

RESEARCH

Open Access



# nRC: non-coding RNA Classifier based on structural features

Antonino Fiannaca \*† , Massimo La Rosa†, Laura La Paglia, Riccardo Rizzo and Alfonso Urso

\*Correspondence:  
antonino.fiannaca@icar.cnr.it

†Equal contributors  
ICAR-CNR, National Research Council of Italy, Via Ugo La Malfa, 90146 Palermo, Italy

## Abstract

**Motivation:** Non-coding RNA (ncRNA) are small non-coding sequences involved in gene expression regulation of many biological processes and diseases. The recent discovery of a large set of different ncRNAs with biologically relevant roles has opened the way to develop methods able to discriminate between the different ncRNA classes. Moreover, the lack of knowledge about the complete mechanisms in regulative processes, together with the development of high-throughput technologies, has required the help of bioinformatics tools in addressing biologists and clinicians with a deeper comprehension of the functional roles of ncRNAs. In this work, we introduce a new ncRNA classification tool, nRC (non-coding RNA Classifier). Our approach is based on features extraction from the ncRNA secondary structure together with a supervised classification algorithm implementing a deep learning architecture based on convolutional neural networks.

**Results:** We tested our approach for the classification of 13 different ncRNA classes. We obtained classification scores, using the most common statistical measures. In particular, we reach an accuracy and sensitivity score of about 74%.

**Conclusion:** The proposed method outperforms other similar classification methods based on secondary structure features and machine learning algorithms, including the RNAcon tool that, to date, is the reference classifier. nRC tool is freely available as a docker image at <https://hub.docker.com/r/tblab/nrc/>. The source code of nRC tool is also available at <https://github.com/lcarPA-TBlab/nrc>.

**Keywords:** ncRNA, Classification, Structural features, Deep learning

## Background

During the last decade, research has shown a growing interest in non-coding RNA (ncRNA). They are small non-coding sequences with the potential to have a functional role in many biological processes and diseases [1] by acting through the regulation of gene expression [2–5]. Different classes of ncRNA have been identified, differing from each other by nucleotide sequence length, folding and function. The most well-known ncRNAs are structural RNA belonging to ribosomal RNA (rRNA) and transfer RNA (tRNA), both involved in translation events [6]. Another interesting class of ncRNA are microRNAs (miRNAs), 18–24 nucleotide long regulatory RNA molecules [7–9]. They can behave as tumour suppressors or oncogenes depending on which target they act upon by altering the standard molecular mechanisms in which their targets are involved [10]. In particular, they interact with target genes through a direct binding to complementary sequences

leading to either mRNA degradation or translational suppression [11]. The final result is the inhibition of the protein product. A miRNA can be considered as an oncogene if its amplification or overexpression down-regulates tumour suppressors or other genes involved in cell differentiation, thereby contributing to cancer formation by stimulating proliferation, angiogenesis, and invasion; whereas the ncRNA molecule will be considered as a tumor-suppressor if it will cause a decrease in oncogene expression [12].

Other ncRNA classes are small nuclear RNAs (snRNA), long non-coding RNAs (lncRNA), silencing RNA (siRNA), riboswitches and internal ribosome entry sites (IRES) [13]. The small nucleolar RNA (snoRNA) molecules, belonging to the snRNA class, participate to post-transcriptional modifications of rRNA, together with small nucleolar ribonucleoproteins (snoRNPs) with whom they are complexed. Dong and colleagues [14] reported a disruption of these RNA molecules in different conditions and cancer diseases [14, 15], they also identified snoRNA U50 as an important factor in the development and/or progression of breast cancer. The lncRNAs are ncRNA longer than 200 nucleotides. Recent works evidence a dysregulated expression pattern of lncRNAs in cancer samples that may be used as independent predictors of patient outcomes [16, 17]. Riboswitches are another class of ncRNA. They are structured non-coding RNA domains that selectively bind metabolites and control gene expression. They can act without the support of proteins, which strengthens the hypothesis of their important role in the regulatory machine [18].

Because of the large number and functions of different ncRNA, their proper identification and classification are a new challenging bioinformatics scenario. Indeed, considering the low percentage of the “discovered ncRNAome” and the lack of knowledge about these non-coding molecules, their classification could help biologists and clinicians in understanding the molecular mechanisms of this regulatory machine. This also implies a need to re-state the principles of basic therapeutic strategies.

The aim of the first works about ncRNA classification was to discriminate between coding and non-coding sequences. To this purpose some bioinformatics tools employ support vector machine (SVM) models [19]: CONC and CPC are prediction tools based on SVM that classify transcripts according to features belonging to coding products [20, 21]. Another interesting classification method, proposed by Lertampaiporn and colleagues [22], uses a hybrid Random Forest (RF) algorithm combined with a logistic-regression model that realises a feature-based discrimination among various ncRNAs. The recent discovery of a “Pandora box” full of a multitude of different biologically functional ncRNA, opened the way to develop resources able to discriminate the different classes of ncRNAs. Various approaches have been applied such as RNA-CODE [23], based on the alignment of short reads, or others based on multi-feature extraction and full-sequence analysis such as RNAcon [24] and GraPPLe [25]. These last methods, in particular, use graph properties (both local and global) of predicted secondary RNA structures together with machine learning algorithms. Their main feature is to identify and extract graph properties that can reflect the functional information of different classes of RNAs. To the best of our knowledge, the RNAcon algorithm currently represents the state-of-the-art classifier of ncRNA classes based on structural features and machine learning techniques. RNAcon considers 20 graph features obtained from the predicted RNA secondary structure and adopts an RF classifier [26].

In this paper, we present nRC (non-coding RNA Classifier), a novel method for the classification of ncRNA sequences belonging to different classes. Our approach uses the structural features extracted from ncRNA secondary structure, rather than the primary structure since it has been demonstrated that the structure of ncRNAs can provide relevant information about their biological functions and therefore their class type [27]. Moreover, we adopted a supervised classification algorithm implementing a deep learning (DL) architecture based on convolutional neural networks (CNNs) [28]. DL represents a successful paradigm for big data analysis, giving a relevant contribution to several fields of medicine and bioinformatics [29]. For instance, the use of DL architectures for the prediction of genomic sequences allows improving the performance of the other standard machine learning methods [30, 31].

In particular, CNNs have been successfully adopted for image classification [32] because they can extract significant features from images at different abstraction levels. Recently, CNNs have also been applied to DNA sequence classification [33] with good results, due to their capability to extract meaningful features even from sequences of symbols. The combination of both structural features and a DL architecture allows us to reach classification scores that outperform other similar classification methods based on secondary structure features and machine learning algorithms like the random forest (RF) [26] and naive Bayes (NB) [34] classifiers.

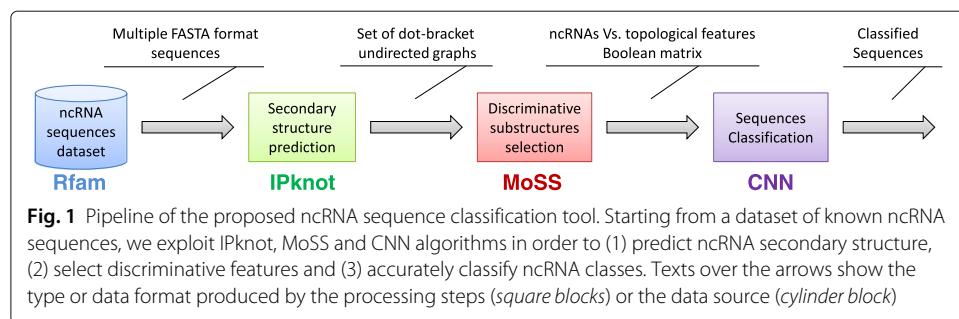
## Methods

### Proposed method

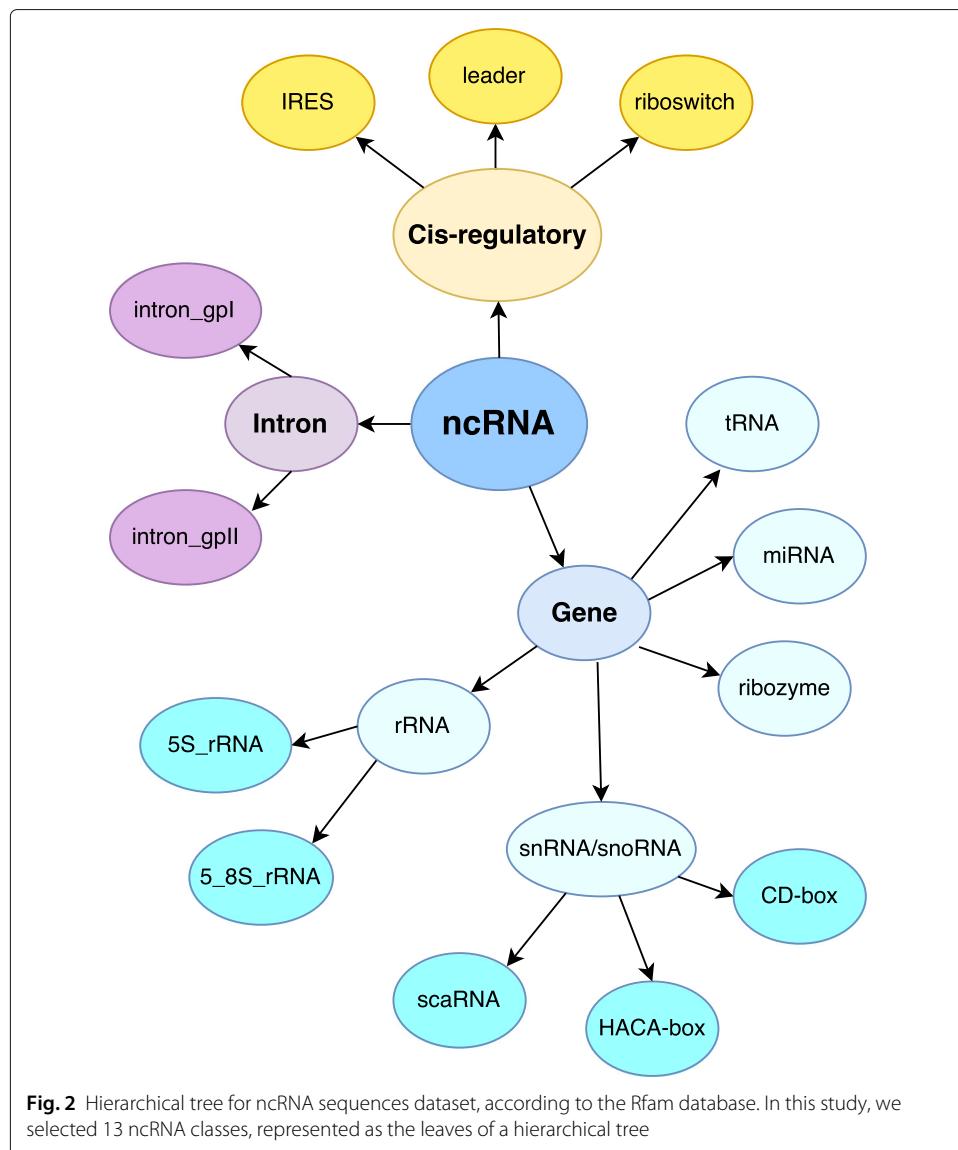
In this section, we introduce the proposed approach for the classification of ncRNA sequences. We classify ncRNA sequences by exploiting a set of discriminative substructures extracted from RNA secondary structures. Starting from a dataset composed of ncRNA fasta sequences belonging to different non-coding classes, we first predict the secondary structure of each sequence (Fig. 1). Then, we identify as features all the discriminative frequent sub-structures extracted from predicted ncRNA secondary structures. Finally, a supervised classification algorithm is trained using as input a ncRNA sequence vs. sub-structures boolean matrix. Each step of the proposed approach, corresponding to a box in Fig. 1, is detailed in the next subsections.

### ncRNA training dataset

To create a consistent and statistically meaningful ncRNA dataset, we followed the approach proposed by Panwar et al. [24], and by Childs et al. [25]. Similar to those studies, we downloaded the ncRNA sequences from the latest version of the Rfam



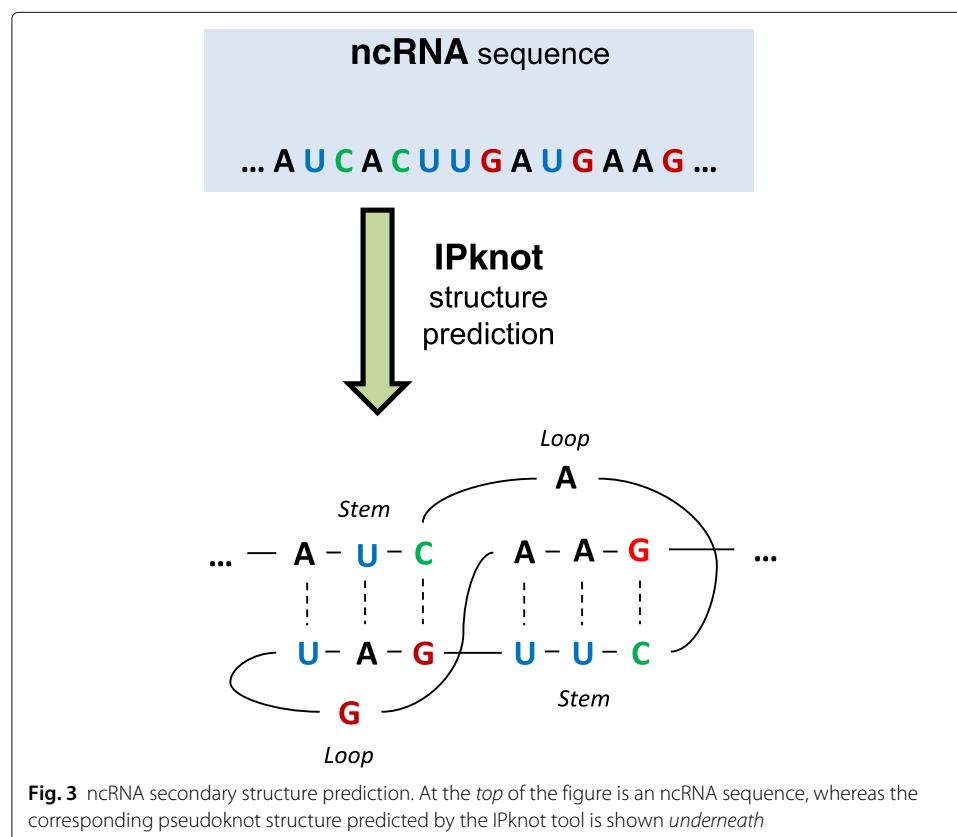
database, release 12 [35]. The Rfam repository represents one of the most complete collections of manually curated RNA sequences, including sequence alignments, annotation and consensus secondary structures. We selected the following 13 ncRNA classes: miRNA, 5S rRNA, 5.8S rRNA, ribozymes, CD-box, HACA-box, scaRNA, tRNA, Intron gpl, Intron gpII, IRES, leader, riboswitch. As will be further explained below, we chose these classes to allow a comparison as fair as possible with RNACon tool. According to Rfam hierarchical organisation among the selected ncRNA classes (Fig. 2), 5S rRNA and 5.8S rRNA belong to rRNA class; CD-box, HACA-box and scaRNA belong to snRNA/snoRNA class; Intron gpl and Intron gpII belong to Intron class. Generally speaking, the leaves of the hierarchical tree represent ncRNA classes used in this study. According to Rfam database, there are three main functional categories for non-coding RNA sequences, i.e. gene, intron or cis-regulatory element. Considering those ncRNA classes, we built a dataset composed of 20% non-redundant sequences



obtained by using the CD-HIT tool [36], as done by Panwar et al. [24]. Finally, to create a balanced dataset, we randomly selected 500 sequences for each ncRNA class, except IRES for which there are only 320 available sequences, to obtain 6320 ncRNA sequences.

#### ncRNA secondary structure prediction

Since the ncRNA dataset reports sequences in fasta format, it does not contain information about the secondary structure of non-coding RNA sequences. As aforementioned in the previous section, the secondary molecular structures can provide a major key for elucidating the potential functions of RNAs and, consequently, could help us to predict if a ncRNA sequence belongs to the same class. For this reason, just as the RNAcon approach, we choose to exploit the IPknot tool [37] for predicting the secondary structure of ncRNA. This tool takes into account all the most important topologies in RNA secondary structures and can provide good predictions in terms of both speed and accuracy with respect to other RNA structure prediction methods [37]. To the best of our knowledge, IPknot is one of the best pseudoknot-free secondary structure prediction tools, since it uses less memory and runs much faster than the other tools, without loss of accuracy [38]. In our study, the most of ncRNA sequences, such as 5S rRNA, tRNA and miRNA, are pseudoknot-free [39]. Figure 3 shows how IPknot can predict a complex secondary structure. As a result, this tool produces a dot-parenthesis format file (representing a graph) for each input sequence.



### Discriminative sub-structure selection

Each dot-parenthesis format file prepared in the previous step can be read as an undirected labelled graph representing the RNA sequence, in which vertices are nucleotides and edges are bonds between two nucleotides. As mentioned before, we have 6320 graphs belonging to 13 ncRNA classes. Of course, we can reasonably suppose there is a sort of similarity among the sequences (graphs) that belong to a particular class. Our hypothesis is that frequent sub-structures (sub-graphs) can act as local features for describing ncRNA sequences because they are probably correlated with the molecular function and, thus, they can be used to identify classes of similar non-coding RNAs.

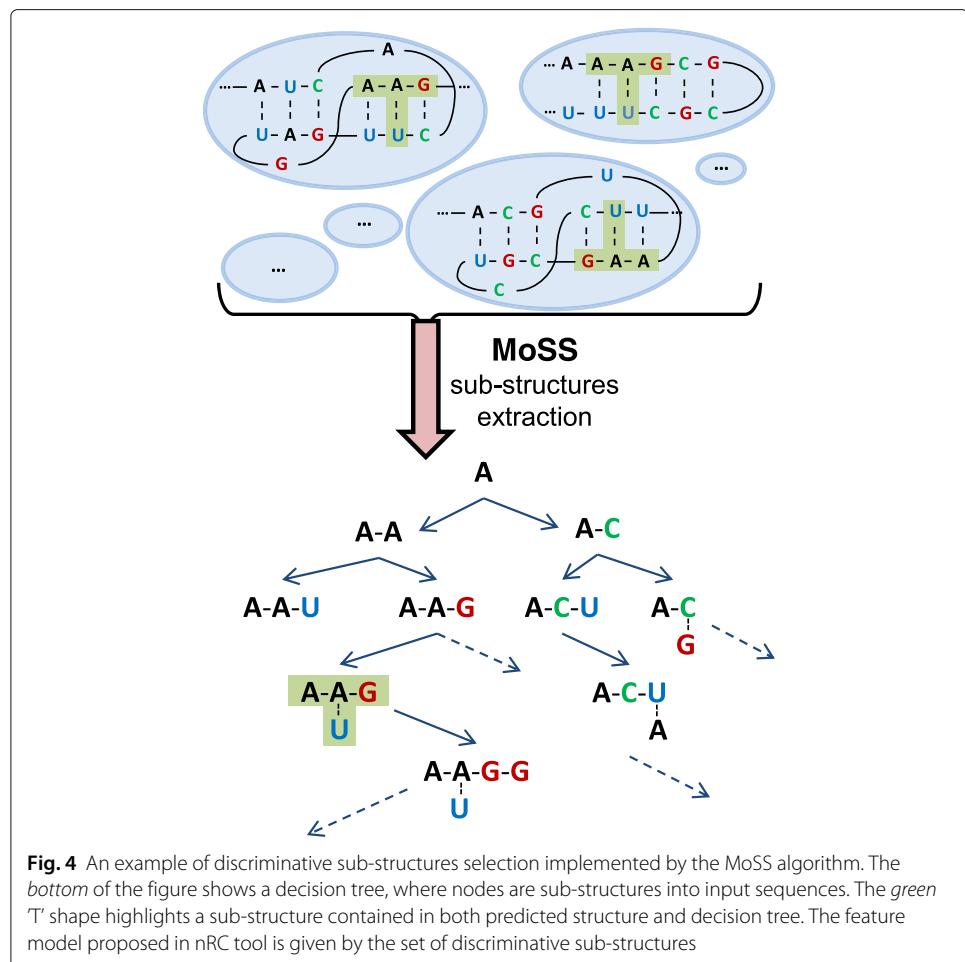
In this context, the selection of molecular sub-structures can be solved in terms of frequent sub-graphs having a certain minimum “support” in a given set of graphs, where the term support identifies the number of graphs containing a sub-graph. To find these sub-graphs, we adopted the molecular substructure miner (MoSS) algorithm [40], which implements a depth-first search strategy. The support expressed as a percentage value is the MoSS parameter that specifies the minimum frequency which a sub-structure must occur to be reported. In any case, since the search of frequent sub-graphs in a set of graphs can produce a very large number of features, the advanced pruning techniques implemented in the MoSS algorithm allows us only to obtain closed frequent sub-graphs. A sub-graph is closed only if its support (i.e., the number of graphs that contain this sub-graph) is higher than the support of all the search tree super-graphs containing this sub-graph. Also, the MoSS algorithm lets the user set the  $m$  minimum and the  $n$  maximum size the sub-structures must have to be taken into account. In the field of molecular compounds, a similar approach was applied to find potential candidates in drug discovery processes [41, 42].

As an example, Fig. 4 shows a search tree (starting from an adenine nucleotide as a seed) created by the MoSS algorithm, when the input is a list of graphs (such as those reported in the top of the figure). In this figure, a sub-graph, i.e. a node of the search tree, is highlighted with a green ‘T’ shape area in both predicted secondary structure and the search tree. That means that this sub-graph is a support for the ncRNA sequence at the top of the figure; if it is also a support for a certain user-determined percentage of input sequences and its super-graph has a lower support, it can be considered as a feature of a ncRNA dataset.

Outcomes of the MoSS algorithm are both the list of closed frequent sub-graphs and, for each graph, the list of its closed sub-graphs. Given  $g$  graphs (ncRNA sequences) and  $s$  sub-graphs (frequent sub-structures), it is possible to define a Boolean matrix  $A(g, s)$  where the element  $(i, j)$  is set to 1 when sub-structure  $j$  is contained in an ncRNA sequence  $i$ .

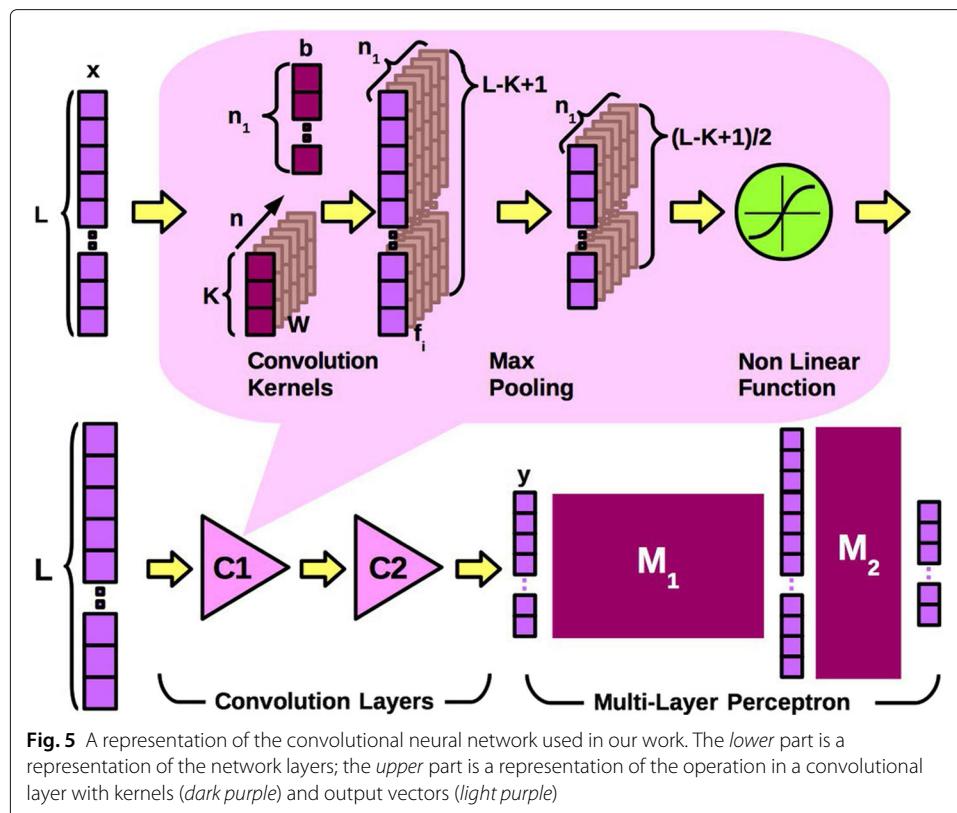
### Classification with deep learning architecture

A machine learning classifier requires a hand-crafted feature selection task to obtain the best representation of the input patterns; this step is crucial for the performances of the classifier. Automatic feature selection is one of the key results of the so-called deep learning neural networks [28, 43, 44]. Le Cun and colleagues demonstrated that feature selection could be obtained from neural network training [32]. The proposed model, called convolutional neural network, was constituted by a set of layers based on convolutional filters and average pooling layers, followed by a multi-layer perceptron. Nowadays



CNN networks are often used for image classification, in these applications the first layers of the network are trained to recognise features constituted by edges or colour details that are assembled to create more sophisticated features used as image descriptors [28]. These image descriptors represent the input of the last, fully connected layers of the network, that implement the classifier.

In the nRC system, each position of the vector obtained by the MoSS subsystem indicates the presence or absence of a structural configuration (Fig. 4). Even if the input vectors are binary and the vector components are not in a particular order, bit configurations can still be used as useful features and assembled to build new, more sophisticated patterns that a CNN can exploit. The neural network used in the nRC system is made by two convolutional layers, C1 and C2, followed by two fully connected layers (Fig. 5). The first convolutional layer C1 of the network learns to recognise features constituted by  $n_1$  groups of these binary values. The dimension of the convolutional kernels or filters in this first layer should be enough to capture interesting patterns but is upper limited by the computational time. In this work kernels from 3 to 8 were tried and  $k = 5$  was used because represents a compromise between length and computational load. The kernels are floating point vectors adjusted during training phase by the learning algorithm. Considering that the input vectors are binary, then an upper limit for the number of kernels  $n_1$  is due to the total number of configurations that can be obtained with  $k$  bits. If  $k = 5$ ,



then the kernels can be 31 at most (excluding the configuration with all zeros values). To maintain a manageable training time we choose to use a  $n_1 = 10$  kernel for the first stage. The second convolutional layer of the network has kernels of the same dimension ( $k = 4$ ) and  $n_2 = 20$ .

A CNN, like the one used in this work, is usually considered “deep” if compared with the commonly used multilayer perceptrons that usually have three layers (input, hidden, and output). If the input pattern is a vector  $x \in \mathbb{R}^L$  and the layer  $C1$  uses a set of  $n_1$  kernels  $w$  of dimension  $k$  ( $w \in \mathbb{R}^k$ ), the convolution output will be a set of  $n_1$  vectors  $f_i \in \mathbb{R}^{L-k+1}$ .

$$f_i = w_i \star x + b_i \quad i = 1, 2, \dots, n_1 \quad (1)$$

where  $\star$  indicates the convolution operator and  $b_i$  is an offset parameter. A logistic function is a non-linear function applied to the output in the proposed application. The output vectors are reduced using the max-pooling operation with a pool of dimension two so that the resulting output vectors will be  $(L - K + 1)/2$ . The max pooling layer compresses the input representation from  $C1$  layer and allows to obtain a more dense representation of the input data. The  $C2$  layer has the same structure but operates with a multi-dimensional input, the output of the  $C2$  layer is constituted by a set of  $n_2$  vectors  $f_j \quad j = 1, 2, \dots, n_2$  given by:

$$f_j = \sum_l w_{j,l} \star x_l + b_j \quad j = 1, 2, \dots, n_2. \quad (2)$$

The  $f_j$  vectors are rearranged in a single vector  $y$ , containing the features extracted from the input pattern. This vector is the input to a fully connected multi-layer perceptron,

with only one hidden layer. The whole network is trained using the stochastic gradient descent algorithm and is implemented in Python using the Theano framework [45, 46].

### Implementation details

According to the introduced pipeline, we integrated the following publicly available algorithms: Ipknot (release 0.0.2), MoSS (release 2.13) and Theano (release 0.8.2). The docker image is based on the operating system Linux Centos (release 7.2.1511). Java (release 1.8.0) and Python (release 2.7.5) were the languages used to implement the nRC tool.

## Results

In this section, we presented the classification results obtained by our classification pipeline. We performed two kinds of experiments: in the first one, we tested nRC tool using a ten-fold cross-validation scheme to find the best configuration in terms of a number of structural features and parameters of the CNN model. In the second one, we validated the best models obtained during the testing phase by considering an independent dataset, downloaded from Rfam database, and consisting of 2600 sequences, not used in the training phase, belonging to the same 13 ncRNA classes as the training dataset. That validation procedure assured us that there is not overfitting with regards to both feature extraction and the learning of the CNN. We introduced both the number of local features and the statistical measures used for testing procedures. Then, since we want to demonstrate that in the proposed pipeline a deep learning architecture can outperform standard classification techniques, we compared the CNN algorithm with 4 of the most known supervised classifiers. Moreover, to test our method against RNAcon tool, the state-of-the-art technique for classification of ncRNA sequences, we introduced an independent validation dataset. Finally, we discussed the obtained results.

### Testing procedures

To evaluate our method for classification of ncRNA sequences, we developed a testing procedure considering different values of the minimum and maximum size of the frequent subgraph fragments extracted by the MoSS tool. Each configuration of those parameters gave, in fact, a different number of structural features used by the classification algorithms. We considered five different configurations, with sub-fragment sizes ranging from two to six, because we are interested in considering local features. The chosen size produced a different number of input features: in particular, we obtained only a few features (about 250) up to many features (about 6000) with regards to the number of sequences in our dataset, i.e. 6320. The min and max size of the MoSS sub-graphs are from 2/4 to 3/6 and the corresponding number of features range from 250 to 6483 (Table 1).

Classification performances have been computed using a ten-fold cross-validation procedure in terms of accuracy, precision, sensitivity, specificity, F-score and MCC. These statistical measures are defined in Table 2.

### Comparison among CNN and other machine learning algorithms

Our proposed classifier based on DL architecture has been compared with four state-of-the-art feature-based algorithms: NB [34], RF [26], k nearest neighbour (kNN) [47] and support vector machine (SVM) [19]. All these algorithms were run using the Weka 3.6.13 platform [48]. As done for the CNN parameters, the one introduced in

**Table 1** Number of input features related to the min (m) and max (n) size of the frequent sub-structures extracted by the MoSS algorithm

MoSS parameters		Number of input features
m/n	Dataset support (%)	
2/4	10	250
4/5	10	1258
3/5	10	1298
4/6	10	6443
3/6	10	6483

“Classification with deep learning architecture” section, we made several trials with different parameter values to establish, for each classification algorithm, the configuration that gave the best performances in terms of evaluation criteria. In detail, on the default algorithm configurations in the Weka platform, we set the following parameters: NB with kernel estimator option, RF with 100 trees and seed = 10, kNN with  $K = 3$ , SVM with gamma = 0.01 and cost = 10. As regards the CNN, the kernel size is  $k = 5$  for both first and second layer; the pool size is 2 for both layers; the number of kernels is  $n_1 = 10$  for the first layer and  $n_2 = 20$  for the second layer. In the fully connected layer, the number of hidden units (columns of  $M_1$  and rows of  $M_2$ ) was 500. We did the first comparison to consider how the accuracy scores change according to the five different numbers of the input features (see Table 1). Our DL approach reaches the highest score of about 74.7% when considering the 6443 features (Fig. 6). The second best classifier is the SVM, with a max accuracy score of about 67.36% when considering 1258 features. The remaining three classifiers did not provide satisfying results.

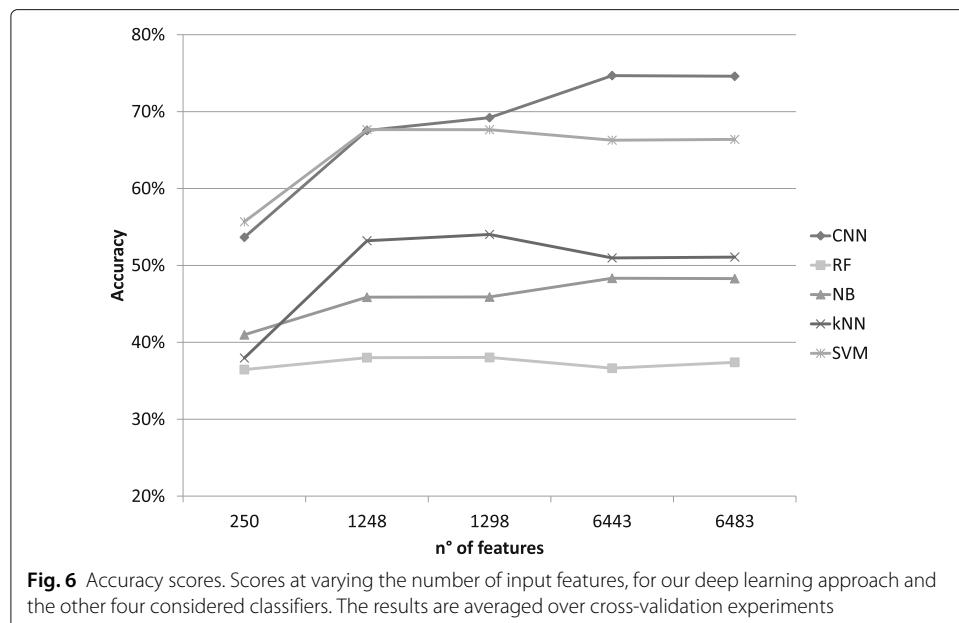
Considering the results obtained with all the classifier algorithms and all the performances indexes, we found that the CNN network results have the lowest standard deviation for all the measures (Table 3). Moreover, the value of all the performance indices increases with the number of input features (Table 4).

During the evaluation procedure, we also compared the execution time among the classification algorithms used in this study. As regards the training phase, the CNN algorithm is significantly more time consuming with respect to the other algorithms, as it is based on DL architecture. Fortunately, in the most case, the classification model is trained only once, so that users can exploit it for classifying new sequences. Conversely, as regards the testing phase, i.e. the classification of new sequences, the CNN is the second fastest

**Table 2** Statistical measures and their formulas used for evaluating the classifiers

Statistical measure	Formula
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Sensitivity	$\frac{TP}{TP+FP}$
Specificity	$\frac{TN}{TN+FP}$
Precision	$\frac{TP}{TP+FP}$
F-score	$\frac{2*TP}{2*TP+FP+FN}$
MCC	$\frac{TP*TN - FP*FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

TP are true positives, TN are true negatives, FP are false positives, FN are false negatives



algorithms just behind the RF. We report the average execution time taken to test classification models in Table 5. All experiments were carried out on a Windows 10 PC, with Intel i7 2.8 GHz CPU and 8 GB RAM.

#### Validation procedure

To further validate our proposed method we performed another classification experiment using an independent dataset whose elements have never been seen by the classifier during the learning phase. We downloaded the validation dataset from Rfam database and is composed of 2600 sequences belonging to the same 13 ncRNA classes as in the original dataset (200 sequences per class). To be more precise, we wanted to demonstrate that both the feature space of size 6443 and the CNN model, learned with the whole training dataset, can generalise the ncRNA class predictions, thus avoiding overfitting. To do that, we first predicted the secondary structure of validation sequences through IPknot, then we represented the sequences of the validation dataset in the same feature space created during the training phase (Fig. 7); finally we evaluated the best CNN model (see Table 4, fourth row) trained with the whole training dataset predicting the ncRNA classes of the

**Table 3** Comparison among five classification algorithms (Alg.) in terms of percentage scores (%) and standard deviations ( $\sigma$ ) of five statistical measures

Alg.	Accuracy		Sensitivity		Specificity		Precision		F-score		MCC	
	%	$\sigma$	%	$\sigma$	%	$\sigma$	%	$\sigma$	%	$\sigma$	%	$\sigma$
CNN	74.69	0.013	74.13	0.010	97.89	0.001	74.09	0.010	74.11	0.010	72.59	0.10
RF	56.60	1.544	56.50	1.539	96.41	0.001	55.69	0.019	56.10	0.037	52.4	0.16
NB	47.31	1.837	48.98	1.844	95.61	0.001	46.80	0.019	47.87	0.037	43.4	0.16
kNN	54.70	1.609	59.17	1.783	96.28	0.001	54.26	0.018	56.61	0.035	51.2	0.15
SVM	67.36	1.991	67.47	1.855	97.11	0.001	73.92	0.673	67.76	0.019	63.1	0.16

For each algorithm, the number of input features providing the best scores has been considered. The results are averaged over cross-validation experiments

**Table 4** Statistical measures for CNN algorithm classifications

Number of features	CNN statistical measures (%)					
	Accuracy	Sensitivity	Specificity	Precision	F-score	MCC
250	53.66	53.18	96.14	52.87	53.03	49.81
1258	67.55	68.58	97.44	68.30	68.44	64.85
1298	69.22	68.61	97.43	68.59	68.60	66.65
6443	74.69	74.13	97.89	74.09	74.11	72.59
6483	74.60	74.01	97.88	73.97	73.99	72.48

According to the MoSS parameter configurations, here we report five sets of features. The results are averaged over cross-validation experiments

validation dataset. The classification results confirm the robustness of the nRC tool with unknown data (Table 6).

#### Comparison between nRC and the RNAcon tool

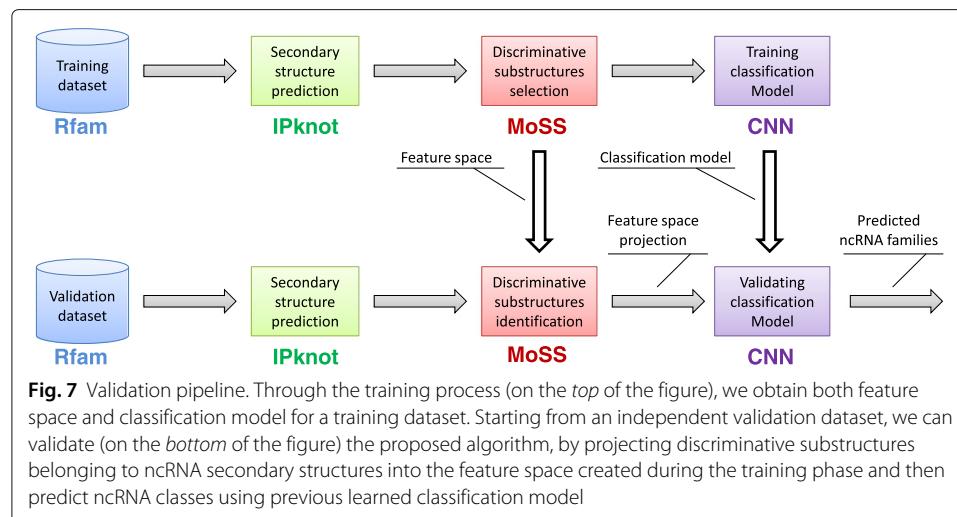
As explained at the end of the “Background” section, the RNAcon tool is the reference classifier of ncRNA sequences that consider structural features and machine learning algorithms. In particular, RNAcon extracts 20 local and global graph properties from the ncRNA predicted secondary structures, and it makes classification using the RF algorithm. Because our proposed method also considers structural features, the frequent sub-graphs, a machine learning classifier, i.e. the DL convolutional network, we made a direct comparison of our results with the ones provided by RNAcon. We used the RNAcon web service available at <http://crdd.osdd.net/raghava/rnacon/>, and we made the comparison considering the validation dataset because it represents an independent dataset for both tools. In particular, we removed from the validation dataset the sequences belonging to the scaRNA class, obtaining this way a set of 2400 sequences, because they are not present in the training dataset of RNACon. Our method outperforms RNAcon, doubling its performances according to accuracy and sensitivity scores when 6443 input features are considered (Fig. 8).

#### Discussion

Because it has been proved that structural properties of the secondary structure of RNA molecules can provide specific information of the biological function of different ncRNA classes [27], we presented a classifier that works on a feature set representing frequent fragments of the RNA molecular structure. That representation, coupled with a classifier based on a DL architecture, allowed us to obtain the best scores when compared to other machine learning algorithms and the RNAcon tool. To analyse in detail the performances of our method, we produced the confusion matrix (Fig. 9), so that it is possible to inspect

**Table 5** Comparison among average execution times of classification models, during the ten-fold cross-validation procedure

Algorithm	Execution time (seconds)
CNN	17.33
NB	25.97
KNN	29.36
RF	2.93
SVM	21.46



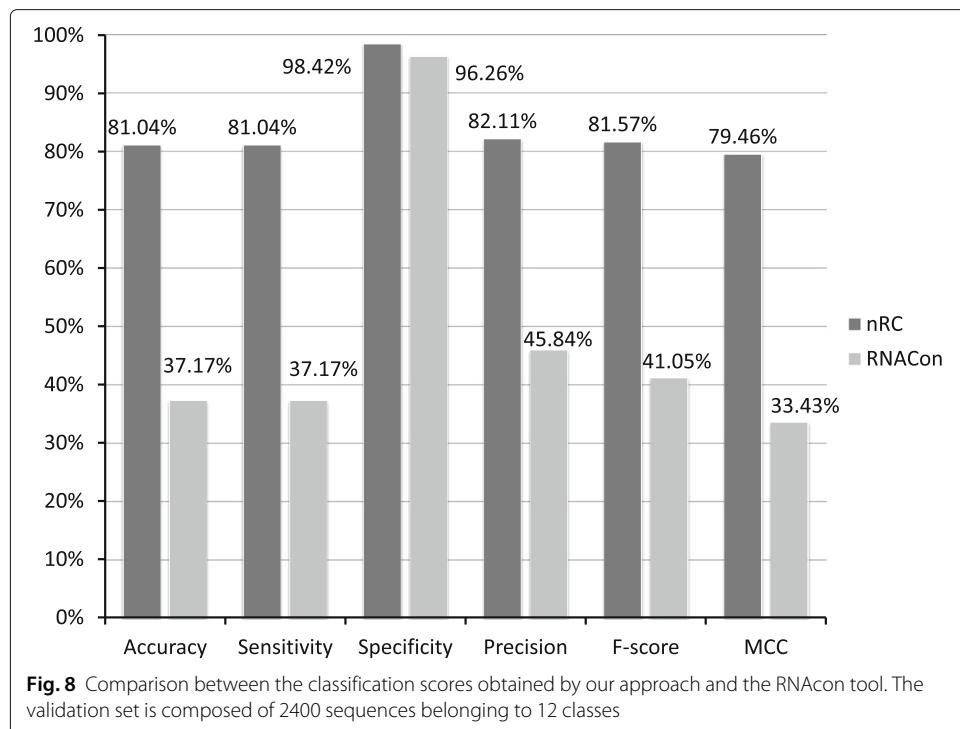
which ncRNA classes our approach better predicted. That confusion matrix has been obtained putting together the single confusion matrices produced at the end of each fold during the testing procedure. For example, we noticed that Intron\_gpI and Intron\_gpII classes are predicted with a sensitivity score of about 95%, whereas miRNA, IRES and HACA-box classes reached sensitivity and precision score of about 50%. We highlighted in red some situations that will need further investigation in the future. For example, the most misclassified miRNAs (9%) are predicted as HACA-box, correspondingly, 8.4% of HACA-box are predicted as miRNA. The same situation happened to the scaRNA class, with 11.4% misclassified as HACA-box, which in turn is predicted as scaRNA in 13% of cases.

As evidence, there is a misrepresentation of some ncRNA classes. As for the CD-box and HACA-box, both classes belong to the same main class group, i.e. they all are snoRNA (see Fig. 2). Even though they have a different global secondary structure, they could share local sub-structures, in fact, they are both involved in the chemical modification of the RNA classes rRNA, tRNA and snRNA after transcription, hypothesising a common link between their function and their structural sub-features. In particular, CD-box RNAs guide methylation events and HACA-box RNAs guide pseudouridylation of the RNA target [49]. Another RNA class belonging to snoRNA class is scaRNAs. The scaRNAs are involved in the modification of RNA polymerase II transcribed spliceosomal RNAs, and they are also defined as composite HACA- and CD-box RNAs, because their conserved domains are the typical motifs of both HACA-box and CD-box [50]. Moreover, similar sub-structures could be found in both miRNAs and some snoRNA, since recent reports have indicated that, despite the differences in size and secondary structure, a human snoRNA and a protozoan snoRNA are associated with Argonautes, processed into small RNAs, and can function as miRNAs [51, 52].

**Table 6** Statistical measures for nRC classification during the validation procedure

Number of features	CNN statistical measures					
	Accuracy	Sensitivity	Specificity	Precision	F-score	MCC
6443	81.81%	81.81%	98.48%	81.50%	81.66%	80.29%

It has been used an independent dataset composed of 2600 sequences belonging to 13 ncRNA classes



As mentioned before, to confirm there is not overfitting with regards to both feature extraction and the learning of the CNN, another confusion matrix (Fig. 10), has also been computed for the experiment with the independent validation dataset. Once again, we noticed the same behaviour as in the previous case, with a similar trend with regards to classification mistakes, such as the miRNA-snoRNA (CD-box and HACA-box) and CD-box-HACA-box misclassifications.

All these evidence let us hypothesise that all these classes of ncRNAs have some shared sub-features on the other analysed ncRNA classes. Because our approach considers these sub-structures as local features, the misclassification among some of the ncRNA classes could be explained by those shared features. Concluding, therefore, in spite of the overall good performances of our classification approach, we need to carry out some further analysis for the ncRNA classes whose sensitivity and a precision score was about 50%. A deeper investigation would allow us to increase the classification scores and to try understanding if and what are the relations between RNA sub-classes, considering, for example, global features as well.

## Conclusions

In this work, we introduce nRC (noncoding RNA Classifier), a new tool for the classification of non-coding RNA sequences. Three steps are the basis of the proposed method: the prediction of ncRNAs secondary structures, the extraction of frequent sub-structures as features and the classification of known ncRNA classes. To implement these processes, we used the IPknot algorithm to predict RNA secondary structures with pseudoknots, the MoSS decision tree pruning algorithm to obtain sub-structures, and a deep learning network architecture, namely a convolutional neural network, as a supervised classifier. Differently to other existing ncRNA classification approaches,

		Reference														Sensitivity
Prediction		5_8S_rRNA	5S_rRNA	CD-box	HACA-box	Intron_gpl	Intron_gplI	IRES	leader	miRNA	riboswitch	ribozyme	scaRNA	tRNA		
5_8S_rRNA	430	7	9	13	1	2	7	10	11	4	4	4	4	4	0.86	
5S_rRNA	1	426	14	21	0	2	8	2	9	10	3	6	6	6	0.88	
CD-box	16	12	333	44	0	3	8	14	53	7	10	24	15	15	0.67	
HACA-box	9	11	31	237	2	1	29	13	42	9	10	65	6	6	0.47	
Intron_gpl	0	0	1	3	479	0	5	2	1	5	22	6	0	0	0.97	
Intron_gplI	4	1	1	2	0	477	2	3	10	6	2	3	6	6	0.95	
IRES	2	5	12	21	0	2	167	4	7	14	10	21	4	4	0.52	
leader	8	3	15	23	2	3	11	399	29	17	2	4	8	8	0.81	
miRNA	17	12	33	45	0	4	10	22	249	16	5	15	23	23	0.51	
riboswitch	4	9	8	11	1	0	16	11	23	382	13	6	9	9	0.76	
ribozyme	3	2	6	10	6	0	22	3	5	10	394	15	5	5	0.79	
scaRNA	1	6	18	57	5	0	27	8	26	9	17	328	6	6	0.64	
tRNA	3	5	19	11	0	3	8	9	32	11	8	1	405	405	0.79	
Precision	0.83	0.84	0.59	0.50	0.91	0.91	0.62	0.78	0.57	0.75	0.84	0.66	0.66	0.78		

**Fig. 9** Confusion matrix related to the predictions of nRC tool over the training dataset (6320 sequences). It has been obtained putting together the single confusion matrices produced at the end of each fold during the testing procedure. Classification of Intron gpl and Intron gplI classes gave the best result; classification of IRES, miRNA and HACA-box classes gave the worst results. In red we highlighted some situations, some of which discussed in Discussion section, that will need further investigation

we (i) created a ncRNAs vs. local topological features Boolean matrix as input data and (ii) adopted a DL architecture for classification. To demonstrate the effectiveness of the proposed approach, we first compared the proposed classifier with four of the most well-known classification algorithms, i.e. RF, NB, kNN and SVM, and then we compared our method with the RNAcon tool, that is the literature reference classifier of ncRNA sequences. Experiments have also been carried out using an independent validation dataset. In both tests, we demonstrated the advantages of using our approach on other strategies, obtaining the highest scores in terms of five different statistical measures, i.e. accuracy, sensitivity, specificity, precision, F-score and MCC. In particular, results demonstrated the proposed method outperformed the state-of-the-art RNAcon approach, doubling its performance in terms of accuracy and sensitivity.

As future work, we are working to train a classification model with much more ncRNA sequences, also belonging to some other well studied ncRNA classes, such as piwi-interacting RNA (piRNA) [53] and circular RNA (circRNA) [54]. In addition,

	Reference														Sensitivity
Prediction	5_8S_rRNA	5S_rRNA	CD-box	HACA-box	Intron_gpl	Intron_gpli	IRES	leader	miRNA	riboswitch	ribozyme	scaRNA	tRNA		
5_8S_rRNA	184	0	3	4	1	0	0	7	2	4	1	0	1		0.92
5S_rRNA	3	178	6	7	0	0	1	1	5	3	1	1	3		0.89
CD-box	2	5	141	12	0	1	2	4	20	3	10	2	3		0.71
HACA-box	1	4	15	120	0	0	10	7	24	6	0	6	2		0.6
Intron_gpl	0	0	0	1	196	0	2	0	0	1	4	1	0		0.98
Intron_gpli	1	2	2	2	0	194	0	3	3	3	0	1	5		0.97
IRES	2	0	1	7	2	0	164	3	4	2	1	2	0		0.82
leader	1	1	7	3	0	0	3	164	11	2	0	3	0		0.82
miRNA	1	1	8	17	0	0	2	9	98	8	3	2	4		0.49
riboswitch	3	4	2	6	0	3	4	1	8	158	4	0	0		0.79
ribozyme	0	0	4	1	1	1	5	0	1	1	168	0	0		0.84
scaRNA	1	0	4	15	0	0	6	0	9	2	3	182	2		0.91
tRNA	1	5	7	5	0	1	1	1	15	7	5	0	180		0.9
Precision	0.89	0.85	0.69	0.62	0.96	0.9	0.87	0.84	0.64	0.82	0.92	0.81	0.79		

**Fig. 10** Confusion matrix related to the predictions of nRC tool over the validation dataset (2600 sequences). We noticed the same behaviour as in the confusion matrix shown in Fig. 9, with a similar trend with regards to classification mistakes, highlighted in red

to improve classification performances, we are planning to test some new secondary structure prediction tools, like those proposed in [55, 56]. Finally, we aim at creating a publicly available web service for the classification of unlabelled non-coding RNA sequences.

#### Abbreviations

CNN: Convolutional neural network; DL: Deep learning; IRES: Internal ribosome entry site; kNN: K nearest neighbour; lncRNA: Long non-coding RNA; miRNA: MicroRNA; MoSS: Molecular sub-structure miner; NB: Naive Bayes; ncRNA: Non-coding RNA; nRC: Non-coding RNA classifier; RF: Random forest; rRNA: Ribosomal RNA; tRNA: Transfer RNA; siRNA: Silencing RNA; snoRNA: Small nucleolar RNA; snoRNP: Small nucleolar ribonucleoproteins; snRNA: Small nuclear RNA; SVM: Support vector machine

#### Acknowledgments

Not applicable.

#### Funding

The publication costs for this article were funded by the CNR Interomics Flagship Project CUP B81J12000980001 “Development of an integrated platform for the application of “omic” sciences to biomarker definition and theranostic, predictive and diagnostic profiles”.

#### Availability of data and materials

The source code of nRC tool is freely available at <https://github.com/IcarPA-TBlab/nrc>, moreover, since the proposed tool requires several dependencies to be installed, a docker image has been released at <https://hub.docker.com/r/tblab/nrc/>. The datasets generated and analysed during the current study are available in the TBLAB repository, [http://tblab.pa.icar.cnr.it/public/nRC/paper\\_dataset/](http://tblab.pa.icar.cnr.it/public/nRC/paper_dataset/).

**Authors' contributions**

AF: project conception, implementation, experimental tests, discussions, assessment, writing. MLR: project conception, implementation, experimental tests, discussions, assessment, writing. LLP: project conception, writing, assessment, discussions. RR: project conception, implementation, discussions, assessment, writing. AU: project conception, discussions, assessment, writing, funding. All authors read and approved the final manuscript.

**Ethics approval and consent to participate**

Not applicable.

**Consent for publication**

Not applicable.

**Competing interests**

The authors declare that they have no competing interests.

**Publisher's Note**

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 5 April 2017 Accepted: 24 July 2017

Published online: 01 August 2017

**References**

1. Esteller M. Non-coding RNAs in human disease. *Nat Rev Genet*. 2011;12(12):861–74.
2. Veneziano D, Nigita G, Ferro A. Computational approaches for the analysis of ncRNA through deep sequencing techniques. *Front Bioeng Biotechnol*. 2015;3:1–6.
3. Mattick JS. The genetic signatures of noncoding RNAs. *PLoS Genet*. 2009;5(4):1000459.
4. Cestarelli V, Fiscon G, Felici G, Bertolazzi P, Weitschek E. Camur: Knowledge extraction from rna-seq cancer data through equivalent classification rules. *Bioinformatics*. 2016;32(5):697.
5. Weitschek E, Felici G, Bertolazzi P. Mala: A microarray clustering and classification software. In: 2012 23rd International Workshop on Database and Expert Systems Applications. New York: IEEE; 2012. p. 201–5.
6. Sonenberg N, Hinnebusch AG. Regulation of Translation Initiation in Eukaryotes: Mechanisms and Biological Targets. *Cell*. 2009;136(4):731–45.
7. Du T, Zamore PD. microPrimer: the biogenesis and function of microRNA. *Dev (Cambridge, England)*. 2005;132(21):4645–52.
8. O'Day E, Lal A. MicroRNAs and their target gene networks in breast cancer. *Breast Cancer Res BCR*. 2010;12(2):201.
9. Obad S, dos Santos CO, Petri A, Heidenblad M, Broom O, Others. OncomiR addiction in an *in vivo* model of microRNA-21-induced pre-B-cell lymphoma. *Nat Genet*. 2011;43:371–8.
10. Croce CM. Causes and consequences of microRNA dysregulation in cancer. *Nat Rev Genet*. 2009;10(10):704–14.
11. Kulkarni V, Nagvi A, Uttamani J, Nares S. MiRNA-Target Interaction Reveals Cell-Specific Post-Transcriptional Regulation in Mammalian Cell Lines. *Int J Mol Sci*. 2016;17(1):72.
12. Shenouda SK, Alahari SK. MicroRNA function in cancer: oncogene or a tumor suppressor? *Cancer Metastasis Rev*. 2009;28(3–4):369–78.
13. Sun X, Du P, Yuan W, Du Z, Yu M, Yu X, Hu T. Long non-coding RNA HOTAIR regulates cyclin J via inhibition of microRNA-205 expression in bladder cancer. *Cell Death Dis*. 2015;6(10):1907.
14. Dong XY, Guo P, Boyd J, Sun X, Li Q, Zhou W, Dong JT. Implication of snoRNA U50 in human breast cancer. *J Gen Genomics*. 2009;36(8):447–54.
15. Liao J, Yu L, Mei Y, Guarnera M, Shen J, Li R, Liu Z, Jiang F. Small nucleolar RNA signatures as biomarkers for non-small-cell lung cancer. *Mol Cancer*. 2010;9(1):1.
16. Gupta RA, Shah N, Wang KC, Kim J, Horlings HM, Wong DJ, Tsai MC, Hung T, Argani P, Rinn JL, Others. Long non-coding RNA HOTAIR reprograms chromatin state to promote cancer metastasis. *Nature*. 2010;464(7291):1071–6.
17. Quinn JJ, Chang HY. Unique features of long non-coding RNA biogenesis and function. *Nat Rev Genet*. 2016;17(1):47–62.
18. Breaker RR. Riboswitches and the RNA world. *Cold Spring Harb Perspect Biol*. 2012;4(2):003566.
19. Scholkopf B, Smola AJ. Learning with Kernels. Cambridge: MIT Press; 2002.
20. Liu J, Gough J, Rost B. Distinguishing protein-coding from non-coding RNAs through support vector machines. *PLoS Genet*. 2006;2(4):29.
21. Kong L, Zhang Y, Ye ZQ, Liu XQ, Zhao SQ, Wei L, Gao G. CPC: assess the protein-coding potential of transcripts using sequence features and support vector machine. *Nucleic Acids Res*. 2007;35(suppl 2):345–9.
22. Lertampaiporn S, Thammarongtham C, Nukoolkit C, Kaewkamnerpong B, Ruengjitchawalya M. Identification of non-coding RNAs with a new composite feature in the Hybrid Random Forest Ensemble algorithm. *Nucleic Acids Res*. 2014;42(11):93–3.
23. Yuan C, Sun Y. RNA-code: a noncoding RNA classification tool for short reads in NGS data lacking reference genomes. *PLoS one*. 2013;8(10):77596.
24. Panwar B, Arora A, Raghava GP. Prediction and classification of ncRNAs using structural information. *BMC Genomics*. 2014;15(1):127.
25. Childs L, Nikolski Z, May P, Walther D. Identification and classification of ncRNA molecules using graph properties. *Nucleic Acids Res*. 2009;37(9):66.
26. Breiman L. Random forests. *Mach Learn*. 2001;45:5–32. dx.doi.org/10.1023%2FA%3A1010933404324.
27. Mathews DH, Turner DH. Prediction of RNA secondary structure by free energy minimization. *Curr Opin Struct Biol*. 2006;16(3):270–8.
28. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*. 2015;521(7553):436–44.

29. Min S, Lee B, Yoon S. Deep learning in bioinformatics. *Brief Bioinform.* 2016;1–19.
30. Chicco D, Sadowski P, Baldi P. Deep autoencoder neural networks for gene ontology annotation predictions. In: Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics. BCB '14. New York: ACM; 2014. p. 533–40. <http://doi.acm.org/10.1145/2649387.2649442>.
31. Alipanahi B, Delong A, Weirauch MT, Frey BJ. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat Biotechnol.* 2015;33(8):831–8.
32. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *IEEE Proc.* 1998;86(11):2278–324.
33. Rizzo R, Fiannaca A, La Rosa M, Urso A. A Deep Learning Approach to DNA Sequence Classification. In: Computational Intelligence Methods for Bioinformatics and Biostatistics, Lecture Notes in Computer Science, vol. 9874. Cham (ZG): Springer International Publishing; 2016. p. 129–40.
34. John GHG, Langley P. Estimating Continuous Distributions in Bayesian Classifiers. In: Besnard P, Hanks S, editors. Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence. Montreal, Quebec, Canada, vol. 1. San Francisco: Morgan Kaufmann; 1995. p. 338–45. 1302.4964.
35. Nawrocki EP, Burge SW, Bateman A, Daub J, Eberhardt RY, Eddy SR, Floden EW, Gardner PP, Jones TA, Tate J, Finn RD. Rfam 12.0: updates to the RNA families database. *Nucleic Acids Res.* 2015;43(D1):130–7.
36. Fu L, Niu B, Zhu Z, Wu S, Li W. CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics.* 2012;28(23):3150–2.
37. Sato K, Kato Y, Hamada M, Akutsu T, Asai K. IPknot: fast and accurate prediction of RNA secondary structures with pseudoknots using integer programming. *Bioinformatics.* 2011;27(13):85–93.
38. Jabbari H, Condon A. A fast and robust iterative algorithm for prediction of rna pseudoknotted secondary structures. *BMC Bioinforma.* 2014;15(1):147.
39. Sperschneider J, Datta A. Dotknot: pseudoknot prediction using the probability dot plot under a refined energy model. *Nucleic Acids Res.* 2010;38(7):103.
40. Borgelt C, Meini T, Berthold M. MoSS: a program for molecular substructure mining. In: Proceedings of the 1st International Workshop on Open Source Data Mining Frequent Pattern Mining Implementations - OSDM '05. New York: ACM Press; 2005. p. 6–15.
41. Di Fatta G, Fiannaca A, Rizzo R, Urso A, Berthold M, Gaglio S. Context-Aware Visual Exploration of Molecular Databases. In: Sixth IEEE International Conference on Data Mining - Workshops (ICDMW'06). New York: IEEE; 2006. p. 136–41.
42. Fiannaca A, La Rosa M, Di Fatta G, Gaglio S, Rizzo R, Urso A. The BioDICE Taverna plugin for clustering and visualization of biological data: a workflow for molecular compounds exploration. *J Cheminformatics.* 2014;6(1):24.
43. Coates A, Ng AY, Lee H. An analysis of single-layer networks in unsupervised feature learning. In: International Conference on Artificial Intelligence and Statistics. 2011. p. 215–23.
44. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. *Science.* 2006;313(5786):504–7.
45. Bastien F, Lamblin P, Pascanu R, Bergstra J, Goodfellow IJ, Bergeron A, Bouchard N, Bengio Y. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop. 2012.
46. Bergstra J, Breuleux O, Bastien F, Lamblin P, Pascanu R, Desjardins G, Turian J, Warde-Farley D, Bengio Y. Theano: a CPU and GPU math expression compiler. In: Proceedings of the Python for Scientific Computing Conference (SciPy). 2010. Oral Presentation.
47. Cover T, Hart P. Nearest neighbor pattern classification. *IEEE Trans Inf Theory.* 1967;13(1):21–7.
48. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA data mining software. *ACM SIGKDD Explor Newslett.* 2009;11(1):10–18.
49. Matera AG, Terns RM, Terns MP. Non-coding RNAs: lessons from the small nuclear and small nucleolar RNAs. *Nat Rev Mol Cell Biol.* 2007;8(3):209–20.
50. Darzacq X, Jády BE, Verheggen C, Kiss AM, Bertrand E, Kiss T. Cajal body-specific small nuclear RNAs: a novel class of 2'-O-methylation and pseudouridylation guide RNAs. *EMBO J.* 2002;21(11):2746–56.
51. Ender C, Krek A, Friedländer MR, Beitzinger M, Weinmann L, Chen W, Pfeffer S, Rajewsky N, Meister G. A human snoRNA with microRNA-like functions. *Mol cell.* 2008;32(4):519–28.
52. Saraiya AA, Wang CC. snoRNA, a Novel Precursor of microRNA in Giardia lamblia. *PLoS Pathog.* 2008;4(11):1000224.
53. Ng KW, Anderson C, Marshall EA, Minatel BC, Enfield KS, Saprunoff HL, Lam WL, Martinez VD. Piwi-interacting rnas in cancer: emerging functions and clinical utility. *Mol Cancer.* 2016;15(1):5.
54. Lasda E, Parker R. Circular rnas: diversity of form and function. *Rna.* 2014;20(12):1829–42.
55. Fiscon G, Paci P, Iannello G. Monster v1.1: a tool to extract and search for rna non-branching structures. *BMC Genomics.* 2015;16(6):1.
56. Fiscon G, Iannello G, Paci P. A perspective on the algorithms predicting and evaluating the rna secondary structure. *J Genet Genome Res.* 2016;3(1):1–7.