High precision in microRNA prediction: a novel genome-wide approach with convolutional deep residual networks

C. Yones, J. Raad, L.A. Bugnon, D.H. Milone, and G. Stegmayer

Research Institute for Signals, Systems and Computational Intelligence, sinc(i), FICH-UNL, CONICET, Ciudad Universitaria UNL, 3000, Santa Fe, Argentina.

Abstract

MicroRNAs (miRNAs) are small non-coding RNAs that have a key role in the regulation of gene expression. The importance of miRNAs is widely acknowledged by the community nowadays and computational methods are needed for the precise prediction of novel candidates to miRNA. This task can be done by searching homologous with sequence alignment tools, but results are restricted to sequences that are very similar to the known miRNA precursors (pre-miRNAs). Besides, a very important property of pre-miRNAs, their secondary structure, is not taken into account by these methods. To fill this gap, many machine learning approaches were proposed in the last years. However, the methods are generally tested in very controlled conditions. If these methods were used under real conditions, the false positives increase and the precisions fall quite below those published. This work provides a novel approach for dealing with the computational prediction of pre-miRNAs: a convolutional deep residual neural network (mirDNN). This model was tested with several genomes of animals and plants, the full-genomes, achieving a precision up to 5 times larger than other approaches at the same recall rates. Furthermore, a novel validation methodology was used to ensure that the performance reported in this study can be effectively achieved when using mirDNN in novel species.
To provide fast an easy access to mirDNN, a web demo is available at http://sinc.unl.edu.ar/web-demo/mirdnn/. The demo can process FASTA files with multiple sequences to calculate the prediction scores and generates the nucleotide importance plots.
Full source code: http://sourceforge.net/projects/sourcesinc/files/mirdnn and https://github.com/cyones/mirDNN.
Contact: gstegmayer@sinc.unl.edu.ar

1 Introduction

MicroRNAs (miRNAs) have a crucial role in post-transcriptional gene regulation, cell growth and other physiological processes (Sarkar *et al.*, 2021). Since their discovery (Bartel, 2004), miRNAs have reshaped the knowledge on gene regulation. They can determine the genetic expression of cells and can influence the state of the tissues. Therefore, finding new miRNAs and inferring their functions are necessary tasks for understanding their roles in gene regulation. Given their importance in promoting or inhibiting certain diseases and infections, the discovery of novel miRNAs is considered one of the most important problems in biology today (Demirci *et al.*, 2017).

MiRNAs are encoded in primary precursors of about 100 nt in length called pri-miRNAs, which are transcribed by RNA polymerase II. After transcription, the chain forms a hairpin-like secondary structure with few internal loops that is recognized and cut by the Drosha enzyme, forming a pre-miRNA. The pre-miRNA is then exported to the cytoplasm and converted by the Dicer enzyme into double stranded RNA (dsRNA). The dsRNA is loaded and separated by the Argonaute protein, binding to the mature miRNA and promoting the expulsion of the complementary strand, to form the RNA-induced silencing complex (RISC). Once loaded into the RISC, the mature miRNA performs its post transcriptional regulation function via binding within the $3'$ untranslated regions (UTR) of messenger RNA (Sarma *et al.*, 2020). To discover new miRNAs, the evaluation of large number of hairpin-like sequences in a full genome context with experimental methods is unfeasible, thus computational methods play an important and increasing role for their prediction (Huan and et al., 2015; Takahashi and et al., 2015; Chen *et al.*, 2019; Bugnon *et al.*, 2019b).

The simplest way of finding miRNAs in a genome is using a local alignment search tool, such as BLAST, looking for regions of similarity between sequences. Having known and publicly available miRNAs of other species, similar sequences can be found by direct comparison, that is, similarity-based homology. However, it is known that RNA sequences can be considered context free languages, which means that direct similarity search can be very limited (Searls, 2002). For example, miRNA precursors (pre-miRNAs) in animals usually have a very stable secondary structure, and although a sequence can be relatively similar to a known animal pre-miRNA, even a small number of mutations can lead to a very different and unstable secondary structure. Another problem of sequence alignment methods is that not all nucleotides are equally important in a pre-miRNA. It is known that the stem region of a sequence is usually highly conserved in comparison to the loop region. A sequence alignment tool would give the same importance to mutations, independently of their positions on the sequence.

The limitations of the alignment-based tools led to the emergence of machine learning (ML) approaches for pre-miRNA prediction (Li *et al.*, 2010; Gomes and et al., 2013; Shukla *et al.*, 2017; Stegmayer *et al.*, 2018; Chen *et al.*, 2019). These methods can learn the intrinsic properties that define a pre-miRNA, besides the raw sequence. Nonetheless, this prediction task has many complex and challenging aspects. First, the number of known pre-miRNAs to be used as positive class is quite limited (Shaker *et al.*, 2020). For example, in the case of the human genome, there are 1,982 pre-miRNAs deposited in miRBase (Kozomara *et al.*, 2018) v22 (release oct 2018), while more than 48 millions of sequences with hairpin-like structure can be extracted from the complete genome. This constitutes a very large class imbalance problem (in the order of 1:24,000), which is very difficult to be handled by any ML model (Bugnon *et al.*, 2019b). Second, in these conditions, existing ML-based methods have a very high rate of false positives, that is, they provide an excessively long list of candidates to novel pre-miRNAs that cannot be validated with wet-experiments (Demirci *et al.*, 2017). Third, the performance measures are always calculated in very controlled conditions. For example, with cross-validation over the known pre-miRNAs of the species under study, a setup which is not possible when studying a new sequenced genome without any confirmed pre-miRNA. Furthermore, most methods are tested on artificially balanced datasets, which leads to the overestimation of the precision. Finally, a very important aspect when looking for candidates in a full genome is the feature extraction required for training a classical ML classifier. A very large number of features have been proposed for pre-miRNAs classification, having the serious disadvantage of being highly dependant on the feature engineering and the manual intervention (de ON Lopes *et al.*, 2014; Yones *et al.*, 2015; Raad *et al.*, 2019).

The emergence of deep learning (DL) models has led to substantial improvements in the field of automatic information extraction and representation, precisely avoiding the feature engineering step (LeCun *et al.*, 2015). Deep models can automatically extract relevant features by themselves, differently to the traditional procedure of hand-made definition of the features, which is extremely time consuming and requires domain experts. Deep learning is inspired by the representation of biological neural networks and it can be considered, today, among the best paradigms of ML for classification (Bengio *et al.*, 2013). It has already been employed for small-RNA feature extraction (Zheng *et al.*, 2019), identification and classification (Amin *et al.*, 2019), showing promising results for genomics analysis also. In fact, DL models had recent success in several sequence classification studies (Zeng *et al.*, 2016; Bugnon *et al.*, 2019b) because empirical analyses have shown that deep models can extract motifs from a set of homologous sequences, which are essential for distinguishing among different types of sequences, such as for example miRNA families (Seo *et al.*, 2018; Tang and Sun, 2019). In Eraslan *et al.* (2019) authors analyze gaps and challenges for DL in genomics, mentioning the need for more DL-based tools capable of handling the real genome-wide scenario.

In this work we propose mirDNN, a novel pre-miRNA prediction algorithm based on deep learning. It has been designed for finding candidate sequences in genome-wide data. The model is a convolutional deep residual neural network (He *et al.*, 2016) that can automatically learn suitable features from the raw data, without a manual feature engineering. It is capable of constructing a model that can successfully learn, automatically, the intrinsic structural characteristics of pre-miRNAs within a sequence. The proposal has been tested with several genomes of animals and plants, and compared with state-of-the-art algorithms.

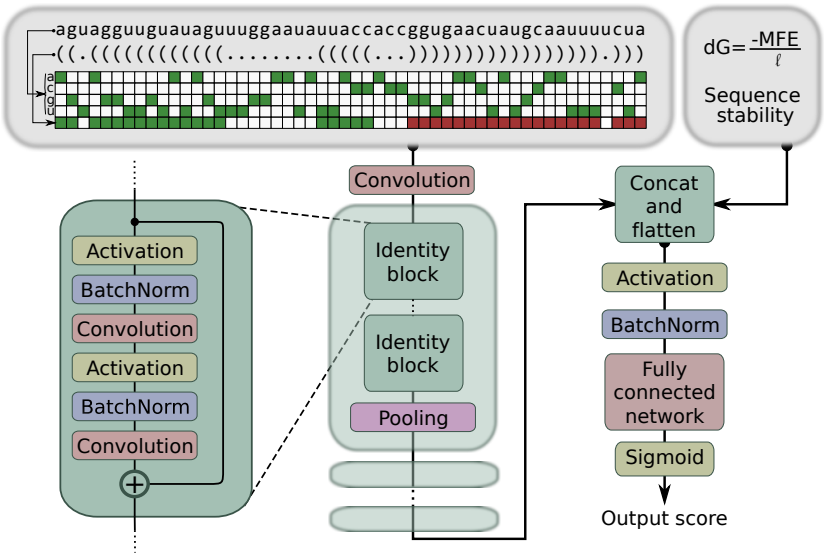

Figure 1: Schematic representation of mirDNN end-to-end architecture. In the matrix, a green cell represents a 1, a red cell represents a -1, and a white cell represents a 0.

2 Deep learning model for microRNA prediction

MirDNN takes as input RNA sequences, their corresponding predicted secondary structure and the minimum free energy (MFE) when folding (Lorenz *et al.*, 2016). The input sequence and its secondary structure is represented as a one-hot-encoding tensor of shape $4 \times L$ (see Figure 1), where L is the maximum sequence length considered. It has 4 rows for the 4 possible nucleotides (A,C,G,U) in each position. Each column represents a nucleotide in the sequence. The tensor size is fixed and completed with zero-padding for sequences of variable length. This tensor can have, in each column, a 1 and three 0s, according to the nucleotide value to represent at each position. For example, if the first nucleotide of the sequence is Adenine, the first element of the column is 1 at the first row and the rest of the column is 0; if the nucleotide is Cytosine, the second element of the column is a 1 and the rest is 0. The secondary structure of the sequence is represented as a tensor of shape $1 \times L$, where the i-th element is 1 if the i-th nucleotide of the sequence is left-paired (a left parentheses in the RNAfold format), -1 if it is right-paired and 0 otherwise. The distinction between left and right paired nucleotides help to identify bulges and loops in the sequence. These two tensors are concatenated over the first dimension to form a tensor of shape $5 \times L$, which is the input tensor to the convolutional neural network (CNN).

The first layer of the network is a one-dimensional convolution of length F with W filters. This layer generates a sequence of length L and W features, which is the width of the network in all the next layers. Then, several stages composed of identity blocks (He *et al.*, 2016) and pooling layers are stacked. The identity blocks allow the model to auto-define the number of convolutional layers needed during training, which avoids optimization of the number of hidden layers. When there are many identity blocks available, the model is capable of automatically selecting how many of them are necessary while non-necessary blocks are just skipped. Each block is composed of two activation functions, two batch normalization layers, and two convolutions that can be seen in the detail (left part) of Figure 1. All these operations are done over the input tensor and the result is summed up to the input of the next identity block. This helps to back-propagate the error during training, allowing the addition of more convolution layers without difficulting the training of the model. After B identity blocks, a pooling layer is used to reduce the length L of the sequence by a factor of 2. After M of these stages, a tensor of shape $W \times L/2^M$ is obtained. This tensor is converted in a one-dimensional vector, and then it passes through activation and batch normalization layers. Then, the input sequence stability, D_G, is appended in order to form a tensor of $WL/2^M + 1$ elements. It is calculated as $D_G = -MFE/\ell$, where ℓ is the original sequence length (not L, which is the zero-padded one). After that, this tensor feeds a fully connected network that generates the corresponding output score for prediction of the input sequence.

For training the mirDNN model, two strategies were used to tackle the imbalance present in genome-wide data. The first one was positive class oversampling, where each training batch is built with a proportion of positive samples that is higher than the real proportion in the dataset. To achieve this, the positive cases are

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (sinc.unl.edu.ar)
C. Yones, J. Raad, L. A. Bugnon, D. H. Milone & G. Stegmayer; "High precision in microRNA prediction: a novel genome-wide approach with convolutional deep residual networks"
Computers in Biology and Medicine, Vol. 134, 2021.

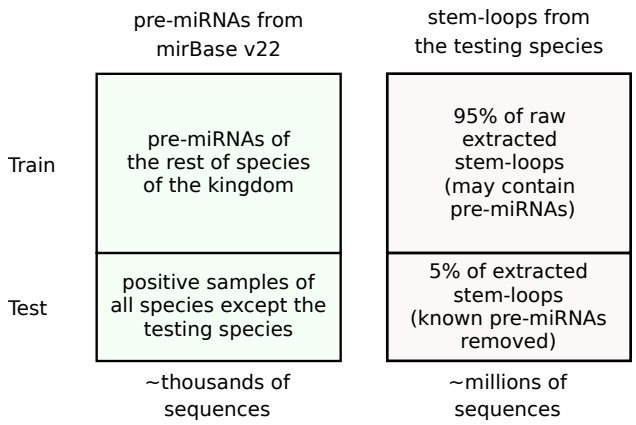

Figure 2: The leave-one-species-out scheme for training and testing the mirDNN model.

sampled with replacement, which means that a positive sample can be selected multiple times to be part of the training set. In this sampling with replacement, each sample is independent and has the same probability of being chosen.

The second strategy was focal loss (FL), proposed by Lin *et al.* (2017). When the negative samples (the majority class) are classified, they generate an error to be back-propagated through the model. Due to the high imbalance, the sum of these contributions is much larger than the contribution of the (few) positive samples, and the model is heavily biased towards the negative class. Thus, it does not learn the positive class correctly. In order to solve this, a FL function is defined as

$$FL(p_t) = -(1 - p_t)^{\gamma} \log p_t, \tag{1}$$

where p_t is the predicted probability (output score) for the sequence under analysis, and the parameter γ can be used to increase or reduce the weight given to those samples that are correctly classified. This way, if the majority of the samples are correctly classified (with a small label error), the error back-propagated is much smaller than with traditional binary cross entropy loss (BCE). Therefore, in an imbalance escenario the model errors for the minority (in this case, the positive) class get more importance and drive the learning of the network.

3 Materials and experimental setup

3.1 Data

Genome-wide data from 4 species was used in this work: *Arabidopsis thaliana*, *Caenorhabditis elegans*, *Anopheles gambiae*, and a much more larger dataset with the sequences of the *Homo sapiens* genome (Bugnon *et al.*, 2019a). First, HextractoR[1] was used to extract all the stem-loop sequences from each genome with standard settings, with window size of 320 nt for plants and 160 nt for animals. The positive samples were taken from mirBase v22. Only the animal sequences were used on test with animal genomes and only the plant sequences were used on the test with *A. thaliana*.

3.2 Experimental setup

In order to have an experimental setup that emulates a real genome-wide situation, for example the case of the genome of a recently sequenced species, a leave-one-species-out scheme was used (see Figure 2). In this scheme, all the pre-miRNAs of each testing species were excluded from the corresponding training sets. This is a very challenging test condition, a much harder problem to solve than the problems usually used in published experimental setups.

For training mirDNN with a full-genome, the positive-class training set has all the well-known pre-miRNAs of the species of one kingdom, without the specific pre-miRNAs of the target species to predict. For example,

[1] https://cran.r-project.org/web/packages/HextractoR

Species	Train		Test	
	Negative	Positive	Negative	Positive
Arabidopsis thaliana	1,289,582	7,855	67,714	326
Caenorhabditis elegans	1,652,486	22,795	86,907	253
Anopheles gambiae	4,062,470	22,918	213,529	130
Homo sapiens	45,796,170	21,131	2,394,897	1,917

Table 1: Number of sequences in train and test partitions of each genome, using the leave-species-out validation schema.

for prediction in the human genome, all well-known pre-miRNAs of animals are used as the positive-class, excluding the human pre-miRNAs. The negative-class for the training set has 95% of the rest of the hairpin-like sequences in the target species genome, excluding all known pre-miRNAs of that species. For testing, the specific known pre-miRNAs of the target species are used, together with the remaining 5% of stem-loops of the genome as negative-class. The total number of sequences in each train and test partition is reported in Table 1.

To select the optimal hyper-parameters for mirDNN, a grid search was performed using the data of three full genomes: *A. thaliana*, *C. elegans* and *A. gambiae*. The training sequences of each genome were split in a 90%-10% training-validation scheme to measure the performance of each hyperparameter set. The hyper-parameters tuned were: the number of filters W (16, 32, 64), the number of identity blocks B on each stacked stage (1,3,5), the filter length F (3, 5), and the loss function (BCE or FL). Two imbalance conditions were used during training: i) minority class oversampling; and ii) natural data imbalance.

In Figure 3, the effect of different parameters over the AUCPR is shown. First, Figure 3 a) shows the effect of different alternatives during the training process. From the figure, it can be concluded that oversampling the positive class generates a significative fall in the AUCPR. Training the model with the natural class imbalance of the dataset leads to the best results. With respect to the loss function, using FL achieves the best results in all cases. Secondly, in Figure 3 b) the AUCPR of different architectures using FL and without oversampling is shown. Here, the differences are not very conclusive: there are several good pairs of parameters. In *C. elegans* and *A. thaliana* the best results were achieved with $B = 3$ and $W = 64$, while in *A. gambiae* this configuration reached the second best AUCPR. Therefore, these values will be used in the final model. Finally, in Figure 3 c) the effect of filter size W on AUCPR (left axis) in the three datasets using FL and without oversampling is shown. As it can be seen, larger filters do not help improving the results, and since filters of length 5 are computationally intensive, filters of length 3 were chosen.

The mirDNN model has been compared with the following methods: i) BLAST alignments against complete pre-miRNAs, using the e-value of each sequence as the prediction score (BLAST version 2.6 was used with default parameters); ii) DeepMir (Tang and Sun, 2019) a CNN model for miRNA family prediction, which obtained one of the best performances in a very recent review of genome-wide methods (Bugnon *et al.*, 2020); iii) miRNAss (Yones *et al.*, 2017), a semi-supervised method based on graphs designed for taking advantage of unlabeled sequences when there are very few positive samples; iv) a classical Random Forest (RF), as the representative of a supervised model for pre-miRNAs prediction (Gudy *et al.*, 2013), which was among the best supervised models according to (Stegmayer *et al.*, 2018); and v) a Gradient Boosting Machine (GBM), which is currently one of the best methods based on ensemble learning (Ke *et al.*, 2017).

The full source code to reproduce the experiments is freely available [2]. Furthermore, to provide fast an easy access to mirDNN, a web demo (Stegmayer *et al.*, 2016) is also available [3].

3.3 Performance measures

The prediction quality of the model was assessed using the classical classification measures of precision (P) and recall (R), defined as

$$P = \frac{TP}{TP + FP} \qquad R = \frac{TP}{TP + FN} \tag{2}$$

[2] http://sourceforge.net/projects/sourcesinc/files/mirdnn
[3] http://sinc.unl.edu.ar/web-demo/mirdnn/

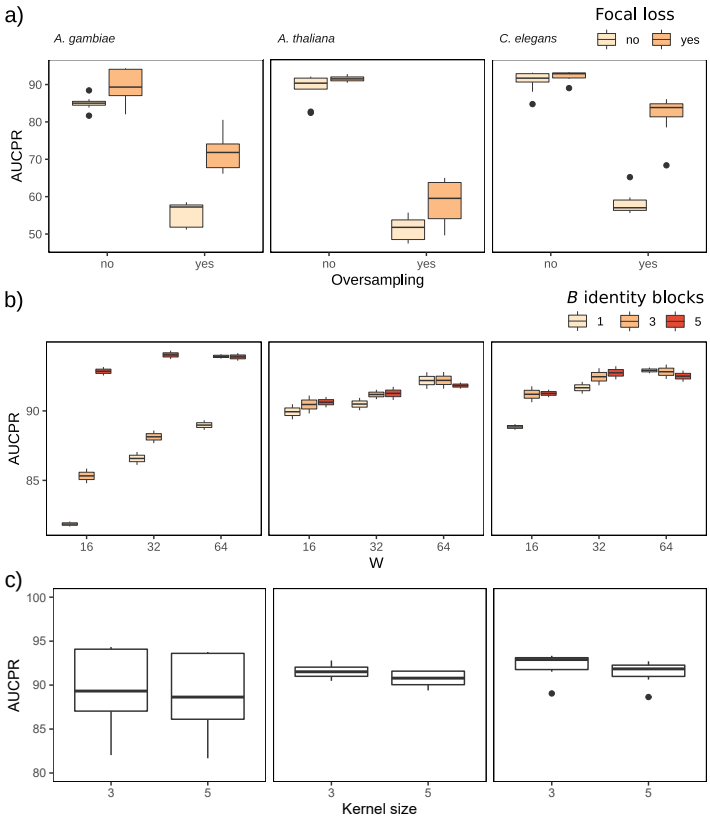

Figure 3: Effect of different hyper-parameters sets in the AUCPR on the 3 genomes: a) effect of focal loss and oversampling; b) different combinations of filters W and number of identity blocks B, using FL without oversampling for training; c) effect of increasing the size of the filters.

where TP, FP and FN are true positive, false positive and false negative predictions, respectively. These measures were used to generate precision recall curves (PRC). It has been shown (Saito *et al.*, 2015) that PRC are preferred over the classical receiver operating characteristic (ROC) curves and area under the ROC curve (AUROC) to assess binary classifiers with highly imbalanced data. When there is a large class imbalance in a dataset, a classifier can reach a good performance in terms of specificity, but can perform poorly in providing good quality candidates, with a large amount of false positives. A PRC can provide a better assessment of performance because it also evaluates the fraction of true positives among the total positive predictions. The area under the precision-recall curve (AUCPR), which is a single numeric summary of the information, will also be reported.

4 Results and discussion

4.1 Comparison with state-of-the-art methods

Figure 4 shows the ROC (top) and PRC (bottom) plots corresponding to mirDNN and the other methods for *A. thaliana*, *C. elegans*, *A. gambiae* and *H. Sapiens* full genomes. The ROC plots show recall in the y-axis and FP rates (FPR) in the x-axis. In the PRC plots, precision is shown in the y-axis and recall in the x-axis. The curves are generated varying the threshold applied to the output score, to separate the positive from the negative samples, calculating the precision and recall for each possible threshold.

According to the ROC plots (Figure 4, top), the methods seem to be similar and equally good for all species. However, it should be noted that due to the high class imbalance present in data, most classifiers can reach low FPR in these plots (i.e. good specificity), even with an unacceptable number of FPs. This is because $FPR = FP/N_{neg}$ and the number of negatives (N_{neg}) is always much larger than the number of FPs in high imbalance, resulting in low FPRs. For example, in Figure 4 c) for *A. gambiae* all classifiers seem to be indistinguishable and equally good (AUROC 0.97 for mirDNN, 0.99 for BLAST, 0.98 for RF and 0.98 for GBM), with the exception of deepMir (AUROC 0.70) and miRNAss (AUROC 0.48), which are the worst

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (sinc.unl.edu.ar)
C. Yones, J. Raad, L. A. Bugnon, D. H. Milone & G. Stegmayer; "High precision in microRNA prediction: a novel genome-wide approach with convolutional deep residual networks"
Computers in Biology and Medicine, Vol. 134, 2021.

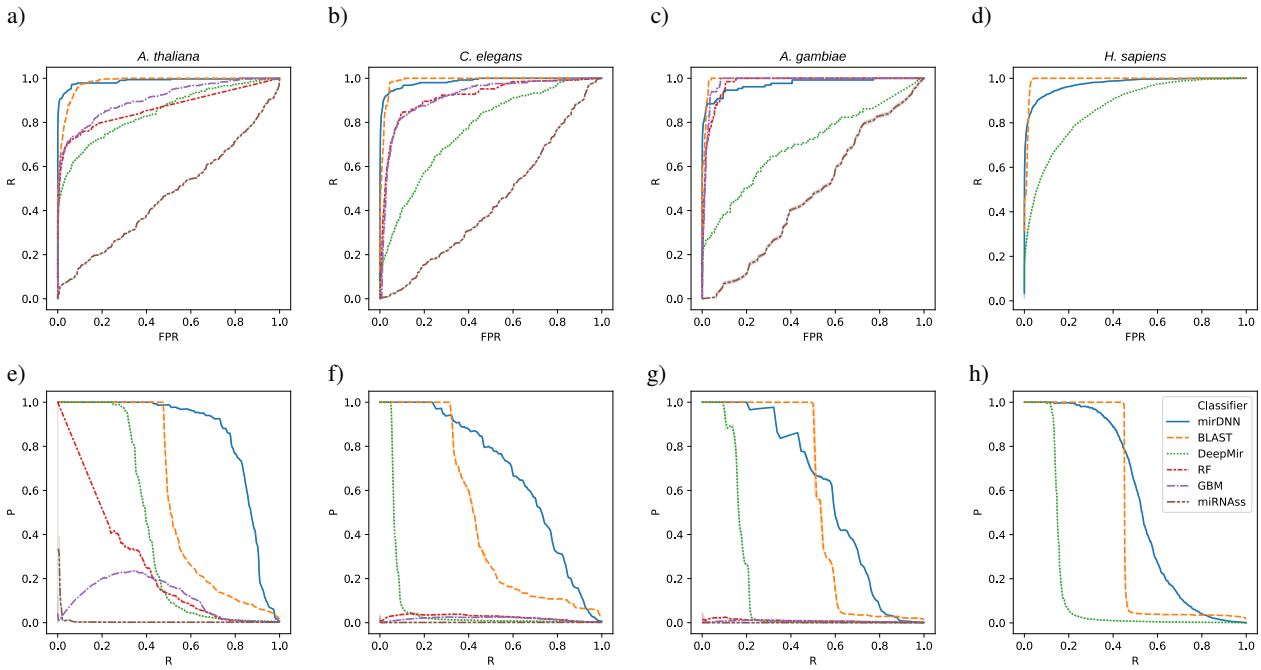

Figure 4: (top) ROC plots for all methods on different genomes: a) *A. thaliana*, b) *C. elegans*, c) *A. gambiae*, d) *H. Sapiens*. (bottom) PRC plots for all methods on different genomes: e) *A. thaliana*, f) *C. elegans*, g) *A. gambiae*, h) *H. Sapiens*

ones. Despite that, when the point of best performance for each method is analyzed in detail, the RF model has 556 FPs and GBM has 2,694 FPs, while mirDNN has only 42 FPs. Similar behavior and AUROC values are observed for the other genomes. Actually, in highly imbalanced data sets, the real difference in performance among models can be seen with the PRC plots.

In Figure 4 e) the PRC plot of *A. thaliana* full genome shows that mirDNN can provide a precision of 0.93 and, at the same time, a recall of 0.70. BLAST reaches a precision of 1.00 up to a recall of approximately 0.50, where it abruptly decreases performance. DeepMir presents a similar behavior but with a lower performance. The feature-based ML methods present the worst results: RF has an almost linearly decreasing trend with a much worse performance than mirDNN and BLAST; while miRNAss and GBM can barely achieve a precision of 0.40 for low recall rates, therefore they are not competitive. For a recall of 0.70, mirDNN has a precision nearly 5 times higher than the second method, BLAST.

In Figure 4 f) for *C. elegans*, the best performing methods are mirDNN and BLAST, while DeepMir, RF, GBM and miRNAss have very low performances. BLAST obtains a very good precision, but only for the positive identical sequences, up to a recall of 0.40 approximately. Above that value, the precision falls. MirDNN has the same precision for the identical sequences, but also maintains a good precision for recalls up to 0.80.

In Figure 4 g) for *A. gambiae*, BLAST again shows high precision for sequences that are extremely similar to the positives class samples (known pre-miRNAs). However, for a recall of about 0.60, precision rapidly decreases to a very low value (under 0.15). DeepMir, again, presents a similar behavior, but with a much lower performance. Meanwhile, mirDNN can maintain a precision of 0.67 for a recall of 0.60. That is, with BLAST there are 152 TP and 769 FP, while mirDNN provides a list of 152 TPs with only 75 FPs.

In order to see the scalability of mirDNN with a larger dataset, it was tested on *Homo sapiens* full genome. Here, mirDNN was compared with BLAST and DeepMir, because the other methods were highly outperformed in the other three genomes. As it can be seen in Figure 4 h), DeepMir achieves a good precision but only for recall below 0.20. BLAST achieves better results, with good precision for recall smaller than 0.45, which could be the approximate size of the set of pre-miRNAs that have very similar sequences in other species. But for recalls higher than 0.45 the precision abruptly falls to very small values. Differently, mirDNN maintains a relatively good precision for recalls up to 0.60. This means that, for example, with a precision of 0.50 mirDNN founds 1,016 true pre-miRNAs while BLAST founds only 863. These results show that the proposed method is really useful for discovering new pre-miRNAs, while the best competitor only allows finding sequences very

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (sinc.unl.edu.ar)
C. Yones, J. Raad, L. A. Bugnon, D. H. Milone & G. Stegmayer; "High precision in microRNA prediction: a novel genome-wide approach with convolutional deep residual networks"
Computers in Biology and Medicine, Vol. 134, 2021.

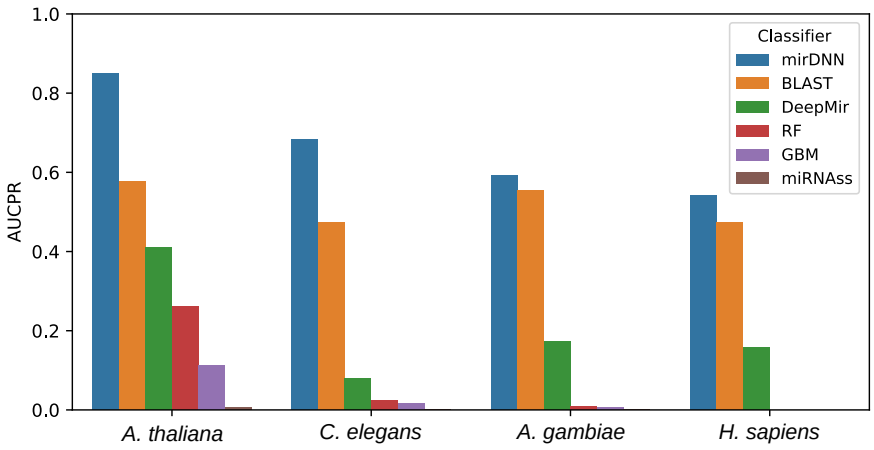

Figure 5: Comparative AUCPR of mirDNN on the four test genomes.

similar to those already known.

Finally, a summary of results with all genomes is presented in Figure 5, comparatively showing the AUCPR for all the methods. It can be clearly seen here that mirDNN has the best results in all genomes, no matter the species. This highlights the capability of the deep model for analyzing complete sequences, exploiting both sequential and structural information, and without the need of hand-crafted features. The second best method is BLAST, which shows better performance than the other classical ML methods. This can be explained due to the fact that many pre-miRNAs are highly conserved in many species. This gives BLAST very good precision for these conserved sequences, and also explains why the precision for BLAST abruptly falls when trying to predict pre-miRNAs that are not very conserved across species. In the case of DeepMir, in spite of being a CNN that learns features directly from the sequence, its performance is much lower than the one achieved by mirDNN. This difference could be due to several reasons. First, the lower number of layers and a simpler architecture in comparison to mirDNN, and without identity blocks. These blocks allow the model to auto-define the number of convolutional layers needed during training, avoiding optimization of this critical hyperparameter. Second, the lack of a strategy to tackle the class imbalance during training in comparison to the positive class oversampling and focal loss of mirDNN, which allow obtaining a better performance in a real genome-wide scenario. The other ML methods, in spite of using features calculated over the full sequence and considering the secondary structure, fall behind the other methods in terms of performance. This could be mainly attributed to the fact that automatically learnt features are more flexible and discriminant than hand crafted features. The convolutions can effectively learn most of the discriminative patterns found in the sequence/structure for each species, while the hand crafted features are fixed for all of them.

4.2 Class activation maps

An interesting insight regarding the mirDNN deep model is that it is possible to know which patterns in the training data were those actually learnt by the model. This can provide useful details on which parts are important in a sequence to produce a pre-miRNA, based on the information coming from a large database of known pre-miRNAs. Given the number of parameters of the mirDNN model (321,315), it would be difficult to obtain interpretations of the predictions directly from them. While the first layers learn patterns related to the prediction tasks, these patterns are then heavily processed by the deepest layers of the model. Therefore, it is important to study its inner behavior, as a whole.

To achieve this, the importance of each nucleotide in the prediction task was measured independently. First, the whole input sequence was evaluated with the trained mirDNN model, and the prediction output was used as reference score for the sequence. Then, all nucleotides were masked, one by one, that is, the corresponding column was converted to an all-zeros vector, and each masked version of the sequence was evaluated again with mirDNN. The decrease in the score for the sequence was used as a measure of each nucleotide importance. Since the mature miRNA is encoded in a 21nt sequence, generally found next to the terminal loop, it could be

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (sinc.unl.edu.ar)
C. Yones, J. Raad, L. A. Bugnon, D. H. Milone & G. Stegmayer; "High precision in microRNA prediction: a novel genome-wide approach with convolutional deep residual networks"
Computers in Biology and Medicine, Vol. 134, 2021.

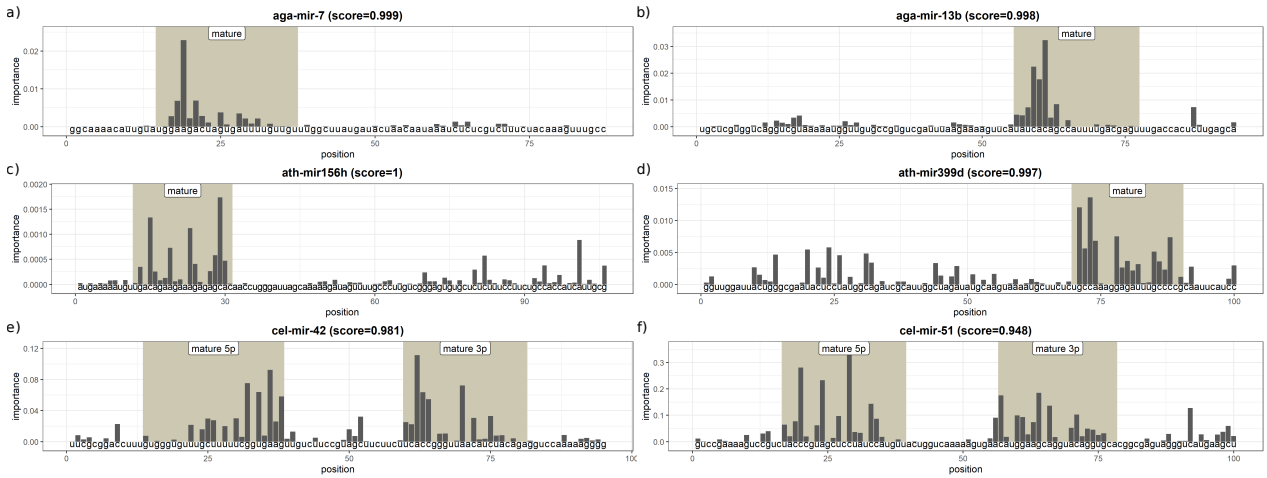

Figure 6: Nucleotide importance of six known pre-miRNA sequences. a) aga-mir-7, b) aga-mir-13b, c) ath-mir156h, d) ath-mir399d, e) cel-mir-42, f) cel-mir-51.

expected that convolutional layers pay particular attention to this section of a sequence.

Figure 6 presents the importance that mirDNN assigns to each nucleotide in six sample sequences of known pre-miRNAs. The vertical axis shows the importance levels of each nucleotide, while the horizontal axis shows the nucleotides of the sequence. In addition, the area of the sequence where the corresponding mature miRNA is located is highlighted. As it can be seen, in the region where the mature is located, the importance values are higher. Figure 6.a) shows how mirDNN gives more importance to the location of the mature in the 5p region. It can be seen in Figure 6.b) how the network assigns more importance to the nucleotides that are in the exact 3p region, known to encode the mature. In particular here, the nucleotides that encode the seed within the mature are those that present the highest level of importance. A similar importance to the mature region can ben seen in Figures 6.c) and 6.d) for another species. In Figure 6.e) and 6.f) are presented two examples of pre-miRNAs that have two mature regions each. In these examples it is clearly shown that both mature regions, 3p and 5p, are recognized with high importance by mirDNN. To access the statistical significance of these result, the distributions of nucleotide importances among all sequences were compared through a t-test with the distribution of those nucleotides that do not lay in the mature region. The difference has shown to be statistically significative with a p-value of 8.3E-13, which shows that this trend is present in all the pre-miRNAs used for testing.

5 Conclusions

This work provided a novel approach based on convolutional deep residual networks for dealing with the computational prediction of pre-miRNAs. The comparative results have shown that this novel approach can achieve a better performance than other machine learning and sequence alignment methods in several genomes. The deep learning model was capable of using, both, the sequential and the structural information of the hairpins, which has significantly improve the predictions in comparison to feature engineering based methods. In addition, a validation methodology closer to a real prediction task has been designed for testing the model leaving out all data from one species in the test partition. An original method to generate nucleotide importance values was also presented, which can be used to get insights about the newly discovered pre-miRNA candidates. In summary, it can be stated that the discovery of pre-miRNAs with mirDNN is fast and scalable, making it suitable to process whole genomes, achieving very high precision.

Acknowledgements

The authors acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

Funding

This work was supported by Agencia Nacional de Promocion Cientifica y Tecnologica (ANPCyT) [PICT 2018 3384 and PICT 2018 2905] and UNL [CAID 2020-115].

References

Amin, N., McGrath, A., and Chen, Y.-P. P. (2019). Evaluation of deep learning in non-coding rna classification. *Nature Machine Intelligence*, **1**, 246–256.

Bartel, D. (2004). MicroRNAs: genomics, biogenesis, mechanism, and function. *Cell*, **116**(1), 281–297.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, **35**(8), 1798–1828.

Bugnon, L., Yones, C., Raad, J., Milone, D., and Stegmayer, G. (2019a). Genome-wide hairpins datasets of animals and plants for novel mirna prediction. *Data in Brief*, **25**, 104209.

Bugnon, L. A., Yones, C., Milone, D. H., and Stegmayer, G. (2019b). Deep neural architectures for highly imbalanced data in bioinformatics. *IEEE Transactions on Neural Networks and Learning Systems*, **31**, 2857–2867.

Bugnon, L. A., Yones, C., Milone, D. H., and Stegmayer, G. (2020). Genome-wide discovery of pre-mirnas: comparison of recent approaches based on machine learning. *Briefings in Bioinformatics*. bbaa184.

Chen, L., Heikkinen, L., Wang, C., Yang, Y., Sun, H., and Wong, G. (2019). Trends in the development of mirna bioinformatics tools. *Briefings in Bioinformatics*, **20**(5), 1836–1852.

de ON Lopes, I., Schliep, A., and de Carvalho, A. (2014). The discriminant power of RNA features for pre-miRNA recognition. *BMC Bioinformatics*, **15**(1), 124+.

Demirci, M. D. S., Baumbach, J., and Allmer, J. (2017). On the performance of pre-microrna detection algorithms. *Nature communications*, **8**(1), 1–9.

Eraslan, G., Avsec, Z., Gagneur, J., and Theis, F. (2019). Deep learning: new computational modelling techniques for genomics. *Nature Reviews Genetics*, **20**, 1.

Gomes, C. and et al. (2013). A review of computational tools in microRNA discovery. *Frontiers in Genetics*, **4**(1), 81–104.

Gudy, A., Szczeniak, M., Sikora, M., and Makalowska, I. (2013). HuntMi: an efficient and taxon-specific approach in pre-miRNA identification. *BMC Bioinformatics*, **14**(1), 83+.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer.

Huan, T. and et al. (2015). Genome-wide identification of microRNA expression quantitative trait loci. *Nature Communications*, **6**(1), 6601.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems*, pages 3146–3154.

Kozomara, A., Birgaoanu, M., and Griffiths-Jones, S. (2018). miRBase: from microRNA sequences to function. *Nucleic Acids Research*, **47**(D1), D155–D162.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, **521**, 436–444.

Li, L., Xu, J., Yang, D., Tan, X., and Wang, H. (2010). Computational approaches for microrna studies: a review. *Mamm Genome*, **21**(1), 1–12.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Lorenz, R., Hofacker, I., and Stadler, P. (2016). RNA folding with hard and soft constraints. *Algorithms for Molecular Biology*, **11**(8), 1–10.

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (sinc.unl.edu.ar)
C. Yones, J. Raad, L. A. Bugnon, D. H. Milone & G. Stegmayer; "High precision in microRNA prediction: a novel genome-wide approach with convolutional deep residual networks"
Computers in Biology and Medicine, Vol. 134, 2021.

Raad, J., Stegmayer, G., and Milone, D. H. (2019). Complexity measures of the mature miRNA for improving pre-miRNAs prediction. *Bioinformatics*, **36**(8), 2319–2327.

Saito, T., Rehmsmeier, M., Hood, L., Franco, O., Pereira, R., and Wang, K. (2015). The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLOS ONE*, **10**(3).

Sarkar, J. P., Saha, I., Sarkar, A., and Maulik, U. (2021). Machine learning integrated ensemble of feature selection methods followed by survival analysis for predicting breast cancer subtype specific miRNA biomarkers. *Computers in Biology and Medicine*, **131**, 104244.

Sarma, A., Phukan, H., Halder, N., and Madanan, M. G. (2020). An in-silico approach to study the possible interactions of mirna between human and sars-cov2. *Computational Biology and Chemistry*, **88**, 107352.

Searls, D. B. (2002). The language of genes. *Nature*, **420**(6912), 211.

Seo, S., Oh, M., Park, Y., and Kim, S. (2018). Deepfam: deep learning based alignment-free method for protein family modeling and prediction. *Bioinformatics*, **34**(13), i254–i262.

Shaker, F., Nikravesh, A., Arezumand, R., and Aghaee-Bakhtiari, S. H. (2020). Web-based tools for miRNA studies analysis. *Computers in Biology and Medicine*, **127**, 104060.

Shukla, V., Varghese, V. K., Kabekkodu, S. P., Mallya, S., and Satyamoorthy, K. (2017). A compilation of Web-based research tools for miRNA analysis. *Briefings in Functional Genomics*, **16**(5), 249–273.

Stegmayer, G., Pividori, M., and Milone, D. H. (2016). A very simple and fast way to access and validate algorithms in reproducible research. *Briefings in Bioinformatics*, **17**(1), 180–183.

Stegmayer, G., Di Persia, L. E., Rubiolo, M., Gerard, M., Pividori, M., Yones, C., Bugnon, L. A., Rodriguez, T., Raad, J., and Milone, D. H. (2018). Predicting novel microRNA: a comprehensive comparison of machine learning approaches. *Briefings in Bioinformatics*.

Takahashi, R. and et al. (2015). Loss of microRNA-27b contributes to breast cancer stem cell generation by activating ENPP1. *Nature Communications*, **6**(1), 7318.

Tang, X. and Sun, Y. (2019). Fast and accurate microrna search using cnn. *BMC bioinformatics*, **20**(23), 1–14.

Yones, C., Stegmayer, G., Kamenetzky, L., and Milone, D. (2015). miRNAfe: a comprehensive tool for feature extraction in microRNA prediction. *BioSystems*, **238**, 1–5.

Yones, C., Stegmayer, G., and Milone, D. H. (2017). Genome-wide pre-miRNA discovery from few labeled examples. *Bioinformatics*, **34**(4), 541–549.

Zeng, H., Edwards, M. D., Liu, G., and Gifford, D. K. (2016). Convolutional neural network architectures for predicting dna–protein binding. *Bioinformatics*, **32**(12), i121–i127.

Zheng, X., Xu, S., Zhang, Y., and Huang, X. (2019). Nucleotide-level convolutional neural networks for pre-mirna classification. *Scientific reports*, **9**(1), 1–6.

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (sinc.unl.edu.ar)
C. Yones, J. Raad, L. A. Bugnon, D. H. Milone & G. Stegmayer; "High precision in microRNA prediction: a novel genome-wide approach with convolutional deep residual networks"
Computers in Biology and Medicine, Vol. 134, 2021.