ncRNA Classification with Graph Convolutional Networks

Emanuele Rossi
University of Cambridge
er513@cam.ac.uk

Federico Monti
USI Lugano
federico.monti@usi.ch

Michael Bronstein
Imperial College London
USI Lugano
m.bronstein@imperial.ac.uk

Pietro Liò
University of Cambridge
pl219@cam.ac.uk

ABSTRACT

Non-coding RNA (ncRNA) are RNA sequences which don't code for a gene but instead carry important biological functions. The task of ncRNA classification consists in classifying a given ncRNA sequence into its family. While it has been shown that the graph structure of an ncRNA sequence folding is of great importance for the prediction of its family, current methods make use of machine learning classifiers on hand-crafted graph features. We improve on the state-of-the-art for this task with a graph convolutional network model which achieves an accuracy of 85.73% and an F1-score of 85.61% over 13 classes. Moreover, our model learns in an end-to-end fashion from the raw RNA graphs and removes the need for expensive feature extraction. To the best of our knowledge, this also represents the first successful application of graph convolutional networks to RNA folding data.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**.

KEYWORDS

RNA, graph convolutional networks, ncRNA classification

ACM Reference Format:
Emanuele Rossi, Federico Monti, Michael Bronstein, and Pietro Liò. 2019. ncRNA Classification with Graph Convolutional Networks. In *Proceedings of DLG@KDD workshop (DLG@KDD '19)* . ACM, New York, NY, USA, 5 pages.

DLG@KDD '19, August 2019, Anchorage, Alaska, US

ACM ISBN 978-1-4503-9999-9/18/06.

1 INTRODUCTION AND RELATED WORK

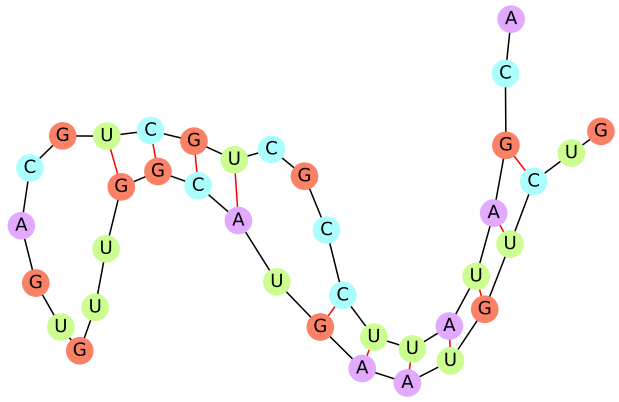

Figure 1: Graph representing the folding of RNA sequence *ACGAUAUUCCGCUGCUGCAGUGUUGGCAUGAAUGUCUG*. Generated using the ViennaRNA package [7]. Black edges represent phosphodiester bonds and red edges represent hydrogen bonds.

ncRNA

RNA, together with DNA, is one of the fundamental carriers of genetic information. While the main function of RNA is in the production of proteins from instructions present in DNA code, RNA has been shown to also carry other important biological functions. In particular, recent findings on cancer research have shifted the attention from protein-coding RNAs to non-coding RNAs, as principal effectors and regulators of tumorigenesis and cancer development [6, 17, 20, 21]. Moreover, certain RNAs have been shown to impact gene expression through fulfilling roles encompassing sensory and scaffolding capacities at various stages of the gene regulation process [13]. We refer to this functional RNA, which is transcribed from DNA but not translated into proteins, as non-protein coding RNA (ncRNA).

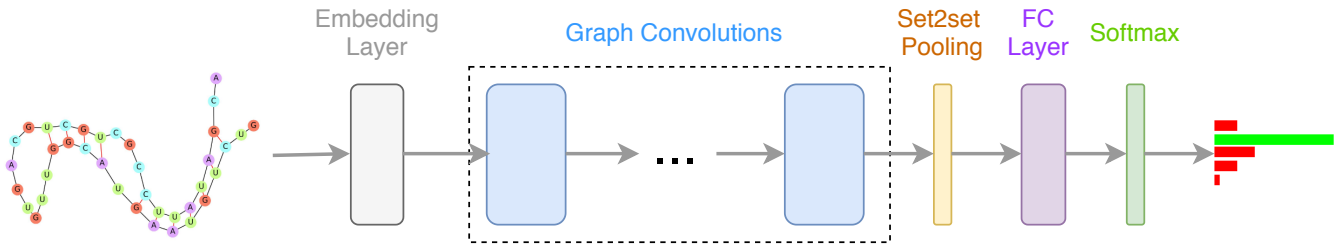

Figure 2: Diagram representing the architecture of our *RNAGCN* model. The input consists of an RNA graph with a nucleotide on each node. Initially, an embedding layer maps each nucleotide into a continuous vector. After that, multiple graph convolutional layers are used to refine the features on each node by propagating information in the graph. The set2set pooling model is then used to aggregate the relevant information from the output of the last convolutional layer and produce a graph-wise representation. A fully-connected layer with softmax activation finally produces the output probability for each class.

At its most basic form, RNA is a sequence of four types of nucleotides: *adenine* (A), *guanine* (G), *cytosine* (C) and *uracil* (U). The sequence of nucleotides forms what is commonly called the RNA *primary structure*. However, it is the folding of the RNA sequence into its *secondary structure* which is more related to its function. The folding is generated by hydrogen bonds between complementary base pairs. The most common occurring base pairs are A-U and G-C, also called Watson-Crick base pairs. However, RNA sometimes presents hydrogen bonds between different bases. All base pairs which do not follow Watson-Crick rules are called Wobble base pairs. Figure 1 shows the graph representing the folding of a short ncRNA sequence. We can observe both Watson-Crick pairs and Wobble pairs (G-U). Black edges represent phosphodiester bonds (between adjacent bases in the original sequence) while red edges represent hydrogen bonds.

ncRNA Classification

A wide variety of different classes, or families, of ncRNA have been identified, which differ by function and structure. Since the identification of drugs targeting the regulatory circuits of ncRNA depends on knowing its family, there has been an increasing interest in the development of methods for ncRNA classification. More traditional methods, such as *RNA-CODE* [22], are based on alignment strategies. Other methods, such as *RNAcon* [15] and *GraPPLe* [3], use standard machine learning classifiers on manually extracted graph properties of the RNA secondary structure. These approaches have shown that graph properties (both local and global) reflect the functional information of different classes of RNAs and are therefore informative for the classification.

More recently, *nRC* [4] uses a convolutional neural network on graph features extracted using MoSS [1], which finds frequent local sub-structures in a set of graphs. In particular, MoSS is used to extract up to 6483 binary features for each input graph, where each feature represents the presence or absence of a particular sub-structure. To the best of our knowledge, *nRC* represents the state-of-the-art approach for ncRNA classification.

Graph Convolutional Networks

Deep learning has recently had a remarkable impact on multiple domains, including natural language processing and computer vision [11]. However, most of popular deep neural models, such as convolutional neural networks (CNNs) [12], only work on grid-structured (Euclidean) data, and are not directly applicable to graphs. For this reason, *nRC* [4] first extracts features from the ncRNA graphs before applying a CNN.

Recently, there has been growing interest in extending deep learning techniques to non-Euclidean data, including graphs [2]. Several models for deep learning on graphs have been developed in the past few years, including graph convolution [10], graph attention [18], mixture models [14] and neural message passing [5].

Our Contribution

We are the first to apply graph convolutional networks on RNA folding data, achieving state-of-the-art results on the task of ncRNA classification with an accuracy of *85.73%* and an F1-score of *85.61%* over 13 classes. Our model is aware of different bond types and uses attention to aggregate information from the most important nodes for the final classification task. Moreover, since it learns directly from the RNA graphs, it removes the need for manual features extraction.

2 BACKGROUND

Most graph convolutional networks model can be interpreted as following a standard framework of *message passing* [5]. In particular, at each layer, the features of a node are updated by aggregating messages from its neighbours. Given a graph

G, with node features x_v and edge features e_{vw}, the update at layer $t+1$ takes the form:

$$m_v^{t+1} = \sum_{w \in N(v)} M_t(x_v^t, x_w^t, e_{vw}) \quad (1)$$

$$x_v^{t+1} = U_t(x_v^t, m_v^{t+1}) \quad (2)$$

where M_t is a learnable function which computes the message from node w to node v, $N(v)$ represents the neighbours of v in the graph, m_v^{t+1} represents the aggregation of all incoming messages for node v, and U_t is a learnable function which updates the features for node v given its previous features and the incoming aggregated message.

In graph classification problems, it is also necessary to produce a global graph representation by aggregating the final features for all nodes. This operation is called *global pooling* and can be defined as:

$$\hat{y} = R(x_v^T | v \in G) \quad (3)$$

where R is a learnable function which is permutation invariant with respect to the order of nodes, and x_v^T represents the features of node v after the last convolutional layer.

3 PROPOSED METHOD

Our model is shown in figure 2. It takes as input a graph corresponding to a folded ncRNA sequence. Mathematically, the ncRNA classification task takes the form of a prediction on a graph G with node features x_v and edge features e_{vw}. In particular, x_v is just a one-hot representation of the nucleotide of node v, and e_{vw} is a one-hot representation of the edge type (either hydrogen bond or phosphodiester bond).

The model consists of one embedding layer which maps each nucleotide to a continuous vector representation, followed by a sequence of graph convolutional layers. In particular, we use a layer similar to the one used in [5], which is able to propagate information differently based on the edge type. Our convolutional layers take the form:

$$x_v' = \text{LeakyReLU}(W x_v + \sum_{w \in N(v)} A(e_{vw}) x_w) \quad (4)$$

where A is a 2-layer multilayer perceptron with LeakyReLU as non-linearity, which produces a projection matrix from edge features e_{vw}. Since our edge features e_{vw} are one-hot encodings, this amounts to learning a different projection matrix for each edge type, allowing the model to spread information differently based on the bond between two nodes. Lastly, W is a matrix of learnable weights.

After the last convolutional layer, a global pooling mechanism is used to obtain a single representation for the whole graph. In particular, we experimented with both the simple sum operator and the more advanced Set2Set model [19],

Dataset	*#Graphs*	*Avg. #Nodes*	*Avg. #Edges*	*#Classes*
train	5670	162.02	210.46	13
val	650	163.30	212.12	13
test13	2600	149.15	193.25	13
test12	2400	147.52	191.13	12

Table 1: Summary of datasets statistics.

which is a permutation invariant global pooling operator based on iterative content-based attention:

$$q_t = LSTM(q_{t-1}^*) \quad (5)$$

$$\tilde{\alpha}_{v,t} = x_v^\intercal q_t \quad (6)$$

$$\alpha_{v,t} = \frac{exp(\tilde{\alpha}_{v,t})}{\sum_{w=1}^{N} exp(\tilde{\alpha}_{w,t})} \quad (7)$$

$$r_t = \sum_{v=1}^{N} \alpha_{v,t} x_v \quad (8)$$

$$q_t^* = q_t \| r_t \quad (9)$$

where q_0 is the zero vector, and q_T^* is the final representation of the graph. At each step t, the output of the LSTM is used to compute attention scores $\alpha_{v,t}$ over all nodes. The new input to the LSTM is the concatenation of the old input with a weighted average of the nodes' features, where the weights are given by the attention scores. We use $T = 10$ steps and 1 layer for the LSTM. The size of the hidden state is the same as the number of output features from the last convolutional layer.

A fully connected layer with softmax activation is finally used to produce the probabilities for each of the 13 ncRNA classes.

Each convolutional layer is followed by batch norm [8]. Dropout [16] has also been used to regularize the model.

4 EXPERIMENTS

Dataset

We used the datasets introduced in [4], which consist of a training dataset of 6320 ncRNA sequences and a test dataset of 2600 sequences. Both dataset contain sequences from 13 different ncRNA classes: *miRNA*, *5S rRNA*, *5.8S rRNA*, *ribozymes*, *CD-box*, *HACA-box*, *scaRNA*, *tRNA*, *Intron gpI*, *Intron gpII*, *IRES*, *leader* and *riboswitch*. While the test dataset is perfectly balanced, the training dataset contains only 320 sequences from the *IRES* class, compared to 500 sequences for all other classes.

In line with [4], we also report results on a different test dataset, obtained by removing all sequences belonging to the *scaRNA* class from the original test dataset. This allows for a comparison with *RNACon* [15], whose publicly available

Metric	*Formula*
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Sensitivity	$\frac{TP}{TP+FN}$
Specificity	$\frac{TN}{TN+FP}$
Precision	$\frac{TP}{TP+FP}$
F1-Score	$\frac{2*TP}{2*TP+FP+FN}$
MCC	$\frac{TP*TN-FP*FN}{\sqrt{(TF+FP)(TP+FN)(TN+FP)(TN+FN)}}$

Table 2: Definition of the metrics used for the evaluation of the models.

Hyperparameter	*Values*
Num. of conv. layers	3, 4, 5, 6, 7
Conv. layers hidden dimension	40, 60, 80, 100
Global pooling type	sum, set2set
Dropout rate	0, 0.1, 0.2, 0.5

Table 3: Hyperparameters of the model which have been tuned, together with the values we tried for each one of them.

model was not trained on the *scaRNA* class. We refer to the original test dataset with 13 classes as *test13* and to the reduced test dataset with 12 classes as *test12*.

In order to tune our model, we further split the original training dataset in two: a validation set with 650 sequences (50 from each class) and a training set with the remaining 5670 sequences. The statistics of the final splits are shown in table 1.

For each sequence, we generate the corresponding folding graph using the ViennaRNA [7] package.

Experimental Setting

The hyperparameters of the model have been tuned on the held-out validation set using early stopping with a patience of 30 epochs. The hyperparameters tuned, together with the values tried for each of them, are described in table 3. For the optimization of the model, we used Adam [9] with a learning rate of 0.0004.

The best performing model on the validation set consists of 5 convolutional layers of dimension 80, the set2set model for global pooling, and uses a dropout rate of 0.1 for regularization.

For the evaluation of the model, we use the same metrics as in [4], which we define in table 2.

Results

We first tested our method on the independent test set with 13 classes (*test13*). The results are shown in the top part of table 4. While *nRC* obtains an accuracy of *81.81%*, our model outperforms it with an accuracy of *85.73%*. We also observe similar improvements in all other metrics.

The bottom part of table 4 shows instead the results on the test dataset with only 12 classes (*test12*). Our model outperforms both *RNACon* and *nRC* on all metrics.

We want to re-emphasize that we test our model on the dataset with 12 classes only to be consistent with [4] and be able to compare our model to *RNACon*, even though we are aware that comparing models trained on different datasets is not rigorous. However, the results show that our model not only outperforms *RNACon* of more than 40% on almost all metrics, but also significantly improves on *nRC*, the previous state-of-the-art method for ncRNA classification, which has been trained and tested on the same data as our model.

5 CONCLUSION

We have presented *RNAGCN*, the first successful application of graph convolutional networks to RNA folding data, which achieves state-of-the-art results on the challenging task of ncRNA classification. Our model combines edge-aware convolutions and an attention-based pooling mechanism. With respect to existing approaches, our model comes with the additional benefit of being trained end-to-end and removing the need for manual feature extraction from the graph.

REFERENCES

[1] C. Borgelt, T. Meinl, and M. Berthold. 2005. 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)*. ACM, New York, NY, USA, 6–15.

[2] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. 2017. Geometric Deep Learning: Going beyond Euclidean data. *IEEE Signal Processing Magazine* 34, 4 (July 2017), 18–42.

[3] L. Childs, Z. Nikoloski, P. May, and D. Walther. 2009. Identification and classification of ncRNA molecules using graph properties. *Nucleic Acids Res* 37, 9 (May 2009).

[4] A. Fiannaca, M. La Rosa, L. La Paglia, R. Rizzo, and A. Urso. 2017. nRC: non-coding RNA Classifier based on structural features. *BioData Mining* 10, 1 (2017), 27.

[5] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. 2017. Neural Message Passing for Quantum Chemistry. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Doina Precup and Yee Whye Teh (Eds.), Vol. 70. PMLR, International Convention Centre, Sydney, Australia, 1263–1272.

[6] F. He, L. Fang, and Q. Yin. 2019. miR-363 acts as a tumor suppressor in osteosarcoma cells by inhibiting PDZD2. *Oncology Reports* (March 2019).

[7] I. L. Hofacker. 2003. Vienna RNA secondary structure server. *Nucleic Acids Res* 31, 13 (01 Jul 2003), 3429–3431.

[8] S. Ioffe and C. Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *CoRR*

Model	*Dataset*	*Accuracy*	*Sensitivity*	*Specificity*	*Precision*	*F1-score*	*MCC*
nRC	*test13*	81.81%	81.81%	98.48%	81.50%	81.66%	80.29%
RNAGCN (ours)	*test13*	**85.73%**	**86.09%**	**98.82%**	**86.09%**	**85.61%**	**84.59%**
RNACon	*test12*	37.17%	37.17%	96.26%	45.84%	41.05%	33.43%
nRC	*test12*	81.04%	81.04%	98.42%	82.11%	81.57%	79.46%
RNAGCN (ours))	*test12*	**85.29%**	**81.06%**	**98.78%**	**87.82%**	**86.30%**	**84.07%**

Table 4: Summary of results on the two independent test datasets with 13 and 12 classes respectively. Results for *nRC* and *RNACon* are taken from [4].

abs/1502.03167 (2015).

[9] D. P. Kingma and J. Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[10] T. N. Kipf and M. Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

[11] Y. LeCun, Y. Bengio, and G. E. Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.

[12] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (Nov 1998), 2278–2324.

[13] T. R. Mercer and J. S. Mattick. 2013. Structure and function of long noncoding RNAs in epigenetic regulation. *Nature Structural &Amp; Molecular Biology* 20 (05 Mar 2013), 300 EP –. Review Article.

[14] F. Monti, D. Boscaini, J. Masci, E. Rodolà, J. Svoboda, and M. M. Bronstein. 2016. Geometric deep learning on graphs and manifolds using mixture model CNNs. *CoRR* abs/1611.08402 (2016).

[15] B. Panwar, A. Arora, and G. PS Raghava. 2014. Prediction and classification of ncRNAs using structural information. *BMC Genomics* 15, 1 (2014), 127.

[16] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* 15, 1 (Jan. 2014), 1929–1958.

[17] X. Tian and Z. Zhang. 2017. miR-191/DAB2 axis regulates the tumorigenicity of estrogen receptor-positive breast cancer. *IUBMB Life* 70, 1 (Dec. 2017), 71–80.

[18] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

[19] O. Vinyals, S. Bengio, and M. Kudlur. 2016. Order Matters: Sequence to sequence for sets. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

[20] H. Wu, J. Li, E. Guo, S. Luo, and G. Wang. 2018. MiR-410 Acts as a Tumor Suppressor in Estrogen Receptor-Positive Breast Cancer Cells by Directly Targeting ERLIN2 via the ERS Pathway. *Cellular Physiology and Biochemistry* 48, 2 (2018), 461–474.

[21] C. Yang, S. Tabatabaei, X. Ruan, and P. Hardy. 2017. The Dual Regulatory Role of MiR-181a in Breast Cancer. *Cellular Physiology and Biochemistry* 44, 3 (2017), 843–856.

[22] C. Yuan and Y. Sun. 2013. RNA-CODE: A Noncoding RNA Classification Tool for Short Reads in NGS Data Lacking Reference Genomes. *PLOS ONE* 8, 10 (10 2013), 1–10.