sincFold: end-to-end learning of short- and long-range interactions in RNA secondary structure

Leandro A. Bugnon[1]*, Leandro Di Persia[1], Matias Gerard[1], Jonathan Raad[1], Santiago Prochetto[1,2], Emilio Fenoy[1], Uciel Chorostecki[3], Federico Ariel[2], Georgina Stegmayer[1], Diego H. Milone[1]

[1]Research Institute for Signals, Systems and Computational Intelligence, sinc(i), FICH-UNL, CONICET, Ciudad Universitaria UNL, 3000, Santa Fe, Argentina. [2] Instituto de Agrobiotecnología del Litoral, CONICET-UNL, CCT-Santa Fe, Ruta Nacional N° 168 Km 0, s/n, Paraje el Pozo, 3000, Santa Fe, Argentina. [3] Faculty of Medicine and Health Sciences, Sant Cugat del Vallès, 08195, Barcelona, Spain

Abstract

Motivation: Coding and non-coding RNA molecules participate in many important biological processes. Non-coding RNAs fold into well-defined secondary structures to exert their functions. However, the computational prediction of the secondary structure from a raw RNA sequence is a long-standing unsolved problem, which after decades of almost unchanged performance has now re-emerged thanks to deep learning. Traditional RNA secondary structure prediction algorithms have been mostly based on thermodynamic models and dynamic programming for free energy minimization. More recently deep learning methods have shown competitive performance compared with the classical ones, but still leaving a wide margin for improvement.
Results: In this work we present sincFold an end-to-end deep learning approach that predicts the nucleotides contact matrix using only the RNA sequence as input. The model is based on 1D and 2D residual neural networks that can learn short- and long-range interaction patterns. We show that structures can be accurately predicted with minimal physical assumptions. Extensive experiments were conducted on several benchmark datasets, considering sequence homology and cross-family validation. sincFold was compared against classical methods and recent deep learning models, showing that it can outperform state-of-the-art methods.
Availability: The source code is available at https://github.com/sinc-lab/sincFold (v0.16) and the web access is provided at https://sinc.unl.edu.ar/web-demo/sincFold
Contact: lbugnon@sinc.unl.edu.ar

1 Introduction

Non-coding ribonucleic acid (ncRNA) molecules have emerged as crucial players in cellular processes, encompassing epigenetics, transcriptional and post-transcriptional regulation, chromosome replication, translation and protein activity and stability [1, 2]. Recent efforts have even explored the clinical potential of ncRNA in diagnostics, vaccines, and therapies [3]. This paradigm shift in our understanding of ncRNA, from being dismissed as ”transcriptional noise” prior to the 1980s to being recognized as regulators of gene expression at multiple levels, has generated an explosion of research in this field over the past few decades [4].

RNA itself consists of an ordered sequence of four basic nucleotides: adenine (A), cytosine (C), guanine (G), and uracil (U). Pairing these bases within an RNA molecule gives rise to its secondary structure, a crucial determinant of its functions and stability [5]. The secondary structure is characterized by hydrogen bonding interactions between complementary base pairs, which typically include the canonical Watson-Crick-Franklin pairs A-U and C-G [6], along with the wobble pair G-U [7]. Basic stem-loop structures, formed by nested base pairs, are commonly observed. However, the secondary structure can also exhibit complex motifs arising from local bonding and long-range sequence interactions.

Despite the growing number of publicly accessible ncRNA sequences, a significant proportion of their true structures remains unknown [8]. Secondary structures can be obtained with sophisticated experimental techniques such as X-ray crystallography, nuclear magnetic resonance [9, 10, 11], enzymatic probing methods such as nextPARS [12], or chemical probing such as DMS-seq [13] and SHAPE-seq [14]. However, all these methods suffer from low resolution and high costs [15]. Consequently, due to its cost-effectiveness the computational prediction has gained substantial relevance in biological research and biotechnological applications.

Traditional computational methods for RNA secondary structure prediction employ a thermodynamic model of base-pair interactions optimized through dynamic programming to identify structures with minimal free energy [16, 17, 18]. Despite being proposed 20 years ago [19], these methods, such as RNAstructure [20], ProbKnot [21], RNAfold [22], LinearFold [23], LinearPartition [24], and Ipknot [25], continue to dominate the field. However, their average base-pair prediction performance remains at around $F_1 = 70\%$ [26].

To surpass this performance ceiling, machine learning (ML) techniques, and particularly deep learning (DL), have emerged as promising alternatives [27]. DL techniques have been widely noticed for structure prediction in proteins with AlphaFold [28] and more recently several methods were presented for RNA secondary structure prediction [25, 29, 30, 31, 32]. However the available RNA datasets are very small compared to proteins, they are highly biased in several ways and pseudoknots are not consistently annotated, being a key factor in RNA structures. In [33] authors state that there are several possible ways to enable the accurate prediction of RNA structures in the near future, such as improving knowledge through more data, diversifying the data used in prediction, and improving the machine learning methods used. In particular DL methods rely less on assumptions about the thermodynamic mechanics of folding, instead adopting a data-driven approach. Consequently, they could be better suited to identify complex structures that defy modeling using traditional techniques. However, recent systematic evaluations of techniques for comparatively assessing their performance on ncRNAs showed that DL has not yet clearly overperformed classical methods [26, 34, 35].

Currently, several DL approaches are available with different architectural designs, input representations, training data and optimization algorithms for parameter adjustments [36]. Among these proposals, SPOT-RNA [30] was the pioneering DL method based on ensembles of convolutional networks (CNNs) and bidirectional Long-short term memory neural networks (LSTM). SPOT-RNA2 [37] improved its predecessor by using predictions from thermodynamic models, evolution-derived sequence profiles and mutational coupling, however requiring multiple sequence alignments. Another hybrid approach was MXfold [38], combining support vector machines and thermodynamic models. Similarly, DMFold [39] and MXFold2 [25] integrated DL techniques with energy based methods. Another method based on both DL and dynamic programming was CDPfold [29], which iteratively computes a matrix representation of possible matchings between bases according to a physical model of base interactions, and then trains a convolutional network over this matrix to predict base pairing probabilities. Upon this, dynamic programming is applied to obtain the final RNA secondary structure.

In more recent years, UFold [31] approached the secondary structure prediction problem using a well-known architecture from image segmentation, the U-Net encoder-decoder [40]. It uses a 2D feature map to encode the occurrences of one of the sixteen possible base pairs between nucleotides for each position in the map, including an additional channel with the matrix representation of possible matchings iteratively computed with the algorithm proposed in CDPfold. The predicted output is the contact score map between the bases of the input sequence, which goes through a post-processing step that involves solving a linear programming problem to obtain the optimum contact map. Interestingly, a very recent method, REDfold [32], reported to outperform UFold. This DL method also utilizes a U-Net encoder-decoder network to learn dependencies among the RNA sequence, together with symmetric skip connections to propagate activation information across layers and output post-processing with constrained optimization.

In this work we present sincFold, a novel end-to-end deep learning method for RNA secondary structure prediction. Our approach is based on ResNet bottlenecks to capture both short- and long-range dependencies in the RNA sequence. Unlike other DL models, we adopt a two-stage encoding process: initially, we model sequence encoding in 1D, enabling the learning of small context features and reducing computational costs; then a pairwise encoding in 2D is incorporated to capture distant relationships. Extensive experimental evaluations on two widely used ncRNA databases demonstrate that sincFold outperforms classical methods and DL state-of-the-art techniques in terms of F_1 performance. We have made the source code for sincFold freely accessible, facilitating its adoption and further development in the research community[1]. Moreover, a web service to test the trained model is provided[2].

[1] Source code available at https://github.com/sinc-lab/sincFold
[2] Web-demo available at https://sinc.unl.edu.ar/web-demo/sincFold

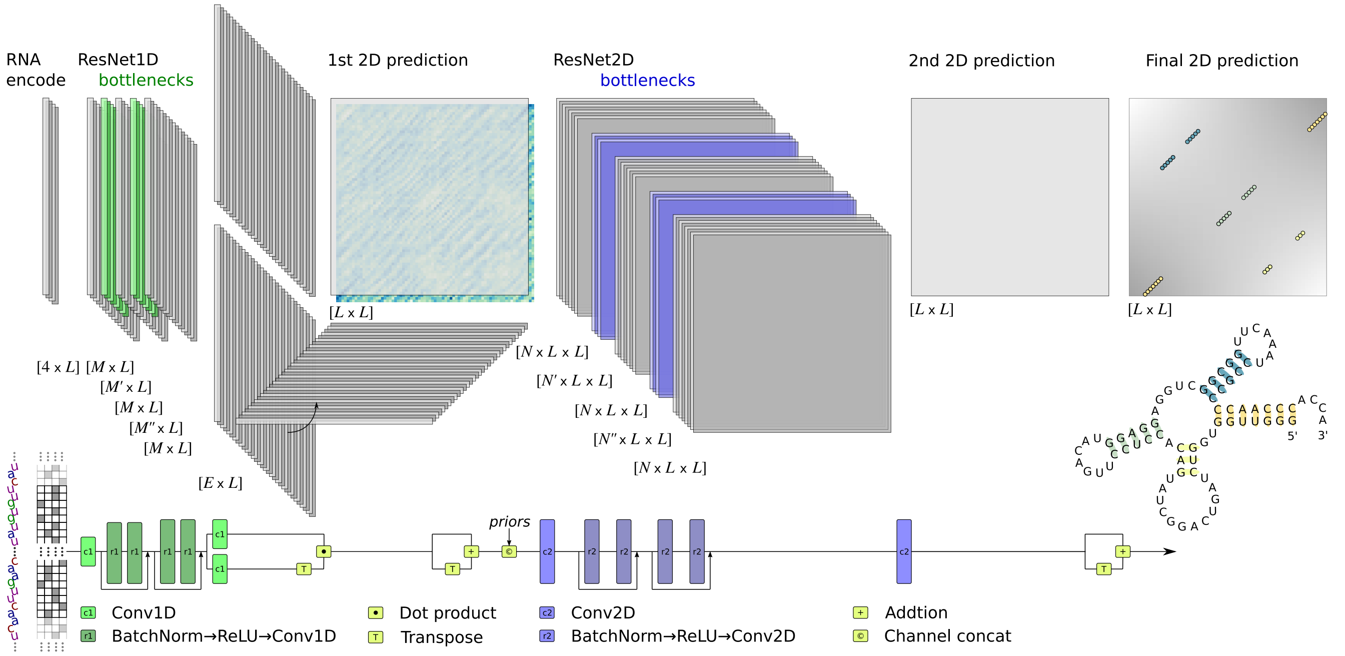

Figure 1: The end-to-end architecture of sincFold. Top: data flow with its shapes and dimensionality in each point of the architecture, from the $[4 \times L]$ one-hot encoded RNA sequence at the input to the $[L \times L]$ connection matrix at the output. Bottom: neural processing blocks depicted as differentiable layers.

2 The sincFold model

In order to obtain a secondary structure prediction from a stand-alone RNA sequence, we propose sincFold. As shown in Figure 1, this novel deep learning architecture is composed of two stages: the first one learns local patterns in 1D encodings, while the second stage can learn more distant interactions in 2D. The figure represents in detail the shapes and dimensions of the data along the pipeline (top) and the neural processing blocks (bottom).

The model takes as input a RNA sequence of length L encoded in one-hot (bottom-left) so that each nucleotide type of the sequence is represented with a vector of size 4 (i.e. a one-hot codification of the 4 canonical nucleotides). The encoded sequence goes through a one-dimensional (1D) convolutional layer that performs a first automatic extraction of low-level features for each nucleotide. Then, identity blocks [41] are stacked in a 1D-ResNet. These blocks allow the model to propagate the signal and reduce vanishing gradient issues, while maintaining the same sequence length. Moreover, the identity blocks make the model capable of auto-defining the number of convolutional layers needed during training. Each block is composed of two batch normalization layers, ReLU activations and convolutional layers in 1D, with bottlenecks in the features (depicted with light green in the figure). Bottlenecks reduce the learnable parameters while helping to learn more relevant features.

After the 1D bottlenecks, a $M \times L$ encoding is obtained, where M is the dimension of the feature vector of each nucleotide. Then two convolutions in 1D produce two compressed encodings of size $E \times L$. A matrix product between one $E \times L$ matrix and the other $E \times L$ matrix transpose is made, obtaining a first "draft" of the contact matrix in 2D ($L \times L$). After that, the matrix is forced to be symmetric by adding its transpose. An additional channel of interaction priors is added at this point, coding different bonding strengths for C-G, U-A and G-U.

Once the information is represented in 2D, the new tensor $L \times L$ will go through a 2D-ResNet stage. Similarly to the 1D-ResNet stage, a 2D-convolutional layer is followed by 2D-ResNet blocks composed of batch normalization layers, ReLU activations and 2D convolutions. After several 2D-ResNet layers with bottlenecks the 2D pairwise encodings are flattened to a $L \times L$ output, and its transpose is added to force symmetry. This output matrix is the final 2D prediction of the secondary structure for the RNA sequence, and the entire model can be trained with a unified cost function. A simple post-processing is applied to find the maximum activation on each row and column, thus retaining only one interaction per nucleotide.

To guide training, we propose a composed loss function

$$\mathcal{L} = \mathcal{L}_\alpha + \lambda_\beta \mathcal{L}_\beta + \lambda_1 \mathcal{L}_1 \tag{1}$$

where $\mathcal{L}_\alpha$ is the cross-entropy loss of the final prediction, $\mathcal{L}_\beta$ is the cross-entropy loss of the model prediction prior to the 2D-ResNet block, and $\mathcal{L}_1$ is a L_1 loss of the predictions used to enforce the contact matrices sparseness. Cross-entropy is computed element by element in the matrix. The weights λ_β and λ_1 of each of these terms are hyperparameters to be adjusted experimentally.

Our proposed architecture is different to existing DL models in several ways. SPOT-RNA converts to a 2D representation but only as a pre-processing stage, by outer concatenation of the one-hot codification of the sequence. In this model, 1D patterns are not learned throughout the sequence. Then, the prediction of structure is obtained with an ensemble of ResNet blocks with dilated convolutions, a 2D-BLSTM (bidirectional long short-term memory) layer and a fully connected block. Furthermore, the SPOT-RNA source code for training is not available and thus it cannot be compared to other methods under the same conditions. MXFold2 has an architecture that models 1D and 2D representations though BLSTM and 2D convolution blocks, respectively. The conversion from 1D to 2D is based on a concatenation of halves of the 1D embeddings, so that different halves appear together in the corresponding coordinates of the $L \times L$ output. The choice of halves as a concatenation block only responds to the need to form a 2D representation, but has no basis in the modeling of structural connections to be predicted. Moreover, as in SPOT-RNA pre-processing, these concatenations do not include any inner products that measure similarity between 1D representations. Finally, it is important to note that MXFold2 is actually a hybrid method, which does not predict a contact matrix but four types of folding scores for each pair of nucleotides. The folding scores are integrated with the free energy parameters of Turner nearest-neighbor model. Then an optimal secondary structure is calculated using classical dynamic programming. Differently from UFold and REDfold, which use the standard U-Net originally proposed in computer vision for image segmentation, and a post-processing step with linear programming, with sincFold we propose a novel full end-to-end architecture that models separately the 1D (short range) and 2D (long range) interaction. It is important to note that in both UFold and REDfold the conversion from 1D to 2D is, as in other models, a pre-processing stage (i.e. it is not part of the DL model). For example, in UFold pre-processing the one-hot codification of the RNA sequence is converted into a 16 channels 'image' via a Kronecker product. In contrast, sincFold learns representations from a 1D sequence, converts them to a 2D representation with a tensorial product and then learns long range interactions through training.

3 Data and performance measures

3.1 Data

In order to evaluate the performance of sincFold, we have run benchmarks with datasets widely used by the community.

RNAstralign dataset [42]: contains 37,149 sequences from 8 large RNA families: 5S rRNAs, Group I Intron, tmRNA, tRNA, 16S rRNA, Signal Recognition Particle (SRP) RNA, RNase P RNA and Telomerase RNA. It is one of the most comprehensive RNA structure datasets available.

ArchiveII dataset [5]: the most widely used benchmark dataset for RNA folding methods, containing RNA structures from 9 RNA families: 5S rRNAs, SRP RNA, tRNA, tmRNA, RNase P RNA, Group I Intron, 16S rRNA, Telomerase RNA and 23S rRNA. The total number of sequences is 3,975.

TR0-TS0 dataset [30]: the same train and test sets as in SPOT-RNA. The learning data are from bpRNA 1.0 (Danaee et al., 2018). It consists in a nonredundant set of RNA sequences with annotated secondary structure from bpRNA34 at 80% sequence-identity cutoff with CD-HIT-EST [43]. This filtered dataset of 13,419 RNAs is randomly divided into 10,814 RNAs for training (TR0), 1,300 for validation (VL0), and 1,305 for an independent test (TS0).

Ablation dataset: in addition, we compiled a dataset of sequences derived from the URS server [44] to be used as a small independent dataset for model optimization. These sequences and secondary structures were extracted from the Protein Data Bank (PDB), consisting of 753 sequences ranging from 8 to 456 nucleotides.

As suggested in [45], sequences longer than 512 nucleotides were filtered to limit the runtime of experiments, leaving 22,611 sequences in the RNAstralign dataset and 3,864 sequences in the ArchiveII dataset. Group I intron RNAs were excluded from the RNAstralign dataset because it included sequences without a unique structure. Thus, in this manuscript we will show results only for sequences with less than 512 nt.

To assess the performance, all DL methods used in this study were re-trained from scratch with the exact same partitions for training and testing. First we perform a k-fold cross-validation with $k = 5$ on the Ablation, ArchiveII and RNAstralign datasets. For the ArchiveII dataset, the original k-fold split provided by the authors was used [5]. Sequences were randomly divided into 5 independent folds of approximately the same size, and each fold was in turn taken as the test data while the remaining folds were taken as the training data. Then, we considered the structural differences between sequences used in training and testing, in order to analyze the impact of homology on performance. In the TR0-TS0 dataset, we use the provided homology-aware partitions with 80% sequence similarity cut-off. Finally, we perform a cross-family analysis (testing on unseen RNA families) using the ArchiveII dataset.

3.2 Performance measures

The focus of performance measures is on the predicted base pairs in comparison to a reference structure [46]. Pairs that are both in the prediction and in the reference structure are true positives (TP), while pairs predicted but not in the true structure are false positives (FP). Similarly, a pair in the reference structure that is not predicted is a false negative (FN), and a pair that is neither predicted nor in the true structure is a true negative (TN). To fully characterize the successes and failures of structure prediction, we use the F_1 score, defined as

$$F_1 = \frac{2TP}{2TP + FP + FN}. \tag{2}$$

The whole RNA structure can be considered as a large interaction network composed of interactions and base stackings [47]. The interaction network fidelity (INF) similarity measure [48] was designed to score the similarity between the interactions of a reference RNA structure and the interactions of a predicted RNA structure. INF is defined as

$$INF = \frac{TN.TP - FN.FP}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \tag{3}$$

resembling the well-known Matthews correlation coefficient. When the prediction reproduces exactly the base interactions of the reference structure, then $|FP| = |FN| = 0$, $|TP| > 0$, and thus $INF = 1$. When the prediction does not reproduce any of the interactions of the reference structure, then $INF = 0$, since $|TP| = 0$.

In [49] it was demonstrated how two structures that share a common feature (for example, a hairpin) with the exact same base pair patterns can achieve $F_1 = 0$. This is because similar base-pair patterns between the two secondary structures can only be shifted, but this will not be reflected by the F_1 score. Thus, the Weisfeiler-Lehman graph kernel (WL) metric was proposed in order to capture graphs structural information by iteratively refining node labels based on their local neighborhoods. The WL metric first assigns to each node (nucleotide) in the graph (secondary structure) a label representing its local structural information. Then, a label propagation step iterates over the nodes and updates their labels based on the labels of their neighboring nodes. Finally, a hash function is computed that aggregates these labels to generate a feature vector. The WL is defined as

$$WL(G_1, G_2) = \Phi(G_1).\Phi(G_2), \tag{4}$$

where $\Phi(G_i)$ represents the feature vector of graph G_i obtained by aggregating the labels through the hash functions. The WL-similarity score is sensitive to both structural and sequence-level alterations.

3.3 Distance measure for secondary structures

It is known that minor changes in RNA sequences can represent significant changes in secondary structure and, conversely, very similar structures can be obtained from quite different sequences. Thus, for analyzing results, the structural distance between data samples is more representative of the prediction challenge than a simple sequence-level distance. For this reason, the structural distance was computed using RNAdistance from the ViennaRNA package [22, 50]. This distance is based on the edit distance of a tree representation, in which the secondary structure is converted into a tree by assigning an internal node to each base pair and a leaf node to each unpaired digit [51]. Then, a tree is transformed into another tree by a series of editing operations with predefined costs. The distance between the two trees is the smallest sum of the costs along an editing path, which is divided by the length of the longest sequence in order to obtain a normalized distance.

4 Results

4.1 Ablation study and hyperparameters exploration

We conducted an ablation study to gain a deeper understanding on the contribution of each of the components of the sincFold architecture. We run several versions of the sincFold model: C1D) a baseline model with only 1D-convolutional networks; R1D) the same model replacing convolutions with 1D-residual blocks and bottlenecks; C1D+C2D) the model with the 2D-stage using only convolutional neural networks; C1D+R2D) replacing convolutions with the 2D-residual blocks in the 2D stage; and R1D+R2D) with residual blocks and bottlenecks in both

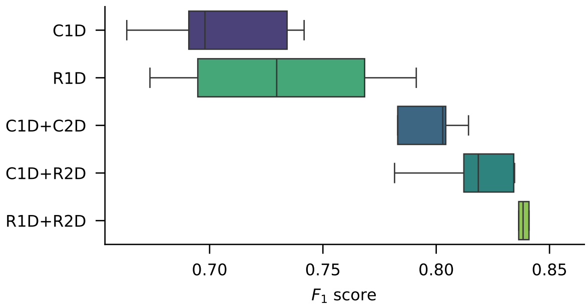

Figure 2: Ablation study on each of the components of the sincFold architecture. Each box has the F_1 scores from a 5-fold cross-validation on the Ablation dataset.

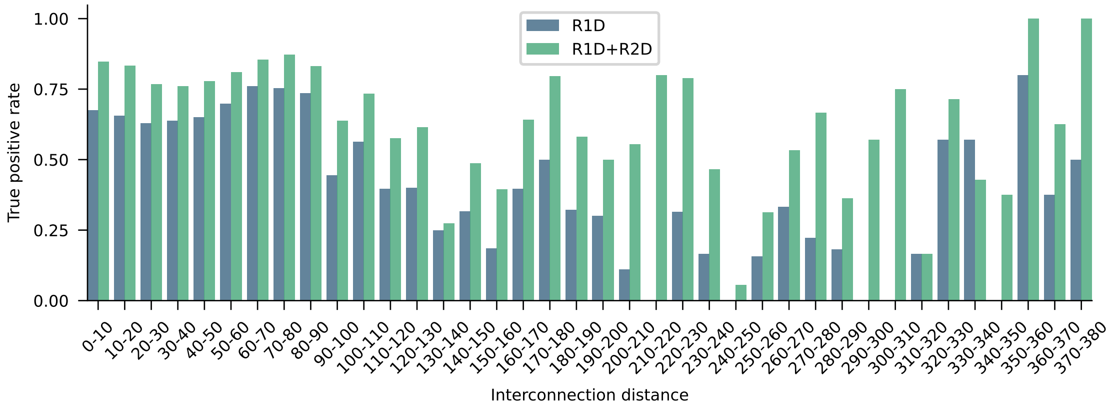

Figure 3: True positive rate for each interaction distance, comparing the model with only the first stage and both stages. Resnet based model with only the first stage (R1D) and both stages (R1D+R2D).

stages. The F_1 scores for each ablated sincFold version, from a 5-fold cross-validation on the Ablation dataset, are shown in the boxplots of Figure 2.

It can be seen that changing the C1D to a R1D block slightly improves the median results, from a median $F_1 = 0.697$ to $F_1 = 0.729$. It can be observed that adding the 2D stage (C2D) to the output of the previous models increased their performance significantly, by 10% for each model. The F_1 raises up to 0.802 with C1D+C2D, and using a ResNet instead of a CNN in the 2D stage (C1D+R2D), performance further improves up to $F_1 = 0.818$. Finally, results are even further improved in the model with ResNet blocks in both stages (R1D+R2D), reaching $F_1 = 0.838$.

After the ablation study we conclude that ResNet blocks effectively improve the generalization capability, in comparison to simple convolutional layers. Figure 3 presents a detailed analysis of the true positive rate of predictions in this dataset along the interconnection distance. Interestingly, when the 2D stage (green) is added to the 1D stage (blue) the model performance improves for all connections, and especially for distances longer than 200 nt.

This shows that in fact the sincFold 2D stage improves the learning of long range dependencies. Moreover, a very interesting property of ResNet blocks is that when there are many blocks available, the model is capable of automatically selecting how many of them are really necessary, skipping the non-necessary blocks during training. This reduces the learnable parameters while helping to learn more relevant features.

Using the best performing sincFold architecture (R1D+R2D), we performed a hyperparameter space search in the Ablation dataset, exploring: batch size, learning rate, the use of learning rate schedule, weights for the loss components λ_β, λ_1, architecture of the 1D-ResNet stage (kernel size and dilation, number of filters and number of layers) and the 2D-ResNet stage (kernel size, number of filters, bottleneck size and number of layers). Parameters were explored randomly [52], and the best configuration was selected for the next experiments (Supplementary Material Figure S1).

4.2 Performance according to test-train structural distance on random partitions

Figure 4 shows the comparative results among classical folding methods (RNAfold, RNAstructure, ProbKnot, IPKnot, LinearPartition-V, LinearFold-V, LinearPartition-C and LinearFold-C), the hybrid method MXfold2, DL based methods (UFold and REDfold) and the proposed sincFold in terms of F_1 for 5-fold cross-validation on the RNAstralign dataset. All DL methods were trained and evaluated from scratch with the same dataset partitions on cross-validation. It can be seen that all classical methods have a performance between 0.633 and 0.712 of F_1. MXFold2 combines DL and thermodynamic models and achieves better performance (median $F_1 = 0.907$). DL methods show even better scores, UFold reaches a median $F_1 = 0.966$ and REDfold arrives at median $F_1 = 0.976$. The proposed method, sincFold, achieves $F_1 = 0.986$. The variance of our method is very small, and the box is not overlapped with the performance of the other DL methods.

Figure 5 shows the comparative results among classical folding methods, hybrid method, DL methods and the proposed sincFold, in terms of F_1 for the ArchiveII dataset. As in the previous result, classical methods have a median performance below $F_1 = 0.620$. In this case, MxFold2 achieves $F_1 = 0.738$, UFold has a median $F_1 = 0.855$ and REDfold arrives at $F_1 = 0.831$. The proposed method sincFold achieves the highest median $F_1 = 0.913$. It can be seen that in both datasets, our proposal achieves a significantly better performance than classical methods and state-of-the-art DL methods.

The detailed performance for each method according to the lengths of the test sequences, from shorter (left) to longer sequences (right) are analyzed in Figure 6. Light blue bars indicate the proportion of each bin of lengths in the dataset. Here it can be seen that for shorter sequences, all methods have average performance above $F_1 = 0.60$, being particularly good at this task all DL methods, with $F_1 > 0.90$. As sequence length is increased, classical methods lower performance while DL methods are less affected, maintaining $F_1 > 0.75$ in most cases and being sincFold always the best for long sequences, achieving $F_1 = 0.85$ for sequences between 300 and 400 nucleotides. At the extreme, for sequences longer than 400 nucleotides, sincFold is still better than other methods despite the few examples available to learn from, achieving a median $F_1 = 0.74$, which is superior to the average performance of classical methods in the shortest sequences.

It is well-known that at random partitions there can be sequences with high similarity between training and testing partitions, thus methods can show overly optimistic results. To have more insights on the sincFold performance in this regard, we report comparative results by analyzing the sequences following the secondary structure distance between test and train partitions, as shown in Figure 7. Instead of just making one single partition with a certain sequence identity level, we have analyzed a full range of structural similarities. The distance between two structures was computed using RNAdistance from the ViennaRNA package [22] as explained in Section 3.3. For each test sequence, the test-train distance was defined as the minimum structural distance between this test sequence and all the sequences in its corresponding training fold. Then, test sequences with similar structural distance were grouped into bins to obtain the x-axis in the figure. Ranges of structural distances are presented from large (very-hard) test-train distances (left) down to low (very-easy) test-train distances (right). Light blue bars indicate the proportion of test sequences in each bin of structural distances.

It can be clearly seen that as the test-train distance diminishes, all methods (including the classical, non-

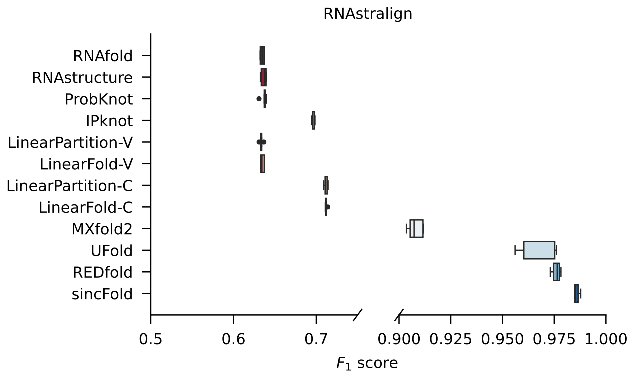

Figure 4: Comparative results among classical folding methods, DL-based folding methods and sincFold, for a 5-fold cross-validation on the RNAstralign dataset. Horizontal scale was adjusted to improve visualization.

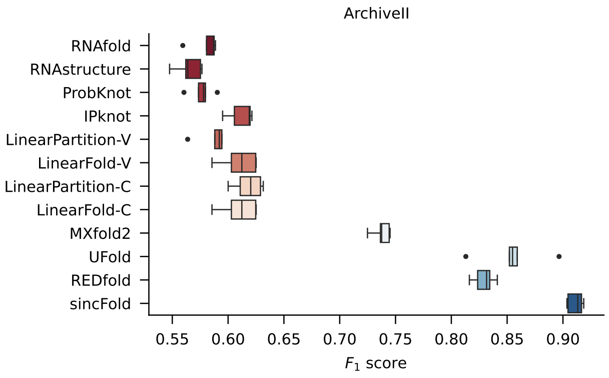

Figure 5: Comparative results among classical folding methods, DL-based folding methods and sincFold, for cross-validation on the ArchiveII dataset.

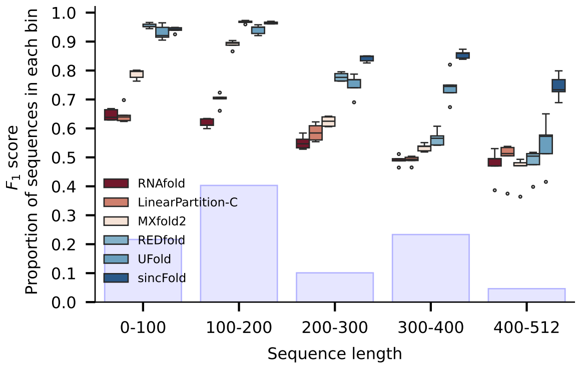

Figure 6: Detailed F_1 performance for each method according to the mean lengths of the sequences, from shorter (left) to longer (right) sequences.

learnable ones) improve performance. This also makes evident that structures on the left side are really harder to predict, even for classical models that are not trained and thus they are agnostic to the test-train structural distances. For the 23 structures in the first 2 bins of distances all methods have median $F_1 < 0.50$. It can be seen that for distances between 0.40 and 0.25, both classical and DL methods have again a low performance $F_1 \in (0.30, 0.60)$. In the middle cases, from 0.25 to 0.20 distance, DL methods are slightly better than classical ones. Finally, for the lowest test-train structural distances (< 0.15), DL methods are clearly better for RNA secondary structure prediction, being sincFold the best method in all cases, improving classical methods from a distance of 0.25 and all other trainable methods from 0.20. These trends can be explained by two facts. First, the abundance of structure samples benefits DL models more than the classical ones because the former have more cases to learn from. Besides, the benefits for the classical methods are indirect since they do not learn, but were developed looking at the most abundant or popular structures that therefore better fit the thermodynamic models. Secondly, based on the advantage of structure abundance for data-driven approaches, sincFold is the one that best takes advantage of the ability to learn from more distant samples, regardless of how much is known about the thermodynamics of the molecules. This is evident even when distance is around 0.25 and thus far from overfitting from training samples.

4.3 Homology-aware validation

For a deeper performance analysis of the methods considering homology between training and testing partitions, in this section we performed experiments with a more rigorous control of homology, instead of using random partitions. Table 1 shows the results of testing models in a non-redundant set of RNA sequences at 80% sequence-identity cutoff (TR0-TS0 partitions). The table reports comparative results on the TS0 test set according to F_1, WL and INF metrics. In all cases, the three metrics are consistent and show that sincFold is the best method to

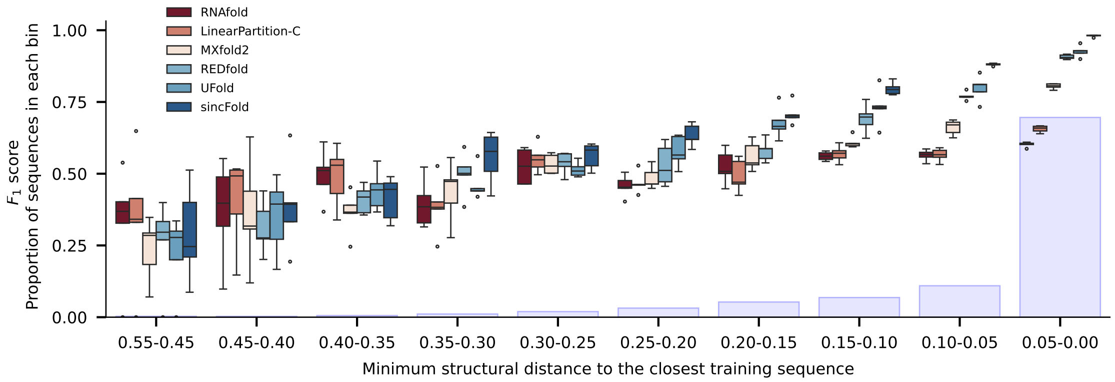

Figure 7: Mean F_1 scores for each method according to test-train structural distance, from large distances (left) to low distances (right). Bars indicate the proportion of each bin in the dataset.

Table 1: Performance for methods on bpRNA. The TR0 partition is used for training and the TS0 partition for testing.

Method	F_1	WL	INF
RNAfold	0.508	0.681	0.520
LinearPartition-C	0.547	0.737	0.556
MXfold2	0.502	0.736	0.509
REDfold	0.621	0.799	0.626
UFold	0.580	0.743	0.590
sincFold	0.631	0.827	0.641

Table 2: Inter-family performance comparison in ArchiveII, between sincFold and classical, hybrid and DL methods.

Method	F_1	WL	INF
RNAfold	0.580	0.699	0.580
LinearPartition-C	0.618	0.738	0.619
MXfold2	0.523	0.669	0.523
REDfold	0.348	0.609	0.366
UFold	0.351	0.602	0.370
sincFold	0.414	0.631	0.432

predict RNA structures with low homology to the training set, and there is almost a 10% performance gap with the classical methods.

For benchmarking inter-family performance, a family-fold cross-validation in the ArchiveII dataset was performed. That is, one family is left out for testing per cross-validation fold and the rest of the families are used for training as in [45]. This eliminates most of the homology to the training set, providing a hard measure of performance and, thus, allows estimating future performance on novel RNAs that do not belong to any well-known family.

Table 2 shows the average inter-family performance comparison with several metrics between sincFold and other classical, hybrid and pure DL methods for RNA secondary structure prediction. It can be seen that all the three metrics used, F_1 score, WL and INF metric, consistently indicate that sincFold is the best DL model to predict novel RNA structures of families of RNA never seen during training. As seen previously, performance of the hybrid method is in-between classical and DL methods. Obviously classical methods obtain the best performance here since they do not fully comply with the cross-family validation. This is because they use constraints and thermodynamic parameters that have been experimentally determined from the hairpin loops and other important structures that were most frequently found in most of the RNA families in this dataset [53, 54, 55, 56, 57]. In terms of F_1, the difference between the best DL method (sincFold) and the best classical method (LinearPartition-C) is 0.191, and 0.193 in the case of INF. However, looking at the WL metric this gap is much lower, being 0.098. This suggests that when measuring the graph structural information of the predictions, DL and classical methods are close in performance for inter-family validation.

Table 3 shows the detailed performance of DL methods for each family in the ArchiveII dataset. Full results for all methods and all measurements for each family can be found in the Supplementary Material, Table S1. The 9 RNA families are characterized in terms of number of samples, average length, structural distance and sequential distance to the other families. It can be seen that when the grp1 family is used as the test set, all methods have a moderate to low performance. The best DL method here achieves $F_1 = 0.429$. This is a family with a very low number of examples, which have a mean sequence length that is longer than most of the other families, with a moderate structural distance to the other families and high sequence distance to the rest of the dataset. In the case of the tmRNA family, all methods have low performance as well, but here both sincFold and UFold achieve the best result. In spite of having more testing examples (and thus the training set is much smaller), the characteristics of this family are similar to grp1 regarding structure and sequence distance to the

Table 3: Inter-family performance detail of the F_1 score in ArchiveII for each RNA family in the comparison between sincFold and other DL RNA secondary structure prediction methods.

	grp1	tmRNA	tRNA	5s	srp	telom.	RNaseP	16s	23s
family size	74	462	557	1283	914	35	454	65	15
ave(sequence length)	375	366	77	119	180	438	332	317	326
ave(min(structural distance))	0.526	0.508	0.445	0.497	0.548	0.531	0.519	0.490	0.574
UFold	**0.429**	0.347	0.492	0.377	0.199	**0.177**	0.421	0.332	0.394
REDfold	0.311	0.274	0.415	**0.472**	0.179	0.112	0.358	0.366	**0.402**
sincfold	0.350	**0.350**	**0.685**	0.439	**0.250**	0.154	**0.443**	**0.392**	**0.402**

training set, while sincFold achieves a similar results. The tRNA family is the one with the lowest sequence length, having a large number of examples. In this case, while the other DL methods have low performance, here sincFold achieves a performance of $F_1 = 0.685$ that is very close to the performance of many classical methods (Table S1). The srp and telomerase testing families are the hardest ones. The srp family, which is indeed very different from the rest of the families regarding sequence distance and structural distance, is better predicted by sincFold. The telomerase family has a very low number of samples, which are indeed very different from the rest of the families regarding mean sequence length (those are the longest sequences, almost double the average). In the case of the 16s and 23s families, also sincFold provides the best predictions.

In summary, sincFold shows improved performance in comparison with the other DL methods in the prediction of tmRNA, tRNA, srp, RNAseP, and 16s families; that is 5 out of 9 families. This is further evidence of the improved generalization capability that sincFold provides relative to the state-of-the-art DL methods.

5 Conclusions

In this work, we presented sincFold, an end-to-end deep learning model that can accurately predict the secondary structure from a RNA sequence without requiring multi-sequence alignments, or any other pre-processing of the input sequences. Local and distant relationships can be learnt effectively using a sequential 1D-2D architecture. Based on ResNet blocks, bottlenecks layers and a 1D-to-2D projection, it has proven to be better suited to identify structures that might defy traditional modeling, while reducing the effective number of trainable parameters. We show that sincFold outperforms other methods even with moderate structural distances between train and testing sequences. Results also show that sincFold, thanks to its capability for capturing a wide range distances in interactions, is significantly better than all other methods for the secondary structure prediction also in longer ncRNA sequences (more than 200 nucleotides). In an inter-family evaluation, sincFold performed better than other state-of-the-art DL approaches, showing that RNA structure predictions can still be improved with trainable methods.

Competing interests

No competing interest is declared.

Funding

We would like to thank to the AWS Cloud Credit for Research, the Ministerio de Producción, Ciencia y Tecnología, Santa Fe (PEICID-2022-075) and the Agencia Nacional de Promoción de la Investigación, el Desarrollo Tecnológico y la Innovación (Ciencia y tecnología contra el hambre) for their support for this work. This work was supported by ANPCyT (PICT 2018 3384, PICT 2022 0086) and UNL (CAI+D 2020 115). We also thank Nvidia for donating Titan XP GPUs used in the research.

References

[1] Zhang, P., Wu, W., Chen, Q., and et al. (2019). Non-coding RNAs and their integrated networks. *Journal of Integrative Bioinformatics*, **16**(3).

[2] Mattick, J., Amaral, P., Carninci, P., and et al. (2023). Long non-coding RNAs: definitions, functions, challenges and recommendations. *Nature Reviews Molecular Cell Biology*, **24**(6), 430–447.

[3] Winkle, M., El-Daly, S. M., Fabbri, M., and et al. (2021). Noncoding RNA therapeutics — challenges and potential solutions. *Nature Reviews Drug Discovery*, **20**(8), 629–651.

[4] Chen, X. and Huang, L. (2022). Computational model for ncRNA research. *Briefings in Bioinformatics*, **23**(6).

[5] Sloma, M. F. and Mathews, D. H. (2016). Exact calculation of loop formation probability identifies folding motifs in RNA secondary structures. *RNA*, **22**(12), 1808–1818.

[6] Watson, J. and Crick, F. (1953). Molecular structure of nucleic acids: A structure for deoxyribose nucleic acid. *Nature*, **171**(4356), 737–738.

[7] Varani, G. and McClain, W. H. (2000). The g·u wobble base pair. *EMBO reports*, **1**(1), 18–23.

[8] Gao, W., Yang, A., and Rivas, E. (2023). Thirteen dubious ways to detect conserved structural rnas. *IUBMB Life*, **75**(6), 471–492.

[9] Spokoini-Stern, R., Stamov, D., Jessel, H., and et al. (2020). Visualizing the structure and motion of the long noncoding rna hotair. *RNA*, **26**(5), 629–636.

[10] Fürtig, B., Richter, C., Wöhnert, J., and et al. (2003). NMR spectroscopy of RNA. *ChemBioChem*, **4**(10), 936–962.

[11] Keel, A. Y., Rambo, R. P., Batey, R. T., and et al. (2007). A general strategy to solve the phase problem in RNA crystallography. *Structure*, **15**(7), 761–772.

[12] Chorostecki, U., Willis, J., Saus, E., and et al. (2021). *Profiling of RNA Structure at Single-Nucleotide Resolution Using nextPARS*, pages 51–62. Springer US, New York, NY.

[13] Ding, Y., Tang, Y., Kwok, C., and et al. (2014). In vivo genome-wide profiling of rna secondary structure reveals novel regulatory features. *Nature*, **505**(7485), 696—700.

[14] Loughrey, D., Watters, K. E., Settle, A. H., and et al. (2014). Shape-seq 2.0: systematic optimization and extension of high-throughput chemical probing of rna secondary structure with next generation sequencing. *Nucleic Acids Research*, **42**, e165 – e165.

[15] Ross, C. J. and Ulitsky, I. (2022). Discovering functional motifs in long noncoding RNAs. *WIREs RNA*.

[16] Zuker, M. and Stiegler, P. (1981). Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, **9**(1), 133–148.

[17] Schroeder, S. J. and Turner, D. H. (2009). Optical melting measurements of nucleic acid thermodynamics. In *Methods in Enzymology*, pages 371–387. Elsevier.

[18] Turner, D. H. and Mathews, D. H. (2009). NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. *Nucleic Acids Research*, **38**(suppl_1), D280–D282.

[19] Mathews, D. H., Sabina, J., Zuker, M., and et al. (1999). Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *Journal of Molecular Biology*, **288**(5), 911–940.

[20] Reuter, J. S. and Mathews, D. H. (2010). RNAstructure: software for RNA secondary structure prediction and analysis. *BMC Bioinformatics*, **11**(1), 1–10.

[21] Bellaousov, S. and Mathews, D. H. (2010). ProbKnot: Fast prediction of RNA secondary structure including pseudoknots. *RNA*, **16**(10), 1870–1880.

[22] Lorenz, R., Bernhart, S. H., Höner, C., Tafer, H., and et al. (2011). ViennaRNA package 2.0. *Algorithms for Molecular Biology*, **6**(1), 1–10.

[23] Huang, L., Zhang, H., Deng, D., and et al. (2019). LinearFold: linear-time approximate RNA folding by 5'-to-3' dynamic programming and beam search. *Bioinformatics*, **35**(14), i295–i304.

[24] Zhang, H., Zhang, L., Mathews, D. H., and et al. (2020). LinearPartition: linear-time approximation of RNA folding partition function and base-pairing probabilities. *Bioinformatics*, **36**(Supplement_1), i258–i267.

[25] Sato, K., Akiyama, M., and Sakakibara, Y. (2022). RNA secondary structure prediction using deep learning with thermodynamic integration. *Nature Communications*, **12**(1), 1–10.

[26] Bugnon, L., Edera, A., Prochetto, S., and et al. (2022). Secondary structure prediction of long noncoding RNA: review and experimental comparison of existing approaches. *Briefings in Bioinformatics*, **23**(4).

[27] Wu, K. E., Zou, J. Y., and Chang, H. (2023). Machine learning modeling of RNA structures: methods, challenges and future perspectives. *Briefings in Bioinformatics*.

[28] Jumper, J., Evans, R., Pritzel, A., and et al. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, **596**(7873), 583–589.

[29] Zhang, H., Zhang, C., Li, Z., and et al. (2019). A new method of RNA secondary structure prediction based on convolutional neural network and dynamic programming. *Frontiers in Genetics*, **10**, 1–10.

[30] Singh, J., Hanson, J., Paliwal, K., and et al. (2019). RNA secondary structure prediction using an ensemble of two-dimensional deep neural networks and transfer learning. *Nature Communications*, **10**(1).

[31] Fu, L., Cao, Y., Wu, J., and et al. (2022). UFold: Fast and accurate RNA secondary structure prediction with deep learning. *Nucleic Acids Research*, **50**(3), e14.

[32] Chen, C.-C. and Chan, Y.-M. (2023). REDfold: accurate RNA secondary structure prediction using residual encoder-decoder network. *BMC Bioinformatics*, **24**(1).

[33] Schneider, B., Sweeney, B., Bateman, A., and et al. (2023). When will RNA get its AlphaFold moment? *Nucleic Acids Research*, **51**(18), 9522–9532.

[34] Flamm, C., Wielach, J., Wolfinger, M., and et al. (2022). Caveats to deep learning approaches to rna secondary structure prediction. *Frontiers in Bioinformatics*, **2**.

[35] Justyna, M., Antczak, M., and Szachniuk, M. (2023). Machine learning for RNA 2D structure prediction benchmarked on experimental data. *Briefings in Bioinformatics*, **24**(3), bbad153.

[36] Zhao, Q., Zhao, Z., Fan, X., and et al. (2021). Review of machine learning methods for RNA secondary structure prediction. *PLOS Computational Biology*, **17**(8), e1009291.

[37] Singh, J., Paliwal, K., Zhang, T., and et al. (2021). Improved RNA secondary structure and tertiary base-pairing prediction using evolutionary profile, mutational coupling and two-dimensional transfer learning. *Bioinformatics*, **37**(17), 2589–2600.

[38] Akiyama, M., Sato, K., and Sakakibara, Y. (2018). A max-margin training of RNA secondary structure prediction integrated with the thermodynamic model. *Journal of Bioinformatics and Computational Biology*, **16**(06), 1840025.

[39] Wang, L., Liu, Y., Zhong, X., and et al. (2019). DMfold: A novel method to predict RNA secondary structure with pseudoknots based on deep learning and improved base pair maximization principle. *Frontiers in Genetics*, **10**, 1–10.

[40] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Lecture Notes in Computer Science*, pages 234–241. Springer International Publishing.

[41] He, K., Zhang, X., Ren, S., and et al. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[42] Tan, Z., Fu, Y., Sharma, G., and Mathews, D. H. (2017). TurboFold II: RNA structural alignment and secondary structure prediction informed by multiple homologs. *Nucleic Acids Research*, **45**(20), 11570–11581.

[43] Fu, L., Niu, B., Zhu, Z., and et al. (2012). CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, **28**(23), 3150–3152.

[44] Baulin, E., Yacovlev, V., Khachko, D., and et al. (2016). URS DataBase: universe of RNA structures and their motifs. *Database*, **2016**, baw085.

[45] Szikszai, M., Wise, M., Datta, A., and et al. (2022). Deep learning models for rna secondary structure prediction (probably) do not generalize across families. *Bioinformatics*, **38**(16), 3892–3899.

[46] Mathews, D. H. (2019). How to benchmark RNA secondary structure prediction accuracy. *Methods*, **162-163**, 60–67.

[47] Magnus, M., Antczak, M., Zok, T., and et al. (2019). Rna-puzzles toolkit: a computational resource of rna 3d structure benchmark datasets, structure manipulation, and evaluation tools. *Nucleic Acids Research*.

[48] Parisien, M., Cruz, J., Westhof, E., and et al. (2009). New metrics for comparing and assessing discrepancies between rna 3d structures and models. *RNA*, **15**(10), 1875–1885.

[49] Runge, F., Franke, J. K. H., Fertmann, D., and et al. (2023). Rethinking performance measures of rna secondary structure problems. *NeuIPs 2023 - Machine Learning in Structural Biology Workshop*, **1**, 1–12.

[50] Fontana, W., Konings, D., Stadler, P., and et al. (1993). Statistics of rna secondary structures. *Biopolymers*, **33**(9), 1389–1404.

[51] Hofacker, I., Fontana, W., Stadler, P., and et al. (1994). Fast folding and comparison of rna secondary structures. *Monatshefte fur Chemie Chemical Monthly*, **125**(2), 167–188.

[52] Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, **13**(10), 281–305.

[53] Mathews, D., Disney, M., Childs, J., and et al. (2004). Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of rna secondary structure. *Proceedings of the National Academy of Sciences*, **101**(19), 7287–7292.

[54] Proctor, D., Schaak, J., Bevilacqua, J., and et al. (2002). Isolation and characterization of a family of stable rna tetraloops with the motif ynmg that participate in tertiary interactions. *Biochemistry*, **41**(40), 12062–12075.

[55] Antao, V., Lai, S., and Tinoco, I. (1991). A thermodynamic study of unusually stable rna and dna hairpins. *Nucleic Acids Research*, **19**(21), 5901–5905.

[56] Antao, V. and Tinoco, I. (1992). Thermodynamic parameters for loop formation in rna and dna hairpin tetraloops. *Nucleic Acids Research*, **20**(4), 819–824.

[57] Groebe, D. and Uhlenbeck, O. (1988). Characterization of rna hairpin loop stability. *Nucleic Acids Research*, **16**(24), 11725–11735.