Computers in Biology and Medicine 159 (2023) 106861

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

NeRNA: A negative data generation framework for machine learning applications of noncoding RNAs

Mehmet Emin Orhan [a], Yılmaz Mehmet Demirci [b], Müşerref Duygu Saçar Demirci [c,*]

[a] Department of Bioengineering, Graduate School of Engineering and Science, Abdullah Gül University, Kayseri, Turkey
[b] Department of Engineering Science, Faculty of Engineering, Abdullah Gül University, Kayseri, Turkey
[c] Department of Bioengineering, Faculty of Life and Natural Sciences, Abdullah Gül University, Kayseri, Turkey

ARTICLE INFO

Keywords:
RNA
Noncoding RNA
Data generation
Machine learning

ABSTRACT

Many supervised machine learning based noncoding RNA (ncRNA) analysis methods have been developed to classify and identify novel sequences. During such analysis, the positive learning datasets usually consist of known examples of ncRNAs and some of them might even have weak or strong experimental validation. On the contrary, there are neither databases listing the confirmed negative sequences for a specific ncRNA class nor standardized methodologies developed to generate high quality negative examples. To overcome this challenge, a novel negative data generation method, NeRNA (negative RNA), is developed in this work. NeRNA uses known examples of given ncRNA sequences and their calculated structures for octal representation to create negative sequences in a manner similar to frameshift mutations but without deletion or insertion. NeRNA is tested individually with four different ncRNA datasets including microRNA (miRNA), transfer RNA (tRNA), long noncoding RNA (lncRNA), and circular RNA (circRNA). Furthermore, a species-specific case analysis is performed to demonstrate and compare the performance of NeRNA for miRNA prediction. The results of 1000 fold cross-validation on Decision Tree, Naïve Bayes and Random Forest classifiers, and deep learning algorithms such as Multilayer Perceptron, Convolutional Neural Network, and Simple feedforward Neural Networks indicate that models obtained by using NeRNA generated datasets, achieves substantially high prediction performance. NeRNA is released as an easy-to-use, updatable and modifiable KNIME workflow that can be downloaded with example datasets and required extensions. In particular, NeRNA is designed to be a powerful tool for RNA sequence data analysis.

1. Introduction

Based on the central dogma of molecular biology, genetic information stored in DNA is transcribed to messenger RNA (mRNA) which would be translated into a protein [1]. In addition to this well-known function of mRNAs, in a living cell, various RNA molecules can be found performing a wide range of actions. Apart from the viruses that have RNA as their genomes and mRNAs, the rest of the RNA molecules identified so far could be labelled as noncoding RNAs (ncRNAs). In the past decades, various ncRNAs have been extensively studied including small ncRNAs, such as transfer RNAs (tRNAs) taking role in translation by carrying amino acids to the growing peptide chain, small nucleolar RNAs (snoRNAs) responsible for RNA modifications, small nuclear RNAs (snRNAs) involved in RNA splicing, and large ones, such as ribosomal RNAs (rRNAs) while relatively new type of RNAs, regulatory ncRNAs have been a popular research area for not only experimental but also computational studies. Categorization of these ncRNAs have also been based on their size; long non-coding RNAs (lncRNAs), which are longer than 200 nt, and small ncRNAs, which are usually shorter than 200 nt [2]. These ncRNAs can be further categorized into various groups, including microRNAs (miRNAs) involved in post-transcriptional gene regulation, circular RNAs (circRNAs) that are generated by a unique splicing reaction known as back-splicing, PIWI-associated small RNAs (piRNAs), and endogenous siRNAs (endo-siRNAs) [3,4]. Mammalian genomes encode thousands of ncRNAs; known examples of human miRNA precursor sequences are around 2000 [5] and according to a recent report, there are ~20,000 annotated lncRNA genes which is quite close to the total number of protein-coding genes in the human genome [6]. Contemplating the impact of ncRNAs in cellular processes and taking into account challenges in experimental identification of all

* Corresponding author.
E-mail address: duygu.sacar@agu.edu.tr (M.D. Saçar Demirci).

https://doi.org/10.1016/j.compbiomed.2023.106861
Received 22 November 2022; Received in revised form 3 February 2023; Accepted 30 March 2023
Available online 11 April 2023

ncRNAs, building reliable computational approaches for ncRNA detection is an essential step for further analysis [7].

Due to cost and laborious steps of experimental detection, computational prediction has become an essential part of ncRNA studies. Among many methods, Machine Learning (ML) based algorithms have been frequently applied. While there are cases where unsupervised approaches are used [8], ML for ncRNA prediction is almost exclusively based on supervised learning in which a classification algorithm is trained for learning from known examples [9].

Since RNA is a self-folding molecule, bases in the RNA sequences tend to form pairs through hydrogen bonds. This folding leads to secondary structure of the RNA which is essential for three-dimensional (3D) structure and function [10]. In general, ML methods begin with obtaining datasets of the RNA sequence and structure to calculate features defining characteristic properties of the RNA class of interest. Then, a classifier is trained to generate rules based on the examples (input data; positive (non-coding RNAs) and negative (non-noncoding examples)). In the last step of the analysis, the model generated by the classifier will be used on unknown samples to label them accordingly.

The overall ML system can be divided into five main modules [11]: 1) Training and testing data, 2) Feature selection, 3) Machine learning algorithm, 4) Training scheme establishment and 5) parameter tuning. Based on the principle of 'garbage in garbage out' in ML [12] and results of early works about impact of ML elements on the overall performance, obtaining high quality datasets seems the most crucial step among the tasks that lead to model establishment. For 2-class classification of ncRNAs, experimentally validated RNA sequences could be used as the positive class. However, there are no standard methodologies to generate high quality negative examples. To address these issues, a novel negative data generation method, NeRNA (negative RNA), is developed. NeRNA is tested with four different ncRNA datasets from different RNA types. Furthermore, a species-specific case analysis is performed to demonstrate performance of NeRNA for miRNA prediction.

NeRNA is released as a KNIME (Konstanz Information Miner) [13] workflow which can be downloaded with example datasets and required extensions.

2. Methods

NeRNA generates datasets based on the RNA sequences, their secondary structures, and chemical properties. This section describes the workflow.

3. Data

We adopted four ncRNAs datasets in this study: (1) Dataset I consisting of known *Homo sapiens, Mus musculus, Bos taurus, Gallus gallus, Oreochromis niloticus, Equus caballus, Glycine max, Monodelphis domestica, Medicago truncatula and Pan troglodytes* miRNA hairpin sequences from miRBase (v22.1) (sample size 650 and bigger species collected from) [14]; (2) Dataset II including tRNA sequences and structures of 101 different organisms from Psi-C Database (v1.0) [15]; (3) Dataset III containing 1000 randomly selected lncRNA sequences from LNCipedia (v5.2) [16], and (4) Dataset IV having 1000 randomly selected mouse circular RNA sequences from circBase (v0.1) [17] (Table 1).

4. Architecture of NeRNA algorithm

Negative RNA generation framework was developed in the KNIME workflow management system. The framework consists of a fasta file reader, secondary structure generator, NeRNA algorithm, and fasta file writer functions. Inside the KNIME framework, R scripts (R version v4.1.0) were created to perform specific functions by using the seqinr package (v4.2.8) [18] and stringr library (v1.4.0).

The majority of the RNA sequences include four bases (A, G, C, and U) that can form base pairs such as A–U, G–C, and G–U. These pairs are represented in secondary structures using parenthesis and point symbols; for that purpose, RNAfold software from the ViennaRNA package (v2.5.1) was used to generate secondary structures [19]. RNAfold is applied with the –noPS option (not drawing the mfe (minimum free energy) structure in postscript.) to datasets I, II, III and IV while circular structure generation option (–circ) is also used for dataset IV. For better representation of the secondary structures, the nucleotides involved in base pairs are shown as A, G, C, and U in Table 2, while non-base paired ones are shown as A′, G′, C′, and U′, respectively.

NeRNA architecture is constructed based on the mathematical representation of biological principles of RNA folding. An overview of the algorithm for generation of negative sequences is shown in Algorithm 1.

Algorithm 1. Generate Negative Sequences

The algorithm starts with an RNA sequence of length n. Next, secondary structure for this sequence is calculated with RNAfold. At the

Table 2
Map from base elements to octal representation of the sequence.

Base	Octal Representation
A	000
G	001
C	010
U	100
A'	011
G'	110
C'	101
U'	111

Table 1
Datasets used for testing NeRNA. Number indicates number of sequences in each dataset. * list of 101 different organisms for tRNA dataset is available in Supplementary File 1, S1.

Datasets	RNA type	Organisms	Number	Sequence Length			Source
				Min	Max	Average	
I	miRNA hairpins	*Homo sapiens*	1917	41	180	81.89	miRBase [14]
		Mus musculus	1234	39	147	82.60	
		Bos taurus	1064	43	149	76.23	
		Gallus gallus	882	48	169	87.36	
		Oreochromis niloticus	812	40	100	61.05	
		Equus caballus	715	52	145	104.61	
		Glycine max	684	54	473	135.92	
		Monodelphis domestica	680	44	111	64.92	
		Medicago truncatula	672	54	910	165.26	
		Pan troglodytes	655	69	148	89.94	
II	tRNA	101*	1110	54	99	77.56	Psi-C Database [15]
III	lncRNA	*Homo sapiens*	1000	202	29066	1496.97	LNCipedia [16]
IV	circRNA	*Mus musculus*	1000	51	29991	1566.49	circBase [17]

Algorithm 1: Generate Negative Sequences

Input: RNA Sequence file R
Output: Negative Sequences Table

for *each* $x \in R$ **do**
  ▷ x: one sequence
  ▷ n_x: length of x
  ▷ Step 1: RNAfold Calculation for x
  f_{fold} = secondary structure of x
  for *each* $i \in \{1, 2, 3, \ldots, n_x\}$ **do**
    ▷ i: index of base
    if $f_{fold_i} =$. **then**
      $x_i = x'_i$
    if $f_{fold_i} =$ (**or**) **then**
      $x_i = x_i$
  ▷ Step 2: Octalization Method
  for *each* $i \in \{1, 2, 3, \ldots, n_x\}$ **do**
    ▷ y: octal representation of x
    $(y_{3i-2}y_{3i-1}y_{3i}) = T(x_i)$ ▷ T:(table)
  ▷ Step 3: Shifting on y
  ▷ z: shifted version of y
  $z_{3n_x} = y_1$
  for *each* $j \in \{1, 2, 3, \ldots, 3n_x - 1\}$ **do**
    $z_j = y_{j+1}$
  ▷ Step 4: Inverse of Octal Representation
  ▷ w: inverse octal representation of z
  for *each* $i \in \{1, 2, 3, \ldots, n_x\}$ **do**
    $w_i = T^{-1}(z_{3i-2}z_{3i-1}z_{3i})$
    if $w_i = A'$ **then**
      $w_i = A$
    if $w_i = G'$ **then**
      $w_i = G$
    if $w_i = U'$ **then**
      $w_i = U$
    if $w_i = C'$ **then**
      $w_i = C$
  Concatenate w_i
Output negative sequences table

start of the negative RNA generation process, the sequence is converted to numerical data using a map from base elements to octal representation (See Table 2). Since every base element corresponds to a three-digit number in our map, the length of numerical representation is 3n. After finding the octal representation of the sequence, the first entry in this representation is moved to the last index, becoming $3n^{th}$ entry in the newly created numerical representation, and the indices for all other entries are decreased by 1; thus, they are shifted backwards in the representation. This representation-shifting process creates a new numerical value, which can be transformed into a sequence of base elements using inverse of the map we have defined since the map is bijective (See Table 2). The outcome of this procedure gives us a sequence in the form of a secondary structure which need not be genuine. For this reason, we convert this sequence into an RNA sequence without the structure information. The sequence we obtain at the end is the novel negative RNA-sequence (Fig. 1).

5. Graphical representation of RNA secondary structures

Zhang et al. created a dynamic 3D graphical representation for RNA structure based on the chemical properties of the bases [20].

- amino group M = {A, C} and keto group K = {G, U},
- purine group R = {A, G} and pyrimidine group Y = {C, U}
- weak H-bonds group W = {A, U} and strong H-bonds group S = {C, G}.

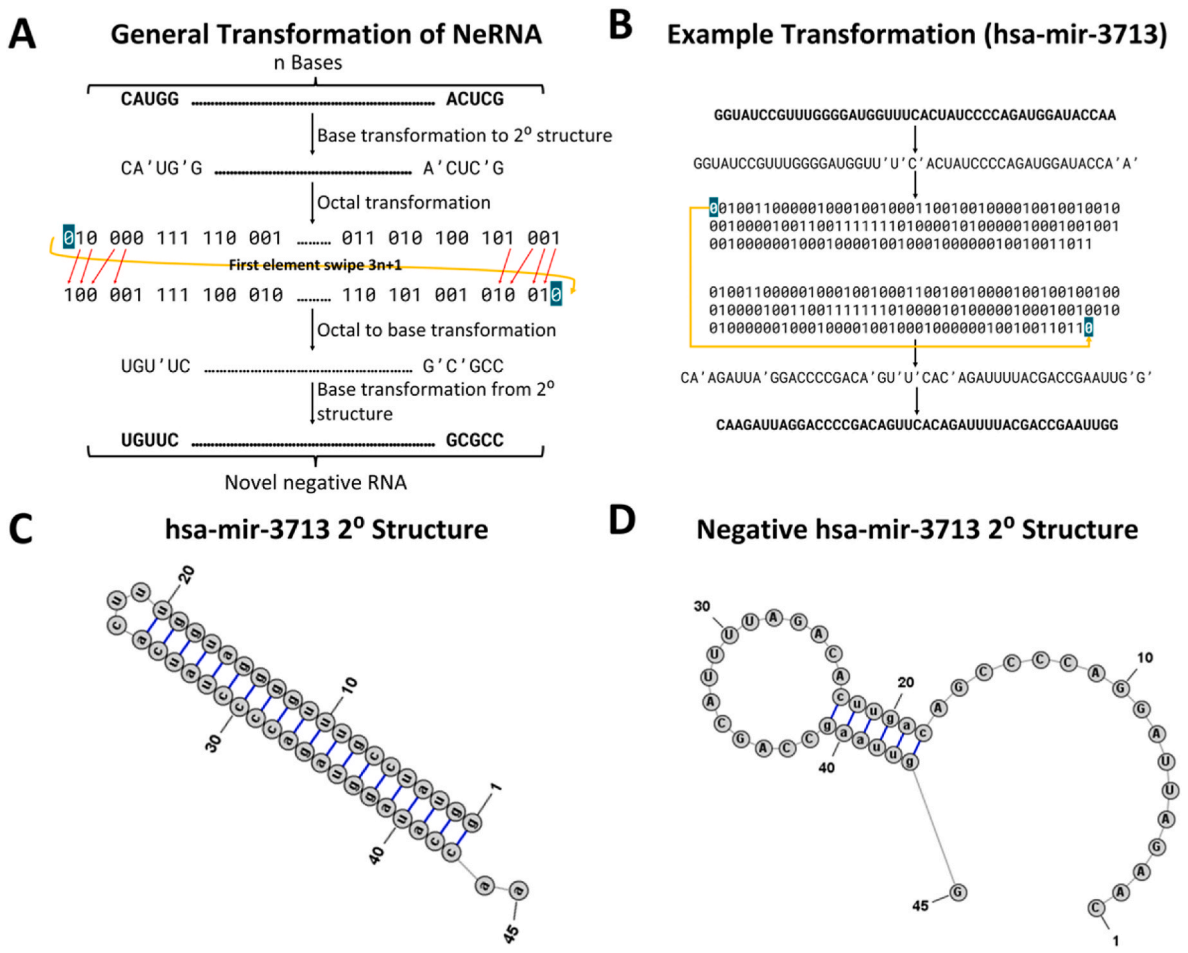

Fig. 1. Novel negative RNA generation example. A) General transformation approach of NeRNA using any given RNA sequences, B) hsa-mir-3713 transformation by NeRNA, C) hsa-mir-3713 secondary structure, D) Secondary structure of negative hsa-mir-3713 sequence obtained from NeRNA. RNAfold is applied for structure generation.

The same base grouping scheme is applied to define three maps $\alpha 1$, $\alpha 2$ and $\alpha 3$ (Table 3), where n is the length of the RNA sequence and i is the index of base in the sequence. For a primer number of p, since the p-adic representation is unique for positive integers, a p-adic approach is used in these maps. Due to the limitations on digits of numbers in KNIME platform, prime numbers 101, 103, 107 and 109 are used with a scale. Three maps considered together will produce a unique representation of the sequence. In order to represent ncRNA secondary structure as vectors, based on the definitions from Table 3, 36-dimensional vector features are calculated as shown in Table 4.

6. Case study

Four case studies are performed to demonstrate how NeRNA improves the generation of negative sequences. The application of NeRNA workflow to create artificial sequences is demonstrated using the sequence samples from lncRNA, circRNA, tRNA, and miRNA (Table 1). In addition, pseudo-human [21] miRNA dataset and NeRNA generated negative human miRNA sequences are used for comparative analysis.

Machine learning-based classifiers like Decision Trees (DT), Random Forest (RF), and Naive Bayes (NB), and deep-learning-based approaches such as multilayer perceptron (MLP), convolutional neural network (CNN), simple feed-forwad neural networks (FNN) are employed to test the efficiency of the NeRNA generated negative sequences. Classification for model generation and predictions on datasets are performed using KNIME. Parameters and setting used for each algorithm are listed in Supplementary File, (Table S1). In order to avoid overfitting and

Table 3
An explanation of three maps.

g_i	$a_1(g_i) = (x_{1i}, y_{1i}, z_{1i})$			g_i	$a_2(g_i) = (x_{2i}, y_{2i}, z_{2i})$			g_i	$a_3(g_i) = (x_{3i}, y_{3i}, z_{3i})$		
	x_{1i}	y_{1i}	z_{1i}		x_{2i}	y_{2i}	z_{2i}		x_{3i}	y_{3i}	z_{3i}
{A or C}	$\sin\left(\frac{2\pi i}{n}\right)$	$\cos\left(\frac{2\pi i}{n}\right)$	1.01^i	{A or G}	$\sin\left(\frac{2\pi i}{n}\right)$	$\cos\left(\frac{2\pi i}{n}\right)$	1.03^i	{A or U}	$\sin\left(\frac{2\pi i}{n}\right)$	$\cos\left(\frac{2\pi i}{n}\right)$	1.07^i
{G or U}	$\sin\left(\frac{2\pi i}{n}\right)$	$-\cos\left(\frac{2\pi i}{n}\right)$	1.03^i	{C or U}	$\sin\left(\frac{2\pi i}{n}\right)$	$-\cos\left(\frac{2\pi i}{n}\right)$	1.07^i	{C or G}	$\sin\left(\frac{2\pi i}{n}\right)$	$-\cos\left(\frac{2\pi i}{n}\right)$	1.09^i
{A' or C'}	$-\sin\left(\frac{2\pi i}{n}\right)$	$\cos\left(\frac{2\pi i}{n}\right)$	1.07^i	{A' or G'}	$-\sin\left(\frac{2\pi i}{n}\right)$	$\cos\left(\frac{2\pi i}{n}\right)$	1.09^i	{A' or U'}	$-\sin\left(\frac{2\pi i}{n}\right)$	$\cos\left(\frac{2\pi i}{n}\right)$	1.01^i
{G' or U'}	$-\sin\left(\frac{2\pi i}{n}\right)$	$-\cos\left(\frac{2\pi i}{n}\right)$	1.09^i	{C' or U'}	$-\sin\left(\frac{2\pi i}{n}\right)$	$-\cos\left(\frac{2\pi i}{n}\right)$	1.01^i	{C' or G'}	$-\sin\left(\frac{2\pi i}{n}\right)$	$-\cos\left(\frac{2\pi i}{n}\right)$	1.03^i

Table 4
The components of 36-dimensional vector.

$x_1^1 = \frac{1}{n}\sum_{i=1}^{n} x_{1i}^{A,C}$,	$y_1^1 = \frac{1}{n}\sum_{i=1}^{n} y_{1i}^{A,C}$,	$z_1^1 = \frac{1}{1.01^n}\sum_{i=1}^{n} z_{1i}^{A,C}$,	$x_1^2 = \frac{1}{n}\sum_{i=1}^{n} x_{1i}^{G,U}$,	$y_1^2 = \frac{1}{n}\sum_{i=1}^{n} y_{1i}^{G,U}$,	$z_1^2 = \frac{1}{1.03^n}\sum_{i=1}^{n} z_{1i}^{G,U}$,
$x_1^3 = \frac{1}{n}\sum_{i=1}^{n} x_{1i}^{A',C'}$,	$y_1^3 = \frac{1}{n}\sum_{i=1}^{n} y_{1i}^{A',C'}$,	$z_1^3 = \frac{1}{1.07^n}\sum_{i=1}^{n} z_{1i}^{A',C'}$,	$x_1^4 = \frac{1}{n}\sum_{i=1}^{n} x_{1i}^{G',U'}$,	$y_1^4 = \frac{1}{n}\sum_{i=1}^{n} y_{1i}^{G',U'}$,	$z_1^4 = \frac{1}{1.09^n}\sum_{i=1}^{n} z_{1i}^{G',U'}$,
$x_2^1 = \frac{1}{n}\sum_{i=1}^{n} x_{2i}^{A,G}$,	$y_2^1 = \frac{1}{n}\sum_{i=1}^{n} y_{2i}^{A,G}$,	$z_2^1 = \frac{1}{1.03^n}\sum_{i=1}^{n} z_{2i}^{A,G}$,	$x_2^2 = \frac{1}{n}\sum_{i=1}^{n} x_{2i}^{C,U}$,	$y_2^2 = \frac{1}{n}\sum_{i=1}^{n} y_{2i}^{C,U}$,	$z_2^2 = \frac{1}{1.07^n}\sum_{i=1}^{n} z_{2i}^{C,U}$,
$x_2^3 = \frac{1}{n}\sum_{i=1}^{n} x_{2i}^{A',G'}$,	$y_2^3 = \frac{1}{n}\sum_{i=1}^{n} y_{2i}^{A',G'}$,	$z_2^3 = \frac{1}{1.09^n}\sum_{i=1}^{n} z_{2i}^{A',G'}$,	$x_2^4 = \frac{1}{n}\sum_{i=1}^{n} x_{2i}^{C',U'}$,	$y_2^4 = \frac{1}{n}\sum_{i=1}^{n} y_{2i}^{C',U'}$,	$z_2^4 = \frac{1}{1.01^n}\sum_{i=1}^{n} z_{2i}^{C',U'}$,
$x_3^1 = \frac{1}{n}\sum_{i=1}^{n} x_{3i}^{A,U}$,	$y_3^1 = \frac{1}{n}\sum_{i=1}^{n} y_{3i}^{A,U}$,	$z_3^1 = \frac{1}{1.07^n}\sum_{i=1}^{n} z_{3i}^{A,U}$,	$x_3^2 = \frac{1}{n}\sum_{i=1}^{n} x_{3i}^{C,G}$,	$y_3^2 = \frac{1}{n}\sum_{i=1}^{n} y_{3i}^{C,G}$,	$z_3^2 = \frac{1}{1.09^n}\sum_{i=1}^{n} z_{3i}^{C,G}$,
$x_3^3 = \frac{1}{n}\sum_{i=1}^{n} x_{3i}^{A',U'}$,	$y_3^3 = \frac{1}{n}\sum_{i=1}^{n} y_{3i}^{A',U'}$,	$z_3^3 = \frac{1}{1.01^n}\sum_{i=1}^{n} z_{3i}^{A',U'}$,	$x_3^4 = \frac{1}{n}\sum_{i=1}^{n} x_{3i}^{C',G'}$,	$y_3^4 = \frac{1}{n}\sum_{i=1}^{n} y_{3i}^{C',G'}$,	$z_3^4 = \frac{1}{1.03^n}\sum_{i=1}^{n} z_{3i}^{C',G'}$.

problems related to imbalanced datasets, one to one ratio of positive and negative datasets are applied. For instance, dataset IV (Table 1) has 1000 mouse circRNA sequences that are used as positive samples, and for each sequence in this dataset only one corresponding negative one is created, leading to 1000 negative examples. Equal amounts of samples from the positive and negative datasets are randomly divided into training (70%) and testing groups (30%). The training dataset is used on six classifiers NB, DT, RF, MLP, CNN, and FNN, and their performance scores are stored for each iteration. Through 1000 iterations of the sampling and learning process, also known as Monte Carlo Cross-Validation [22], to evaluate model performances, recall, precision, sensitivity, specificity, F-measure and accuracy scores are measured (Supplementary File 2). This machine learning workflow ensures that each classifier uses identical datasets in each iteration, guaranteeing a fair evaluation.

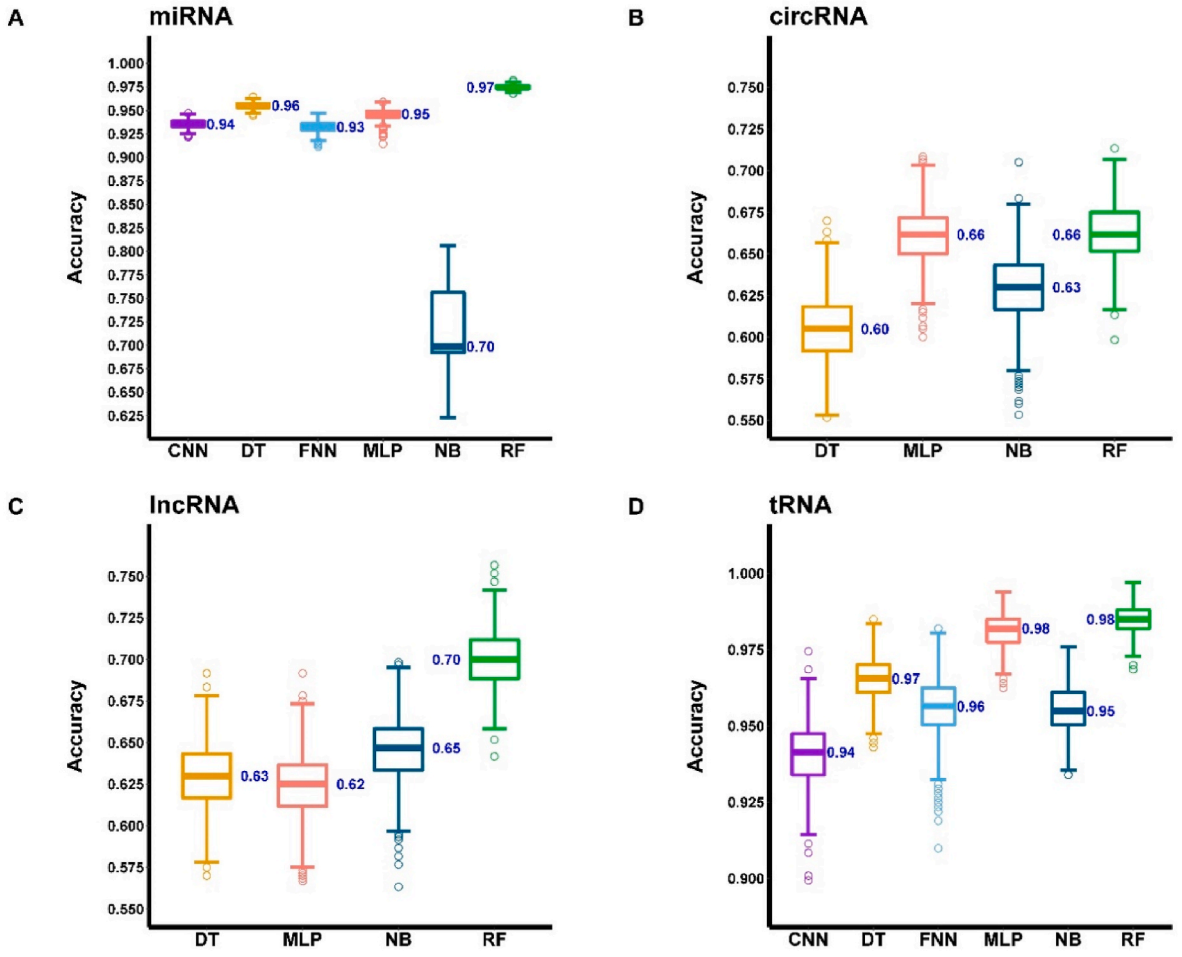

Fig. 2. Performance evaluation of Convolutional Neural Network (CNN), Decision Tree (DT), Feed-Forward Neural Network (FNN), Multilayer Perceptron (MLP), Naive Bayes (NB), and Random Forest (RF).The x-axis indicates the classifier, and the y-axis shows accuracy scores. Box-plots represents accuracy values from 1000 models. Datasets of known ncRNAs and their corresponding NeRNA generated negatives are used. A) miRNA, B) lncRNA, C) tRNA and D) circRNA results show that RF produces better performance. The values for Q2 (median) percentiles are displayed in the boxes.

7. Results

In this study, we develop NeRNA, a novel method for negative sequence generation powered by an *ab initio* approach. NeRNA is released as a KNIME workflow which can be installed from supplementary files.

Machine learning can be used for deciphering how negative RNA sequences affect the performance of classification for model generation. Therefore, various classification algorithms such as RF, DT, NB, MLP, FNN, and CNN are applied to four ncRNA groups' sequences; miRNA, lncRNA, tRNA, and circRNA (Fig. 2). However, CNN and FNN models had problems to learn from examples of lncRNA and circRNA without normalization, thus they are not shown in Fig. 2B and C (Decimal scaling normalized results of CNN and FNN on lncRNA and circRNA datasets are shown in Supplementary File 1, Fig. S1.)

R libraries, ggplot2 (v3.3.5) [23], ggpubR (v0.4.0) and pROC (v1.18.0) [24] are used to generate graphs for model learning statistics. Among the six classifiers, Random Forest seems to be showing better performance in terms of accuracy for all ncRNA classes (Fig. 2). While Decision Tree models appears to be slightly better than Multilayer Perceptron in miRNA sequences (Fig. 2A), Multilayer Perceptron models are showing higher performance to some extent in circRNA (Fig. 2B). Moreover, models are further evaluated using the receiver operating characteristic (ROC) curve score (Fig. 3). The area under the curve (AUC) values on the ROC curve graphs indicate that similar to accuracy assessment, Random Forest models achieve higher performance.

For demonstrating how NeRNA generated negative sequences perform compared to the well-known negative datasets, pseudo hairpin sequences are used (Fig. 4). In order to achieve this, secondary structures and 36-dimensional features of each sequence are calculated. Afterwards, the KNIME classification workflow is used with the same settings. Based on the results of all classification algorithms, it appears that NeRNA-based negative-human miRNA sequences seem to have higher performances (Fig. 4).

In order to test, if the performance of NeRNA could be affected by the organism of the data source, species-specific analysis has been

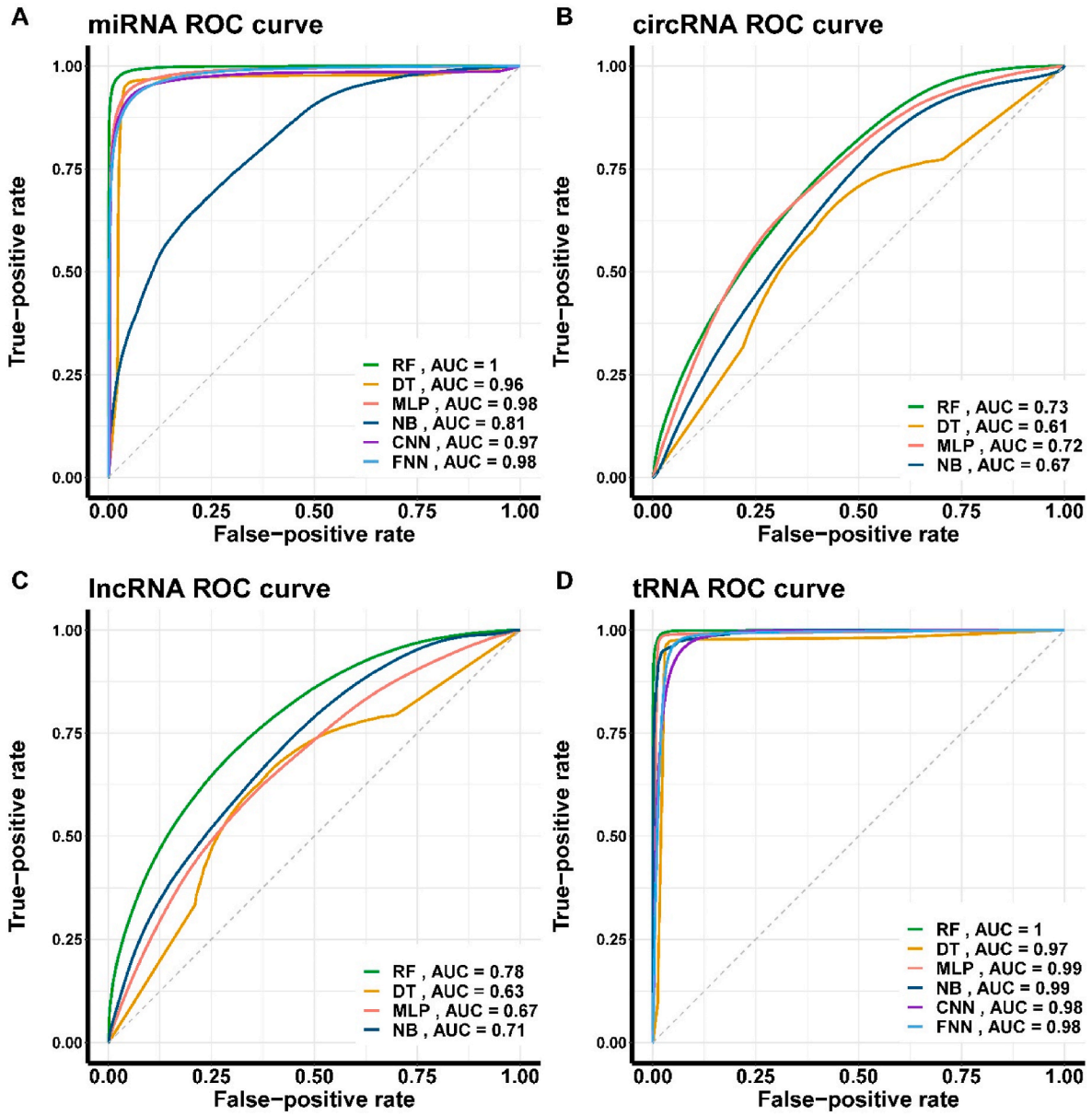

Fig. 3. ROC curve and AUC value graphs. Convolutional Neural Network (CNN), Decision Tree (DT), Feed-Forward Neural Network (FNN), Multilayer Perceptron (MLP), Naive Bayes (NB), and Random Forest (RF) classifiers are used for analysis.

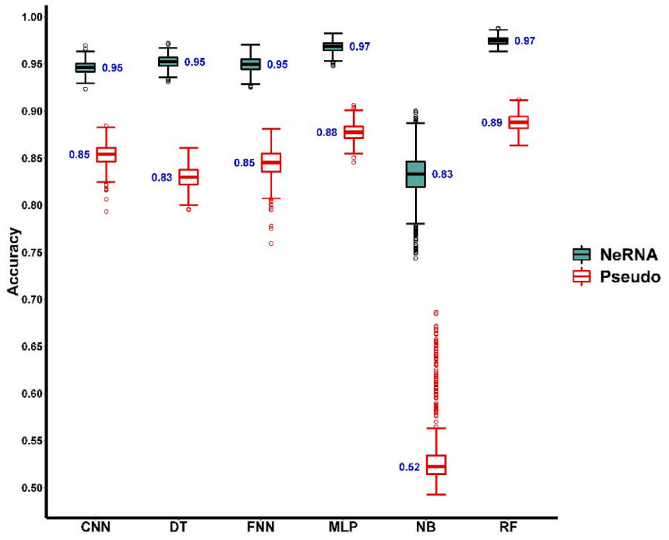

Fig. 4. Box plots of classifier accuracy scores trained with NeRNA and pseudo miRNA negative sequences. Median values are displayed in the middle of the box plots. Convolutional Neural Networks, Decision Trees, Feed-Forward Neural Networks, Multilayer Perceptron, Naive Bayes, and Random Forest are represented by CNN, DT, FNN, MLP, NB, and RF, respectively. As positive data, miRNA hairpin sequences (Table 1) are used in both cases.

performed by using miRNA precursor sequences of ten different species (Table 1). According to the results shown in Fig. 5, in line with the previous findings, Random Forest is the highest performance classifier, followed by MLP and DT in all species.

8. Discussion

Noncoding-RNAs are of major interest since they are involved in various cellular processes including translation (tRNA), post-transcriptional gene regulation (miRNA), X chromosome inactivation (lncRNA) and acting as miRNA sponges (circRNA). Moreover, some of them take role in disease conditions and can be used as disease markers and/or new therapeutic agents. Many existing tools attempt to identify new ncRNAs through the applications of machine learning algorithms, while very few of them take into account of negative samples for learning. In general, positive training datasets are obtained from databases listing predicted or experimentally determined examples of individual ncRNA classes. Considering the many to many relationships between majority of ncRNAs (e.g., miRNA) and their possible targets, it is not possible to co-express all of them for experimental validation, thus, all available negative datasets are arbitrary.

In our previous studies, we endeavored to delineate the impact of ML elements for miRNA prediction [11] and assess the quality of available negative datasets [25]. The easiest way to construct a negative learning dataset which is applied in many studies, is random shuffling of known miRNA sequences. However, this leads to a relatively easy model generation for many classifiers and predictions based on these models could be misleading. In addition, the process of shuffling of the overall sequence is very unlikely to happen in the cell. Other existing negative datasets are composed of either coding sequences [26] or pseudo hairpins from human RefSeq genes that do not have any known experimental support for alternative splicing (AS) events [21].

Functions of ncRNAs are mostly associated with not only their sequences but also with their specific structures. In the case of tRNAs and miRNA precursors, their conserved structures are very essential for their recognition and proper action. In order to transform RNA sequence and structure information into numerical parameters that can be used for training ML methods, several approaches have been proposed. Yao et al. developed a novel 2D graphical representation of RNA secondary structure [27] and there are also 3D representation methods based on the sequence and base chemical information [28]. While the former could suffer from the loss of information due to its nonuniqueness in representing an RNA, the latter method is space-demanding, mainly for long sequences [20]. Since ncRNAs have major differences in length, we adapted a modified version of the dynamic 3D graphical representation method firstly proposed by Zhang et al. [10,20]. The features calculated through this approach are based on the sequence, structure and chemical properties of the RNA thus they can be applied to any ncRNA.

For testing the efficiency of our novel negative datasets, examples from different ncRNA classes are used as positive datasets (Table 1). Among the four classes of ncRNAs analyzed, apparently the smaller ones; tRNAs and miRNAs seem to be better distinguished by the classification algorithms (Fig. 2). We believe that there are two main reasons for this situation. Firstly, tRNAs and miRNA precursors (hairpins) have highly conserved structures and disruption of them through the negative data generation process creates a big difference between the original sequence and its corresponding negative RNA. Another explanation is that lncRNAs and circRNAs have a wide range between their minimum and maximum sequence lengths with an average of 1500 nucleotides (Table 1). Comparing to lncRNAs, circRNAs have a specific circularly shaped structure so it was expected that it would be showing more similar results to tRNAs and miRNAs. Interestingly, the accuracy scores for circRNA analysis show slightly lower values than lncRNAs. We suspect that when the sequence length is above a certain level (e.g. 1000 nt) the influence of the secondary structure might be decreasing. In order to overcome this issue, the dimension of the vector space defining the characteristics of these longer sequences might be increased.

In the miRNA case study, since pseudo dataset is commonly applied for measuring the performance of ML approaches for miRNA identification, it was also used for comparing our novel negative miRNA dataset. As it is shown in Fig. 4, with the same settings (positive dataset, features, classifiers), negative miRNA sequence examples generated by our approach resulted in models having higher accuracy scores.

Due to the limitations of some external tools that have been used while building the NeRNA framework and the nature of RNA molecules, there are certain restrictions in the process. For instance, tRNAs have various post-transcriptionally modified nucleotides like pseudouridine (Ψ) [29]. Unfortunately, it is not possible to account for all those unusual bases while calculating the secondary structures. Thus, unmodified bases and their structures obtained from the database are used (Table 1) [15]. Although current version of NeRNA generates a single negative example per given RNA sequence, it is possible to increase these ratio if needed.

In this work, a novel system for standardization of negative dataset generation of RNA sequences is proposed. It has been shown that integrating sequence, structure and mathematical repreeantation of RNA samples could contribute to generation of high-quality negative samples. Comparison analysis results of the miRNA classification showed that NeRNA generated negative sequences are superior to existing datasets. Although, our method is designed for noncoding RNAs, it is also possible to apply it on coding RNAs, as well as pathogenic and/or viral RNA sequences.

Availability

NeRNA framework, R scripts, datasets and guidelines for the usage of the workflow are available at the GitHub repository (https://github.com/Mehmeteminorhan/NegativeRNA).

Declaration of competing interest

None declared.

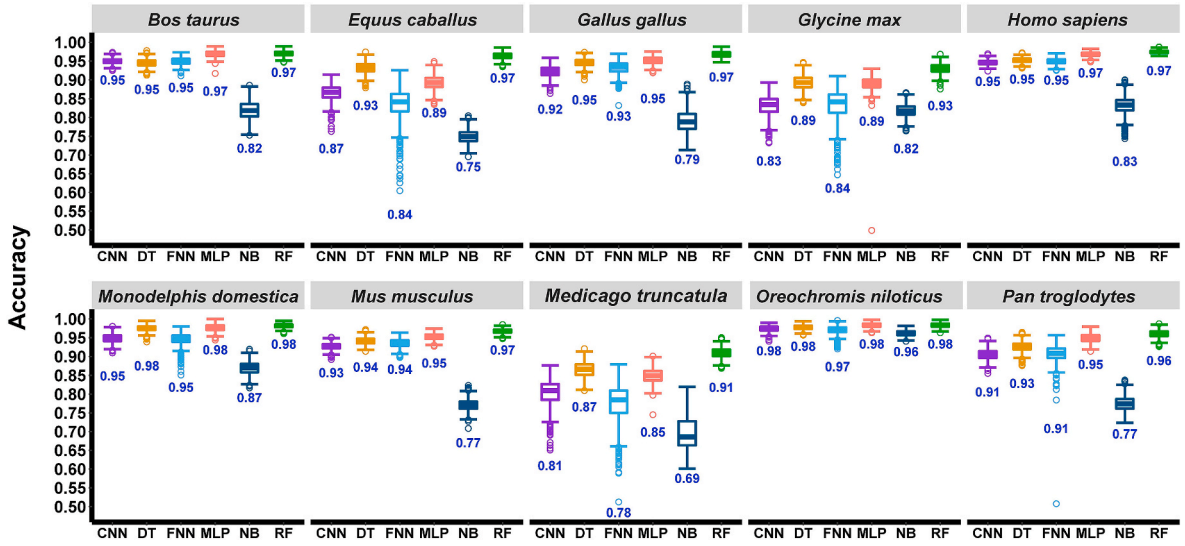

Fig. 5. Species – specific performance comparison. The y-axis shows accuracy scores, and the x-axis indicates the classifier. Convolutional Neural Network (CNN), Decision Tree (DT), Feed-Forward Neural Network (FNN), Multilayer Perceptron (MLP), Naive Bayes (NB), and Random Forest (RF) classifiers are used for analysis. Q2 (median) percentiles values are displayed at the bottom of the boxes. The names of each species are presented at the top, respectively.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.compbiomed.2023.106861.

References

[1] V.L. Robinson, Rethinking the central dogma: noncoding RNAs are biologically relevant, Urol. Oncol. Sem. Orig. Invest. 27 (2009) 304–306, https://doi.org/10.1016/j.urolonc.2008.11.004.

[2] Y. Su, H. Wu, A. Pavlosky, L.-L. Zou, X. Deng, Z.-X. Zhang, A.M. Jevnikar, Regulatory non-coding RNA: new instruments in the orchestration of cell death, Cell Death Dis. 7 (2016), https://doi.org/10.1038/cddis.2016.210 e2333–e2333.

[3] X.-D. Fu, Non-coding RNA: a new frontier in regulatory biology, Natl. Sci. Rev. 1 (2014) 190–204, https://doi.org/10.1093/nsr/nwu008.

[4] Y.M. Demirci, M.D. Saçar Demirci, Circular RNA–MicroRNA–MRNA Interaction Predictions in SARS-CoV-2 Infection, vol. 18, 2021, pp. 45–50, https://doi.org/10.1515/jib-2020-0047.

[5] S. Griffiths-Jones, R.J. Grocock, S. van Dongen, A. Bateman, A.J. Enright, miRBase: microRNA sequences, targets and gene nomenclature, Nucleic Acids Res. 34 (2006) D140–D144, https://doi.org/10.1093/nar/gkj112.

[6] A. Frankish, M. Diekhans, I. Jungreis, J. Lagarde, J.E. Loveland, J.M. Mudge, C. Sisu, J.C. Wright, J. Armstrong, I. Barnes, A. Berry, A. Bignell, C. Boix, S. Carbonell Sala, F. Cunningham, T. Di Domenico, S. Donaldson, I.T. Fiddes, C. García Girón, J.M. Gonzalez, T. Grego, M. Hardy, T. Hourlier, K.L. Howe, T. Hunt, O.G. Izuogu, R. Johnson, F.J. Martin, L. Martínez, S. Mohanan, P. Muir, F. C.P. Navarro, A. Parker, B. Pei, F. Pozo, F.C. Riera, M. Ruffier, B.M. Schmitt, E. Stapleton, M.-M. Suner, I. Sycheva, B. Uszczynska-Ratajczak, M.Y. Wolf, J. Xu, Y. T. Yang, A. Yates, D. Zerbino, Y. Zhang, J.S. Choudhary, M. Gerstein, R. Guigó, T.J. P. Hubbard, M. Kellis, B. Paten, M.L. Tress, P. Flicek, Gencode 2021, Nucleic Acids Res. 49 (2021) D916–D923, https://doi.org/10.1093/nar/gkaa1087.

[7] M.D. Saçar Demirci, in: J. Allmer, M. Yousef (Eds.), Computational Detection of Pre-microRNAs, Springer US, New York, NY, 2022, pp. 167–174, https://doi.org/10.1007/978-1-0716-1170-8_8.

[8] L. Heikkinen, M. Kolehmainen, G. Wong, Prediction of microRNA targets in Caenorhabditis elegans using a self-organizing map, Bioinformatics 27 (2011) 1247–1254, https://doi.org/10.1093/bioinformatics/btr144.

[9] B.-T. Zhang, J.-W. Nam, Supervised learning methods for MicroRNA studies, in: Mach. Learn. Bioinforma., John Wiley & Sons, Inc., 2008, pp. 339–365, https://doi.org/10.1002/9780470397428.ch16.

[10] M.D. Saçar Demirci, MicroRNA prediction based on 3D graphical representation of RNA secondary structures, Turkish J. Biol. 43 (2019) 274–280, https://doi.org/10.3906/biy-1904-59.

[11] M.D. Saçar Demirci, J. Allmer, Delineating the impact of machine learning elements in pre-microRNA detection, PeerJ 5 (2017), e3131, https://doi.org/10.7717/peerj.3131.

[12] Z.R. Yang, Machine Learning Approaches to Bioinformatics, World Scientific Publishing, Toh Tuck Link, Singapore, 2010.

[13] M.R. Berthold, N. Cebron, F. Dill, T.R. Gabriel, T. Kötter, T. Meinl, P. Ohl, C. Sieb, K. Thiel, B. Wiswedel, KNIME: the Konstanz Information Miner, SIGKDD Explor., 2008, pp. 319–326, https://doi.org/10.1007/978-3-540-78246-9_38.

[14] S. Griffiths-Jones, miRBase: microRNA sequences and annotation, Curr. Protoc. Bioinf. Chapter 12 (2010), https://doi.org/10.1002/0471250953.bi1209s29. Unit 12.9.1-10.

[15] M.P. Sajek, T. Woźniak, M. Sprinzl, J. Jaruzelska, J. Barciszewski, T-psi-C: user friendly database of tRNA sequences and structures, Nucleic Acids Res. 48 (2020) D256–D260, https://doi.org/10.1093/NAR/GKZ922.

[16] P.J. Volders, J. Anckaert, K. Verheggen, J. Nuytens, L. Martens, P. Mestdagh, J. Vandesompele, LNCipedia 5: towards a reference set of human long non-coding RNAs, Nucleic Acids Res. 47 (2019) D135–D139, https://doi.org/10.1093/NAR/GKY1031.

[17] P. Glažar, P. Papavasileiou, N. Rajewsky, circBase: a database for circular RNAs, RNA 20 (2014) 1666–1670, https://doi.org/10.1261/rna.043687.113.

[18] D. Charif, J.R. Lobry, SeqinR 1.0-2: A Contributed Package to the R Project for Statistical Computing Devoted to Biological Sequences Retrieval and Analysis, 2007, pp. 207–232, https://doi.org/10.1007/978-3-540-35306-5_10.

[19] I.L. Hofacker, Vienna RNA secondary structure server, Nucleic Acids Res. 31 (2003) 3429–3431, https://doi.org/10.1093/nar/gkg599.

[20] Y. Zhang, H. Huang, X. Dong, Y. Fang, K. Wang, L. Zhu, K. Wang, T. Huang, J. Yang, A dynamic 3D graphical representation for RNA structure analysis and its application in non-coding RNA classification, PLoS One 11 (2016) 1–15, https://doi.org/10.1371/journal.pone.0152238.

[21] K.L.S. Ng, S.K. Mishra, De novo SVM classification of precursor microRNAs from genomic pseudo hairpins using global and intrinsic folding measures, Bioinformatics 23 (2007) 1321–1330, https://doi.org/10.1093/bioinformatics/btm026.

[22] Q.S. Xu, Y.Z. Liang, Monte Carlo cross validation, Chemometr. Intell. Lab. Syst. 56 (2001) 1–11, https://doi.org/10.1016/S0169-7439(00)00122-2.

[23] R.A.M. Villanueva, Z.J. Chen, ggplot2: Elegant Graphics for Data Analysis, second ed., vol. 17, 2019, pp. 160–167, https://doi.org/10.1080/15366367.2019.1565254. Https://Doi.Org/10.1080/15366367.2019.1565254.

[24] X. Robin, N. Turck, A. Hainard, N. Tiberti, F. Lisacek, J.-C. Sanchez, M. Müller, pROC: an open-source package for R and S+ to analyze and compare ROC curves, BMC Bioinf. 12 (2011) 77, https://doi.org/10.1186/1471-2105-12-77.

[25] M.D. Saçar Demirci, J. Baumbach, J. Allmer, On the performance of pre-microRNA detection algorithms, Nat. Commun. 8 (2017) 330, https://doi.org/10.1038/s41467-017-00403-z.

[26] L. Wei, M. Liao, Y. Gao, R. Ji, Z. He, Q. Zou, Improved and promising identification of human MicroRNAs by incorporating a high-quality negative set, IEEE ACM Trans. Comput. Biol. Bioinf 11 (2013) 192–201, https://doi.org/5C261F39-0F1D-4AE0-923F-CEF4DA8E06AE.

[27] Y.H. Yao, B. Liao, T.M. Wang, A 2D graphical representation of RNA secondary structures and the analysis of similarity/dissimilarity based on it, J. Mol. Struct. THEOCHEM. 755 (2005) 131–136, https://doi.org/10.1016/j.theochem.2005.08.009.

[28] W. Zhu, B. Liao, K. Ding, A condensed 3D graphical representation of RNA secondary structures, J. Mol. Struct. THEOCHEM. 757 (2005) 193–198, https://doi.org/10.1016/j.theochem.2005.04.042.

[29] P.G. Foster, L. Huang, D.V. Santi, R.M. Stroud, The structural basis for tRNA recognition and pseudouridine formation by pseudouridine synthase I, Nat. Struct. Biol. 7 (2000) 23–27, https://doi.org/10.1038/71219.