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

Constance Creux[1,2], Farida Zehraoui[1], François Radvanyi[2] and Fariza Tahi[1,*]

[1] Université Paris-Saclay, Univ Evry, IBISC, 91020, Evry-Courcouronnes, France
[2] Molecular Oncology, PSL Research University, CNRS, UMR 144, Institut Curie, Paris, France
* Corresponding author: fariza.tahi@univ-evry.fr

**ABSTRACT**

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 application to non-coding RNA classification. Multiple novel architectures have been developed, but these advancements are not covered by current literature reviews. We present an exhaustive comparison of the different methods and datasets proposed in the state-of-the-art. Moreover, the literature lacks objective benchmarks. We perform experiments to fairly evaluate the performance of various tools for non-coding RNA classification on popular datasets. We also measure computation time and $CO_2$ emissions. With regard to these results, we assess the relevance of the different architectural choices and provide recommendations to consider in future methods. Datasets and reproducible codes are available at `https://evryrna.ibisc.univ-evry.fr/evryrna/ncBench`.

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

## 1 Introduction

The definition of non-coding RNA (ncRNA) classes of 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 still mainly focused on protein-coding genes. It is in the 2000s, when efforts like the Human Genome Project [2] highlighted that 98% of the genome is non-coding, that attention started to turn to ncRNA. Novel classes of ncRNAs 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. In order to handle large amounts of data, multiple computational ncRNA classifiers have been developed over the years.

There are many classes of ncRNAs, which share characteristics and 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 22 nt 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 subdivided. LncRNAs particularly suffer from this lack of definition. RNAs belonging to this class are disparate and can have multiple functions. They are mostly defined by their length over 200 nt and their position in the genome (antisense, intronic...). Due to this lack of precision, ncRNA classifiers tend to focus on classes of small ncRNAs (sncRNAs). We represent widely accepted classes in Figure 1.

The term 'ncRNA classification' can have different meanings in the literature. It is sometimes used to refer to the prediction of specific ncRNA classes [6, 7, 8, 9, 10, 11, 12, 13, 14, 15]. It can also refer to multi-class classification, taking as input a set of ncRNAs and associating each one to a ncRNA class, which is the scope of this review.

The prediction of specific classes has been widely covered in reviews [16, 17, 18, 19]. However, the literature lacks comparisons of current multi-class ncRNA classifiers. In an article published in 2017 [20], earlier methods for ncRNA classification are presented. At the time, approaches relied mainly on alignment algorithms and standard machine learning. Methods have now evolved: deep learning (DL) has demonstrated increased performance in various application domains, including ncRNA classification where DL is now the prevalent approach. Only one work, published in

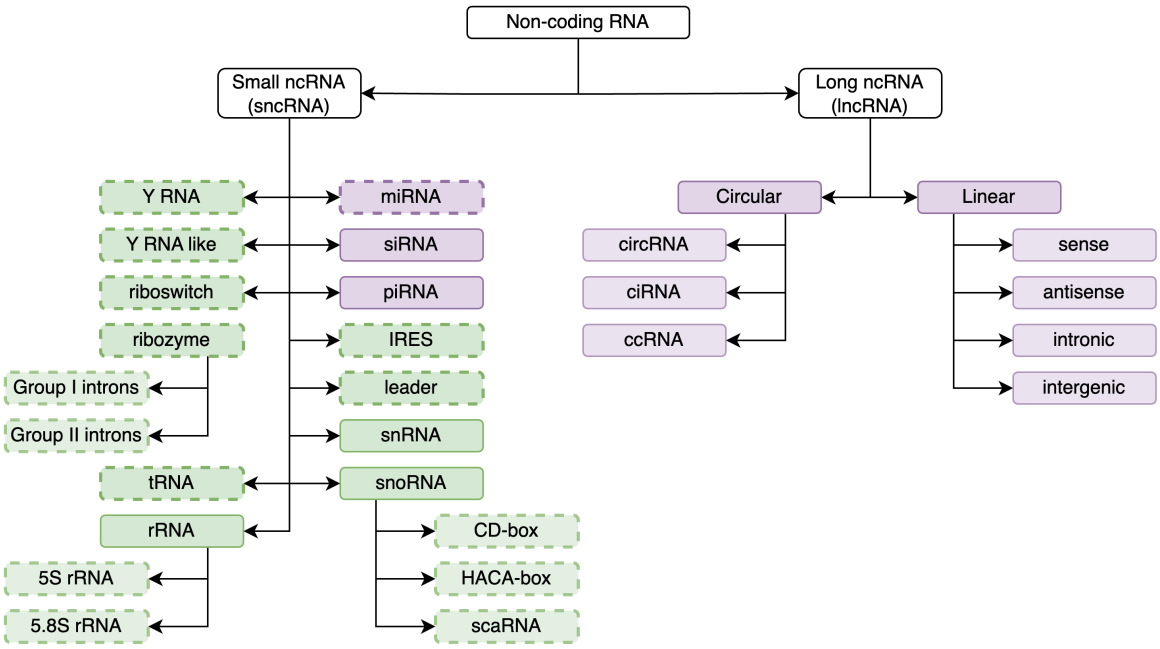


Figure 1: **Taxonomy of ncRNA classes.** Widely accepted classes of small ncRNAs and long ncRNAs are represented. Purple boxes are associated with regulatory ncRNAs, and green boxes with housekeeping RNAs. Classes of interest in current ncRNA classification datasets are in boxes with a dashed outline.

2019 [17], covers the use of DL for multi-class ncRNA classification. We count twelve DL-based ncRNA classifiers in the state-of-the-art; eleven of them were presented after its publication and have not been covered in other literature.

This lack of reviews is an issue in the field, as the performance comparisons proposed in publications accompanying the different tools are unreliable. Indeed, comparisons between methods are often made without re-executing tools and instead simply using the scores announced in previous publications. In these instances, results are not always obtained with comparable experimental protocols. Moreover, the main dataset used for evaluation presents a data leakage problem; thus, reported performance is biased.

We propose to fill a gap in the literature by presenting an exhaustive review of the different DL approaches for multi-class ncRNA classification. We describe the choices in DL architecture, and explain the differences between datasets used for evaluation. Moreover, we conduct a fair comparison of performance between current tools. We evaluate global and per-class prediction results, and measure computation time as well as $CO_2$ emissions.

This paper is organized as follows: the first section presents the state-of-the-art of DL for ncRNA classification, exhaustively describing current classifiers and the accompanying datasets. In the next section we present the results of our comparison of different tools in terms of global and per-class performance, as well as computation time and $CO_2$ emissions. The section that follows discusses these observations. The Methods section presents the details of our experiments.

## 2 Deep learning for ncRNA classification

### 2.1 State-of-the-art classifiers

Earlier methods for ncRNA classification were primarily based on alignment algorithms [21, 22, 23, 24] or standard machine learning algorithms [25, 26, 27]. Feature extraction in these methods is a particularly crucial step, with a high impact on performance. Current ncRNA classifiers are all based on deep learning: most methods are based on Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), or a combination or both.

Applications of DL to ncRNA classification started with multiple CNN-based methods: nRC [28], ncRDeep [29], ncrna-deep[1] [30], RPC-snRC [31], ncRDense [32] and imCnC [33]. nRC pioneered the use of DL in the field. As input, ncRNAs are represented by vectors containing information on the presence in their secondary structure of discriminative sub-structures. Two convolutional layers extract relevant information, and Fully-Connected Layers (FCLs) are used for classification. ncRDeep 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 ncrna-deep, the number of convolutional layers rises to five. Authors tested multiple sequence encodings (space-filling curves or one-hot encoding of k-mers[2] of length 1, 2 or 3). 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, and fed to three consecutive Dense blocks (based on DenseNets, multiple convolutional layers with dense connections), with a final FCL for classification. ncRDense 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 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.

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 [34] and NCYPred [35]. ncRFP uses a bidirectional Long Short-Term Memory RNN (bi-LSTM). 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 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, and it is these 3-mers that are encoded.

Several methods combine both approaches, first using RNNs for sequence representation, from which CNNs then extract relevant information. This is the case for ncDLRES [36], ncDENSE [37] and MFPred [38]. ncDLRES 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 Residual Neural Network (ResNet), which is composed of multiple convolutional layers with skip connections. ncDENSE further evolves that architecture, replacing the dynamic LSTM by a dynamic bidirectional Gated Recurrent Unit (bi-GRU), and the ResNet by a DenseNet. MFPred proposes a rich input representation: 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 is replaced by a Squeeze and Excitation ResNet (SE ResNet), which models interdependencies between the four matrices of learned sequence encodings.

Two existing methods do not fall into the above categorization: RNAGCN [39] and Graph+DL [40], both based on graphs. Taking the secondary structure as input, RNAGCN uses a Graph Neural Network (GNN) 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 Graph+DL[3] is based on features extracted from the secondary structure. Authors propose a protocol to obtain a rich graph-theory-based representation of structures. Other than the extensive feature preparation step, the method is based on a simple DL network of five FCLs.

We present in Figure 2 a chronological view of the methods for ncRNA classification described above. As we can see in the figure, most methods are based on CNNs, RNNs, or a combination of the two. Moreover, approaches can be separated based on the type of input data they require. A majority of DL ncRNA classifiers use the sequence as input, three use the secondary structure, and two use the sequence in combination with the structure or with epigenetic data.

### 2.2 State-of-the-art datasets

Along with the tools from the previous section, multiple datasets have been proposed for evaluation of DL ncRNA classifiers in recent publications. We identify five.

The first dataset was proposed in 2017 by Fiannaca et al. [28]. Sequences were downloaded from Rfam 12 [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, this dataset has

[1]This method is unnamed, we refer to it using the name of its GitHub repository.

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

[3]This method is unnamed, we refer to it by naming its two components.

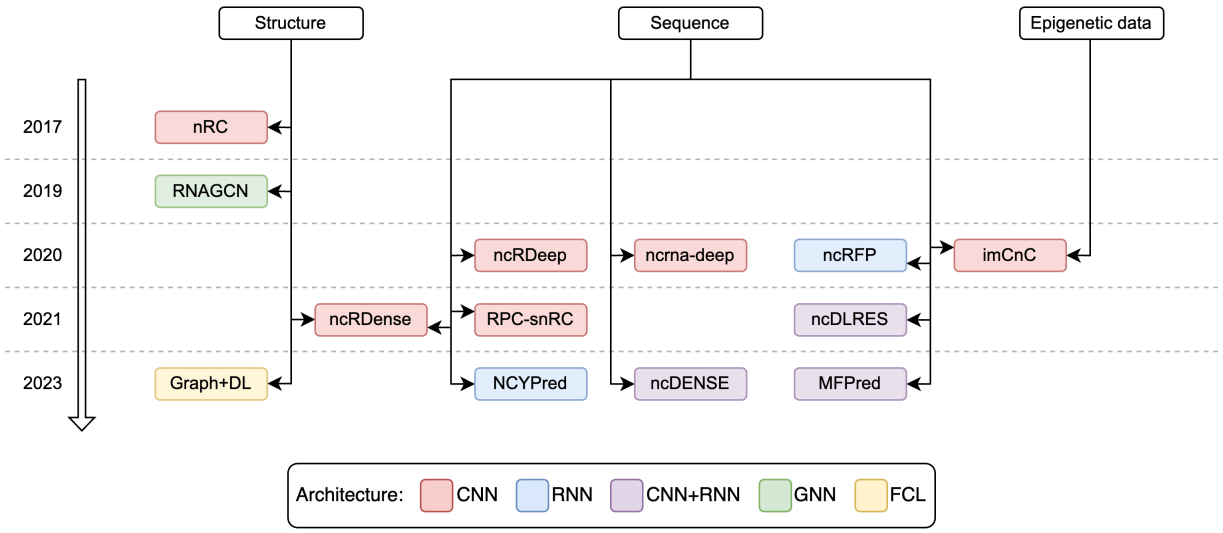


Figure 2: **Overview of existing approaches for ncRNA classification.** Methods are represented in chronological order. We specify the type of input data used (secondary structure, sequence, or epigenetic data) and the type of DL architecture (CNN, RNN, both, or other).

been used by all state-of-the-art methods. However, it presents a data leakage problem: 347 ncRNAs are present in both the training and the test set (representing respectively 5.5% and 13.3% of the sets).

Boukelia et al. [33] constructed a second dataset in 2020. This dataset contains 42,000 sequences of human ncRNAs downloaded from Rfam 13 [43]. It also includes four types of epigenetic modifications (from DeepBlue [44] and HHMD [45] databases). In total, 16 sncRNA classes are represented.

The same year, Noviello et al. [30] 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. A total of 306,016 sequences are present in the dataset.

In 2023, a dataset was proposed by Lima et al. [35]. Employing a similar protocol as Fiannaca et al. [28] but with updated data from Rfam 14.3 [46], authors were able to multiply by five the number of ncRNAs in the dataset, obtaining 45,447 ncRNAs. 13 ncRNA classes are represented.

Sutanto et al. [40] also proposed a dataset in 2023. They obtained sequences from Rfam 14.1 [46] and removed those that could not be used by their structure prediction tool of choice. Then, CD-HIT [42] was used to remove redundancy in the dataset. In total, the dataset comprises 59,723 RNAs representing 12 classes.

The comparison between state-of-the-art datasets can be summed up to the origin of data, diversity of sequences, and classes represented:

- As one of the largest databases on ncRNAs, all datasets are obtained from Rfam, with recent releases containing more sequences. Boukelia et al.'s dataset [33] also integrates other databases for epigenetic data.
- In three cases ([28, 40, 35]), CD-HIT is used to remove sequences with similarity over 80%. This step restricts the size of the dataset, but only removes examples that would have been redundant while maintaining diversity. In some datasets, sequences containing degenerate nucleotides are removed ([30, 35]).
- Most datasets represent around 13 commonly-used sncRNA classes, grouping ncRNAs that share a function while not being too specific. Only Noviello et al.'s dataset [30] differs, including many more classes.

## 3 Results

In this section, we compare the performance of different available state-of-the-art ncRNA classifiers: nRC [28], RNAGCN [39], ncrna-deep [30], MFPred [38], ncRDense [32], and NCYPred [35]. For this purpose, three datasets are used: Dataset1, Dataset1-nd and Dataset2. Dataset1 is the one by Fiannaca et al. [28], in which we fix the data

leakage issue. Dataset1-nd is the same dataset in which we remove sequences containing degenerate nucleotides. Dataset2 is the one presented by Lima et al. [35] (see Supplementary Materials Table 1). The selection of tools and datasets for this benchmark, as well as the experimental protocol, are detailed in the Methods section.

### 3.1 Overall performance

We assess in Figure 3 the prediction performance of state-of-the-art ncRNA classifiers on Dataset1, Dataset1-nd and Dataset2, using the following multi-class metrics: Accuracy, MCC, F1-score, Precision, Recall, and Specificity. Results are, in general, better on Dataset2. This can be explained by the fact that Dataset2 is much larger than Dataset1(-nd). Models are trained on a larger variety of ncRNAs, making the prediction of new examples easier. Also of note, since Dataset1-nd is very similar to Dataset1 (it is only slightly smaller due to the removal of sequences with degenerate nucleotides), it is normal that obtained scores are close. nRC, RNAGCN, ncrna-deep and MFPred can be retrained, while ncRDense and NCYPred can only be used for prediction. This difference is visible in the results: the latter tools only perform well on the dataset they were originally trained on (Dataset1 for ncRDense and Dataset2 for NCYPred). This is particularly visible looking at Precision, Recall and F1-score. Opposingly, tools that were retrained have consistent performances across datasets. We notice that ncrna-deep reaches the highest scores across the different metrics and for the three datasets. MFPred and RNAGCN also obtain high scores on Dataset2, surely benefiting from the increased amount of data.

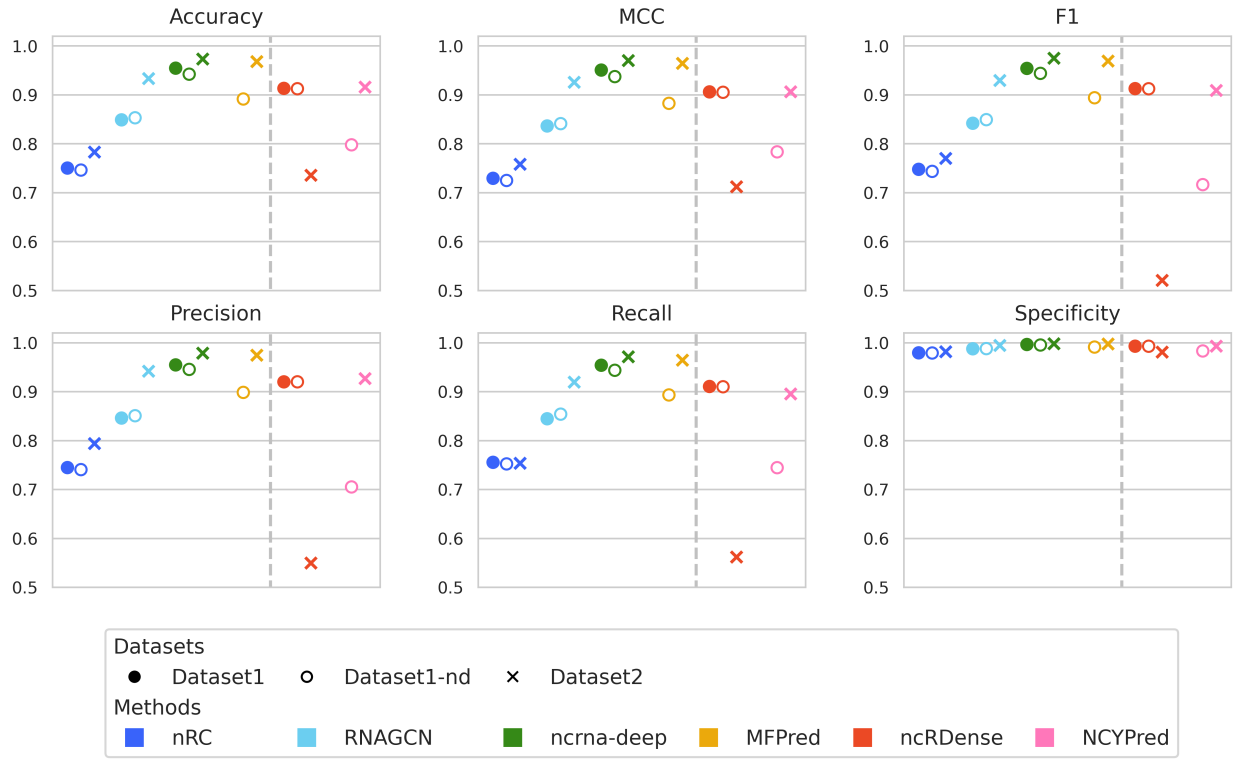


Figure 3: **Comparison of overall performance of the state-of-the-art methods on the test sets of Dataset1, Dataset1-nd and Dataset2.** The shape of the markers designates the datasets, and each color corresponds to a method. Tools on the right of the dashed line, ncRDense and NCYPred, could not be retrained before prediction. No results are shown for MFPred and NCYPred on Dataset1 since these two tools do not process degenerate nucleotides. The tools are sorted chronologically on each of the two sides separated by the dashed line.

We compare the performance we measured for each tool to the performance reported in its original publication, and illustrate the difference in Figure 4. All publications use the same formula for MCC, therefore, we base our comparison on this metric. nRC, RNAGCN, ncrna-deep, and ncRDense all use Dataset1, and MFPred and NCYPred use Dataset1-nd (albeit the biased versions with data leakage). Results in our experiments were significantly worse than what was originally reported by the various publications. These drops in performance are partially explained by the correction of the data leakage problem. For NCYPred, our lower measured performance could also be due to model training. In their evaluations, authors retrained their tool on Dataset1-nd, which we could not do, only having access to the web

server version. This means that two classes from Dataset1-nd are unknown by the model and cannot be predicted. Only two methods, MFPred and NCYPred, reported results on Dataset2 as it is very recent. These reported results seem more reliable as we only observe negligible differences in MCC.

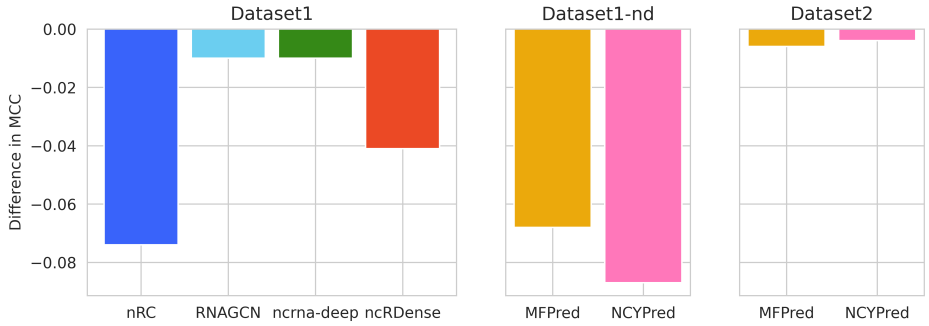


Figure 4: **Difference between MCC values measured in our benchmark and those reported by each state-of-the-art method.** A negative value of $-x$ signifies that the value we measured is lower by $x$ than the one originally reported.

### 3.2 Performance on individual classes

A great challenge in ncRNA classification is achieving high predictions across classes. In Figure 5, we present the accuracy of predictions for each class. Here, another explanation for the better results on Dataset2 than Dataset1(-nd) that were noticed in the previous section is presented: the classes included in each dataset differ, which can impact performance. The two classes included solely in Dataset1(-nd) (IRES and scaRNA) seem more difficult to predict, thus lowering overall scores. We observe once again that highest performance is obtained by ncrna-deep, seconded by MFPred. The difference between classes is striking for nRC, which is able to predict some classes quite well (Intron-gpII, 5S-rRNA), but performs poorly on other classes (miRNA, HACA-box). Aside from the classes it was not trained on (IRES and scaRNA), NCYPred obtains similar performances to other tools on Dataset1-nd. 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, underscoring the fact that not allowing to retrain a tool can be detrimental. It is interesting that, although recent tools make generally very accurate predictions, the more problematic classes are the same as those for earlier methods, such as miRNA or HACA-box. Wrongly classified miRNAs tend to be predicted as CD-box (see Supplementary Materials Figures 2-4): there is literature explaining possible similarities between the two classes [47, 48]. HACA-box tends to be mispredicted as CD-box or scaRNA. These are subclasses of snoRNAs, which can explain the confusion.

### 3.3 Computation time and $CO_2$ emissions

We compare the computation time and $CO_2$ emissions of benchmark methods, which we measured when executing the codes. RNAGCN, ncrna-deep and MFPred were executed on the same GPU machine (NVIDIA GeForce RTX 3090). nRC only supports CPU computation and was executed on Intel Xeon(R) W-1250 CPU @ 3.30GHz. $CO_2$ emission estimations were conducted with CodeCarbon [49, 50]. We can use it for RNAGCN, ncrna-deep and MFPred, as source codes are all in Python, but not for nRC which is written in a mix of Java and Python.

Table 1 shows the computation time as well as the $CO_2$ emissions of the state-of-art methods on Dataset2 (the largest). 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. For MFPred, the lack of speed is due to model complexity: the model is large and contains considerably more layers than the other methods. The $CO_2$ emissions of these methods seem to be proportional to 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, etc. Moreover, ncRDense cannot predict more than 1,000 ncRNAs at once, further complicating the measure of prediction time. As an order of magnitude, for the same 1,000 ncRNAs, we obtained predictions in around 20 seconds with NCYPred and 30 seconds with ncRDense. No preprocessing is required. $CO_2$ emissions cannot be measured.

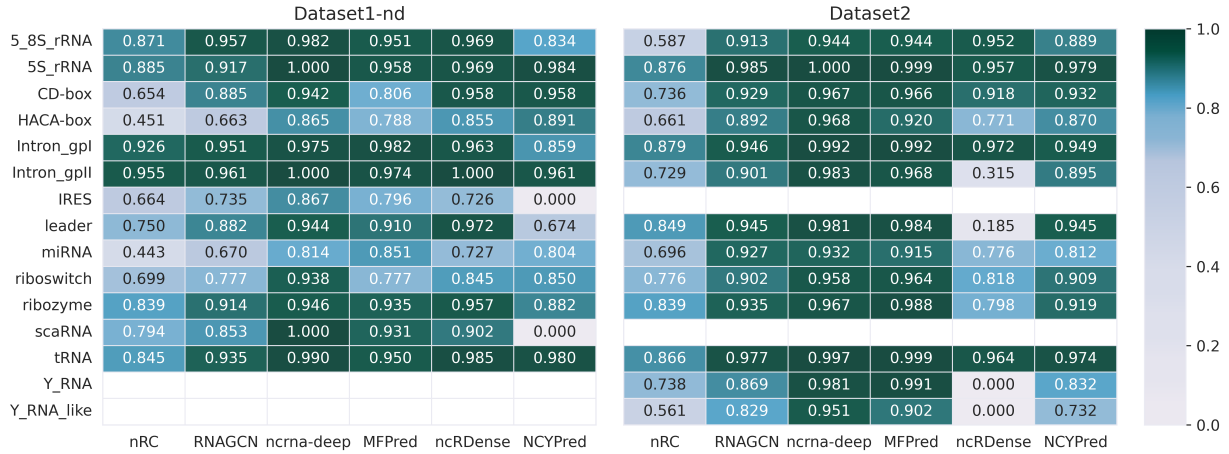


Figure 5: **Comparison of accuracy of prediction of each ncRNA class obtained by state-of-the-art tools on Dataset1-nd and Dataset2.** Light colors correspond to lower accuracies, while colors tending towards dark green represent the best results. Results for Dataset1 are comparable to those for Dataset1-nd and are presented in Supplementary Materials Figure 1.

| | **Computation time** | | | **Emissions** |
|---|---|---|---|---|
| | **Preprocessing** | **Training** | **Prediction** | (in $gCO_2eq$) |
| **nRC** | 3h 06mn | 9h 04mn | 57s | - |
| **RNAGCN** | 13mn 22s | 1h 05mn | 2.5s | 17.8 |
| **ncrna-deep** | 10.8s | 4mn 11s | 2.2s | 1.7 |
| **MFPred** | 2mn 53 | 9h 18mn | 1mn 3s | 233.8 |

Table 1: **Comparison of computation times and $CO_2$ emissions on Dataset2.** The computation time is calculated for preprocessing, training and prediction, while the $CO_2$ emission is calculated for training. Times and emissions are computed for the hyperparameter sets selected in Table 4. (See Supplementary Materials Table 2 for Dataset1 and Dataset1-nd).

## 4 Discussion

From the different results presented in the previous section, the current recommendation for users looking to classify ncRNAs would be to use the ncrna-deep tool, as it obtains the best results on benchmark datasets while being the fastest. MFPred, although more recent than ncrna-deep, performs slightly less well than the latter, and its execution time is longer. Moreover, it cannot be used on sequences containing degenerate nucleotides. This goes against the comparisons made in MFPred's publication, and emphasizes the necessity of unbiased benchmarks such as the one presented here, and the importance of employing comparable experimental protocols. nRC pioneered the use of DL for ncRNA classification and was the tool of reference for years, but is now outperformed by more recent tools. RNAGCN, also obtaining lower performances, brought to the field a different point of view as its GNN-based architecture stands out from the rest of the state-of-the-art. If the goal is to classify new ncRNA datasets, we would advise against using web servers that do not provide the possibility to perform training, as it impacts performance negatively.

Several recommendations can be formulated with regard to the development of future ncRNA classifiers:

- The first choice to make in the architecture is how to represent data. nRC and RNAGCN choose to use the secondary structure. This is based on the widely accepted idea that the function of ncRNAs is linked to their structure [51], an approach adopted by many earlier ncRNA classifiers [25, 27]. It is interesting that this experimental observation does not translate particularly well to DL approaches, as methods that use the structure do not give the best results in our benchmark. This can be explained by the fact that secondary structure prediction is a difficult task, and while algorithms have improved over the years, their accuracy is still limited. Using newer, more accurate structure prediction algorithms could be a way to improve results in the future.
- 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 low for ncrna-deep, high for MFPred, but ncrna-deep performs better. The results obtained in our benchmark suggest that the simple representation should in fact be preferred, as the time gain is considerable without any effect on performance.

- 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. For example, earlier ncRNA classifiers ([52, 26, 23, 53]) used RNA-seq data. The inclusion of epigenetic data by imCnC [33] 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 research, in which computational ncRNA classifiers could be extremely useful.
- Finally, there is an additional consideration regarding the availability of tools. In the description of state-of-the-art datasets, we noted an increase in dataset size and variability in the classes included. In this context, it is important to propose tools that can be applied to and evaluated on new datasets. The best practice would be to propose both a web server and to share source code. The web server would provide ease of use. By offering the possibility of retraining the model, it would not be limited to one dataset. Moreover, a well-documented source code could be used in bigger projects and with more flexibility.

# 5 Methods

## 5.1 Literature search

The scope of our study is multi-class non-coding RNA classifiers that are based on deep learning. To make sure we included all relevant literature, we performed the following search from Digital Science's Dimensions platform (`https://app.dimensions.ai`): the terms *(ncRNA OR non-coding RNA) AND (deep learning) AND (classification OR classes OR families)* had to be present in the title and/or abstract. On February 14th, 2024, this search yielded 567 results. From these, a lot of results were not directly linked to our problem statement (e.g. a lot of publications were about lncRNA-disease association prediction, or cancer patient classification). We carefully checked that no relevant title had been forgotten from our study.

## 5.2 Tool selection

We present in Table 2 information on the accessibility of all DL-based ncRNA classifiers. Most publications propose an online version of their tool, as a web server or by sharing source code. Web servers have the advantage of requiring no installation and being easy to use. However, they are sometimes limited: for example, here, no web server can be retrained on different datasets. If source code is available, the model can be trained on different datasets, and computational power can be increased. Nonetheless, executing the code is sometimes complicated, especially if documentation is insufficient.

We choose to assess the quality of prediction of six methods: nRC [28], RNAGCN [39], ncrna-deep [30], ncRDense [32], NCYPred [35] and MFPred [38]. Graph+DL [40] and RPC-snRC [31] 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 [33], 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 [29] and ncRDense [32], and on the other hand ncRFP [34], ncDLRES [36], ncDENSE [37] and MFPred [38]. In both cases, we use the latest version: ncRDense and MFPred.

## 5.3 Dataset selection

Out of the five state-of-the-art datasets described previously, three have been made available publicly: Fiannaca et al.'s dataset [28], Noviello et al.'s dataset [30], and Lima et al.'s dataset [35]. We present in Table 3 information regarding the size of these datasets, as well as the number of times they were used in other publications.

We decide to focus on Fiannaca et al.'s dataset and Lima et al.'s dataset. Noviello et al. present their dataset with the goal of predicting the function of ncRNAs. As such, the classes represented are numerous, and can describe very precise functions. We consider these classes to be too specific for our presented problem of ncRNA classification into broader, more general functional groups (or classes). In our experiments, we refer to three datasets:

| Method | Usage |
|---|---|
| **nRC** [28] | ***Web server: X – Source code: ✓ – Training: ✓***<br>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** [39] | ***Web server: X – Source code: ✓ – Training: ✓***<br>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** [34] | ***Web server: X – Source code: ✓ – Training: ✓***<br>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** [33] | ***Web server: X – Source code: ✓ – Training: ✓***<br>GitHub repository: `https://github.com/BoukeliaAbdelbasset/imCnC`. No instructions or examples of how to use the tool are given. |
| **ncRDeep** [29] | ***Web server: ✓ – Source code: X – Training: X***<br>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** [30] | ***Web server: X – Source code: ✓ – Training: ✓***<br>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. |
| **ncRDense** [32] | ***Web server: ✓ – Source code: X – Training: X***<br>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** [36] | ***Web server: X – Source code: ✓ – Training: ✓***<br>GitHub repository: `https://github.com/linyuwangPHD/ncDLRES`. No examples of how to use the tool, but very basic instructions are given. |
| **RPC-snRC** [31] | ***Web server: X – Source code: X – Training: X***<br>The GitHub repository linked in the paper no longer exists. |
| **NCYPred** [35] | ***Web server: ✓ – Source code: ✓ – Training: ✓***<br>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** [37] | ***Web server: X – Source code: ✓ – Training: ✓***<br>GitHub repository: `https://github.com/ck-fighting/ncDENSE`. No examples of how to use the tool are given. |
| **MFPred** [38] | ***Web server: X – Source code: ✓ – Training: ✓***<br>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. |
| **Graph+DL** [40] | ***Web server: X – Source code: X – Training: X***<br>Source code is not linked. Data is not available from the proposed link in the publication. |

Table 2: **Description of the availability and ease-of-use of DL ncRNA classification tools.** Tools are presented in chronological order of publication. Each cell indicates if the tool is accessible through a web server, if source code can be downloaded, and if the tool can be retrained. All links have been checked at the time of writing (February 2024).

| Source | Nb Instances | Nb Classes | Lengths | Nb Uses |
|---|---|---|---|---|
| **Fiannaca et al.** [28] | 8,920 | 13 | 38-1182 | 12 |
| **Noviello et al.** [30] | 306,016 | 88 | 1-80 | 1 |
| **Lima et al.** [35] | 45,447 | 13 | 42-500 | 2 |

Table 3: **Description of state-of-the-art available datasets for ncRNA classification.** Datasets are sorted by date. The number of instances and classes in each dataset is given. We also give the range of sequence lengths. The last column indicates how many times the dataset has been used by state-of-the-art ncRNA classifiers.

- Dataset1: the dataset by Fiannaca et al. in which we fix the data leakage issue. Specifically, we remove from the test set the 347 sequences that were also present in the training set. The training and test sets contain respectively 6,320 and 2,253 ncRNAs.
- Dataset1-nd: a version of Dataset1 in which we remove sequences containing degenerate nucleotides, as these cannot be handled by some methods. The training and test sets contain respectively 6,161 and 2,189 ncRNAs.
- Dataset2: the dataset by Lima et al., in which we confirmed there was no data leakage. It does not include any sequence with degenerate nucleotides. The training and test sets contain respectively 31,801 and 13,646 ncRNAs.

### 5.4 Experimental protocol

nRC, RNAGCN, ncrna-deep and MFPred were retrained before predicting each test set. When possible, we tested different hyperparameters, which are presented in Table 4. This could not be done for MFPred as parameter values are fixed directly in the code, and no guidelines are given on how to change them or what to change. ncRDense and NCYPred were only accessible to us as web servers that could not be retrained, 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 layers. |
| **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 4: **Overview of tested hyperparameters.** Parameters that gave the best results on Dataset1 and Dataset2 are denoted by $^{(1)}$ and $^{(2)}$ respectively. The models chosen on Dataset1 were also used for Dataset1-nd. Default values were used for parameters not listed.

Predictions were obtained on the held-out test set of each dataset. MFPred and NCYPred cannot handle degenerate nucleotides, therefore their performance is not measured on Dataset1.

Overall performance is measured with Accuracy, MCC (Matthews Correlation Coefficient), F1-score, Precision, Recall (or Sensitivity) and Specificity. The last four metrics are binary metrics for which 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:

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

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

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

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

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

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

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 $c$.

## 6 Funding

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

## 7 Availability

The datasets presented in this work and the codes to reproduce this benchmark are available on the EvryRNA platform: `https://evryrna.ibisc.univ-evry.fr/evryrna/ncBench`.

## References

[1] Antonin Morillon. Non-coding RNA, Its History and Discovery Timeline. pages 1–24. January 2018.

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

[3] Taewan Kim and Carlo M. Croce. MicroRNA: trends in clinical trials of cancer diagnosis and therapy strategies. *Experimental & Molecular Medicine*, 55(7):1314–1321, July 2023.

[4] Zhixia Zhou, Zhan Wang, Jie Gao, et al. Noncoding RNA-mediated macrophage and cancer cell crosstalk in hepatocellular carcinoma. *Molecular Therapy - Oncolytics*, 25:98–120, June 2022.

[5] Xueman Chen, Rong Luo, Yunmei Zhang, et al. Long noncoding RNA DIO3OS induces glycolytic-dominant metabolic reprogramming to promote aromatase inhibitor resistance in breast cancer. *Nature Communications*, 13(1):7160, November 2022.

[6] J. Brayet, F. Zehraoui, L. Jeanson-Leh, et al. Towards a piRNA prediction using multiple kernel fusion and support vector machine. *Bioinformatics*, 30(17):i364–i370, September 2014.

[7] Van Du T. Tran, Sebastien Tempel, Benjamin Zerath, et al. miRBoost: boosting support vector machines for microRNA precursor classification. *RNA*, 21(5):775–785, May 2015.

[8] Christophe Tav, Sébastien Tempel, Laurent Poligny, et al. miRNAFold: a web server for fast miRNA precursor prediction in genomes. *Nucleic Acids Research*, 44(W1):W181–W184, July 2016.

[9] 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.

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

[11] Yajun Liu, Yulian Ding, Aimin Li, et al. Prediction of exosomal piRNAs based on deep learning for sequence embedding with attention mechanism. In *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 158–161, December 2022.

[12] Minghua Li and Chun Liang. LncDC: a machine learning-based tool for long non-coding RNA detection from RNA-Seq data. *Scientific Reports*, 12(1):19083, November 2022.

[13] Jonathan Raad, Leandro A Bugnon, Diego H Milone, et al. miRe2e: a full end-to-end deep model based on transformers for prediction of pre-miRNAs. *Bioinformatics*, 38(5):1191–1197, March 2022.

[14] Guillaume Postic, Christophe Tav, Ludovic Platon, et al. IRSOM2: a web server for predicting bifunctional RNAs. *Nucleic Acids Research*, 51(W1):W281–W288, July 2023.

[15] Vivek Rajendran, Harishchander Anandaram, Sachin Kumar S, et al. A Comparative Analysis of Machine Learning and Deep Learning Approaches for Circular RNA Classification. In *2023 6th International Conference on Contemporary Computing and Informatics (IC3I)*, volume 6, pages 1026–1034, September 2023.

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

[17] 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.

[18] Dalwinder Singh and Joy Roy. A large-scale benchmark study of tools for the classification of protein-coding and non-coding RNAs. *Nucleic Acids Research*, 50(21):12094–12111, November 2022.

[19] Tea Ammunét, Ning Wang, Sofia Khan, et al. Deep learning tools are top performers in long non-coding RNA prediction. *Briefings in Functional Genomics*, 21(3):230–241, May 2022.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

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

[25] 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.

[26] 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.

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

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] Emanuele Rossi, Federico Monti, Michael Bronstein, et al. NcRNA classification with graph convolutional networks. In *Proceedings of DLG@KDD workshop*, pages 17–21, Anchorage, Alaska, US, May 2019.

[40] Kevin Sutanto and Marcel Turcotte. Assessing Global-Local Secondary Structure Fingerprints to Classify RNA Sequences With Deep Learning. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 20(5):2736–2747, September 2023.

[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] Markus Brameier, Astrid Herwig, Richard Reinhardt, et al. Human box C/D snoRNAs with miRNA like functions: expanding the range of regulatory RNAs. *Nucleic Acids Research*, 39(2):675–686, January 2011.

[48] Motoharu Ono, Michelle S. Scott, Kayo Yamada, et al. Identification of human miRNA precursors that resemble box C/D snoRNAs. *Nucleic Acids Research*, 39(9):3879–3891, May 2011.

[49] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, et al. Quantifying the carbon emissions of machine learning. *Workshop on Tackling Climate Change with Machine Learning at NeurIPS 2019*, 2019.

[50] Kadan Lottick, Silvia Susai, Sorelle A. Friedler, et al. Energy Usage Reports: Environmental awareness as part of algorithmic accountability. *Workshop on Tackling Climate Change with Machine Learning at NeurIPS 2019*, 2019.

[51] 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.

[52] 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.

[53] 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.