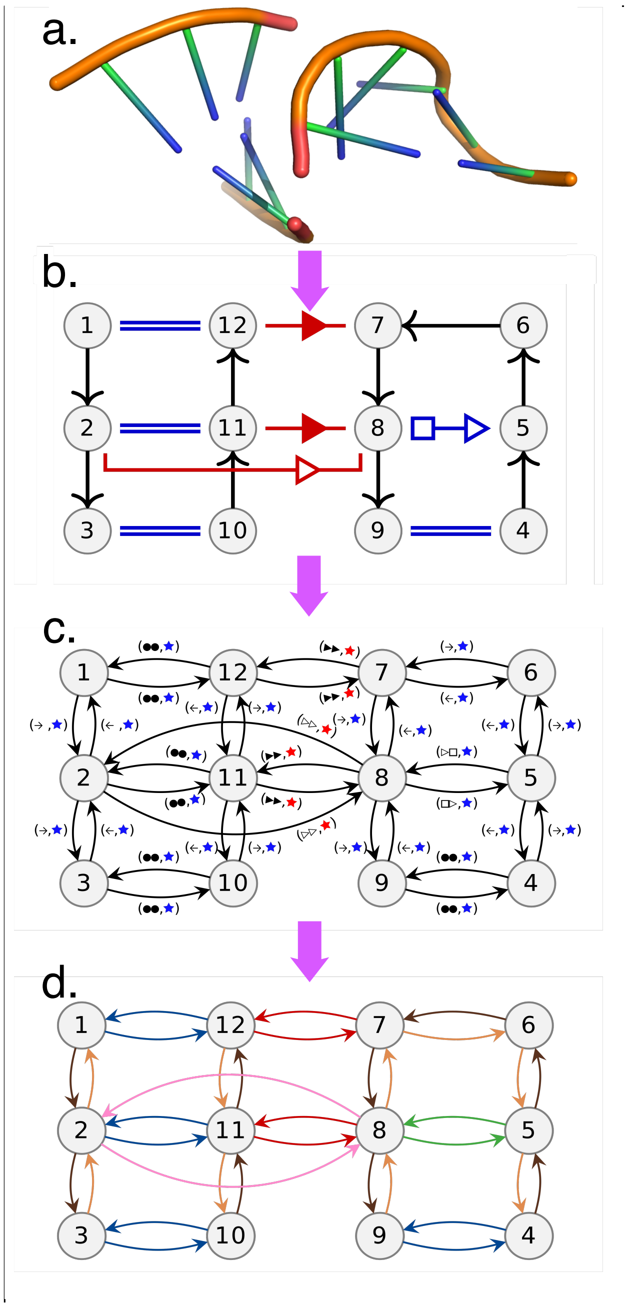

Fig. 2. From 3D structure to directed edge-labelled graph In this figure we illustrate the transition from the 3D structure (a) to RNA 2D structure graph (b) and finally directed edge-labelled graph (c) with a simple RNA structure. Each edge label of the directed edge-labelled graph is a pair whose first element represent the type of interaction (using the same symbols as in the RNA 2D structure graph) while the second denote the local (blue) vs. long-range (red) property of the interaction (using the same colors as in the RNA 2D structure graph). Moreover, the set of edge labels forms a directed proper edge-coloring, as illustrated with the last panel (d).

Details about the duplication procedure and the different versions are provided in section 2.1 of the supplementary material.

2.4 Graph Matching Algorithms

In this section we briefly present our 3 algorithms, the 3 problems they solve and how we take advantage of the PEC. Formal and complete descriptions are provided in the supplementary material (sections 1.2, 1.3 and 1.4).

2.4.1 Definitions & Notations

Two graphs $G = \{V_G, E_G\}$ and $H = \{V_H, E_H\}$ are isomorphic *iff* there is a bijection b from V_G to V_H that respects the edges and their labels. A graph $G = \{V_G, E_G\}$ is a *subgraph* of graph $H = \{V_H, E_H\}$ *iff* there exists at least one injection i from V_G to V_H that respects the edges and their labels.

Given two graphs G,H, a graph $S = (V_S, E_S)$ is a *common subgraph* of G and H if it is a subgraph of G and a subgraph of H. A common subgraph S of G and H is *maximal iff* for all S' subgraph of G and H, $S \subset S' \implies S = S'$. All three algorithms take two properly edge-colored graphs $G = \{V_G, E_G\}$ and $H = \{V_H, E_H\}$ as an input. For any color c, the sets of c-colored edges are noted E_{Gc} and E_{Hc}.

2.4.2 Using the PEC when extending a matching

The three algorithms presented in this paper revolves around exploiting the constraints that the PEC places on the matching of the two graphs. In all three algorithms, matching the two graphs is done by starting with a minimal match and then extending it through the neighbours of the already matched nodes. This strategy is common and usually requires to test all permutations between the two sets of neighbours. However, the constraint of respecting the PEC only leaves at most a single valid affectation of the neighbours, as illustrated in figure 3. As a consequence, the complexity of the extension process is linear in the number of nodes (since the number of colors is fixed, cf. section 1.2.3 of the the supplementary material).

2.4.3 Graph Isomorphism Algorithm:

The *Graph Isomorphism* problem consists in determining if two properly edge-colored graphs G and H are isomorphic. Our Graph Isomorphism Algorithm determines the color c that minimizes the product $|E_{G,c}| \times |E_{H,c}|$. Then, for all pairs of edges $(\{g_1, g_2\}, \{h_1, h_2\}) \in E_{G,c} \times E_{H,c}$, the algorithm launch an extension with the matching $\big((g_1, h_1), (g_2, h_2)\big)$ as starting point. The two graphs are isomorphic *iff* any matching can be extended into a bijection of V_G and V_H that respects the edges and their coloring. As we mentionned previously, the extension process is in $\mathcal{O}(|C| \times n)$ (assuming $n = |V_G| = |V_H|$, if not, G and H are trivially not isomorphic) and the number of starting point is capped by $\mathcal{O}(n^2/|C|)$ resulting in a $\mathcal{O}(n^3)$ complexity for the algorithm (cf. section 1.2.3 of the the supplementary material).

2.4.4 Subgraph Isomorphism Algorithm:

The *Subgraph Isomorphism* problem consists in, given two properly edge-colored graphs G and H, determining if G is a subgraph of H. Our Subraph Isomorphism Algorithm is derived from our Graph Isomorphism Algorithm, the difference between the two being that G is a subgraph of H *iff* any matching can be extended into an injection of V_G in V_H that respects the edges and their coloring. The complexity is the same as the Graph Isomorphism Algorithm: $\mathcal{O}(n^3)$ with $n = min(|V_G|, |V_H|)$ (cf. section 1.3.3 of the the supplementary material).

2.4.5 All Maximal Common Subgraphs Algorithm:

The *All Maximal Common Subgraphs* problem consists in finding all maximal common subgraphs between two properly edge-colored graphs G and H (please not that this differs slightly from the *maximal common subgraph* problem which usually consists in finding the largest common subgraph). This algorithm relies on the same extension strategy than the two previous ones. However, unlike the two previous problem, encountering a discrepancy during the extension does not implies that this extension can be abandoned (as illustrated in Fig. 4). Instead, it suggests the existence of an alternative way of matching the graphs by considering the nodes in a different order than in the current extension. As we are looking for all maximal common subgraphs, this alternative has to be explored as well. As a consequence, we designed an unconventional backtracking mechanism. For any new discrepancy encountered, we launch a new extension with a list of constraints such that this new extension will explore the alternative suggested by the discrepancy. Such an extension can also encounter new discrepancies and so on and so forth. Figure 5 illustrates this process and a complete description of this mechanism is provided in sections 1.4.2 of the supplementary material as well as a formal proof of its correctness in section 1.4.3.

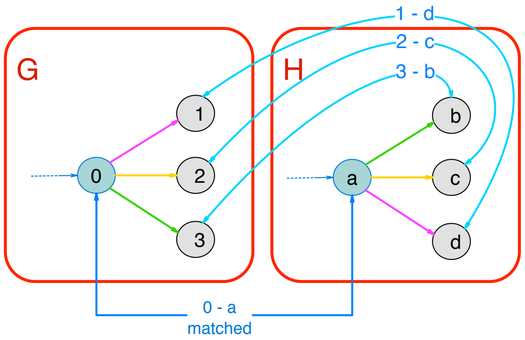

Fig. 3. Impact of proper edge-coloring on graph-matching This figure displays a piece of two graphs (G on the right and H on the left) in which the nodes 0 and a are already matched together. The next step is to match their neighbours. In the generic case, all permutations have to be tested. On the contrary, in the example displayed, the colors of the edges limit the options to consider to a single one.

2.5 From common subgraphs back to RNA structures

By transposing the RNA structures to graphs and using our algorithms, we are thus able to obtain the set of *All Maximal Common Subgraphs* contained in any dataset. However, as the number of common subgraphs grows exponentially with the size of the graphs, we expect this set to be unsuited for any practical study. As a consequence, we designed our method to extract a subset of all the structural elements contained in the dataset, defined by the user through rules or restrictions. Our method extracts and organize such subset from the set *All Maximal Common Subgraphs*

2.5.1 Recurrent Structural Element (RSE)

We call *Recurrent Structural Elements* (RSEs) any recurrent subgraph of RNA 2D Structure Graphs (i.e. observed in at least two RNAs of the dataset). A RSE is formally defined as a pair $(S, \mathfrak{O})$ with:

- $S = \{V_S, E_S\}$ a connected graph with the properties of a RNA 2D structure graph
- $\mathfrak{O}$ a collection of *occurrences*. An *occurrence* records an observation of S in the dataset. We represent an *occurrence* as a pair (G, i) with $G = \{V_G, E_G\}$ a RNA 2D structure graph and i an injection from V_S to V_H such that:
$n, n' \in V_S, \{n, n'\} \in E_{l,S} \implies \{i(n), i(n')\} \in E_{l,G}$
- $\exists (G, i), (H, i') \in \mathfrak{O}$ s.t. $G \neq H$ (i.e. it should be *recurrent*)

This minimal definition encompass a broad diversity of structural elements. As a consequence, we expect that any study of structural elements to be focalized on a subset rather than all RSEs. We call such a subset a *class of RSEs*. A class is defined by a set of rules/restrictions R, which are to be designed by the users to invalidate structural elements that fall out of the scope of the study they are conducting.

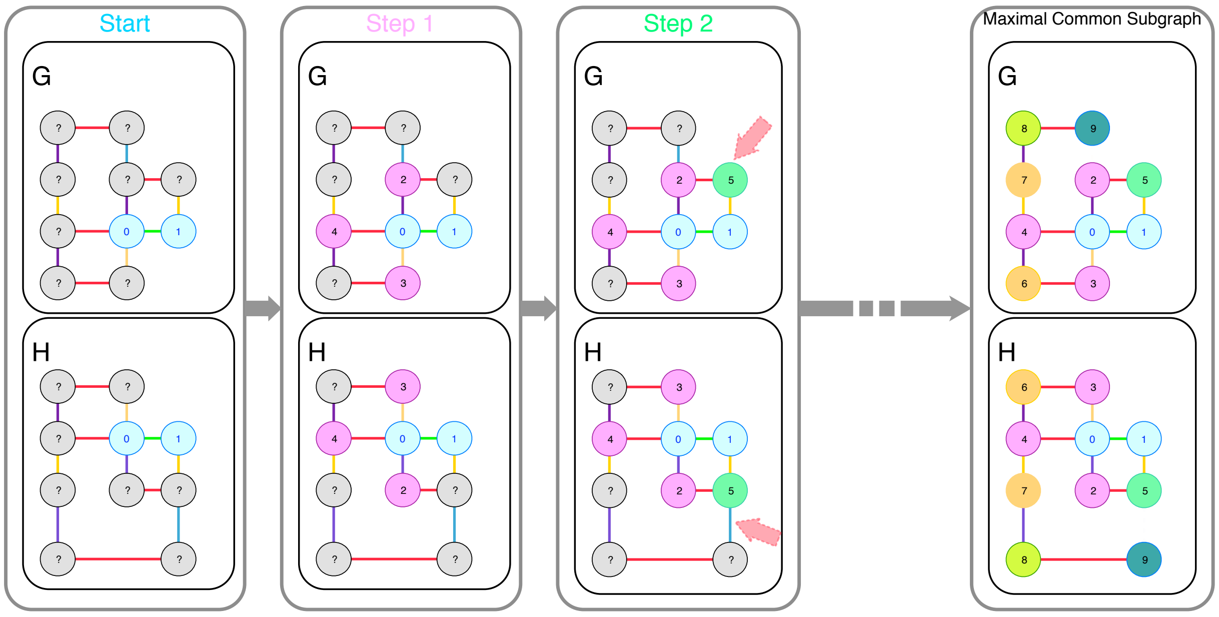

Fig. 4. Illustration of the extension process This figure illustrates the extension process from a "starting point" (here $((g_0, h_0), (g_0, h_0))$, in blue). We first consider the neighbours of g_0 and h_0 (in purple). Thanks to the PEC, there is only one way to match. We then consider the neighbours of g_1 and h_1 (in green). We match g_5 and h_5 but discover that the their neighbourhoods are not compatible. At this point the behaviours of the three algorithms differ. This discovery implies that the matching cannot be extended to cover all of G so the Graph Isomorphism and Subgraph Isomorphism will abandon it and pass on to another "starting point". The All Maximal Common Subgraphs on the contrary will take note of this discrepancy and keep extending the matching nevertheless. This extension will output a maximal common subgraph of G and H and a new branch will be created to explore the alternative solution suggested by the discrepancy found.

Our method can handle any set R that can be translated into a filtering function $f : G \rightarrow C_{RSE}$ with G a graph that shares the same properties as an RNA 2D structure graph and C_{RSE} the collection of RSEs in G that respects the rules in R.

We also designed our method to offer the possibility of providing a second filtering function $f' : G \rightarrow G'$ that takes a RNA 2D structures graphs G in the dataset and output another graph G', which is a subgraph of G without the edges and nodes in G that already infringe a rule of R (and thus have cannot possibly be part of any valid RSE). f' is optional as it only improves performances by reducing the search space.

2.5.2 Extraction of RSEs

For every pair of RNA 2D Structure Graphs in the dataset (after the application of f' if provided), we use our algorithm solving the *maximal common subgraphs* problem to extract the set of all maximal common subgraphs between the two graphs (as illustrate in Fig. 6). The filtering function f (derived from the rules defining the class of RSEs currently extracted) is applied to each maximal common subgraph found. The sets of RSEs obtained are gathered and clustered using our *graph isomorphism* algorithm. This process involves non trivial yet incidental mechanisms which we describe in section 2.2 of the the supplementary material.
Please note that our implementation relies on parallelization to improve the performances by distributing the pairs of graphs to process (cf. section 2.3 of the the supplementary material).

2.5.3 Network of RSEs

To study how similar RSEs are one to another, we organize the set of RSEs into a network $G = \{V, E\}$. A node in V represent RSE. An edge $e = \{r_1, r_2\}$ from RSEs $r_1 = (S_1, \mathfrak{O}_1), r_2 = (S_2, \mathfrak{O}_2)$, is in E iff S_1 is a subgraph of S_2. We rely on our *subgraph isomorphism* algorithm to build this network.

3 Applications & Results

In this section, we present three applications of our method to three different classes of RSEs and the corresponding results.

3.1 Dataset

All three applications use the same dataset of RNA structures: the non-redundant RNA database maintained on RNA3DHub [18] on Sept. 9th 2016, version 2.92. It contains 845 all-atom molecular complexes with a resolution of at worse 3Å. From these complexes, we retrieved all RNA chains also marked as non-redundant by RNA3DHub. The basepairs were annotated for each chain using `FR3D`. Because `FR3D` cannot analyse modified nucleotides or those with missing atoms, our present method does not include them either. If several models exist for a same chain, the first one only was considered.

To distinguish between local and long-range interactions, we define a secondary structure from the ensemble of canonical CWW interactions. This task can be ambiguous for pseudoknotted and large structures. We used the K2N algorithm [26] from the PyCogent library [10]. A case that can not be treated by K2N is when a nucleotide is annotatedÂ as having two CWW interactions. Since this is rare, we decided to keep the interaction belonging to the largest stack.

3.2 Interaction networks over 2+ SSEs

The `CaRNAval` project [22] aimed at extracting RNA structural motifs containing non-canonical base pairs and long range interactions involving exactly 2 SSEs: the *Recurrent Interaction Networks* (RINs). The algorithm developed in `CaRNAval` to extract RINs is also graph based and relies on a greedy algorithm. That algorithm generates seeds (basically minimal common subgraphs) and tries enlarges them step by step. The decision of

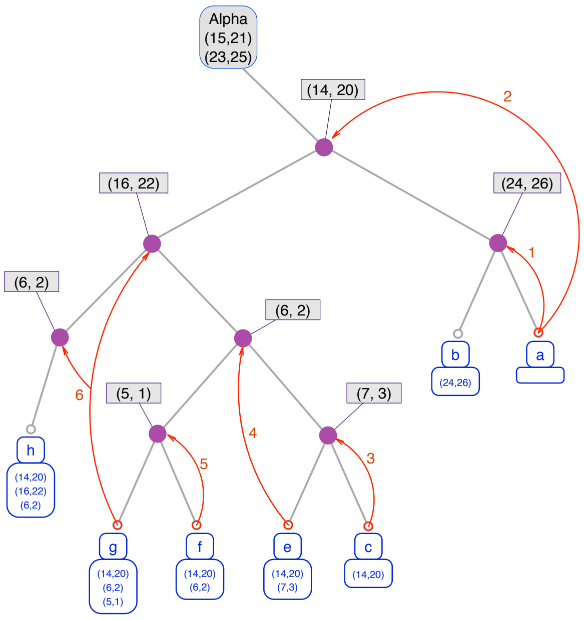

Fig. 5. Exploration tree with backtracking This figure displays the exploration tree representing a posteriori the relation between the different branches created. In this tree, the root is a starting point (i.e. the nodes that are already matched at the start of an exploration) and each leaf is a maximal common subgraph. Each path from the root to a leaf describes an exploration. For instance, the node (14,20) corresponds to the action of matching the node 14 from G to the node 20 of H. All the leafs in the right subtree have matched 14 to 20 and all the ones in the left subtree did not. Please note that only the nodes with a left child are represented, all other nodes have been collapsed since they bear no information. The first exploration always produces the right most maximal common subgraph. This exploration encountered two conflicts and the algorithm thus produced two new branches which respectively were instructed not to add (24,26) and not to add (14,20). The first of the two produced another maximal common subgraph without any trouble but the second encountered another conflict and so on and so forth.

limiting RINs to exactly two SSEs was both sound as it is a property of known motifs this project was looking for (such as A-minors for instance) but it was also necessary given the performances of the greedy algorithm. On the contrary, our method does not need such limitation: we can work with any number of SSEs and are thus able to extract more structures. Moreover, despite working on a generalization of the problem studied in CaRNAval, we still process the same dataset more than 50 times faster.

In this section, we introduce *Generalized Recurrent Interaction Networks* (GRINs), a generalization of RINs without a limitation on the number of SSEs. We then describe how we applied our generic method to the problem of extracting GRINs through the presentation of the filtering function f_{GRIN}. Finally, we compare the results of the generic method over the one initially used in CaRNAval through both the sets of structures extracted and the performances.

3.2.1 Generalized Recurrent Interaction Networks (GRIN)

The *GRINs* are a class of RSEs and a generalization of the RINs. GRINs have to include *at least* 2 SSEs while RINs had to include *exactly* 2. A GRIN is a pair $\{S, \mathfrak{O}\}$, where S is a *canonical graph* representing the interactions network while $\mathfrak{O}$ is the collection of occurrences. A GRIN, in addition to the constraints that defines RSEs, must respect the following ones:

1. each node in the canonical graph S belongs to a cycle in the undirected graph induced by S. (The undirected graph induced by S is obtained by replacing every directed edge by a undirected edge and merging those between the same nodes.)

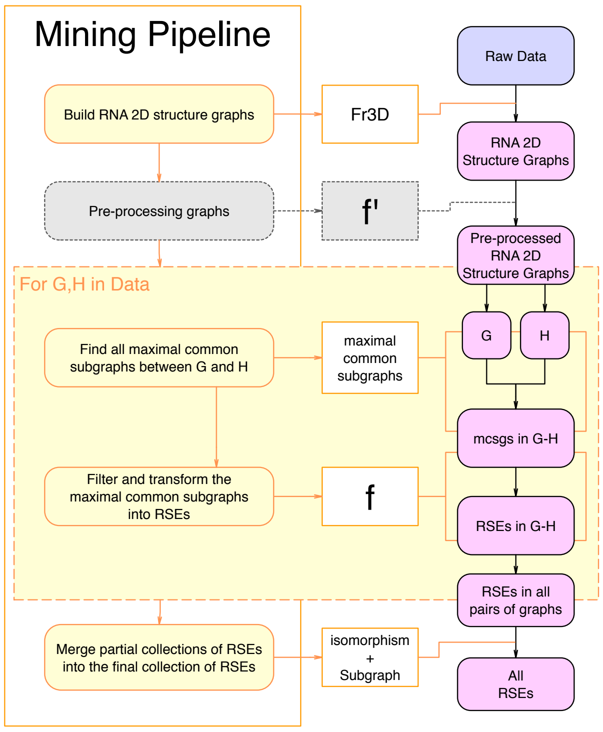
Mining Pipeline
Build RNA 2D structure graphs
Pre-processing graphs
For G,H in Data
Find all maximal common subgraphs between G and H
Filter and transform the maximal common subgraphs into RSEs
Merge partial collections of RSEs into the final collection of RSEs
Fr3D
f'
maximal common subgraphs
f
isomorphism + Subgraph
Raw Data
RNA 2D Structure Graphs
Pre-processed RNA 2D Structure Graphs
G
H
mcsgs in G-H
RSEs in G-H
RSEs in all pairs of graphs
All RSEs

Fig. 6. Simplified display of the full pipeline The RNA 2D structure graphs given as input are pre-processed for the sake of optimization. Each pair of graphs in the pre-processed data is then given to the maximal common subgraphs algorithm as input and the output is post-processed into partial sets of RSEs. All partial sets of RSEs are finally merged into the complete set of RSEs which is the output of the whole pipeline.

2. Each node in S is involved in a canonical or a non-canonical interaction (*i.e.* no nodes with only backbone interactions)
3. If two nodes, a and b in S, form a local canonical base pair, there exists a node c in S such that c is a neighbour to a or b, and c is involved in a long-range or non-canonical interaction. In other words we do not extend stacks whose nucleotides are involved in canonical base pairs only.
4. S contains at least two long-range interactions, i.e. four edges labeled as long-range since each interaction is described with two directed edges.

Each of the above constraints is justified as follows:

1. This condition is enforces the cohesiveness of the interaction network by preventing danglings that would create variations of little interest.
2. The interaction networks are intended to capture a representation of the geometry. Non interacting nucleotides do not have geometric constraints.
3. Stacks of canonical base pairs (i.e. at least two consecutive cWW with no other interaction) form the core of the structure and are either embedded in the secondary structure with little geometric variation or result from the folding of the tertiary structure (co-axial stacking between helices, loop-loop interactions or pseudo-knots) with often a larger geometric variation.
4. This is a property observed in all known interaction networks like the A-minor and the ribose zipper.

Those rules are inherited from CaRNAval and are, especially for rule #4, partially arbitrary. We conserve those rules in the definition of the GRINs, including #4, as a first step. Indeed, relaxing too many constraints in a single step would have made any comparison of the results and validation of our method difficult. However, we will be relaxing rule #4 in a second step (cf. section 3.3).

Filtering function f_{GRIN}: We translate the rules defining GRINs into the filtering function f_{GRIN} required for our pipeline by converting each rule into a corresponding filter. We provide the details of this conversion in the Sup. Mat. The rules defining the GRINs class happen to be directly transposable to the input. As a consequence we can also use f_{GRIN} as a pre-filtering function f'_{GRIN}.

3.2.2 Comparison of the results

Comparison between RINs and GRINs The original version of CaRNAval presents 337 RINs and a total of 6056 occurrences. From the same dataset, our new method has extracted 557 GRINs and a total of 7709 occurrences. Amongst the 337 original RINs, 243 are isomorphic to a GRIN. Of the remaining 94 RINs, 88 are found inside larger GRINs, i.e. the canonical graph of the RIN is a subgraph of canonical graph of at least one GRIN. To put it differently, those 88 RINs are still captured but are always found inside "larger contexts" that could not be perceived before because of the limitation on the number of SSEs. Now that we relaxed this constraint, the "larger contexts" are now captured as new GRINs that "assimilated" those 88 RINs. We elaborate further on the question of the SSEs in subsection 3.2.4. For the same reason, the numbers of observations of the 243 RINs/GRINs common to both versions have changed for 81 of them (+4 observations in average). All the signal captured by the original version of CaRNAval is present in the new results: all observations of any of those 331 RIN is covered by at least one observation of a GRIN.

The 6 last RINs are not represented in the new collection and neither are they included in a new GRIN. 4 of them were already invalid and should not have passed the filters of the previous version, their absence actually validates our method. The 2 last RINs are a special case: they both have only 2 observations with both observations inside a single RNA chain. We chose not to test a RNA 2D structure graph against itself in the new method and so are not capturing those two RINs. Please note that we could test a graph against itself, it would only require to add the constraint that "a node should not me matched with itself" to the Maximal Common Subgraph Algorithm algorithm.

3.2.3 Network of GRINs

Let us now compare the RINs networks with the GRINs ones (cf. subsection 2.5.3:*Network of RSEs*). The network formed by the RINs consists of 3 main connected components and named after a characteristic RIN they contained. They are the Pseudoknot mesh, the A-minor mesh and the Trans W-C/H mesh, respectively containing 59, 196 and 22 RINs. The remaining RINs are shared between 25 other components, each of size 1 to 4.

In contrast, the network of GRINs only has 16 components, twelve less. It suggests that the newly found interaction networks connects RINs components together. This claim is supported by the fact that, in the network of GRINs, the Pseudoknot and A-minor meshes have merged into one containing 482 GRINs. This new giant mesh contains all the elements in the two main meshes of `CaRNAval` plus another 230 GRINs. The Trans W-C/H mesh remains disconnected and gains 16 elements for a total of 38 GRINs.

3.2.4 GRINs and SSEs

The main artificial constraint on RINs was their restriction to exactly two SSEs. While biologically justificable, it allowed to strongly constrain the problem making the previous method computable on a large server. By removing this constraint, we observe GRINs covering a varied amount of SSEs. We show in Fig 7 the distribution of SSEs in the GRINs and of their occurences.

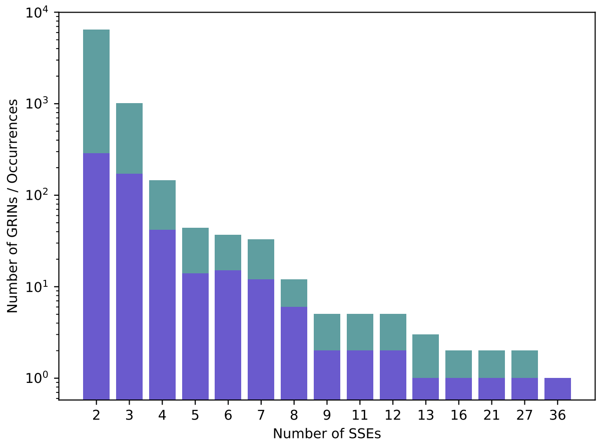

Fig. 7. Distribution of GRINs (in blue) and all their occurrences (in green) over the different numbers of SSEs.

Since GRINs can be over an arbitrary number of SSEs, occurences of the same base pair network can cover a different number of them. We show in Table. 1 that it is not the case. Out of the 557 GRINs, 435 had all of their occurences span the same number of SSEs. There are 116 that can be over two different number of SSEs, and only 6 GRINs have their occurences cover three different number of SSEs.

Variation in number of SSEs	0	1	2
Numbers of GRINs	435	116	6

Table 1. **GRINs and variation on SSEs span** For each GRIN we compute how the number of SSEs covered varies between the occurences. A value of 0 means that all occurences are over the same number of SSEs while ±1 (resp. ±2) means that the GRIN can span two different number of SSEs (resp. three).

3.2.5 Large GRINs

While the largest RIN has 26 nodes, a GRIN can potentially encompass an entire molecule. There are 64 GRINs with more than 26 nodes, amongst them 4 have above 100 nodes, the largest GRIN containing 293 nucleotides. Those new giants are found in structures of ribosomal subunits. The existence of those GRINs shows that the dataset we are using contains extremely similar structures. The RNA3DHub non-redundant RNAs can still share a considerable portions of their geometry, on up to 293 connexe nucleotides. As a consequence, we might have to update our method, either by modifying our definition of GRIN to limit their size or by adding an additional screening to the dataset.

3.3 Generic Interaction Networks (GINs)

In the previous section we created GRINs as a generalization of RINs. A natural way to relax even further the problem is to remove the constraint on the 2 required long range interactions. We call *Generic Interaction Networks* (GINs) the class obtained from the GRINs by removing rule #4 (cf. definition of GRINs in 3.2.1). While it is a simple modification, trivial to implement, the search space increases drastically.

3.3.1 Collection of GINs

Our methods finds 920 GINs for a total of 12 239 occurences. All 557 GRINs have their canonical graph isomorphic to the canonical graph of a GIN. The RINs to GRINs transition was done by allowing more than 2 SSEs, which opened the possibility of finding new larger "including" structures. In contrast removing the constraint on the number of long range interaction does not.

We show in Fig. 8 the distribution of the GINs and of their occurences in function of the number of long range interactions they have. Amongst the remaining 363 GINs, 222 contain no long range interaction and 141 have exactly 1. Those represent 39% of the GINs and 37% of the occurrences.

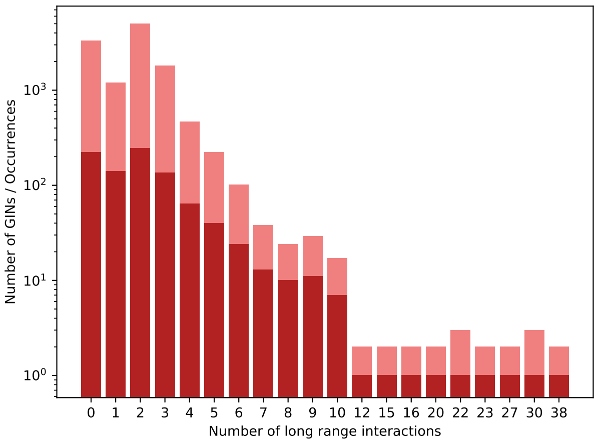

Fig. 8. Distribution of GINs (in red) and all their occurrences (in rose) over the different numbers of long range interactions they contain.

In Fig. 9 we show the distribution of the number of SSEs that are covered by the GINs. Compared to previously, most GINs span two SSEs. This shift from the previous, more constrained, results is due to the 222 GINs with no long range.

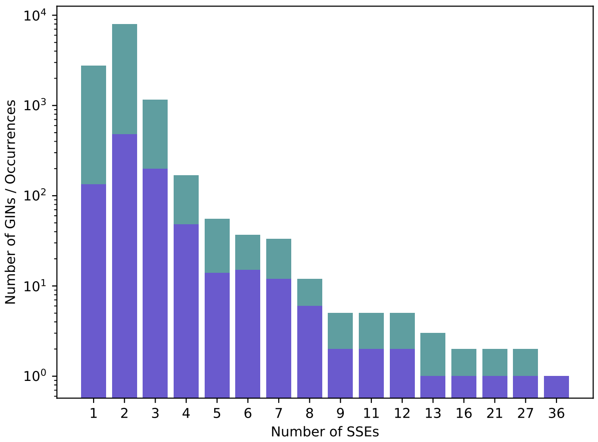

Fig. 9. Distribution of GINs (in blue) and all their occurrences (in green) over the different numbers of SSEs.

As previously, for any given GIN the occurences span a consistent number of SSEs. As we show in Table 2, the same trend as for the GRINs is followed.

Variation in number of SSEs	0	±1	±2
Numbers of GINs	754	159	7

Table 2. Variation in the number of SSEs over the occurrences of the same GIN (Cf. Table 1). Those numbers show that the variation in the number of SSEs amongst the occurrences of GINs is both uncommon and limited, even more than with GRINs, albeit slightly (82% of GINs with no variation vs 78% of GRINs).

3.3.2 Network of GINs

Almost all the structures are connected together. There are 888 GINs connected toge ther in a giant component. Of the remaining 32 GINS, twenty three are singletons, and it remains 6 mini components.

3.3.3 Performances

Reproducing the CaRNAval dataset we tested the validity of our method and its performances. As all the RINs found and all their occurrences were present in the collection of GRINs, it shows that our method captured strictly more signal than the previous one. In term of performances, the runtime dropped from around ~330 hours to ~200 minutes (both are total runtime over the same 20 cores, for the same dataset), despite solving a more general problem. Relaxing the problem to a maximum by computing the GINs took 19 hours, to analyze the entire non-redundant database of RNA structures.

3.4 Applications to RNA 3D module-based RNA structure prediction

As described earlier, the methods described in this paper were implemented with rules independent from the isomorphism, subgraph and maximal common subgraph algorithms, in order to allow some modifications of the rules to extend the range of applications of caRNAval.

This modularity can namely be applied to the problem of RNA 3D structure prediction. *RNA 3D modules* are small RNA substructures involved in structural organization and ligand binding processes. They can be defined with rules similar to the ones describing RINs, with two major differences. First, RNA modules do not need to include long range interactions, and many of the well characterized modules are entirely local, namely the kink-turn and g-bulged modules. Second, unlike RINs, RNA modules are defined by both their structure and sequence profiles rather than exclusively the former.

RNA 3D modules can be leveraged in the prediction of a full 3D structure. The fragment-based method implemented by Parisien and Major in MC-Sym[17] constructs a full 3D structure from an augmented secondary structure by mapping the components of this secondary structure to a database of 3D structure fragments. The prediction of 3D modules has been shown to improve this class of methods by providing more informative fragments, namely in RNA-MoIP[21]. Further progress has since been made in this direction with recent improvements in RNA 3D modules identification in sequences[29][24].

The main limitation of this type of method remains the difficulty of assembling a strong dataset of modules. RNA modules are typically identified by searching RNA 3D structure for recurrent subgraphs, a task to which caRNAval should be able to contribute. Unfortunately, as of now, no fragment-based method has been able to integrate long-range modules into a 3D structure prediction pipeline, and the published version of caRNAval cannot be applied to the discory of common subgraphs without long range interactions as its execution time would explode.

However, the modularity of the methods previously presented, as well as the improved complexity allow for the tackling of this problem. The implemention of those methods constitutes the first software able to discover both long-range and local RNA modules and as such, a significant step towards more accurate fragment-based prediction of 3D structure from sequence.

4 Conclusion

In this paper we present a novel algorithm that can find arbitrarily large recurrent structural elements (RSEs) between two RNA structures, represented as graphs. By leveraging a proper edge coloring of those graphs, we improve drastically on previous methods, and allow for the first time to identify modules arbitrarily large.

We show that we are a hundred time faster than our previous method, `CaRNAval`. The gain in efficiency allows to relax the constraints and search for generic interaction networks (GIN), which can span any number of SSEs, and have any number of long range interaction, even none.

In `CaRNAval` the network of found modules had three clear main components. We show that the network of found GINs is a massive components linking together more that 95% of the recurrent structures together. This can be key to understand how those structural feature emerged and where propagate, or for the design of artificial RNAs.

References

[1] Alberto Apostolico, Giovanni Ciriello, Concettina Guerra, Christine E. Heitsch, Chiaolong Hsiao, and Loren Dean Williams. Finding 3D motifs in ribosomal RNA structures. *Nucleic Acids Research*, 37(4):e29, 2009.

[2] Sri Devan Appasamy, Hazrina Yusof Hamdani, Effirul Ikhwan Ramlan, and Mohd Firdaus-Raih. InterRNA: a database of base interactions in RNA structures. *Nucleic acids research*, 44(D1):D266–D271, 2015.

[3] Grzegorz Chojnowski, Tomasz Waleń, and Janusz M Bujnicki. RNA Bricks—a database of RNA 3D motifs and their interactions. *Nucleic Acids Research*, 42(D1):D123–D131, 2014.

[4] José Almeida Cruz and Eric Westhof. Sequence-based identification of 3D structural modules in RNA with RMDetect. *Nature methods*, 8(6):513–519, 2011.

[5] Mahassine Djelloul and Alain Denise. Automated motif extraction and classification in RNA tertiary structures. *RNA*, 14(12):2489–2497, 2008.

[6] Carlos M Duarte, Leven M Wadley, and Anna Marie Pyle. RNA structure comparison, motif search and discovery using a reduced representation of RNA conformational space. *Nucleic Acids Research*, 31(16):4755–4761, 2003.

[7] Patrick Gendron, Sébastien Lemieux, and François Major. Quantitative analysis of nucleic acid three-dimensional structures. *Journal of molecular biology*, 308(5):919–936, 2001.

[8] Anne-Marie Harrison, Darren R South, Peter Willett, and Peter J Artymiuk. Representation, searching and discovery of patterns of bases in complex RNA structures. *Journal of computer-aided molecular design*, 17(8):537–549, 2003.

[9] Hung-Chung Huang, Uma Nagaswamy, and George E Fox. The application of cluster analysis in the intercomparison of loop structures in RNA. *RNA*, 11(4):412–423, 2005.

[10] Rob Knight, Peter Maxwell, Amanda Birmingham, Jason Carnes, J Gregory Caporaso, Brett C Easton, Michael Eaton, Micah Hamady, Helen Lindsay, Zongzhi Liu, Catherine Lozupone, Daniel McDonald, Michael Robeson, Raymond Sammut, Sandra Smit, Matthew J Wakefield, Jeremy Widmann, Shandy Wikman, Stephanie Wilson, Hua Ying, and Gavin A Huttley. PyCogent: a toolkit for making sense from sequence. *Genome Biology*, 8(8):R171, 2007.

[11] Neocles B Leontis, Jesse Stombaugh, and Eric Westhof. Motif prediction in ribosomal RNAs lessons and prospects for automated motif prediction in homologous RNA molecules. *Biochimie*, 84(9):961–973, 2002.

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

[13] Neocles B Leontis and Eric Westhof. Analysis of RNA motifs. *Current opinion in structural biology*, 13(3):300–308, 2003.

[14] Aurélie Lescoute, Neocles B Leontis, Christian Massire, and Eric Westhof. Recurrent structural RNA motifs, isostericity matrices and sequence alignments. *Nucleic Acids Research*, 33(8):2395–2409, 2005.

[15] Aurélie Lescoute and Eric Westhof. The A-minor motifs in the decoding recognition process. *Biochimie*, 88(8):993–999, 2006.

[16] Aurélie Lescoute and Eric Westhof. The interaction networks of structured RNAs. *Nucleic Acids Research*, 34(22):6587–6604, 2006.

[17] Marc Parisien and Francois Major. The MC-Fold and MC-Sym pipeline infers RNA structure from sequence data. *Nature*, 452(7183):51, 2008.

[18] Anton Petrov. *RNA 3D Motifs: Identification, Clustering, and Analysis*. PhD thesis, Bowling Green State University, 2012.

[19] Anton I Petrov, Craig L Zirbel, and Neocles B Leontis. WebFR3D—a server for finding, aligning and analyzing recurrent RNA 3D motifs. *Nucleic acids research*, 39(suppl_2):W50–W55, 2011.

[20] Anton I Petrov, Craig L Zirbel, and Neocles B Leontis. Automated classification of RNA 3D motifs and the RNA 3D motif atlas. *RNA*, 19(10):1327–1340, 2013.

[21] Vladimir Reinharz, François Major, and Jérôme Waldispühl. Towards 3d structure prediction of large rna molecules: an integer programming framework to insert local 3d motifs in rna secondary structure. *Bioinformatics*, 28(12):i207–i214, 2012.

[22] 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, 2018.

[23] Karen Sargsyan and Carmay Lim. Arrangement of 3D structural motifs in ribosomal RNA. *Nucleic Acids Research*, 38(11):3512–3522, 2010.

[24] Roman Sarrazin-Gendron, Hua-Ting Yao, Vladimir Reinharz, Carlos Gonzalez Oliver, Yann Ponty, and Jerome Waldispuhl. Stochastic sampling of structural contexts improves the scalability and accuracy of rna 3d modules identification. *bioRxiv (accepted to RECOMB 2020)*, page 834762, 2019.

[25] 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(1):215–252, 2008.

[26] Sandra Smit, Kristian Rother, Jaap Heringa, and Rob Knight. From knotted to nested RNA structures: a variety of computational methods for pseudoknot removal. *RNA*, 14(3):410–416, 2008.

[27] Leven M Wadley and Anna Marie Pyle. The identification of novel RNA structural motifs using COMPADRES: an automated approach to structural discovery. *Nucleic Acids Research*, 32(22):6650–6659, 2004.

[28] Cuncong Zhong, Haixu Tang, and Shaojie Zhang. RNAMotifScan: automatic identification of RNA structural motifs using secondary structural alignment. *Nucleic Acids Research*, 38(18):e176–e176, 2010.

[29] Craig L Zirbel, James Roll, Blake A Sweeney, Anton I Petrov, Meg Pirrung, and Neocles B Leontis. Identifying novel sequence variants of rna 3d motifs. *Nucleic acids research*, 43(15):7504–7520, 2015.