# Computational methods for structural and functional annotation of RNA sequences

Literature review

Roman Sarrazin-Gendron
Supervisor : Jérôme Waldispühl

January 10, 2020

## 1 RNA biology

Ribonucleic Acid (RNA) is a nucleic acid, a family which also includes DNA. It is essential to life and found in every living cell. RNA molecules are chains of monomers known as nucleotides, attached to a rigid phosphate backbone. RNA can thus be represented as a sequence of the four types of nucleotides: adenosine (A), cytosine (C), guanine (G) and uracil (U). In its most denatured state, RNA is an unstructured chain, and its nucleotides are only linked through the backbone, allowing a string representation, known as sequence. Unlike the double stranded DNA, it is most often found single stranded and folded on itself. To achieve its function, the RNA strand folds into a specific 3D structure closely linked to that function and determined by its sequence. First, compatible nucleotides form sets of hydrogen bonds. These very stable interactions are called canonical base pairs, which form a scaffold of stems and loops, known as secondary structure. Then, non-canonical interactions arise and define a tertiary structure.

In this review, computational approaches to infer functional information from RNA sequences will be presented, with a focus on methods that leverage the identification of specific substructures associated with functions of interest.

## 2 RNA structure and structural modules

When folding into a three-dimensional structure, the nucleotides that are not part of canonical base pairs tend to form non-canonical base pairs, building networks of interactions that stabilize the RNA and define the 3D structure. Therefore, most of the information specific to the 3D structure is associated to unpaired regions of the secondary structure, known as loops. We refer to the networks of non-canonical base pairs formed by the nucleotides of a loop as RNA structural modules, key building blocks of the three-dimensional structure. Unfortunately, non-canonical base pairs are quite unstable, and their energies could not be quantified experimentally. Consequently, while energy models perform well in secondary structure prediction tasks, they cannot currently solve the folding problem beyond that secondary structure. Hence, computational approaches to predict a full structure from sequence, one of the key problems in the field of RNA function prediction, are typically statistical in nature.

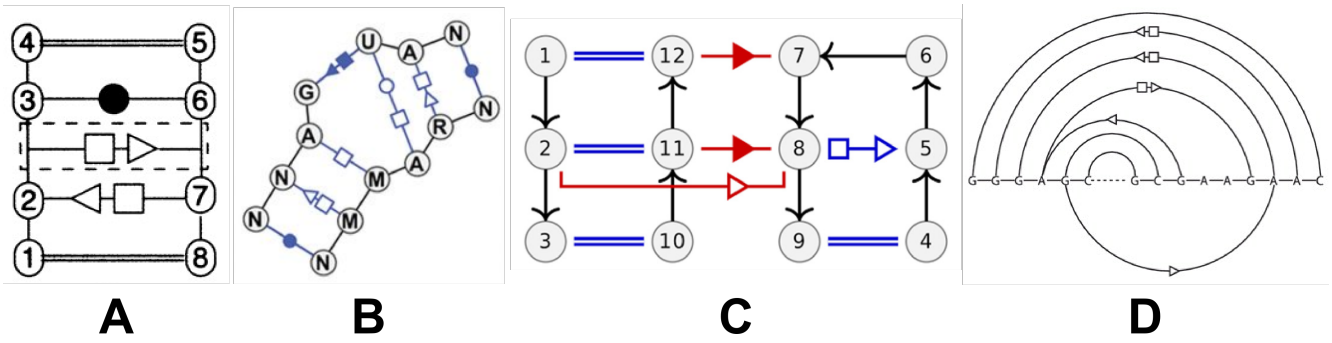



Figure 1: RNA structural modules in A) `Rna3dMotif` [1], B) `RNA 3D Motif Atlas` [2], C) `CaRNAval` [3], D) `RNAMotifScan` [4]. Edge labels (shapes) represent the base pair type.

A first step towards a solution could be to start from a secondary structure obtained through energy-based methods, and annotate its loops with structural module predictions from statistical methods. This process requires identification, classification and prediction of RNA modules.

## 2.1 Modeling RNA structural modules

In order to classify RNA modules, some consistent representation allowing comparisons is required.

The most common representation is an undirected graph, in which nodes encore nucleotides and edges, interactions between nucleotides. Those interactions can represent any type of molecular contact, namely backbone adjacency, canonical and non-canonical base pairs, as well as various types of stackings. Base pairs and stacking types are indicated with specific labels based on their orientation. This compact notation is namely used in `Rna3Dmotif` [1], the `RNA 3D Motif Atlas` [2], and in `caRNAval` [3], as seen in Figure 1. This model is the most flexible in terms of the range of module types it can encode.

The representation used in `RNAmotifscan` [4] is slightly different. That approach considers a motif as a sequence of nested base pairs (see Figure 1, section 2.2.5), allowing some crossing base pairs. This method has the upside of allowing fast computational methods when there are few crossing base pairs, at the cost of not being able to efficiently model structures with many crossing base pairs.

## 2.2 Identifying structural modules in RNA structures and clustering them

The first core task associated with RNA modules is to build a dataset of functionally significant modules from known RNA 3D structures. This can be done manually for some specific modules, but there are thousands of known RNA structures and hundreds are crystallized every year. At this scale, automated methods are necessary. All the methods described in this section rely on the same core approach and assumption: a single substructure appearing in many different full structures would be very unlikely to be found by chance. Thus, by converting atomic structures to the type of graph described in section 2.1, we can formulate the problem of finding recurrent subgraphs in a database of large graphs.

### 2.2.1 Classifying by largest common subgraph `Rna3Dmotif`

The first method for identifying modules in RNA structures was presented in `Rna3Dmotif` [1], by Djelloul and Denise in 2008. This software aims at identifying functionally significant tertiary structure interactions in secondary structure loops. This is achieved in two steps:

1. Extract all secondary structure loops from all known RNA structures
2. Compare the loops using some similarity function, cluster them, and label some clusters as *functional modules* to build a catalogue.

Loops are first identified in the secondary structure of all known 3D structures. Then, for each loop, the non-canonical base pairs present in that region of the full 3D structure graph were added. The loops were compared all-to-all with a similarity scored based on the computation of the largest extensible common non-canonical subgraph (LECNS), inspired from Valiente's algorithm for graph isomorphism (2002) [5]. This similarity function is defined as follows:

$$\mathrm{sim}(G_1, G_2) = \begin{cases} \frac{\|\mathrm{LECNS}(G_1,G_2)\|}{max(\|G_1\|,\|G_2\|)} & \text{if } \|\mathrm{LECNS}(G_1, G_2)\| > 1 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Where $G_1$ and $G_2$ are the compared graphs, and $\|G\|$ denotes the non-canonical size of a graph $G$, defined as its number of non-canonical edges. The sim function is thus an estimation of the proportion of non-canonical vertices in common between the graphs.

The clustering starts with hierarchical clustering with average linkage based on $\mathrm{sim}(G_i, G_j)$. Then, four experimentally well-known module clusters were used as references to define a minimum similarity threshold to separate tentative clusters. A representative common subgraph, including all non-canonical edges shared by more than half of the members of the cluster, was built for each cluster. Finally, this consensus graph was compared to all input loops to retrieved missed candidates that would belong in the cluster. The 3D root mean-squared deviation (RMSD) of atomatic positions was used as validation. RMSD simply measures the average distance between each atoms of two superposed molecules. Djelloul and Denise showed their graph-based clustering identified clusters of structures with low RMSD, and released the first RNA module catalogue.

### 2.2.2 Identifying maximum cliques with the `RNA 3D Motif Atlas`

The `RNA 3D Motif Atlas` (Petrov et al, 2013) [2] builds on this result by tackling the same problem of grouping secondary structure loops by non-canonical interaction patterns.

Rather than counting shared edges in graphs, the `RNA 3D Motif Atlas` relies on all against all structural alignments in 3D using an iterative geometrical search. To focus on the most important features, nodes that are not part of a base pairing or stacking interaction are not considered. The alignment results are organized in a square matrix of size $NxN$, where $N$ is the number of loops to cluster. This matrix is then converted to a large graph where each node is an entry of the matrix, and is linked by a weighted edge to all the entries it could be aligned to. Subgraphs of maximally connected nodes are known as cliques, and identifying clusters of loops in such graph equates to finding the largest cliques. This is done using the algorithm presented in `R3D Align` (Rahrig et al, 2010) [6]. This iterative method finds the largest clique, removes all members of the clique from the main graph and looks for the next largest clique. If two cliques have the same size, the tie is broken by taking the most tightly connected one.

This process, more consistent than the previous approach with its structural alignment, allows for tighter clusters; while non-canonical network graphs are predictive of the 3D structure, the best alignments are still obtained directly from 3D comparison. However, 3D comparison remains computationally expensive, and can only be performed accurately on small graphs, being restricted to specific types of local loops. The `RNA 3D Motif Atlas` is available online, and is updated every month as new RNA structures are released.

### 2.2.3 Computing maximum subgraph isomorphisms with `caRNAval`

All approaches presented so far are limited to predicting networks of immediately adjacent interactions. However, many key base pairs occur between remote nucleotides. Reinharz et al were the first to tackle the problem of identifying recurrent long range interaction cycles in RNA structures. Their algorithm, `CaRNAval` [3], released in 2018, uses the same graph representation as `Rna3Dmotif` (Figure 1), but leverages an improved mining method: given two structure graphs, the algorithm aims at finding the maximal common base pair network, or maximal subgraph isomorphism, a NP-hard problem. The solution presented is exponential in the worst case, but performed well enough on this specific problem to be executed on all known structures.

Before execution, specific labels are added to edges joining distant nucleotides in the full structure graphs. Then, for two full structure graphs $G_1$ and $G_2$, we start from a small common subgraph of $G_1$ and $G_2$. Then, the subgraph is iteratively extended one edge at a time by a neighboring edge present in both $G_1$ and $G_2$ until it can no longer be extended. Because long range interactions are specifically targeted, the initial common subgraph is the list of shared edges that are labeled as connecting distant nodes. A maximal common subgraph fitting the requirements of containing at least two long range interactions, and constituting a cycle is returned if any is found. This is repeated for all pairs of graphs.

With this method, `CaRNAval` was used to parse 845 structures, and identified 337 recurrent interaction networks adding up to 6056 occurrences. While expensive in cost in theory, and limited in scope to long-range interaction networks that constitute cycles, `CaRNAval` is highly complementary to the `RNA 3D Motif Atlas`.

### 2.2.4 Alignment-based clustering with `RNAMotifScan`

Breaking rank with methods based on common subgraph comparison, `RNAMotifScan` (Zhong et al, 2010) [4] uses a linear representation of canonical and non-canonical base pairs (Figure 1). The main upside of this method is its compatibility with traditional alignment-like dynamic programming algorithms. These algorithms allow for quantitative alignment score, which can be leveraged for distance-based clustering.

`RNAMotifScan` aims at assigning an alignment score to two RNA loops by solving two subproblems:

- Matching base pair types and base pairing nucleotides with respect to the compatibility of the physical properties of each type of base pair.
- Matching the nucleotides that do not participate in base pairs.

The first subproblem ensures that graphs that are aligned together do have a similar 3D structure. The 3D structure of a base pair is determined by its base pair type and its nucleotides. This fulfills the main objective of clustering structural modules. The second subproblem equates

to a simple sequence alignment problem: if two loops have the same 3D structure, then their sequences can be expected to be alignable to some degree. If that is not the case, it is likely the wrong base pairs have been aligned. Here, we will present a simplified, high level version of the algorithm.

All RNA loops have a base pair between the first and last nucleotide, and everything else can be considered "nested" within this base pair. In this context, we call enclosed base pair the base pair that is nested within the enclosing base pair. When aligning a structural loop query to a loop of interest, the dynamic programming algorithm recurses through enclosed base pairs until no match can be found, and then backtracks. At each steps, it executes as follows:

- Case A: at least one loop has no enclosed base pair within the enclosing base pair.
  - assign a sequence alignment score.
- Case B: within the enclosing base pair is at least one enclosed base pair, for both loops. For all pairs of enclosed base pairs (one per loop):
  - Assign a score to the similarity between the two base pairs.
  - Assign a sequence alignment score for the regions inside the enclosing base pair but outside the enclosed base pair. If there is a positional asymmetry between the two loops, apply a penalty.
  - Recursion; call the algorithm on the pair of enclosed base pairs.

While this method is elegant and appears to have more sensitivity than the state of the art, it has the same limitations as most dynamic programming algorithms applied to RNA: an expensive computational cost ($O(n^2m^2)$ where $m$ and $n$ are the number of base pairs of the two compared loops), and a relative lack of flexibility compared to statistical methods. The authors are currently working on more situational weights and penalties, as the one-size-fits-all presented in the paper leads to errors.

### 2.2.5 Graph-based alignment-free clustering with `GraphClust`

So far, the approaches presented require RNA modules to be cycles when clustering them. It must also be stated that none of these approaches scales particularly well, as they cannot avoid quadratic complexity. In practice, however, they should eventually be applied to thousands of structures. To specifically address the issue of fast and accurate clustering RNA sequence-structure pairs, Heyne et al presented a new approach, `GraphClust` [7] for fast structural clustering in 2012. Their method is designed to work on sequences with unknown structures by rapidly predicting a representative sample of potential secondary structures, but the same pipeline can be applied to short sequences with known secondary structures, which is the data type tools like `Rna3Dmotif` and the `RNA 3D Motif Atlas` use. `GraphClust` then converts the structured sequences to graphs, similar to the ones used in those two methods. The core novelty of the approach is how those graphs are used; rather than using alignment or enumeration-based comparisons, fast clustering is performed on vector representations of those graphs. In particular, a neighborhood subgraph pairwise distance kernel (`NSPDK`), suitable for large datasets with discrete labels, is used.

The `NSPDK` operates over all pairs of neighborhood subgraphs between the two graphs. Here, the neigborhood subgraph of a given graph $G(V, E)$, $N_r^v(G)$, is a subgraph of $G$, rooted in $v$, including all vertices at distance$\leq r$ from the root, where $r$ is some integer $\geq 0$ describing the

radius of the neighborhood (Figure 2a). We define a neighborhood-pair relation $R_r, d$, which holds when the distance between the roots of two neighborhood subgraphs of the same radius $r$ is exactly equal to some integer $d$. Here, the distance is simply the length of the shortest path between two nodes. To count isomorphic subgraphs, we can then define a kernel based on the inverse of the relation $R_{r,d}$. This new relation returns all possible pairs of subgraphs respecting $r$ and $d$. A kernel iterating over all instances given by a relation is referred to as a *decomposition kernel*.

$$K(G, G^{'}) = \sum_{r} \sum_{d} \sum_{\substack{A,B \in R_{r,d}^{-1}(G) \\ A^{'},B^{'} \in R_{r,d}^{-1}(G^{'})}} \mathbf{1}(A \cong A^{'}) \mathbf{1}(B \cong B^{'})$$

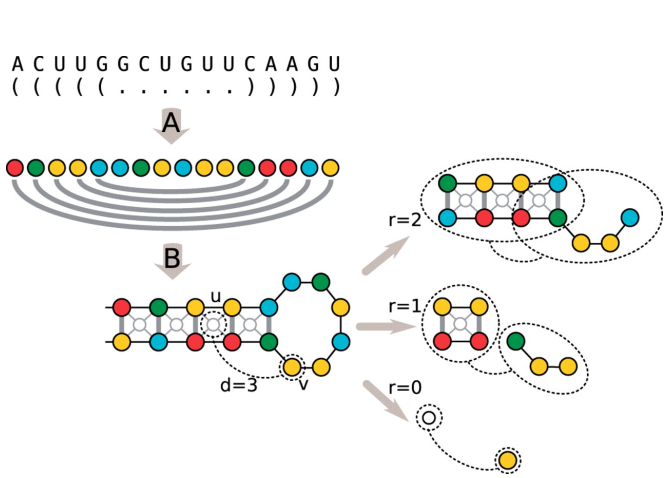
Where $\mathbf{1}(A \cong A^{'})$ is an indicator expression returning 1 if $A$ and $A^{'}$ are isomorphic. This brings back the issue of computing an isomorphism, which is computationally expensive. To tackle this issue, the authors use an efficient method to encode graphs as strings based on the distances between pairs of vertices. If two graphs are isomorphic, they will have the same code. This idea, introduced by De Grave and Costa in 2010 [8], is then further pushed to an explicit feature encoding to a sparse vector, where each feature key is an integer encoding a pair of subgraphs with $r < r^*$ at distance $d < d^*$), and the value associated with that key is the count of occurrences of pairs identical to that one . This remains efficient because the number of features is not exponential in the number of nodes, as $r$ and $d$ can be bound maximum values $r^*$ and $d^*$.

To map this vector to an easy to cluster integer, the authors suggest an idea based on approximate nearest neighbor search solved by locality-sensitive hashing. A hash function is considered locality-sensitive when the probability of a collision is higher for objects that are close-by than for objects that are far apart. All sparse vectors were binarized by setting all non-null components to 1 to allow for the computation of a Jaccard similarity, defined as the ratio of the number of features the instances have in common over the total number of features. In this context, such metric is sufficient because the vectors are extremely sparse, which means having common features is very significant. A min-hash function is then built, with a set of random hash functions mapping integers to integers. Conveniently, the min-hash function is an unbiased estimator for Jaccard similarity and can be computed in $O(1)$, by checking, for two integers $x$ and $y$, what proportion of the time $h_i(x) = h_i(y)$ over $N$ hash functions $h_i$. The k-neighborhood of some RNA vector is then assembled by finding a set of vectors that consistently return the same min-hash values, then sorting this set by `NSPDK` distance, and returning the $k$ most similar.

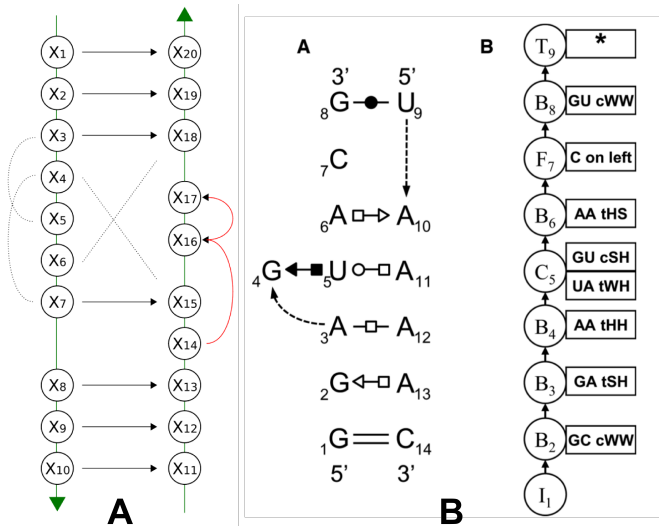
This very fast method which executes in linear time constituted a great improvement over the state of the art for clustering sequences, and illustrates the vast potential contributions of graph embeddings to the task of RNA structure clustering. However, it does have the limitation of only working on short consecutive sequences, which means it cannot represent long range modules mined by `CaRNAval`. It also only accepts nested base pairs, which means it cannot leverage the information from non-canonical base pairs. For these reasons, `GraphClust` does not render the previously mentioned algorithms for 3D comparison and graph comparison methods obsolete.

## 2.3 Identifying structural modules in RNA sequences

In the previous section, we have described methods to discover structural modules in 3D structures and classify them. This section focuses on the connected task of identifying modules in sequences, in order to help predict its 3D structure or infer functional knowledge.




(a) `GraphClust` [7] graph representation and subgraph extraction.




(b) Graphical models in A)`RMDetect` [9] and B) `JAR3D` [10].

Figure 2: Overview of the core graph-based models in `GraphClust`, `RMDetect` and `JAR3D`.

### 2.3.1 RNA modules as Bayesian networks in `RMDetect`

Cruz and Westhof presented the first method for identifying RNA modules in sequences, `RMDetect` [9], in 2011. Their approach to estimate the probability $\Pr(\theta \mid \omega)$ of the presence of a module $\theta$ in a given sequence $\omega$ relied on a statistical model encoding the sequence signal and dependence relationship between the partners of a base pair. An intuitive way to leverage this relationship is by converting the graph representation of a module to a Bayesian network, where each node has a probability distribution over 4 nucleotides conditioned over its parents. Bayesian networks must be directed, so all edges were given a direction matching the natural order of the sequence (Figure 2b). The parameters of the distribution at each node are learned from a multiple sequence alignment, a list of aligned sequences known to adopt the typical structure of the same module. Then, an input sequence and secondary structure are scanned, and nucleotides from the regions for which the secondary structure matches the architecture of the module are scored with the Bayesian network to evaluate how well they can accommodate the base pairs required to form the module. This approach scales linearly in the number of modules and requires a lot of manual work in the model building, which makes it poorly suited for full genome scans. However, it was able to identify new occurrences of a few important modules and established the ground work for the task of RNA module identification in sequence.

### 2.3.2 Discovering new variants of known structural modules with `JAR3D`

While `RMDetect` laid important ground work in the field, its core relies on the assumption that all instances of a multiple sequence alignment associated with a modules adopt the same 3D conformation. Unfortunately, this is not true, and further work needs to be done to only include sequences that can adopt the characteristic 3D structure of the module when training model.

Zirbel et al tackled this issue in 2015 with `JAR3D` [10]. This software takes advantage of the identification and classification work performed by the `RNA 3D Motif Atlas`, developed by the same group. Starting from clustered and aligned modules, `JAR3D` aims at predicting whether an

input loop matches one of the entries of the RNA 3D Motif Atlas, and provide a score quantifying the quality of that match. Another issue with the RMDetect model needs to be addressed: the requirement for the graph to be directed. This direction doesn't directly affect the results, but it causes the last nodes of the graph in sequence order to have a lot more parents than others, and require a large number of sequences to satisfyingly fill the multi-dimensional table. This is not justified by biology, as there is no reason for nodes in the second half of the sequence to have significantly more parents than nodes in the first half. JAR3D tackles this by using a different type of model, a hybrid of stochastic context-free grammars (SCFG) and Markov random fields (MRF). A grammar can be described as a set of rules used to generate sequences, in this case sequences of nucleotides. A stochastic grammar has probabilistic rules rather than deterministic. A grammar is considered context-free when the generation of an element does not depend on its position in the sequence. In this context, the SCFG includes a set of rewrite rules used to generate structural module components, and then a structural module as a sequence of components like base pairs or clusters (Figure 2b). Those clusters cannot, however, be fully modeled by the SCFG, since they include interactions between multiple nucleotides, like base triples or non-nested base pairs. Clusters of interacting nodes cannot match rules assuming nested base pairs, but a MRF is suitable to model them. MRFs are undirected graphs designed to represent dependence relationships that directed models cannot represent, particularly between highly connected subgraphs, which corresponds to the nucleotide clusters SCFGs cannot represent, making the two model types highly synergistic. These SCFG/MRF models are then trained on the RNA 3D Motif Atlas information on sequence variability and isostericity. Isostericity, in the context of RNA, denotes the measurement of the variability in the 3D structure of a base pair depending on the nucleotides forming it. Each of the 12 existing classes of base pairs is represented by a $4x4$ matrix storing the 3D discrepancy between each possible nucleotide pair and the pair that is actually observed in the 3D structure of reference. This information is used to learn the probability of each pair of nucleotides, or how frequently they should be generated at this position. Once the models are trained, the software compares an input sequence to a stored model by attempting to generate the input sequence with the model, and returning a score associated with the relative probability of the model generating this sequence.

JAR3D can detect more sequence instances of modules than RMDetect, and causes fewer false positives. It is also more flexible in the architectures it can encode with its ability to "remove" nucleotides that do not contribute to the 3D structure of interest from the model, whereas RMDetect requires continuous sequences. However, JAR3D is limited by its data; by design, it can only work with the modules included in the RNA 3D Motif Atlas, which only include the most simple classes of local RNA substructures. Moreover, it cannot be used to search sequences for the loops of interest. It is a pure scoring tool that requires the loop as a direct input.

### 2.3.3 Efficiency improvements with metaRNAmodules and genome-wide applications

Due to the limitations of RMDetect and JAR3D, there was still a need for a method to search for many modules on many sequences. Theis et al presented two approaches to solve this problem. First, with metaRNAmodules [11], a method for automated and fast identification of RNA structural modules, they addressed the issue of required manual operations upstream of executing RMDetect. metaRNAmodules parses the 3D structure space for loops, extracts and compares those loops and obtains a set of internal loops, a specific type of local RNA module. At this step, it is close to a subset of a Rna3Dmotif dataset. Then, the pipeline maps the clustered structures to existing multiple sequence alignments, and automatically trains RMDetect models on those alignments,

putting this software tool on comparable footing to JAR3D in terms of automation. However, neither software was scalable enough to tackle genome search at this point.

This is the second problem Theis and al attempted to tackle in 2015, by releasing a pipeline for genome-wide module mining [12]. The input to this pipeline is a long, full genome multiple sequence alignment of $N$ sequences. After some pre-processing of the alignment to remove evolutionarily insignificant regions, the alignment is sliced into windows of size 40 to 120, in order to achieve an order of length that is solvable by energy models, which cannot reliable predict secondary structure for sequences longer than 150 nucleotides. Each window is evaluated with RNAz, a software for the prediction of structured RNAs (presented in section 3.2). Regions that are likely to adopt conserved structural modules and present sufficient predicted thermodynamic stability are flagged for further analysis. At this point, 142517 windows have been selected. RNAz provides a consensus structure for the structured regions it identifies. Local loops are extracted from this structure and scored against a wide dataset of modules, including the entirely of the RNA 3D Motif Atlas, using RMDetect and JAR3D. Modules identified by either or both software are returned.

This pipeline was tested against shuffled windows, and showed a significant enrichment of structural modules in the RNAz-identified regions. The joint prediction of the same module by RMDetect and JAR3D significantly reduced the false discovery rate, making this pipeline a promising proof of concept for genome-wide applications of RNA module mining.

# 3 Fragment-based RNA 3D structure and gene function prediction from sequence

In the previous section, we have covered the state of the art for identifying structurally relevant fragments in RNA structures, classifying them and predicting their presence in new sequences. In this section, we will discuss some direct and indirect applications of these methods, as well as an overview of other methods based on the general idea of leveraging specific structure fragments to infer functional information.

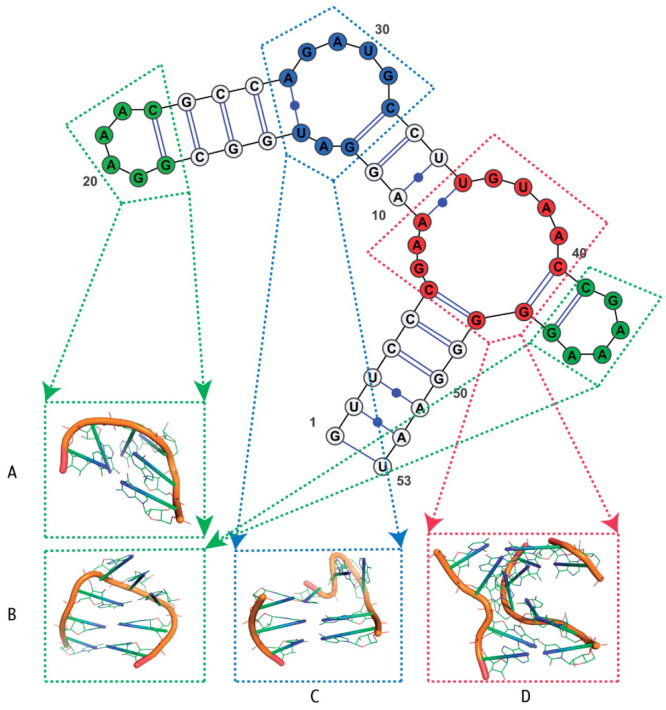
## 3.1 3D structure construction

The most direct application of RNA structural module prediction is for the construction of three-dimensional structures from sequence only. For this task, correctly predicting a secondary structure and an ensemble of structural modules is essential.

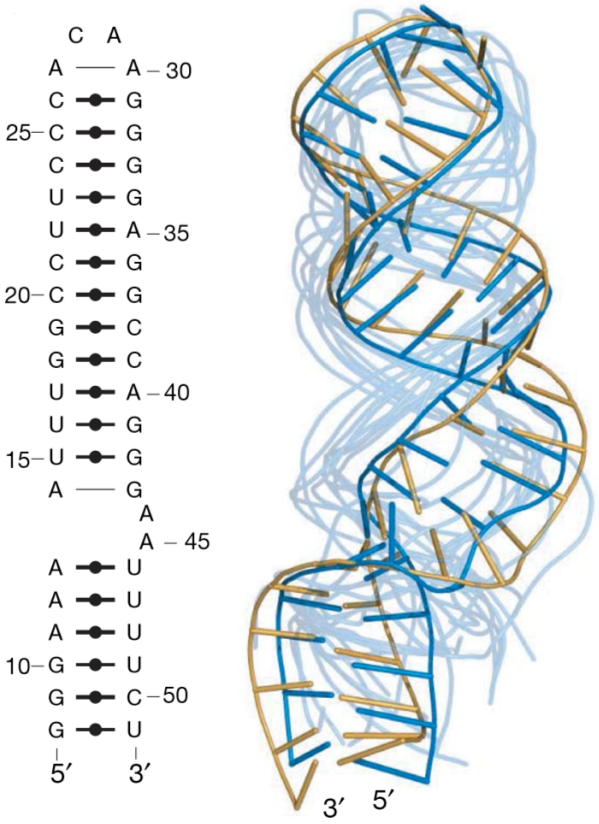
### 3.1.1 Fragment-based method for building full structures

The founding software for fragment-based 3D structure prediction is the MC-Fold-MC-Sym [13] pipeline, published in *Nature* in 2007 by Parisien and Major. This pipeline takes as input a single RNA sequence. MC-Fold outputs an augmented secondary structure, sent as input to MC-Sym which returns a 3D structure. The core novelty of this pipeline lies in the latter, the first software to take as input (mostly) 2D information and output an atomic structure.

The method converts input information about secondary structure, and the non-canonical base pair patterns inside loops. We can call such information an augmented secondary structure. The software will attempt to map each small component of this augmented secondary structure to a small 3D structure from a large database of fragments of full atomic structures. This mapping




(a) Matching augmented secondary structure components to 3D fragments, using `MC-Sym`[14].




(b) `MC-Sym` [9] builds many 3D structures and then selects the most energetically favorable one.

Figure 3: Overview of the `MC-Sym` workflow

is performed via a Las Vegas algorithm, a variant of Monte Carlo algorithms which stochastically searches a space but never returns an incorrect result. At each step, a match is attempted, and we assess whether it respects the initial constraints (Figure 3a). If it does not, another match is attempted. If not match succeeds, the software backtracks to the previous component. This is repeated until a full mapping is reached, and then the process is restarted to visit as many structures as possible within an allocated time, typically twelve hours. Each mapping contains enough information to achieve a structure. After undergoing one more step of energy minimization to optimize the structure within the new constraints provided by the mapping, the optimal mapping is selected (Figure 3b).

`MC-Sym` has limited computational efficiency as it struggles to model sequences longer than 75 nucleotides, but achieved results comparable to the state of art of other classes of approaches like comparative methods, which build a 3D structure by finding sequences similar to the input and copying their structure. The strongest selling point of the fragments methods is that it can be incrementally improved by learning more about the prediction of specific structural components from sequence, and improving the augmented secondary structure with the output of software such as `JAR3D` and `RMDetect`.

### 3.1.2 Annotation of 2D structure with 3D information with `forgi`

It follows that extracting tertiary structure knowledge from sequence and secondary structure is a relevant task to support 3D structure construction, but also for comparing sequences based on structural features. We have stated that secondary structure prediction has been successfully tackled with energy models (although there is still a lot of ongoing work on that problem), and presented methods for the prediction of interactions within loops. The intuition was that a secondary structure is composed of rigid, predictable stems regions, and unpaired loop regions for which we needed additional information from RNA modules. However, while we know enough about the position of the atoms within the stems to generate a reasonable 3D structure, we are still lacking information about the relative position of the stems, which is not fixed. Predicting angles between stems and potential stacking patterns remains a problem.

Thiel et al presented a solution to this problem in 2019 with `forgi 2.0` [15], a python package able to predict interactions between stem regions of a secondary structure.Stems, also known as helices, can be roughly modeled as vectors, and as such, their relative geometry can be modeled by five parameters: a separation vector, which represents a 3D distance between the two vectors, and the respective angle of each stem. `forgi` collects noisy 3D structure data, then calls a software to compute a secondary structure. Then, the start and end positions of each secondary structure component is stored, and vectors are fitted to stems. The package also offers many quality of life tools for structural data processing and overall represents a significant step towards fully automated 3D structure pipelines.

With the tools presented in this section and the previous, we have discussed methods to denoise structural data, predict a secondary structure, predict the interactions within loosely structured regions of the secondary structure, predict the interactions between tightly structured regions, and leverage all these constraints to generate atomic coordinates, covering a full pipeline for structure prediction from sequence.

## 3.2 Functional annotation of genes

3D structure prediction is not the only application of structured region analysis. Functionally significant structural networks can also be used to identify related sequences, discover structured RNA molecules in genomes, and predict interactions between RNA and other molecules. While approaches to solving these problems can stray away from the strict definition of RNA modules discussed previously, they are still focused on leveraging the structural information of specific, functional regions to advance our understanding of gene function.

### 3.2.1 Classifying RNA sequences into structural families

One of the major current problems with the number of available sequences, which has reached the order of a quadrillion nucleotides, is to group them by function. It is definitely not feasible to test all sequences in a laboratory to notice they appear to fold into the same structure. It not even feasible to rely on energy models to fold all known sequences and group the ones with similar secondary structure. More specifically, there is a need to group RNA molecules of the same structure and function into *families* of related RNAs. Eddy and Durbin [16] presented in 1994 the first major step towards solving this problem with a new model for RNA structure patterns that would allow fast pattern matching on new sequences, and later released `infeRNAl` [17], a software leveraging this approach to identify and align similar sequences. Their key contribution

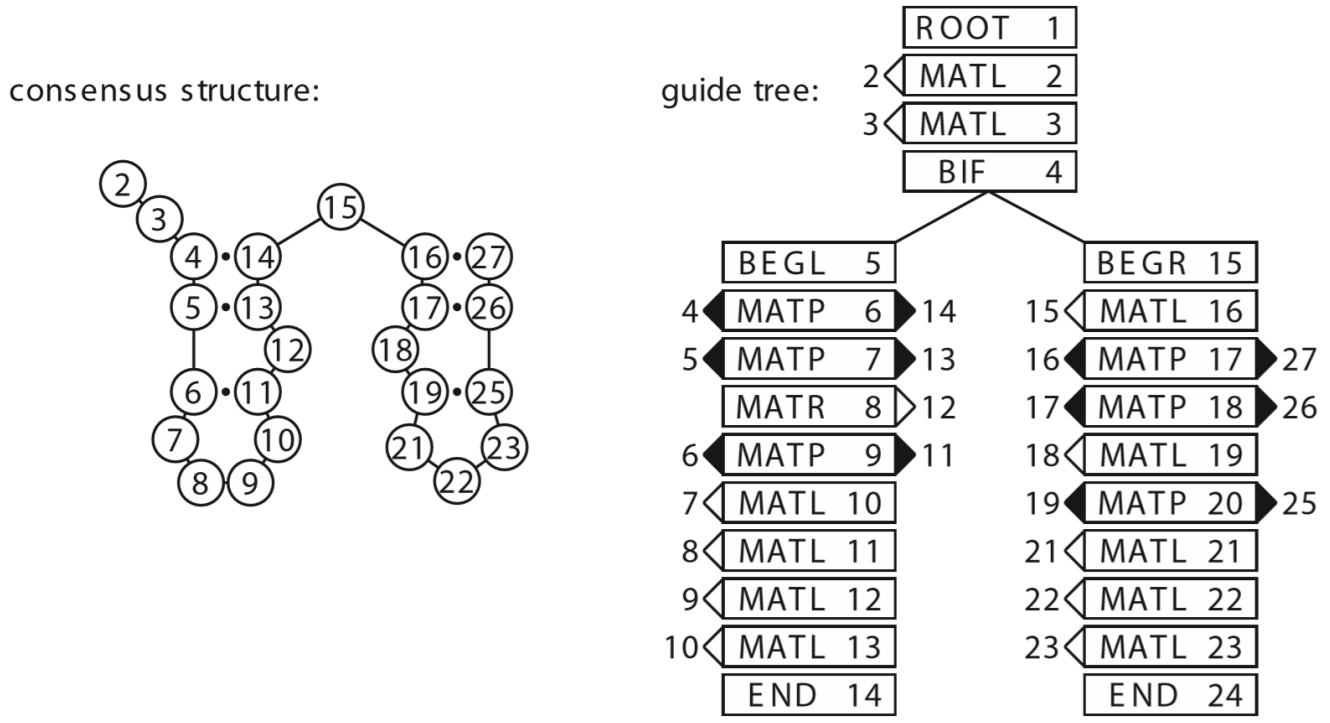



Figure 4: `infeRNAl` [17] consensus secondary structure and derived guide tree. Note the bifurcation at position 4. The other nodes contain either loop opening states, base pairs, or single nucleotides. The numbers on the right figure match the nucleotides on the left figure.

is a covariance model (CM), based on a profile stochastic context-free grammar (pSCFG). Given an alignment of RNA sequences with the same structure in which conserved bases are placed in the same column, the CM will learn generation rules able to generate sequences in the alignment, or similar to the alignment, and thus recognize new sequences belonging to the same family. The sequence and structure information are jointly encoded as an ordered tree. Trees are very commonly used to model RNA structure because they take advantage of nested base pairs; a base pair "includes" further base pairs, which allows starting from a root, defining children as every structural component enclosed in the first base pair, and assigning states to children nodes. Here, a state is defined as an element of the structure, which includes, namely, base pairs and single nucleotides, but also bifurcations (Figure 4). Building the tree will progressively lead to decomposing the initial sequence of nested base pairs and nucleotides into emissions of base pairs and single bases at the leaves. Additionally, a probability can be attributed to each state given the input alignment (i.e. the probability of emitting a specific nucleotide at a specific position), and to state transitions. The desired output is two-fold. We want the probability $\Pr(\omega \mid \theta)$ of a sequence $\omega$ being generated by a CM $\theta$, but also the optimal alignment of the sequence to the CM, in order to add it to the family if the match is conclusive. Both can be achieved simultaneously by computing the likelihood score on the growing sequence as the tree is being constructed and storing those scores in a table. A simple backtrack will then yield the optimal alignment.

While the final scoring is performed by the pSCFG, the software released by the authors, `InfeRNAl`, uses a Hidden Markov Model (HMM) derived from the pSCFG as a screening tool. The HMM can be thought of as a special case of the pSCFG where there would be no bifurcations (one base pair enclosing multiple base pairs) and no base pairs. Indeed, the HMM is not able to

test sequences for covariance between the partners of a base pair. However, it can quickly test for rough sequence signal, i.e. whether the searched sequence contains a subsequence of nucleotides which match the CM in terms of order and identity. This combination of SCFGs and HMMs implemented in `infeRNAl` is able to scan in polynomial time, which allowed the construction of hundreds of RNA structural families in the `Rfam` database. `InfeRNAl` is a very powerful software with many capabilities, most of which go beyond the scope of this review. Its main limitation is the hard requirement for nested base pairs, which means it cannot search for motifs including 3D structure information as non-canonical base pairs are rarely entirely nested.

### 3.2.2 Discovering new functional RNA sequences

In section 2.3.2, we have discussed the relevance of knowing *where to look* when mining genomes for structural information. Indeed, a vast majority of the RNA sequences in the genome will not lead to non-coding RNAs, the type of RNA in which the sequence primarily defines the structure, and the structure, the function. In many RNAs, the majority of the sequence is a copy of DNA information being transported, and as such, searching for structural modules in this type of sequence would not be relevant. This is the problem `RNAz` [18], published by Gruber et al in 2006, aims at solving. `RNAz` takes as input a multiple sequence alignment and outputs a prediction about whether the alignment represents a non-coding RNA, and, if relevant, a consensus secondary structure. This is achieved by two independent criteria: thermodynamic stability and structure conservation. Thermodynamic stability can be estimated by a z-score, a measure for the number of standard deviations by which the energy of the consensus structure of the alignment deviates from the average energy of a set of random sequences. In practice, this is very slow as it requires energy computations on many random sequences, and can be done faster and accurately by training a Support Vector Machine (SVM) on a large number of sequences with known stability, and using this SVM to estimate the mean and standard deviation from nucleotide composition of the input. Evolutionary conservation is evaluated by computing the structure conservation index (SCI). To compute this index, a consensus secondary structure is predicted for all sequences with the constraint that it must match the structure of all other sequences in the alignment, and is then compared to the individual, unconstrained secondary structure of each sequence. The SCI is defined as the ratio of the consensus folding energy to the average unconstrained energy of every sequence. If the alignment is perfect, the sequences should all individually fold into the same structure, and this ratio should equal to 1. The normalized Shannon entropy is also computed. This measure is defined as the sum of the Shannon entropies of each column $i$ of the alignment, divided by the number of columns $L$. We define $p_\alpha^i$ the observed frequency of character $\alpha$ in column $i$

$$H = -\frac{1}{L}\sum_{i}^{L}\sum_{\alpha\in \mathrm{A,C,G,U,\text{-}}} p_\alpha^i \log_2 p_\alpha^i$$

Finally, The estimated z-score of the sequences (from the first SVM step), the SCI and the Shannon entropy are passed to a classification SVM which returns an estimation of the probability Pr(structured RNA | sequence). `RNAz` was tested against shuffled sequences and outperformed the state of the art, but the reference false discovery rate for this task remains around 50%, leaving significant room for improvement.

### 3.2.3 Leveraging evolutionary information in RNA families

Multiple sequence alignments similar to those obtained from the families constructed by tools like `infeRNAl` contain evolutionary information which can be leveraged to better understand biological results. in 2016, Reinharz et al presented `aRNhAck`[19], an approach to combine biological results from the mutate-and-map protocol to multiple sequence alignments. The mutate-and-map (MaM) protocol consists of obtaining RNA-probing data for a molecule and many of its one-point mutants. Probing is an experimental method that consists of testing the reactivity of each nucleotide, which provides a rough estimate of which nucleotides are engaged in base pairs. A one-point mutant is a modified version of the initial molecule in which one nucleotide has been changed. The idea of the MaM protocol is thus to estimate the structural effects of mutations at different positions of the sequence, to identify the most important contributors to the structure. The structural disruption effect of different mutations was measured with the $l^2$ norm between the non-mutated probing profile $R$ and the mutated profile $R_i$ over a window of size $2\lambda + 1$ around the mutated position $i$:

$$\Delta(R, R_i) = \sqrt{\sum_{k=i-\lambda}^{i+\lambda} (R[k] - R_i[k])^2}$$

Here, profiles are lists of length equal to the length of the sequence, containing real values quantifying the reactivity at each position, so we are computing a distance between vectors. Multiple sequence alignments can also be parsed for information about which positions are most essential to the structure. In nature, mutations happen relatively frequently. A mutation at a position that is not essential will carry through time. However, a mutation at a position where a base pair is essential is disruptive, and will either lead to loss of function (and eventually lack of reproduction), or to the appearance of a compensatory mutation where, given a mutation to such nucleotide, the base pair partner would also mutate to rescue the base pair. This phenomenon can be observed as a high covariation between specific columns of an alignment. This covariation between two columns $x$ and $y$ can be evaluated with the normalized point-wise mutual information measure (NPMI), based on the probabilities Pr(.) estimated from the frequencies in those columns.

$$\mathrm{NPMI}(x, y) = \frac{\log \frac{\Pr(x,y)}{\Pr(x)\Pr(y)}}{-\log \Pr(x, y))} \in [-1, 1]$$

Because secondary structure covariation had already been extensively studied and does not require support from the MaM workflow,a proximity filtering was added to highlight long-range interactions. Thus, only pairs of positions which were significantly disruptive according to MaM and showed significant covariance according to NPMI, but were also distant enough in secondary structure were returned. `aRNhack` showed significant enrichment of RNA positions interacting with other molecules among the pairs of co-variating positions returned by the pipeline, a promising result for further work on combining biological assays and multiple sequence alignments to infer the functional role of specific positions in RNA molecules.

### 3.2.4 Predicting biomolecular partners of an RNA from sequence

All algorithms presented so far rely on an explicit secondary structure to make predictions. However, this is not a necessity. The sequence is already highly predictive of the secondary structure, so in theory, sequence-only methods could work for some tasks. This is especially true for the identification of protein-binding sites in RNA. Many proteins interact with specific sequence motifs

(a short sequence of nucleotides), and the nucleotides at these positions are often very conserved. Nevertheless, the secondary structure around those positions still requires a specific conformation to allow the protein to interact, so a high level knowledge of sequence interaction is helpful. Kazan et al presented `RNAcontext` [20] in 2010, a simple but successful mathematical model for the prediction of protein interactions in RNA sequences. Their model takes as input a sequence and a secondary structure label for the type of secondary structure context, and returns binding probability with some specific protein. The authors use four different contexts: paired (P), hairpin loop (L), unstructured region (U), and miscellaneous (M).

Let $\Theta = \{\Phi, \Gamma, \beta_s, \beta_p, K\}$ define the model parameters. Here, $K$ represents the width of the binding site. $\Phi$ a position weight matrix of size 4 x $K$, with the expected frequency of each nucleotide for each position of the binding site. $\Gamma$ is a vector of structure annotations, with a probability value for each type of context (P, L, U or M), for each position. The two $\beta$ terms are bias terms, respectively for sequence affinity and structural context. Given an input sequence $s$, we can compute the probability of seeing at least one subsequence of $s$ be bound by the protein. This will involve scoring all subsequences (or k-mers) of size $K$. We can define this probability as 1 minus the probability of any k-mer being matched. This function takes as input a sequence $s$, a profile matrix $p$ in which each column contains a discrete probability distribution over P,L,M,U, representing the structural context of each nucleotide of the sequence, and finally a set of model parameters $\Theta$.

$$f(s, p, \Theta) = 1 - \prod_{t=0}^{|s|-K} 1 - N(s_{i+1:i+K}, p_{i+1:i+k}, \Theta)$$

Where $N$ is an estimation of the probability of the k-mer from positions $i+1$ to $i+K$ to be bound. This probability will be estimated by scoring its sequence and structural context as follows:

$$N(s_{i+1:i+K}, p_{i+1:i+k}, \Theta) = \sigma(\beta_s + \sum_{j=1}^{K} \Phi_{s_j,j}) + \sigma(\beta_p + \sum_{a \in \{\text{P,L,M,U}\}} \Gamma_a \sum_{j=1}^{K} p_{a,j})$$

where $\sigma$ is the logistic function. The first term is an evaluation of the sequence assuming perfect structural context. It is simply the sum of the trained sequence bias term and the sum of frequencies for each nucleotide in the k-mer $s$. The second term models the structural context. Here, the structural context at each position is evaluated against the model parameter $\Gamma$.

The model is trained on sequence-structure pairs that are known to bind proteins, using affinity data ranging from -1 (never binding) to 1 (always binding), by modeling this affinity as a linear function of the sequence $s_i$'s score output by the model: $\hat{r}^i = \alpha f(s^i, \Theta) + b$. We then attempt to minimize a least squares cost function over N sequences $s_{0:N}$:

$$E(\Theta, \alpha, b) = \sum_{i}^{N} (r^i - \hat{r}^i)^2 + \delta(\sum_{\Phi_k \in \Phi} \Phi_k^2 + \sum_{\Gamma_k \in \Gamma} \Gamma_k^2)$$

Where the regularization term is scaled by a small constant $\delta$ to ensure a unique global minimum. `RNAcontext` was shown to constitute an improvement over the state of the art in protein binding prediction from sequence, all of which relied on explicit secondary structure modeling, successfully demonstrating the merits of denoising secondary structure signal by only including broad context.

# 4 Conclusion

Predicting 3D structure and function from sequence in RNA is still a budding field, and a wide variety of approaches have shown success on specific subproblems. We have reviewed some of those approaches and can observe that despite significant progress, a core issue remains: there are vast amounts of data that need exploring, but RNA folding is a difficult task that requires a lot of flexibility to achieve high accuracy. So far, high-throughput methods tend to be at best acceptable filtering tools, and high accuracy methods tend to be low-throughput. Thus, there is still significant work to be done to either improve or combine those two classes of approaches.

# References


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

[2] 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.

[3] 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.

[4] 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.

[5] Gabriel Valiente. Tree isomorphism. In *Algorithms on Trees and Graphs*, pages 151–251. Springer, 2002.

[6] Ryan R Rahrig, Neocles B Leontis, and Craig L Zirbel. R3d align: global pairwise alignment of rna 3d structures using local superpositions. *Bioinformatics*, 26(21):2689–2697, 2010.

[7] Steffen Heyne, Fabrizio Costa, Dominic Rose, and Rolf Backofen. Graphclust: alignment-free structural clustering of local rna secondary structures. *Bioinformatics*, 28(12):i224–i232, 2012.

[8] Kurt De Grave and Fabrizio Costa. Molecular graph augmentation with rings and functional groups. *Journal of chemical information and modeling*, 50(9):1660–1668, 2010.

[9] José Almeida Cruz and Eric Westhof. Sequence-based identification of 3d structural modules in rna with rmdetect. *Nature methods*, 8(6):513, 2011.

[10] 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.

[11] Corinna Theis, Christian Höner zu Siederdissen, Ivo L Hofacker, and Jan Gorodkin. Automated identification of rna 3d modules with discriminative power in rna structural alignments. *Nucleic acids research*, 41(22):9999–10009, 2013.

[12] Corinna Theis, Craig L Zirbel, Christian Höner Zu Siederdissen, Christian Anthon, Ivo L Hofacker, Henrik Nielsen, and Jan Gorodkin. Rna 3d modules in genome-wide predictions of rna 2d structure. *PloS one*, 10(10):e0139900, 2015.

[13] Marc Parisien and Francois Major. The mc-fold and mc-sym pipeline infers rna structure from sequence data. *Nature*, 452(7183):51, 2008.

[14] 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.

[15] 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.

[16] Sean R Eddy and Richard Durbin. Rna sequence analysis using covariance models. *Nucleic acids research*, 22(11):2079–2088, 1994.

[17] Eric P Nawrocki and Sean R Eddy. Infernal 1.1: 100-fold faster rna homology searches. *Bioinformatics*, 29(22):2933–2935, 2013.

[18] Andreas R Gruber, Sven Findeiß, Stefan Washietl, Ivo L Hofacker, and Peter F Stadler. Rnaz 2.0: improved noncoding rna detection. In *Biocomputing 2010*, pages 69–79. World Scientific, 2010.

[19] Vladimir Reinharz, Yann Ponty, and Jérôme Waldispühl. Combining structure probing data on rna mutants with evolutionary information reveals rna-binding interfaces. *Nucleic acids research*, 44(11):e104–e104, 2016.

[20] Hilal Kazan, Debashish Ray, Esther T Chan, Timothy R Hughes, and Quaid Morris. Rnacontext: a new method for learning the sequence and structure binding preferences of rna-binding proteins. *PLoS computational biology*, 6(7):e1000832, 2010.