COMPARISON AND BENCHMARK OF DEEP LEARNING METHODS FOR NON-CODING RNA CLASSIFICATION

Constance Creux
IBISC
Université Paris-Saclay, Univ Evry
Evry-Courcouronnes, France
constance.creux@univ-evry.fr

Farida Zehraoui
IBISC
Université Paris-Saclay, Univ Evry
Evry-Courcouronnes, France
farida.zehraoui@univ-evry.fr

François Radvanyi
Molecular Oncology
PSL Research University, CNRS, UMR 144,
Institut Curie
Paris, France
francois.radvanyi@curie.fr

Fariza Tahi
IBISC
Université Paris-Saclay, Univ Evry
Evry-Courcouronnes, France
fariza.tahi@univ-evry.fr

ABSTRACT

The grouping of non-coding RNAs into functional classes started in the 1950s with housekeeping RNAs. Since, multiple additional classes were described. The involvement of non-coding RNAs in biological processes and diseases has made their characterization crucial, creating a need for computational methods that can classify large sets of non-coding RNAs. In recent years, the success of deep learning in various domains led to its successful application to non-coding RNA classification. Multiple novel architectures have been developed, but these advancements are not covered by current literature reviews. We propose a comparison of the different approaches and of ncRNA datasets proposed in the state-of-the-art. Then, we perform experiments to fairly evaluate the performance of various tools for non-coding RNA classification. With regard to these results, we assess the relevance of the different architectural choices and provide recommendations to consider in future methods.

***K*eywords** non-coding RNA, non-coding rna classification, deep learning

Introduction

The classification of non-coding RNAs (ncRNAs) in groups with similar functions started in the 1950s, with the discovery of tRNAs (transfer RNAs) and rRNAs (ribosomal RNAs) [1]. The description of other ncRNA classes involved in cell maintenance followed, but research was still mainly focused on protein-coding genes. In the 2000s, efforts like the Human Genome Project [2] highlighted that 98% of the genome is non-coding. Consequently, attention started to turn to ncRNA. Novel classes of ncRNA with regulatory functions, like miRNA (microRNA) or lncRNA (long ncRNA), were described. Studies have shown that ncRNAs are implicated in various biological processes and diseases [3, 4, 5], underscoring their potential as biomarkers and therapeutic targets, and thus, the need to characterize them. Relative to the fact that most of the genome is non-coding, the number of well-known ncRNAs is low. This context creates a need for computational methods that can rapidly handle a large amount of ncRNAs. To this aim, different computational ncRNA classifiers have been developed over the years.

The term 'classification' of ncRNAs can have different meanings in the literature. It can refer to the *identification* of ncRNAs, i.e., separating coding from non-coding RNAs [6, 7, 8, 9]. It can also refer to the identification of one specific class of ncRNA in a dataset, or *class-specific prediction* [10, 11, 12]. A third use of the term is for *multi-class*

classification, taking as input a set of ncRNAs and associating each one to a ncRNA class, which this review focuses on.

There are many classes of ncRNAs, and multiple levels of description. Generally, ncRNAs are sorted into relatively large groups that share characteristics and should perform a common function. For example, tRNAs have a distinctive three-leafed clover structure, and their function is to carry amino acids to the ribosome for protein synthesis. miRNAs are around 22nt long, their precursors have a hairpin structure, and they serve as mRNA silencers. To this day, there is no exhaustive ncRNA classification: new classes can be discovered, and existing classes can be further characterized. One ncRNA class suffers particularly from this lack of definition - lncRNAs, which are defined as sequences longer than 200nt. Bar this criterion, RNAs belonging to this class are disparate and can have multiple functions. Due to this lack of precision around lncRNAs, ncRNA classification methods tend to focus on classes of small ncRNAs (sncRNAs) such as the ones cited earlier: rRNAs, tRNAs, miRNAs...

Multiple reviews have described the different ncRNA classification approaches over the years [13, 14]. However, since 2017, there has been a shift in the type of methods employed, with the use of deep learning (DL) becoming prevalent. To date, DL approaches for ncRNA classification have been presented in different publications, but no unbiased overview has been given. Moreover, in terms of performances, published benchmarks present several flaws: most papers use the results introduced in older articles instead of re-executing tools, even when experimental protocols are not comparable (some results are obtained on a held-out test set, others are averaged over ten-fold cross-validation). These issues render the performances presented incomparable.

This paper intends to fill a gap in the literature, providing a review of the different computational approaches, as well as a fair comparison of performance. The following section describes the different DL methods developed for ncRNA classification. The datasets used in the literature are then presented, before providing a performance comparison of state-of-the-art methods. We discuss the relevance of the different approaches in the next section, before some concluding remarks.

1 ncRNA classification methods state-of-the-art

In a 2017 review by Zhang et al. [14], three categories of methods are described:

- Homology-based methods, which classify ncRNAs based on evolutionary conservation [15, 16, 17]. These are based on sequence or structure alignment algorithms. These methods are limited, as they can only be used for well-known and well-conserved classes.
- *De novo* methods, based on feature information from the sequence or secondary structure. Until a few years ago, methods were based on standard Machine Learning (ML) algorithms, like Support Vector Machines [18] and Random Forest [19]. These rely on feature extraction, with the choice of features being critical.
- Transcriptional sequencing-based methods were developed to take advantage of the data created by Next-Generation Sequencing technologies. These methods could also be based on alignment or standard ML algorithms, but taking as input raw sequencing data [20, 21, 22, 23].

With the development of DL, *de novo* methods rose in popularity, taking inspiration from fields like computer vision and natural language processing. One important deviation from earlier methods is that feature extraction, a step that used to have a great impact on performance, is no longer required for DL methods: the network itself can extract relevant information from a simple representation of the data. Existing approaches are summarized in Figure 1, and details are given below.

The success of CNNs (Convolutional Neural Networks) motivated the development of multiple CNN-based methods for ncRNA classification: nRC, ncRDeep, ncrna-deep, RPC-snRC, ncRDense and imCnC. nRC [24], published in 2017, was the first method to use deep learning in the context of ncRNA classification. As input, ncRNAs are represented by vectors, containing information on the presence of discriminative sub-structures in the secondary structure (identified by the MoSS algorithm [25]). The method is based on the succession of two convolutional layers to extract relevant information, and FCLs (Fully-Connected Layers) for classification. Proposed three years later, ncRDeep [26] is another method based on two convolutions followed by FCLs. However, the input is a simple representation of the sequence, a matrix created from one-hot encodings of nucleotides. In the method by Noviello et al. [27] (which we call ncrna-deep), the number of convolutional layers rises to five. Authors tested multiple sequence encodings (space-filling curves or one-hot encoding of 1-mers, 2-mers or 3-mers[1] present in the sequence). A one-hot encoding of 1-mers, which corresponds to the same representation as ncRDeep, was selected for its superior results. This representation is also used by RPC-snRC [28], and fed to three consecutive Dense blocks (based on DenseNets, multiple convolutional layers with dense connections), with a final FCL for classification. ncRDense [29] is an evolution of

ncRDeep. In addition to sequence information, the method includes a second input: a matrix with multiple representations of nucleotides and one-hot encoding of the secondary structure. Both inputs are processed with independent DenseNets and merged. A final convolutional layer is followed by FCLs for classification.

imCnC [30] stands out as a multi-source CNN. Based on the idea that ncRNAs are related to chromatin states and epigenetic modifications, the approach builds a CNN for each input source (such as the sequence and different epigenetic information), before merging representations to classify samples with FCLs.

RNNs (Recurrent Neural Networks) are popular models for text representation. RNA sequences can be viewed as sentences, making the use of RNNs relevant to extract meaningful information from ncRNAs, as done by the methods ncRFP and NCYPred. ncRFP [31] uses a bi-LSTM (bidirectional Long Short-Term Memory), a subtype of RNNs. When learning a representation for a nucleotide, it is able to take into account close and distant elements, parsing the sequence both forward and backward. An attention mechanism allocates more weight to important information, and FCLs are used for classification. NCYPred [32] also uses a bi-LSTM and FCLs for classification. The main difference between the two is that in ncRFP, each nucleotide is encoded separately, while in NCYPred, the sequence is first transformed into overlapping 3-mers[1], and it is these 3-mers that are then encoded.

Several methods combine both approaches, using RNNs for sequence representation, from which relevant information is extracted by CNNs. This is the case for ncDLRES, ncDENSE and MFPred. ncDLRES [33] is an updated version of ncRFP's architecture. The bi-LSTM is replaced by a dynamic LSTM, which only parses the sequence in the forward sense, but has the advantage of allowing inputs of varying length. For classification, the FCLs are replaced by a ResNet (Residual Neural Network), which is composed of multiple convolutional layers with skip connections. ncDENSE [34] further evolves that architecture, replacing the dynamic LSTM by a dynamic bi-GRU, and the ResNet by a DenseNet. Dynamic bi-GRUs (Gated Recurrent Units) are RNNs that are faster than LSTMs. MFPred [35] represents inputs quite intricately: sequences are encoded into four sets of features, each processed by a dynamic bi-GRU, reshaped into matrices, and fused. Compared to ncDENSE, the DenseNet part is replaced by a variant of ResNets, called SE ResNet (Squeeze and Excitation ResNet), which models interdependencies between the four matrices of learned sequence encodings.

Two existing methods do not fall into the above categorization: RNAGCN and RAG+DL, both based on graphs. Taking the secondary structure as input, RNAGCN [36] uses a GNN (Graph Neural Network) with FCL for classification. GNNs are able to learn meaningful representations of non-Euclidean data. RNAGCN is, to date, the only method directly using ncRNA secondary structure without manual feature extraction, removing a source of error. The approach by Sutanto et al. [37] (which we call RAG+DL) is based on features extracted from the secondary structure. A detailed protocol is described to obtain a rich graph-theory-based representation of structures, called RNA-As-Graphs (RAG). Other than the extensive feature preparation step, the method is based on a simple DL network of five FCLs.

[1] A k-mer is a subsequence of k nucleotides.

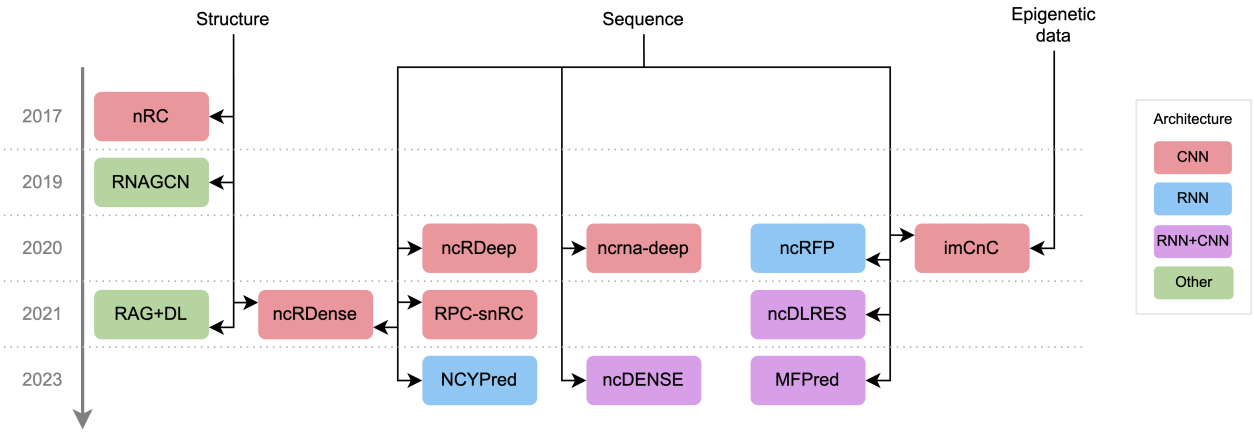

Figure 1: **Overview of existing approaches for ncRNA classification.** Methods are represented depending on the type of data they use (secondary structure, sequence, or other), and the type of deep learning architecture they are based on (CNN, RNN, both, or other).

To summarize, as seen on Figure 1, non-coding RNAs are often represented by their sequences. Their structures can also be used. Only one method uses another source of data, namely epigenetic marks. Two strategies emerge for input representation: the first is to encode data as a simple one-hot encoded matrix, while the other is to employ an RNN to learn a more complex representation. Classification is then usually performed with a CNN. Two state-of-the-art methods do not rely on CNNs or RNNs, and employ either a GNN or a simple succession of FCLs.

2 State-of-the-art datasets

ncRNAs can be viewed through different lenses. Their sequences, secondary and tertiary structures, as well as their expressions, or epigenetic modifications, all carry information. However, these different data types can be more or less accessible. Most recent datasets for ncRNA classification focus on the sequence. It has the advantage of being available in online databases in high quantity (e.g., almost 35 million sequences in the non-coding RNA database RNAcentral [38]). It can also be used to obtain the secondary structure. Indeed, due to a lack of experimentally validated structures for ncRNAs, it is common to predict secondary structure from the sequence. This can be done using external algorithms [39, 40].

Five datasets have been proposed in the literature to evaluate the different methods presented in the previous section. These are summarized in Table 1, and their curation is detailed below.

Source	Size	# Cl	# Uses	Lengths
nRC [24]	8,920	13	12	38-1182
imCnC [30]	42,000	16	1	-
ncrna-deep [27]	306,016	88	1	1-80
RAG+DL [37]	59,723	12	1	-
NCYPred [32]	45,447	13	2	42-500

Table 1: **Summary of the different state-of-the-art datasets for ncRNA classification.** Datasets are sorted in order of publication, with the name of the tool they were published with. The number of instances and classes in each dataset is given. We indicate how many times the dataset has been used by state-of-the-art ncRNA classification methods. Finally, we give the range of sequence lengths for accessible datasets.

The first dataset was proposed in 2017 by Fiannaca et al. [24], along with the nRC tool. Sequences were downloaded from the 12th release of the Rfam database [41] and the CD-HIT tool [42] was used to remove redundant sequences. A random selection was carried out to obtain (almost) balanced classes, with a final size of 8,920 sncRNAs spanning 13 classes. Since its proposal, nRC's dataset has been used by other tools to assess and compare performance: in fact, all state-of-the-art methods from the previous section use nRC's dataset, except for RAG+DL. The appeal of this dataset is that classes are (almost) balanced, and there is a reasonable number of classes to represent a variety of sncRNAs while returning intelligible results. However, this dataset presents a major flaw: 347 ncRNA are present in both the training and the test set (representing respectively 5.5% and 13.3% of the sets). This kind of data leakage casts doubt upon the performances reported by the various tools.

Boukelia et al. [30] constructed a second dataset in 2020, as their method imCnC does not solely rely on ncRNA sequences, but also includes other data sources that needed to be collected. This dataset contains 42,000 sequences of human ncRNAs downloaded from Rfam's 13th release [43]. It also includes four types of epigenetic modifications: DNA methylation, histone methylation, histone acetylation, and binding factors and regulators related to histones (from DeepBlue [44] and HHMD [45] databases). In total, 16 sncRNA classes are represented, though no information is given for the balance between classes. This dataset only includes human ncRNAs, while multiple species are represented in the other datasets presented in this section. The inclusion of epigenetic data is novel, and is a benefit: the authors' experiments show a gain in performance when including epigenetic information in addition to the sequence. This means that epigenetic data is relevant for ncRNA classification, and could be used to further characterize the different classes. The drawback of this dataset is the fact that it is not publicly available.

The same year, Noviello et al. [27] proposed a dataset in which ncRNAs are classified into 88 Rfam families. These families are defined as groups with RNAs that are evolutionarily conserved, have evidence for a secondary structure, and have some known function. They can sometimes be extremely specific. These families can be distributed among what authors call 'macro-classes', which correspond to the level of description given by the other datasets. A total of 306,016 sequences are present in the dataset, making it the largest state-of-the-art dataset.

In 2021, Sutanto et al. [37] curated a new dataset with sequences from Rfam 14.1 [46]. Duplicates were removed, as well as sequences that could not be processed with the tool they use to predict secondary structure, as authors are

interested in secondary structure motifs. Then, CD-HIT [42] was used to remove redundancy in the dataset. In total, the dataset comprises 59,723 RNAs representing 12 classes. This dataset is not available publicly.

The most recent dataset, from 2023, was proposed by Lima et al. [32] along with the tool NCYPred. It was built following the same protocol as nRC's dataset but with updated data. Using the release 14.3 of Rfam [46], authors were able to multiply by 5 the number of ncRNA in the dataset, obtaining 45,447 ncRNAs split into a training set and a test set. Note that in this dataset, classes are not balanced (some are represented by more than 5,000 examples, while others have less than 500). Another difference with nRC's dataset is that authors excluded two classes (scaRNA and IRES), citing the poor results of their method on these classes. Instead, two new classes were added (Y RNA and Y RNA-like). This dataset was used by one other recent method, MFPred.

The comparison between state-of-the-art datasets can be summed up to three key points: origin of data, redundancy and diversity of sequences, and class definition. As one of the largest databases on ncRNAs, all datasets use Rfam as their starting point, with more recent releases containing more sequences. imCnC's dataset [30] also integrates other databases for epigenetic data. Three of the five datasets ([24, 37, 32]) use the CD-HIT tool to remove redundant sequences. CD-HIT uses a predefined threshold to create groups of similar sequences, and removes all but one sequence per group from the dataset. In all cases, the threshold was set to 80%. While this step restricts the size of the dataset, it only removes examples that would not have been informative, while maintaining diversity. Models trained on these datasets could therefore be more efficient without sacrificing performance. Some datasets remove sequences containing degenerate nucleotides ([27, 32]), while the others do not. As degenerate nucleotides can be present in sequences obtained from RNA-sequencing, it can be interesting to represent them in these curated datasets as well. As for the classification task, most datasets represent around 13 commonly-used ncRNA classes, that group ncRNAs that share a function while not being too specific. Only ncrna-deep's dataset [27] differs, taking into a larger amount of classes. While providing a more precise functional annotation, these classes have the disadvantage of not all being on the same level of detail, with some classes being more broad and others very specific. Classes are more equal in that regard in the other proposed classification.

3 Benchmark

This section presents a comparison of different DL tools for ncRNA classification in terms of accessibility and performance.

We select state-of-the-art datasets to perform this comparison. Out of the five datasets described, three have been made available publicly: nRC's dataset, NCYPred's dataset, and ncrna-deep's dataset. We exclude the latter as we want to show classification results on different well-known ncRNA classes, and classes in this dataset are too specific and numerous. As the only two datasets that were used by more than one state-of-the-art tool, we keep nRC's dataset and NCYPred's dataset in our benchmark, referring to these as Dataset1 and Dataset2. As stated, Dataset1 presents an issue of data leakage, which we fix by removing from the test set the 347 sequences that were present in both training and test sets. We also confirmed that the same issue was not present in Dataset2.

Most state-of-the-art approaches propose an online version of their tool, as presented in Table 2. Either source code is shared and has to be installed by users, or web servers are created to be used directly. Sharing source code has the advantage of allowing users to re-train the model to adapt it to their own dataset. However, executing the code involves several steps that could cause issues, and can require a certain expertise. With web servers, users can upload their ncRNAs of interest, and a class prediction will be returned, without re-training the method. Web servers are easier to use as they require no installation, but they might be less effective on new data.

3.1 Experimental protocol

In the following, we assess the quality of prediction of six methods: nRC [24], RNAGCN [36], ncrna-deep [27], ncRDense [29], NCYPred [32] and MFPred [35]. RAG+DL [37] and RPC-snRC [28] are not available and thus cannot be tested. For a fair comparison, we need to be able to compare tools on the same datasets – this cannot be done for imCnC [30], the only tool to require epigenetic data, thus it is not included in this comparison. Finally, for tools that were developed by the same group, ample comparisons between old and new versions are already available. This concerns on one hand ncRDeep [26] and ncRDense [29], and on the other hand ncRFP [31], ncDLRES [33], ncDENSE [34] and MFPred [35]. In both cases, we chose to use the latest version: ncRDense and MFPred.

Out of the methods included, nRC, RNAGCN, ncrna-deep and MFPred were re-trained before predicting each dataset. Moreover, we tested different hyperparameters for nRC, RNAGCN and ncrna-deep, as listed in Table 3, and reported the best results. The same could not be done for MFPred as parameter values are fixed directly in the code, and no

Method	Usage
nRC [24]	***Web server: X – Source code: ✓ – Training: ✓*** Docker container: `https://hub.docker.com/r/tblab/nrc/`, GitHub repository: `https://github.com/IcarPA-TBlab/nrc`. The tool is easy to use: no pre-formatting of data is required, the commands are documented and an example is given.
RNAGCN [36]	***Web server: X – Source code: ✓ – Training: ✓*** GitHub repository: `https://github.com/emalgorithm/ncRNA-family-prediction`. The method is easy to use, as the process is documented. However, to use the tool on a new dataset, an outside tool has to be used to obtain secondary structures, which then need to be formatted correctly before using RNAGCN.
ncRFP [31]	***Web server: X – Source code: ✓ – Training: ✓*** Though not mentioned in the article, a GitHub repository exists: `https://github.com/linyuwangPHD/ncRFP`. Regarding ease of use, the documentation provided is short and does not clearly show which commands to run. Variables and hyperparameters are fixed inside the code and cannot be easily modified.
imCnC [30]	***Web server: X – Source code: ✓ – Training: ✓*** GitHub repository: `https://github.com/BoukeliaAbdelbasset/imCnC`. No instructions or examples of how to use the tool are given.
ncRDeep [26]	***Web server: ✓ – Source code: X – Training: X*** Web server: `https://nsclbio.jbnu.ac.kr/tools/ncRDeep/`. Although not mentioned on the site, there is a limit of 1,000 ncRNA per prediction task. Moreover, it is not possible to export results.
ncrna-deep [27]	***Web server: X – Source code: ✓ – Training: ✓*** GitHub repository: `https://github.com/bioinformatics-sannio/ncrna-deep`. Dataset preparation has to be done separately, but the process is explained and illustrated in the documentation and in the code.
RAG+DL [37]	***Web server: X – Source code: X – Training: X*** The repository containing the dataset and source code no longer exists.
ncRDense [29]	***Web server: ✓ – Source code: X – Training: X*** Web server: `https://nsclbio.jbnu.ac.kr/tools/ncRDense/`. Although not mentioned on the site, there is a limit of 1,000 ncRNA per prediction task. Moreover, it is not possible to export results.
ncDLRES [33]	***Web server: X – Source code: ✓ – Training: ✓*** GitHub repository: `https://github.com/linyuwangPHD/ncDLRES`. No examples of how to use the tool, but very basic instructions are given.
RPC-snRC [28]	***Web server: X – Source code: X – Training: X*** The GitHub repository linked in the paper no longer exists.
NCYPred [32]	***Web server: ✓ – Source code: ✓ – Training: ✓*** Web server: `https://www.gpea.uem.br/ncypred`, GitHub repository: `https://github.com/diegodslima/NCYPred`. The web server is easy to use and results can be exported. Many elements are given in the source code, but due to missing files, the code cannot be used.
ncDENSE [34]	***Web server: X – Source code: ✓ – Training: ✓*** GitHub repository: `https://github.com/ck-fighting/ncDENSE`. No examples of how to use the tool are given.
MFPred [35]	***Web server: X – Source code: ✓ – Training: ✓*** GitHub repository: `https://github.com/ck-fighting/MFPred`. Dataset preparation has to be done separately with scripts provided. Many files are in the repository, but the documentation is lacking, making it difficult to know how to use the tool.

Table 2: **Description of the availability and ease-of-use of ncRNA classification tools.** Each cell indicates if the tool is accessible through a web server, if source code can be downloaded, and if the tool can be re-trained. All links have been checked at the time of writing (September 2023).

directions are given on how to change them or what to change. ncRDense and NCYPred were only accessible to us as web servers, so they could not be re-trained, and were only used for prediction.

Method	Hyperparameters
nRC	Range of size of substructures {(2-4), (3-5)$^{(1,2)}$}, number of kernels in the first {10$^{(2)}$, 20$^{(1)}$} and the second {10, 20$^{(1,2)}$} convolutional layer.
RNAGCN	Size {64$^{(2)}$, 80$^{(1)}$, 128} and type {MPNN, GIN$^{(2)}$, GCN, GAT$^{(1)}$} of graph convolutions.
ncrna-deep	Sequence encoding: {1-mer$^{(1,2)}$, 2-mer, snake, hilbert, morton}.

Table 3: **Overview of tested hyperparameters.** Parameters that gave the best results on Dataset1 are denoted by $^{(1)}$, and same with $^{(2)}$ for Dataset2. Default values were used for parameters not listed.

Predictions were made on the provided test sets for Dataset1 and Dataset2. All samples were used, except in the case of NCYPred and MFPred, which cannot handle degenerate nucleotides, meaning that some samples had to be removed from Dataset1 for these tools.

Performance is measured with Accuracy, MCC (Matthews Correlation Coefficient), F1-score, Precision, Recall (or Sensitivity) and Specificity. For the last four metrics, we use the macro averaging generalization for multi-class problems, where scores are computed on each class separately, then averaged. The six scores are defined as follows:

$$Accuracy = \frac{\frac{1}{C}\sum_{c=1}^{C} TP_c}{S} \tag{1a}$$

$$\text{MCC} = \frac{(\sum_{c=1}^{C} TP_c) * S - \sum_{c=1}^{C}(p_c * t_c)}{\sqrt{(S^2 - \sum_{c=1}^{C} p_c^2)(S^2 - \sum_{c=1}^{C} t_c^2)}} \tag{1b}$$

$$\text{F1-score} = \frac{1}{C}\sum_{c=1}^{C} \frac{TP_c}{TP_c + 0.5(FP_c + FN_c)} \tag{1c}$$

$$\text{Precision} = \frac{1}{C}\sum_{c=1}^{C} \frac{TP_c}{TP_c + FP_c} \tag{1d}$$

$$\text{Recall} = \frac{1}{C}\sum_{c=1}^{C} \frac{TP_c}{TP_c + FN_c} \tag{1e}$$

$$\text{Specificity} = \frac{1}{C}\sum_{c=1}^{C} \frac{TN_c}{TN_c + FP_c} \tag{1f}$$

where C is the number of classes and S the number of samples. Respective of class c, TP_c are True Positives, FP_c are False Positives, TN_c are True Negatives and FN_c are False Negatives. p_c is the number of times class c was predicted, and t_c is the true number of members of class c.

3.2 Overall performance

Figure 2 presents the results obtained on both datasets, according to the six metrics defined previously.

As a general trend looking at both datasets, a divide can be observed between tools that were re-trained (nRC, RNAGCN, ncrna-deep and MFPred) and those only accessible as web servers (ncRDense and NCYPred). The former all have consistent performances across datasets, with more recent methods performing the best. For the latter, performance is dependent on the dataset they were trained on. ncRDense was trained on Dataset1, on which it is the third-best performing. However, on the previously unseen Dataset2, it is outperformed by all other methods. NCYPred obtains an average performance on Dataset2, on which it was trained, but it cannot classify the unknown Dataset1 as well as the other methods. Overall, ncrna-deep and MFPred consistently reach the highest scores, across the different metrics and for the two datasets.

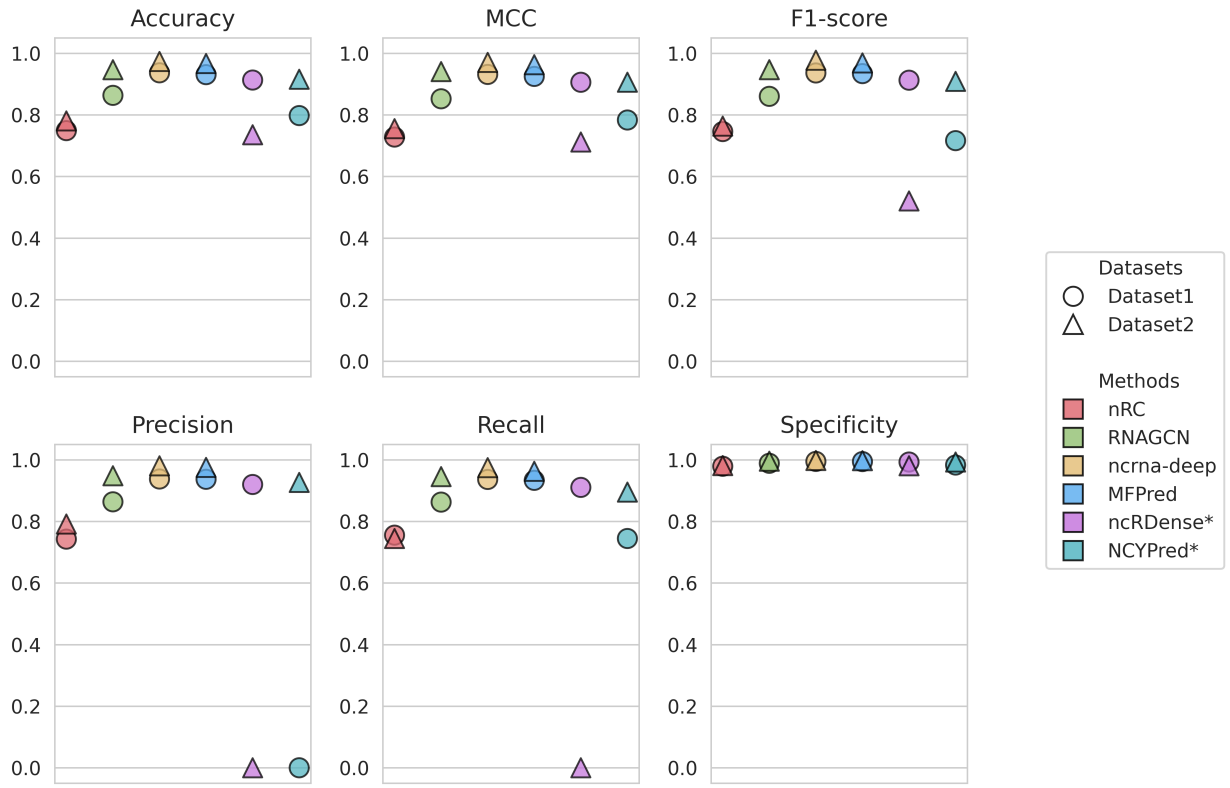

Figure 2: **Comparison of overall performance of the state-of-the-art methods on both test sets.** nRC, RNAGCN, ncrda-deep and MFPred were re-trained on each dataset before prediction, while ncRDense and NCYPred (marked with an asterisk) could only be used for prediction.

Both datasets represent similar sncRNA classes, yet results are, in general, much better on Dataset2. This can be due to multiple factors. The first is that Dataset2 is much larger than Dataset1. Models are trained on a larger variety of ncRNAs, making the prediction of new examples easier. Another reason is that sequences in Dataset2 do not contain degenerate nucleotides, which can be more difficult to handle for certain models. As these can be present in sequences obtained from RNA-seq experiments, depending on the use case, it might be beneficial to use a classifier that is robust to these types of nucleotides. Finally, as the description of ncRNA classes is an open problem, the set of classes that should be included in datasets is not fixed. While Dataset1 and Dataset2 are mostly comparable, each includes two classes not present in the other dataset. This can impact reported performance: for example, the two classes included solely in Dataset1 seem more difficult to predict, thus lowering overall scores. Therefore, while results on Dataset2 are quite satisfying, it is important to keep in mind that they relate to one set of ncRNA classes, which does not include all existing ncRNA classes.

Each tool in this benchmark was presented in a publication that showed experimental results. We compare reported performance to the one we measured, and illustrate the difference in Figure 3. All publications use the same formula for multi-class MCC, therefore, we base our comparison on this metric. nRC, RNAGCN, ncrna-deep, MFPred, ncRDense and NCYPred all use Dataset1 (albeit the biased version with data leakage). Since it is more recent, only MFPred and NCYPred show results on Dataset2.

Results on Dataset1 were significantly worse in our experiments than what was originally reported by the various tools. These drops in performance are likely due, in part, to the correction of the data leakage problem. Interestingly, after our hyperparameter search, the results we obtain for RNAGCN are slightly better than the ones reported by the method. The performance of NCYPred is significantly worse in our experiment. This could be due in part to model training: in their benchmark, authors re-trained their tool on Dataset1, which we could not do, only having access to the web server version. This means that two classes from Dataset1 are unknown by the model and cannot be predicted. Moreover, sequences containing degenerate nucleotides cannot be predicted by NCYPred, further lowering the scores.

On Dataset2, reported results seem more reliable as we only observe negligible differences in MCC.

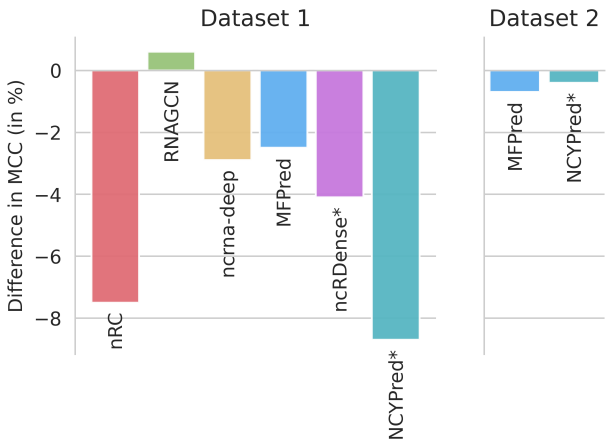

Figure 3: **Difference between MCC reported by each state-of-the-art method and MCC measured in our benchmark.** A negative value of $-x\%$ signifies that reported MCC was $x\%$ higher than the value we measured. We re-trained nRC, RNAGCN, ncrna-deep and MFPred, while for ncRDense and NCYPred (marked with an asterisk) the web server was used.

3.3 Performance on individual classes

A great challenge in ncRNA classification is achieving high predictions across classes. When evaluating multi-class ncRNA classification tools, it is important to analyze class-specific predictions and see how the method fares on different classes. Figure 4 presents the F1-scores obtained for each class on both datasets using the formula:

$$\text{F1-score} = \frac{TP}{TP + 0.5(FP + FN)} \tag{2}$$

Figure 4 emphasizes that more recent methods make more accurate predictions: ncrna-deep and MFPred make very accurate predictions for almost all classes. RNAGCN is also able to reach high performances on Dataset2, however, it struggles for some classes on Dataset1, perhaps needing a larger amount of examples to characterize them. The difference in F1-score between classes is quite striking for nRC, which is able to predict some classes quite well, such as Intron-gpI and 5S-rRNA, but cannot characterize well enough some other classes like miRNA (especially on Dataset1) or HACA-box. Aside from the classes it was not trained on (IRES and scaRNA), NCYPred's performances are on par with that of other tools on Dataset1. The same cannot be said for ncRDense, which does not only fail to predict the classes it was not trained on (Y RNA and Y RNA-like), but also obtains relatively poor performances for most other classes in Dataset2. This emphasizes the point that not allowing users to re-train a tool can be detrimental.

It is interesting to see that, although recent tools make generally very accurate predictions, the more problematic classes are the same as those for earlier methods: miRNA, IRES or HACA-box to name a few.

3.4 Computation time

In the following, we compare the computation time of benchmark methods, which we measure when running the codes. nRC, RNAGCN, ncrna-deep and MFPred were executed on the same machine (NVIDIA GeForce RTX 3090), with GPU acceleration for all methods but nRC which only supports CPU computation. These are presented in Table 4.

Method	Prep	Train	Pred
nRC	-	20m 27s (±23s)	6m 37s (±34s)
RNAGCN	6s	2m 59s (±1.5m)	2.8s (±1.7s)
ncrna-deep	0.29s	27s (±8s)	0.3s (±0.2s)
MFPred	4s	35m 37s (±39s)	5s (±1s)

Table 4: **Comparison of preprocessing (Prep), training (Train) and prediction (Pred) times.** Times are computed for the hyperparameter sets selected in Table 3. Times are given for 1,000 ncRNAs, averaged over both datasets. Note that preprocessing cannot be separated from training / prediction in nRC and cannot be computed separately.

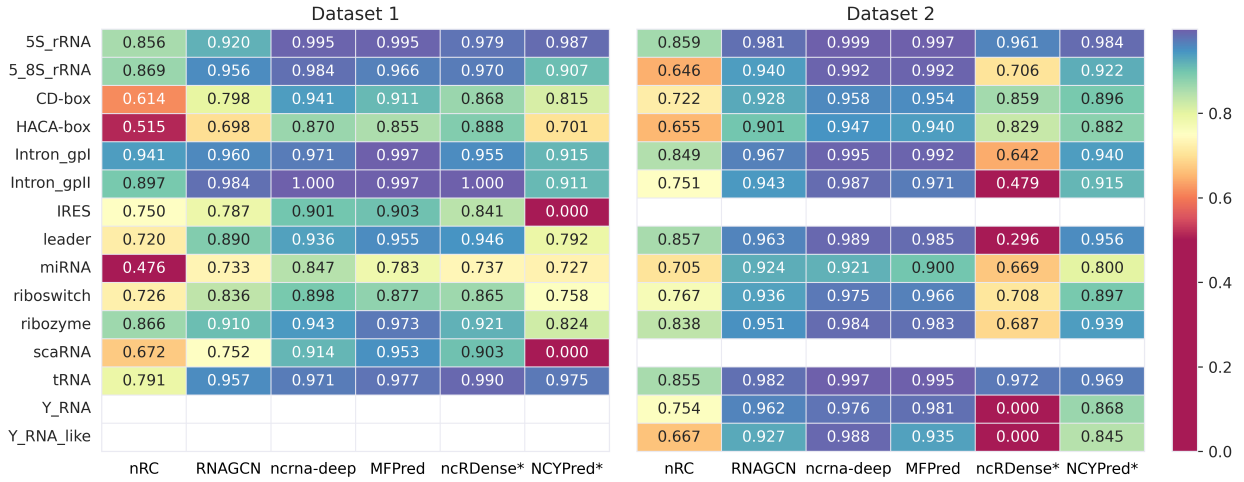

Figure 4: **Comparison of F1-score for each ncRNA class on both datasets.** nRC, RNAGCN, ncrda-deep and MFPred were re-trained on each dataset before prediction, while ncRDense and NCYPred (marked with an asterisk) could only be used for prediction.

State-of-the-art methods require preprocessing - i.e., transforming input sequences to a format that can be handled by the model. This is an additional step to be done by users before using the tool, except for nRC for which preprocessing is included in the pipeline and cannot be done separately from training / prediction.

nRC and MFPred have considerably longer runtimes than RNAGCN and ncrna-deep. For nRC, this can be explained by the method's lack of GPU support, as computations on CPU take much longer. For MFPred, the lack of speed is largely due to the complexity of the model: as presented previously, the data is processed by four different bi-GRUs, before passing through the large architecture that is the SE ResNet. This represents considerably more layers than what is used by the other methods, which naturally leads to a longer training time.

For the web servers ncRDense and NCYPred, it is difficult to measure prediction time precisely, as it depends on multiple factors: network connection, user speed, ... Moreover, ncRDense cannot predict more than 1,000 ncRNAs at once, further complicating the measure of prediction time. As an order of magnitude, for 1,000 ncRNAs, we obtained predictions in around 20 seconds with NCYPred and 30 seconds with ncRDense. No preprocessing is required.

Overall, ncrna-deep stands out, only requiring a fraction of other methods' runtime.

4 Discussion

This section discusses the relevance of architectural choices in state-of-the-art methods with regard to the results obtained.

The first choice to make in the development of the architecture is how to represent data. All methods included in the benchmark take RNA sequences as input, but apply different transformations. For nRC and RNAGCN, the sequence is temporary information, only used to predict the secondary structure. The secondary structure is then used to classify ncRNAs. This approach is based on the widely accepted idea that the function of an ncRNA is linked to its structure [47], and was adopted by many earlier ncRNA classification methods [18, 19]. It is interesting to see that this experimental observation does not translate particularly well to computational approaches, as methods that use the structure do not give the best results in our benchmark. A potential explanation could be that the used representation of secondary structure is not informative. Structure prediction tools propose multiple potential secondary structures, and the structure minimizing free energy is selected. However, it is not necessarily the structure that the ncRNA will take in organisms. Perhaps representing RNAs with an ensemble of possible structures instead of just one could improve classification scores.

Among methods that directly use the sequence, two representation approaches emerge in state-of-the-art methods: using a simple matricial representation of the sequence, or learning a meaningful representation of the sequence with RNN-based networks. These two approaches are represented in our benchmark, with ncrna-deep using a one-hot encoding of the sequence, and MFPred using four RNNs to extract different information from the sequence. Representation complexity is very different inncrna-deep and MFPred, but surprisingly, their performances are almost

identical. Table 4 suggests that the simple representation should in fact be preferred, as the time gain is considerable without any effect on performance.

One lack in state-of-the-art methods is the explainability of results. It is known that some classes are harder to predict than others, leading to the common practice of including per-class performance scores in publications, but no tool to this date provides result explanation. Explainable artificial intelligence [48, 49] could be used to highlight the feature patterns of the different ncRNA classes, which would help gain a better understanding of classes.

While results have shown that the sequence is enough to obtain good classification results, this should not be a limit in the development of new ncRNA methods. Using secondary structure as input was the state-of-the-art in ncRNA classification for many years, perhaps with new approaches competitive scores could be obtained. Other ways of handling the structure could be tested. For example, earlier ncRNA classification methods ([20, 21, 22, 23]) used sequencing data. The inclusion of epigenetic data by imCnC [30] also seemed promising. Indeed, while performance is an important aspect of ncRNA classification, an additional benefit in research could be to obtain a better characterization of classes – which includes describing them by patterns found at multiple levels, not just in the sequence. As stated in the introduction, the description of ncRNA classes and their functions is an ongoing process, in which computational ncRNA classifiers could be extremely useful.

Finally, there is an additional consideration regarding the evolution of ncRNA classes. The definition of the different classes is not fixed and is still being refined. As we discover new types of ncRNAs or redefine the characteristics of existing classes, ncRNA classification datasets are bound to evolve. In this context, it is important to propose tools that can be applied to and evaluated on new datasets. This means developing tools that are accessible, easy-to-use, and well-documented, and that have the option to be re-trained. Furthermore, the ability to discover new classes, or detect similarities between existing classes, could even be integrated into new ncRNA classification approaches.

Conclusion

The combination of a growing interest in ncRNAs over the past 30 years and recent outstanding performances of deep learning methods led to the development of DL-based methods for ncRNA classification. Novel architectures were proposed in the past six years, most being based on CNNs or RNNs. Most datasets represent ncRNAs by their sequence, and can include different ncRNA classes. In this publication, we review the different state-of-the-art contributions of the past six years. We also compare performance in an unbiased benchmark, including overall and class-specific prediction, as well as runtime. Finally, comparing the different architectures and their results allows us to identify potential recommendations for the future of computational ncRNA classification.

Funding

With financial support from ITMO Cancer of Aviesan within the framework of the 2021-2030 Cancer Control Strategy, on funds administered by Inserm.

References

[1] Antonin Morillon. *Long Non-coding RNA : The Dark Side of the Genome*. Elsevier Science, 2018.

[2] Eric S. Lander. Initial impact of the sequencing of the human genome. *Nature*, 470(7333):187–197, February 2011.

[3] Yu Li and Kris V. Kowdley. MicroRNAs in Common Human Diseases. *Genomics, Proteomics & Bioinformatics*, 10(5):246–253, October 2012.

[4] Jiri Sana, Petra Faltejskova, Marek Svoboda, et al. Novel classes of non-coding RNAs and cancer. *Journal of Translational Medicine*, 10:103, May 2012.

[5] Yiwen Fang and Melissa J. Fullwood. Roles, Functions, and Mechanisms of Long Non-coding RNAs in Cancer. *Genomics, Proteomics and Bioinformatics*, 14(1):42–54, February 2016.

[6] Noorul Amin, Annette McGrath, and Yi-Ping Phoebe Chen. Evaluation of deep learning in non-coding RNA classification. *Nature Machine Intelligence*, 1(5):246–256, May 2019.

[7] Ludovic Platon, Farida Zehraoui, Abdelhafid Bendahmane, et al. IRSOM, a reliable identifier of ncRNAs based on supervised self-organizing maps with rejection. *Bioinformatics*, (17):i620–i628, September 2018.

[8] Junghwan Baek, Byunghan Lee, Sunyoung Kwon, et al. LncRNAnet: Long non-coding RNA identification using deep learning. *Bioinformatics*, 34(22):3889–3897, November 2018.

[9] Jianghui Wen, Yeshu Liu, Yu Shi, et al. A classification model for lncRNA and mRNA based on k-mers and a convolutional neural network. *BMC bioinformatics*, page 469, September 2019.

[10] Mohamed Chaabane, Robert M. Williams, Austin T. Stephens, et al. CircDeep: Deep learning approach for circular RNA classification from other long non-coding RNA. *Bioinformatics*, 36(1):73–80, January 2020.

[11] Seunghyun Park, Seonwoo Min, Hyun-Soo Choi, et al. Deep recurrent neural network-based identification of precursor microRNAs. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[12] Anouar Boucheham, Vivien Sommard, Farida Zehraoui, et al. IpiRId: Integrative approach for piRNA prediction using genomic and epigenomic data. *PLOS ONE*, 12(6):e0179787, June 2017.

[13] Paul M. Krzyzanowski, Enrique M. Muro, and Miguel A. Andrade-Navarro. Computational approaches to discovering noncoding RNA. *WIREs RNA*, 3(4):567–579, 2012.

[14] Yi Zhang, Haiyun Huang, Dahan Zhang, et al. A Review on Recent Computational Methods for Predicting Noncoding RNAs. *BioMed Research International*, 2017:e9139504, May 2017.

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

[16] Andreas R. Gruber, Richard Neuböck, Ivo L. Hofacker, et al. The RNAz web server: Prediction of thermodynamically stable and evolutionarily conserved RNA structures. *Nucleic Acids Research*, 35(Web Server issue):W335–W338, July 2007.

[17] Stinus Lindgreen, Paul P. Gardner, and Anders Krogh. MASTR: Multiple alignment and structure prediction of non-coding RNAs using simulated annealing. *Bioinformatics*, 23(24):3304–3311, December 2007.

[18] Yan Karklin, Richard F. Meraz, and Stephen R. Holbrook. Classification of non-coding RNA using graph representations of secondary structure. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pages 4–15, 2005.

[19] Bharat Panwar, Amit Arora, and Gajendra PS Raghava. Prediction and classification of ncRNAs using structural information. *BMC Genomics*, 15(1):127, February 2014.

[20] Mario Fasold, David Langenberger, Hans Binder, et al. DARIO: A ncRNA detection and analysis tool for next-generation sequencing experiments. *Nucleic Acids Research*, 39(Web Server issue):W112–117, July 2011.

[21] Yuk Yee Leung, Paul Ryvkin, Lyle H. Ungar, et al. CoRAL: Predicting non-coding RNAs from small RNA-sequencing data. *Nucleic Acids Research*, 41(14):e137, August 2013.

[22] Cheng Yuan and Yanni Sun. RNA-CODE: A Noncoding RNA Classification Tool for Short Reads in NGS Data Lacking Reference Genomes. *PLOS ONE*, 8(10):e77596, October 2013.

[23] Pavankumar Videm, Dominic Rose, Fabrizio Costa, et al. BlockClust: Efficient clustering and classification of non-coding RNAs from short read RNA-seq profiles. *Bioinformatics*, 30(12):i274–i282, June 2014.

[24] Antonino Fiannaca, Massimo La Rosa, Laura La Paglia, et al. NRC: Non-coding RNA Classifier based on structural features. *BioData Mining*, 10(1):27, August 2017.

[25] Christian Borgelt, Thorsten Meinl, and Michael Berthold. MoSS : A Program for Molecular Substructure Mininig. *First publ. in: OSDM 2005: proceedings of the First International Workshop on Open Source Data Mining / Bart Goethals ... New York : ACM Press, 2005, pp. 6-15*, August 2005.

[26] Tuvshinbayar Chantsalnyam, Dae Yeong Lim, Hilal Tayara, et al. ncRDeep: Non-coding RNA classification with convolutional neural network. *Computational Biology and Chemistry*, 88, October 2020.

[27] Teresa Maria Rosaria Noviello, Francesco Ceccarelli, Michele Ceccarelli, et al. Deep learning predicts short non-coding RNA functions from only raw sequence data. *PLoS Computational Biology*, 16(11):e1008415, November 2020.

[28] Muhammad Nabeel Asim, Muhammad Imran Malik, Christoph Zehe, et al. A Robust and Precise ConvNet for Small Non-Coding RNA Classification (RPC-snRC). *IEEE Access*, 9:19379–19390, November 2021.

[29] Tuvshinbayar Chantsalnyam, Arslan Siraj, Hilal Tayara, et al. ncRDense: A novel computational approach for classification of non-coding RNA family by deep learning. *Genomics*, 113(5):3030–3038, July 2021.

[30] Abdelbasset Boukelia, Anouar Boucheham, Meriem Belguidoum, et al. A Novel Integrative Approach for Non-coding RNA Classification Based on Deep Learning. *Current Bioinformatics*, 15(4):338–348, June 2020.

[31] Linyu Wang, Shaoge Zheng, Hao Zhang, et al. ncRFP: A novel end-to-end method for non-coding RNAs family prediction based on Deep Learning. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pages 1–1, March 2020.

[32] Diego de S. Lima, Luiz J. A. Amichi, Maria A. Fernandez, et al. NCYPred: A Bidirectional LSTM Network With Attention for Y RNA and Short Non-Coding RNA Classification. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 20(1):557–565, January 2023.

[33] Linyu Wang, Xiao-dan Zhong, Shuo Wang, et al. ncDLRES: A novel method for non-coding RNAs family prediction based on dynamic LSTM and ResNet. *BMC Bioinformatics*, 22(1):447, September 2021.

[34] Kai Chen, Xiaodong Zhu, Jiahao Wang, et al. ncDENSE: A novel computational method based on a deep learning framework for non-coding RNAs family prediction. *BMC Bioinformatics*, 24(1):68, February 2023.

[35] Kai Chen, Xiaodong Zhu, Jiahao Wang, et al. MFPred: Prediction of ncRNA families based on multi-feature fusion. *Briefings in Bioinformatics*, page bbad303, August 2023.

[36] Emanuele Rossi, Federico Monti, Michael Bronstein, et al. NcRNA classification with graph convolutional networks. *arXiv*, pages 17–21, May 2019.

[37] Kevin Sutanto and Marcel Turcotte. Assessing the Use of Secondary Structure Fingerprints and Deep Learning to Classify RNA Sequences. In *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 42–49. IEEE Computer Society, January 2021.

[38] RNAcentral Consortium. RNAcentral 2021: Secondary structure integration, improved sequence search and new member databases. *Nucleic Acids Research*, 49(D1):D212–D220, January 2021.

[39] Ronny Lorenz, Stephan H Bernhart, Christian Höner zu Siederdissen, et al. ViennaRNA Package 2.0. *Algorithms for Molecular Biology*, 6(1):26, December 2011.

[40] Kengo Sato, Yuki Kato, Michiaki Hamada, et al. IPknot: Fast and accurate prediction of RNA secondary structures with pseudoknots using integer programming. *Bioinformatics*, 27(13):i85–i93, July 2011.

[41] Eric P. Nawrocki, Sarah W. Burge, Alex Bateman, et al. Rfam 12.0: Updates to the RNA families database. *Nucleic Acids Research*, 43(Database issue):D130–D137, January 2015.

[42] Weizhong Li and Adam Godzik. Cd-hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22(13):1658–1659, July 2006.

[43] Ioanna Kalvari, Joanna Argasinska, Natalia Quinones-Olvera, et al. Rfam 13.0: Shifting to a genome-centric resource for non-coding RNA families. *Nucleic Acids Research*, 46(Database issue):D335–D342, January 2018.

[44] Felipe Albrecht, Markus List, Christoph Bock, et al. DeepBlue epigenomic data server: Programmatic data retrieval and analysis of epigenome region sets. *Nucleic Acids Research*, 44(Web Server issue):W581–W586, July 2016.

[45] Yan Zhang, Jie Lv, Hongbo Liu, et al. HHMD: The human histone modification database. *Nucleic Acids Research*, 38(Database issue):D149–154, January 2010.

[46] Ioanna Kalvari, Eric P Nawrocki, Nancy Ontiveros-Palacios, et al. Rfam 14: Expanded coverage of metagenomic, viral and microRNA families. *Nucleic Acids Research*, 49(D1):D192–D200, January 2021.

[47] Stefanie A. Mortimer, Mary Anne Kidwell, and Jennifer A. Doudna. Insights into RNA structure and function from genome-wide studies. *Nature Reviews Genetics*, 15(7):469–479, July 2014.

[48] Xiaohan Xing, Fan Yang, Hang Li, Jun Zhang, Yu Zhao, Mingxuan Gao, Junzhou Huang, and Jianhua Yao. An Interpretable Multi-Level Enhanced Graph Attention Network for Disease Diagnosis with Gene Expression Data. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 556–561, December 2021.

[49] Haitham A. Elmarakeby, Justin Hwang, Rand Arafeh, Jett Crowdis, Sydney Gang, David Liu, Saud H. AlDubayan, Keyan Salari, Steven Kregel, Camden Richter, Taylor E. Arnoff, Jihye Park, William C. Hahn, and Eliezer M. Van Allen. Biologically informed deep neural network for prostate cancer discovery. *Nature*, 598(7880):348–352, October 2021.