Precursor microRNA Identification Using Deep Convolutional Neural Networks

Binh Thanh Do [*1, 2], **Vladimir Golkov**[1], **Goktug Erce Gurel**[1], **and Daniel Cremers**[1]

[1]Technical University of Munich, Germany

[2]Hanoi University of Science and Technology, Vietnam

{binhtd.hust, goktugercegurel}@gmail.com, {vladimir.golkov, cremers}@tum.de

Abstract

Precursor microRNA (pre-miRNA) identification is the basis for identifying microRNAs (miRNAs), which have important roles in post-transcriptional regulation of gene expression. In this paper, we propose a deep learning method to identify whether a small non-coding RNA sequence is a pre-miRNA or not. We outperform state-of-the-art methods on three benchmark datasets, namely the `human`*,* `cross-species`*, and* `new` *datasets. The key of our method is to use a matrix representation of predicted secondary structure as input to a 2D convolutional network. The neural network extracts optimized features automatically instead of using the enormous number of handcrafted features as most of existing methods do. Code and results are available at* `https://github.com/peace195/miRNA-identification-conv2D`*.*

1. Introduction

Ribonucleic acids (RNAs) are biomolecules involved in many biological processes. RNA is assembled as a chain of monomers called nucleotides. There are four types of RNA nucleotides that serve as arbitrarily arranged building blocks of the nucleotide chain: *adenine* (A), *cytosine* (C), *guanine* (G), and *uracil* (U). MicroRNAs (miRNAs) are a type of non-coding RNA molecule, usually about 20-23 nucleotides long. Over the last three decades, they have been detected in a large number of organisms such as humans [5] and plants [22], as well as in viruses [36]. They bind to target messenger RNAs (mRNAs) to inhibit the translation of mRNAs to proteins [3]. Their importance in their role in gene regulation plays a part in diseases such as cancer [19, 10, 45], and they are good targets for disease markers and therapeutics [45]. Thus, miRNAs identification is a crucial task in medical treatments. But it is difficult to identify miRNAs directly because they are short. Most studies focus on computational methods for identifying pre-miRNAs. Primary transcripts called pri-miRNAs are processed to form precursor miRNAs (pre-miRNAs) and then to mature miRNAs. Identifying pre-miRNAs is easier in comparison to miRNAs because pre-miRNAs are a lot longer (approximately 80 nucleotides) and they have a hairpin loop structure with more structural features. Pre-miRNAs identification is a classification task, yielding the output "positive" or "negative". It is potentially a hard task due to the enormous amount of possible sequences that can be arranged using 4 nucleotide types A, C, G and U. Moreover, the number of explored pre-miRNAs is much smaller than the number of pseudo hairpins. The pseudo-hairpins are RNA sequences, which have similar hairpin loop structure to pre-miRNAs but do not contain mature miRNAs. Hence, we have to cope with a class imbalance problem. On the other hand, there are a lot of structural and biological features inside the input sequences and we do not know which is really needed for pre-miRNAs identification.

Previous work on miRNA and pre-miRNA identification has been based on handcrafted rules (MIReNA [31]) or machine learning. Machine learning methods have been increasingly popular during the last decade and demonstrated to be the most promising, with tools such as HuntMi [15], miRBoost [42], CSHMM [1], microPred [4], miPred [34], triplet-SVM [46], Mirann [37], DP-miRNA [43], deepMiRGene [35]. They predict secondary structure, which forms intricate base-pairing interactions within RNA sequence (see Fig. 1AB for an example), with standard methods such as RNAfold [18], GTfold [32], and CyloFold [6]. Then they extract a lot of handcrafted features, some of which are Watson-Crick nucleotide pairing (A-U, C-G), loop length [43, 42], sequence length [43], dinucleotide pair frequencies [43, 20, 42, 4, 34], trinucleotide pair frequencies (constituting 64 features) [43, 20], melting temperature [43], mini-

[*]Work done as an exchange student of ERASMUS+ Key Action 1 program between HUST and TUM.

mum free energy [44, 43, 9]. These features are used as inputs to machine learning methods such as support vector machines (SVM) [42, 4, 46], random forests [34], neural networks [37, 43, 35, 20] and hidden Markov models [1]. For example, Ref. [20] uses 98 of the aforementioned features for the input of their neural network. DP-miRNA [43] also uses neural network with 58 extracted characteristic features based on sequence composition and secondary structure predicted by RNAfold software, folding measures including various formulations of physical energy. One of the most feature-rich methods is Ref. [9], which uses around 900 features. The authors showed that the most significant factor in pre-miRNA identification is the secondary structure. Secondary structures are distinctive and many features can be extracted from them. The state-of-the-art method deepMiRGene [35] uses one-hot encoding of so-called dot-bracket notation to represent RNA secondary structure as input to a neural network. The problem with this representation is that the information contained therein is rather "entangled". There is no single data entry that indicates whether two nucleotides $\{i, j\}$ are paired with each other. Finding out whether they are paired requires parsing much of the entire secondary structure representation. This parsing is achieved by training a neural network with memory and attention mechanisms. However, "outsourcing" such a known meaningful information disentanglement to the learning usually results in suboptimal disentanglement and a more difficult overall learning task [27]. The biggest advantage of deepMiRGene [35] is they do not need any handcrafted features.

Deep Learning can extract the high-level hidden features of the input very well and get good performance on many tasks such as image classification, face and speech recognition [28], molecular function prediction [33], protein secondary [39] and tertiary structure prediction [2], protein contact prediction [13]. Inspired by the success of deep learning and the importance of secondary structure, we propose an end-to-end deep learning method using the given input sequence and its secondary structure. For the sequence data, people often use LSTM [17] neural networks because they can handle information that is far away in the sequence effectively. In addition, it is suitable for data with variable-sized such as RNA. However, in this paper, we are going to use a convolutional neural network in order to classify RNA sequences as pre-miRNA. We use global pooling to allow variable-sized input of the convolutional neural network to lead to a fixed-sized class prediction. The key of our approach is to encode the secondary structure in the pairing matrix format (also known as *dot plot*, see [8, 12, 30, 41]). In the pairing matrix, we specify the minimum free energy of the sequence and the interactions between nucleotides directly. The results show that our algorithm with this new technique performs better than other state-of-the-art methods on the benchmarks, namely the so-called `human`, `cross-species` and `new` datasets [35, 44]. When we compared our results with and without pairing matrix, we saw that the results are much worse without pairing matrix. Our contributions are:

- Propose a new method to represent input sequences using multi-channels format.
- Develop a convolutional network (ConvNet) for that type of input and outperform state-of-the-art results.
- Compare between variable-sized inputs and fixed-sized inputs for the ConvNet.

2. Our approach

Our approach contains two steps. Firstly, we represent the given input sequence by a multi-channels format based on sequence one-hot encoding and pairing matrix of its secondary structure. Secondly, we propose a ConvNet for that representation to extract features and classify the output automatically. Our proposed ConvNet (Algo. 1) is designed to adapt the variable size of inputs by using a global pooling layer. For comparison, we also build a fixed-sized inputs ConvNet (Algo. 2) by padding 0 into all inputs, which leads them have the same size.

2.1. Input representation

Secondary structure is a great feature for pre-miRNA identification [21, 20]. In this paper, we represent the secondary structure using pairing matrix format as given in Fig. 1C, E, which is the minimum free energy, stands for interaction and 0 stands for non-interaction.

Each nucleotide A, C, G, U of the input sequence is represented using one-hot encoding, which is a binary vector of dimension 4. Thus, we have the dictionary $\{A \mapsto (1,0,0,0), C \mapsto (0,1,0,0), G \mapsto (0,0,1,0), U \mapsto (0,0,0,1)\}$, yielding an $L \times 4$ array that represents the entire sequence of length L in one-hot encoding. Each RNA input sequence of length L is represented as follows (see also Fig. 1 for an example):

- Use RNAfold [18] to predict the secondary structure and minimum free energy.
- Represent the predicted secondary structure as a $L \times L$ *pairing matrix*, where position (i, j) indicates whether the ith nucleotide interacts with the jth nucleotide or not.
- Reshape the $L \times 4$ one-hot encoded sequence to $L \times 1 \times 4$ and replicate it horizontally to shape $L \times L \times 4$.
- Reshape the $L \times 4$ one-hot encoded sequence to $1 \times L \times 4$ and replicate it vertically to shape $L \times L \times 4$.

- Concatenate the three aforementioned arrays along the channels dimension (third dimension), yielding $L \times L \times (1+4+4)$.

By using this data representation, we ensure that every $1 \times 1 \times 9$ "pixel" (i, j) of the $L \times L \times 9$ array contains the entire available information about the nucleotide pair (i, j), namely the nucleotide type at position i and at position j as well as whether the two nucleotides are paired. Moreover, the neighborhood of that "pixel" contains the entire available information about sequence neighborhoods of positions i and j and the pairing between these neighborhoods. The presence and arrangement of such neighborhood-patterns is characteristic of certain RNA types. A convolutional network is appropriate for extracting high-level information from such local patterns (as shown for $L \times L$ representations of proteins [13]), and (with appropriate network layers) consolidating such information across the entire molecule. Fig. 1 is an example of our preprocessing step for the sequence "GCCCUUGGCA", where Fig. 1F is the output of this step.

2.2. Network architecture

Our network architecture is shown in Fig. 2. It contains three convolutional layers followed by two fully connected layers. For variable-sized inputs (Algorithm 1), because of variety length L of the input sequence, we set the batch size equals 1 and use a global pooling layer to control the size of the output, which will be fed into fully connected layers. We normalize the outputs to $[0, 1]$ by using a softmax layer at the end. It is calculated as follows, where 2 is the number of labels:

$$\mathrm{softmax}(x)_i = \frac{\exp(x_i)}{\sum_{i'=1}^{2} \exp(x_{i'})}. \tag{1}$$

The model is trained by minimizing the cross-entropy loss function:

$$\mathcal{L} = -\sum_i \left(\alpha y_i \log \hat{y}_i + \beta(1-y_i)\log(1-\hat{y}_i)\right), \tag{2}$$

where $y_i \in \{0, 1\}$, $\hat{y}_i$ is the output of the softmax layer, and $\alpha, \beta \in \mathbb{R}$ are the weights for penalizing class imbalance.

We set learning rate as 0.001, and use the adaptive subgradient method (AdaGrad) [11] for learning the model parameters. We also use soft one-hot encoding such as $[0.2\ 0.8], [0.1\ 0.9]$ instead of $[0\ 1]$, Adam stochastic optimization [24], AlexNet [25] and ResNet [16] architectures, different batch size, dropout [40], and L_2 weight decay [26] but results are still the same and sometime even worse. In the Fig. 3, our model can fit the training data because the training accuracy score tends to increase to 100%, whereas the training loss tends to decrease to 0.0. The validation accuracy and loss change direction after 40 epochs. Thus, we decide to top at 40th epoch when training in order to avoid overfitting.

In order to compare between variable-sized inputs (Algo. 1) and fixed-sized inputs (Algo. 2), we pad $L \times L \times 9$ input matrix to $400 \times 400 \times 9$ by zeros before putting it to ConvNet. The results are shown in the next section.

Algorithm 1 Variable-sized inputs

```
function BACKPROPAGATION(network, TrainSet)
    batchSize ← 1
    count ← 0
    batchGrad ← 0
    while True do
        input ← BatchInput(TrainSet, batchSize)
        label ← BatchLabel(TrainSet, batchSize)
        grad ← ComputeGradient(input, label)
        batchGrad ← batchGrad + grad
        count ← count + 1
        if count = 64 then
            UpdateNetworkParameter(batchGrad)
            count ← 0
            batchGrad ← 0
```

Algorithm 2 Fixed-sized inputs

```
function BACKPROPAGATION(network, TrainSet)
    batchSize ← 64
    while True do
        input ← BatchInput(TrainSet, batchSize)
        label ← BatchLabel(TrainSet, batchSize)
        grad ← ComputeGradient(input, label)
        UpdateNetworkParameter(grad)
```

3. Experiments

3.1. Datasets

In our experiments, we used datasets used in [35, 44], which were obtained from miRBase [14], NCBI (`https://www.ncbi.nlm.nih.gov/`), fRNAdb [23], NONCODE [7] and snoRNA-LBME-db [29]. We have three benchmark datasets which consist of human, cross-species, and new RNA sequences.

Table 1 shows the number of positive and negative samples in each of the datasets. In total, there are 3230 positive and 23934 negative samples, which leads to a class imbalance. To cope with the class imbalance, we tried to assign more weight for positive samples in the cross entropy loss function.

In Fig. 4, we can see the histogram of sequence lengths in our dataset. While the mean length of RNA sequences is 96, there are still sequences with more than three hun-

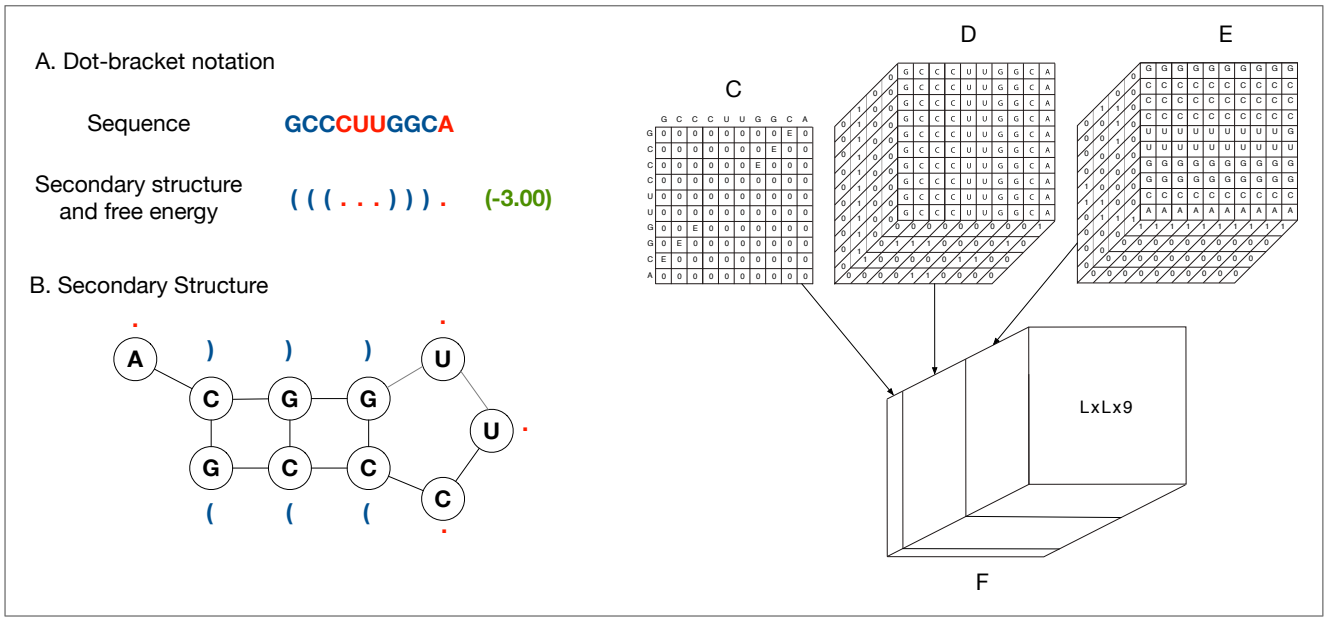

Figure 1. Example for preprocessing step of "GCCCUUGGCA" sequence. (A) Sequence of a pre-miRNA and the dot-bracket notation of its secondary structure, -3.00 is the minimum of free energy. In dot-bracket notation, each corresponding "()" pair represents an interaction nucleotide pair. (B) is the secondary structure of the given sequence. (C) is the pairing matrix based on the secondary structure of sequence where E is the minimum free energy. (D) and (E) are vertical and horizontal one-hot encoded sequence. (F) is the combination of (C), (D) and (E) by concatenating.

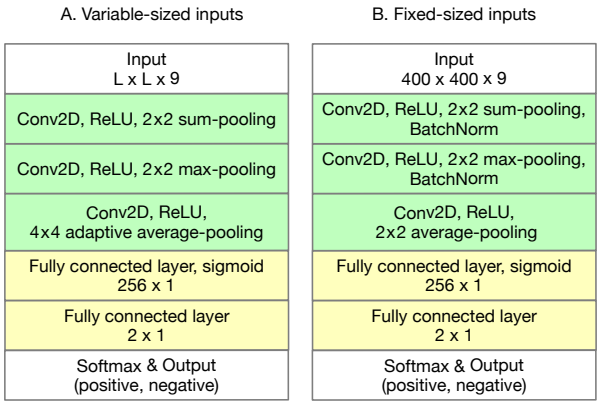

Figure 2. Our proposed ConvNet architecture. For variable-size inputs, we use adaptive average-pooling before feeding data into fully connected layer.

dred nucleotides. The longest and shortest RNA sequences consist of 398 and 45 nucleotides, respectively.

In our model building phase, 90% of the human and cross-species datasets is used as training set. The remaining 10% of human and cross-species datasets has been separated to be used as test set after the model was built, as well as the full new dataset. The training set is splitted into 5 folds for cross validation. We have not used new dataset at

Type / Dataset	Human	Cross-species	New
Positive samples	863	1677	690
Negative samples	7422	8266	8246

Table 1. Statistics of the datasets. We have to cope with a class imbalance problem.

any point until after the cross validation were finished. The new dataset is only used for testing with a model trained by the cross-species dataset as Tran et al. [44] proposed.

3.2. 5-fold cross validation and test performance

3.2.1 Experiment settings

We use 5-fold cross validation for evaluating models. After building the models, we test the success of the model against previous work done in the field. We used 10% of human and cross-species datasets, and all of the new dataset. For comparison, we used sensitivity (SE), specificity (SP), positive predictive value (PPV), F-score, g-mean, Area Under Receiver Operating Characteristic (AUROC), and Area Under Precision-Recall (AUPR). The results are shown in the Table 2, 3, 4. The results of miRBoost, CSHMM, triplet-SVM, microPred, MIReNA and deepMiRGene are referred from [35]. Formulas for metrics described are caculated as following: TP: $\sum$ true positive, TN: $\sum$ true negative, FP: $\sum$ false positive, FN: $\sum$ false negative, SE = TP / (TP + FN), SP = TN / (TN + FP), PPV (precision) = TP / (TP +

Methods	SE	SP	PPV	F-score	g-mean	AUROC	AUPR
miRBoost (CV)	0.803	0.988	0.887	0.843	0.891	-	-
CSHMM (CV)	0.713	0.777	0.559	0.570	0.673	-	-
triplet-SVM (CV)	0.669	0.986	0.851	0.749	0.812	0.957	0.854
microPred (CV)	0.763	**0.989**	**0.888**	0.820	0.869	0.974	0.890
MIReNA (CV)	0.818	0.943	0.624	0.708	0.878	-	-
deepMiRGene (CV)	0.799	0.988	0.885	0.839	0.888	0.984	0.915
deepMiRGene-rerun (CV)	0.818	0.977	0.806	0.811	0.894	0.975	0.876
Proposed fixed-sized (CV)	**0.878**	0.978	0.827	0.849	**0.926**	0.984	0.915
Proposed variable-sized (CV)	0.835	0.985	0.868	**0.851**	0.907	**0.985**	**0.922**
Without pairing matrix (CV)	0.722	0.938	0.594	0.646	0.823	0.909	0.726
miRBoost (test)	0.884	0.969	0.768	0.822	0.925	-	-
CSHMM (test)	0.616	0.978	0.768	0.684	0.777	-	-
triplet-SVM (test)	0.744	**0.992**	0.914	0.821	0.859	0.947	0.830
microPred (test)	0.779	0.988	0.882	0.827	0.877	0.980	0.892
MIReNA (test)	0.826	0.941	0.617	0.706	0.881	-	-
deepMiRGene (test)	0.822	**0.992**	**0.919**	0.868	0.903	0.981	0.918
deepMiRGene-rerun (test)	0.767	0.989	0.892	0.825	0.871	0.981	0.900
Proposed fixed-sized (test)	**0.930**	0.984	0.870	**0.899**	**0.957**	0.983	**0.946**
Proposed variable-sized (test)	0.884	0.991	0.916	**0.899**	0.936	**0.986**	0.934
Without pairing matrix	0.628	0.956	0.621	0.624	0.775	0.892	0.669

Table 2. Results on human dataset. We outperform state-of-the-art in 5 out of 7 measurements.

Methods	SE	SP	PPV	F-score	g-mean	AUROC	AUPR
miRBoost (CV)	0.861	0.977	0.884	0.872	0.917	-	-
CSHMM (CV)	0.826	0.576	0.533	0.564	0.524	-	-
triplet-SVM (CV)	0.735	0.967	0.819	0.775	0.843	0.943	0.869
microPred (CV)	0.825	0.975	0.875	0.848	0.897	0.970	0.873
MIReNA (CV)	0.766	0.952	0.765	0.765	0.854	-	-
deepMiRGene (CV)	0.886	**0.982**	**0.911**	**0.898**	0.933	**0.985**	0.927
deepMiRGene-rerun (CV)	0.883	0.970	0.861	0.871	0.926	0.981	0.929
Proposed fixed-sized (CV)	**0.903**	0.978	0.894	**0.898**	**0.940**	**0.985**	**0.936**
Proposed variable-sized (CV)	0.881	0.981	0.906	0.893	0.930	0.983	**0.936**
Without pairing matrix (CV)	0.785	0.928	0.704	0.737	0.853	0.926	0.828
miRBoost (test)	0.856	0.844	0.526	0.651	0.850	-	-
CSHMM (test)	0.749	0.960	0.791	0.769	0.848	-	-
triplet-SVM (test)	0.760	0.977	0.870	0.812	0.862	0.952	0.908
microPred (test)	0.814	0.985	0.919	0.863	0.896	0.963	0.906
MIReNA (test)	0.796	0.950	0.764	0.780	0.870	-	-
deepMiRGene (test)	0.900	0.983	0.913	0.906	0.940	0.984	**0.955**
deepMiRGene-rerun (test)	0.862	0.984	0.917	0.889	0.921	0.984	0954
Proposed fixed-sized (test)	**0.904**	0.982	0.910	**0.907**	**0.942**	0.983	0.951
Proposed variable-sized (test)	0.880	**0.988**	**0.936**	**0.907**	0.933	**0.985**	0.950
Without pairing matrix	0.695	0.955	0.758	0.725	0.815	0.914	0.798

Table 3. Results on cross-species dataset. We achieve comparative results with deepMiRGene and outperform other methods.

FP), F-score = 2TP / (2TP + FP + FN), g-mean = $\sqrt{\mathrm{SE} \cdot \mathrm{SP}}$.

In our comparison metrics, higher sensitivity means the proportion of actual positives that are correctly identified is high. Higher specificity means the proportion of actual negatives that are correctly identified is high. Higher PPV means more precision. AUROC measures the success of

Methods	SE	SP	PPV	F-score	g-mean	AUROC	AUPR
miRBoost	**0.921**	0.936	0.609	0.733	0.928	-	-
CSHMM	0.536	0.069	0.046	0.085	0.192	-	-
triplet-SVM	0.721	**0.981**	0.759	0.740	0.841	0.934	0.766
microPred	0.728	0.970	0.672	0.699	0.840	0.940	0.756
MIReNA	0.450	0.941	0.392	0.419	0.650	-	-
deepMiRGene	0.917	0.964	0.682	0.782	0.941	**0.981**	0.808
deepMiRGene-rerun	0.899	0.968	0.700	0.787	0.933	0.980	0.798
Proposed fixed-sized	0.917	0.967	0.696	0.792	**0.942**	0.979	**0.864**
Proposed variable-sized	0.859	**0.981**	**0.779**	**0.817**	0.918	0.979	0.818
Without pairing matrix	0.855	0.938	0.535	0.658	0.896	0.958	0.798

Table 4. Results on new dataset. We outperform state-of-the-art in 4 out of 7 measurements, especially for F-score.

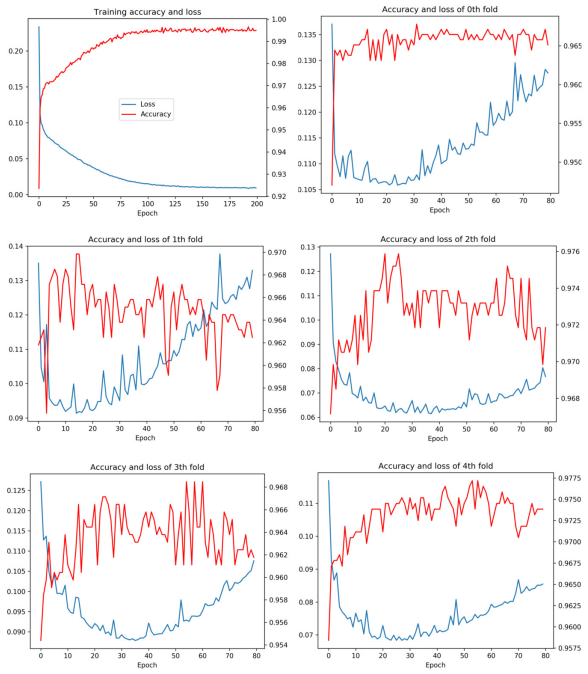

Figure 3. Accuracy and loss curve of human training and 5-fold validation datasets, where red stands for accuracy curve and blue stands for loss curve.

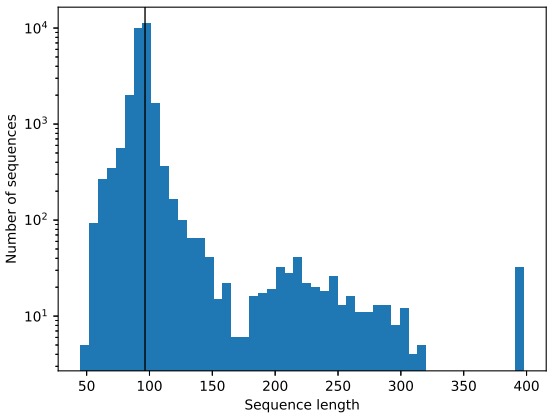

Figure 4. Histogram of length of precursor microRNA sequences. The sequence length is variable, where the max, min and average are 45, 398 and 96.

distinguishment between two groups, positive and negative in our case. Since we are dealing with a skewed dataset (there are a lot more negative sequences), AUPR is important because it gives an informative picture of the performance.

3.2.2 5-fold cross validation results

Looking at the 5-fold cross validation results, our approach shows outperformed results on human dataset and comparative results on cross species dataset, especially for sensitivity and F-score. Especially in the human dataset, the sensitivity (0.878) is 6% better than the best of others while the F-score is also better than others. Therefore, it means our approach will get promising results on test phase and it could handle the imbalance data as well.

3.2.3 Test results

Looking at the test results, we can see that our proposed method outperforms the previous methods significantly. In human dataset, the test results of every metric except two (SP and PPV) are better than all other methods. For cross-species dataset, it outperforms other methods in SE, F-score and g-mean. In new dataset, it achieves the best overall performance with highest scores in F-score, g-mean and AUPR. The SP and PPV is strong relative to each other. When SP is very high, the PPV also high too. In human and cross-species dataset, we achieve best SE. That is why

the SP and PPV decrease. But in all dataset, we achieve the best F-score (0.899 for human and 0.907 for cross-species). Even in metrics where our methods does not have the best score, it is really close to the best in almost all the cases. In the new dataset, we can see that our method gives us an improvement on deepMiRGene.

We can also see how deep learning approaches without handcrafted features, deepMiRGene and our model, perform compared to previous machine learning methods. We can see they have better results in human dataset in every metric except for SP, where triplet-SVM gives the same score with deepMiRGene. For both CV and test parts of cross-species, deep learning approaches perform better than all of the previous methods. For new dataset, miRBoost gives the best result in terms of SE and triplet-SVM has the same SP with our proposed model, but in every other category, deep learning approaches have better results. Considering all of the datasets and results, we can see that deep learning methods perform better than machine learning solutions.

When we compare deepMiRGene with our proposal, we observe that our approach gives better results in all test datasets. We rerun their model with the code they provided, and also included the rerun results in our tables. For human dataset, our approach gives improvements in every metric except for SP and PPV. For cross-species dataset, our approach gives better results in terms of every metric other than AUPR. Lastly, for the new dataset, our model performs better in every metric except for AUROC. The results when training on cross-species dataset and testing on new dataset demonstrate that our approach still adapts to mix species pre-miRNA generalization.

It is also important to compare our three approaches with each other. It is easy to observe that without pairing matrix, the results are much worse in each dataset and each metric. Therefore, the pairing matrix is the most important part of our method. Fixed-sized and variable-sized inputs give similar results for most of the categories. Fixed-sized inputs has dimensions of $400 \times 400 \times 9$ but in variable-sized, the dimensions are $L \times L \times 9$, where L is the sequence length. With fixed-sized inputs, we need more RAM and GPU to train than variable-sized inputs. So that, we encourage to use variable-sized inputs for this task.

Considering all the results and comparisons with other methods, the results show that our newly proposed method gives the best performance in identifying pre-miRNAs.

4. Discussion and conclusions

The results when using the pairing matrix are much better than just using the input sequence. It means the pairing matrix and secondary structure play an important role in miRNA identification. At that time, the feature space is bigger. Therefore, we need a big dataset for a good data generalization. In addition to small number of samples, the classes are also imbalanced. Thus it is difficult for deep learning model to learn. We could improve the identification by collecting more data then using bigger neural networks such as AlexNet [25], ResNet [16].

We proposed a new method to represent the RNA input sequence using the pairing matrix which is generated from its secondary structure. With the pairing matrix, our convolutional neural network architecture can extract the structural diversity characteristics, minimum free energy, and hairpin structures easily. We obtain state-of-the-art results, especially for F-score, on three benchmark datasets `human`, `cross-species` and `new`. In the future, we want to try using long short term memory neural networks (LSTMs) [17] and attention mechanism for this input representation because LSTMs seem good for machine learning tasks with sequence data. Moreover, we will try to expand our method to other tasks which are relative to miRNA such as miRNA target prediction and miRNA function prediction [38].

References

[1] S. Agarwal, C. Vaz, A. Bhattacharya, and A. Srinivasan. Prediction of novel precursor miRNAs using a context-sensitive hidden Markov model (CSHMM). *BMC bioinformatics*, 11(1):S29, 2010.

[2] M. AlQuraishi. End-to-end differentiable learning of protein structure. *bioRxiv*, 265231, 2018.

[3] D. P. Bartel. MicroRNAs: genomics, biogenesis, mechanism, and function. *Cell*, 116(2):281–297, 2004.

[4] R. Batuwita and V. Palade. microPred: effective classification of pre-miRNAs for human miRNA gene prediction. *Bioinformatics*, 25(8):989–995, 2009.

[5] I. Bentwich, A. Avniel, Y. Karov, R. Aharonov, S. Gilad, O. Barad, A. Barzilai, P. Einat, U. Einav, E. Meiri, et al. Identification of hundreds of conserved and nonconserved human microRNAs. *Nature genetics*, 37(7):766, 2005.

[6] E. Bindewald, T. Kluth, and B. A. Shapiro. Cylofold: secondary structure prediction including pseudoknots. *Nucleic acids research*, 38(suppl_2):W368–W372, 2010.

[7] D. Bu, K. Yu, S. Sun, C. Xie, G. Skogerbø, R. Miao, H. Xiao, Q. Liao, H. Luo, G. Zhao, et al. Noncode v3. 0: integrative annotation of long noncoding RNAs. *Nucleic acids research*, 40(D1):D210–D215, 2011.

[8] A. Churkin and D. Barash. Rna dot plots: an image representation for rna secondary structure analysis and manipulations. *Wiley Interdisciplinary Reviews: RNA*, 4(2):205–216, 2013.

[9] M. D. S. Demirci, J. Baumbach, and J. Allmer. On the performance of pre-microRNA detection algorithms. *Nature communications*, 8(1):330, 2017.

[10] W. Dong, C. Qiu, H. Zhang, J. Wang, Q. Cui, and Y. Yin. Human microRNA oncogenes and tumor suppressors show significantly different biological patterns: from functions to targets. *PLoS ONE*, 5(9), 2010.

[11] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[12] W. M. Fitch. Locating gaps in amino acid sequences to optimize the homology between two proteins. *Biochemical genetics*, 3(2):99–108, 1969.

[13] V. Golkov, M. J. Skwark, A. Golkov, A. Dosovitskiy, T. Brox, J. Meiler, and D. Cremers. Protein contact prediction from amino acid co-evolution using convolutional networks for graph-valued images. In *Annual Conference on Neural Information Processing Systems (NIPS)*, Barcelona, Spain, Dec. 2016.

[14] S. Griffiths-Jones, R. J. Grocock, S. Van Dongen, A. Bateman, and A. J. Enright. miRBase: microRNA sequences, targets and gene nomenclature. *Nucleic acids research*, 34(suppl_1):D140–D144, 2006.

[15] M. W. S. M. S. Gudy, Adam and I. Makaowska. HuntMi: an efficient and taxon-specific approach in pre-miRNA identification. *BMC bioinformatics*, 14(1), 2013.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[17] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[18] I. L. Hofacker. Vienna RNA secondary structure server. *Nucleic acids research*, 31(13):3429–3431, 2003.

[19] M. D. Jansson and A. H. Lund. MicroRNA and cancer. *Molecular oncology*, 6(6):590–610, 2012.

[20] L. Jiang, J. Zhang, P. Xuan, and Q. Zou. BP neural network could help improve pre-miRNA identification in various species. *BioMed research international*, 2016, 2016.

[21] P. Jiang, H. Wu, W. Wang, W. Ma, X. Sun, and Z. Lu. Mipred: classification of real and pseudo microRNA precursors using random forest prediction model with combined features. *Nucleic Acids Research*, 35(suppl 2):W339–W344, 2007.

[22] M. W. Jones-Rhoades, D. P. Bartel, and B. Bartel. MicroRNAs and their regulatory roles in plants. *Annu. Rev. Plant Biol.*, 57:19–53, 2006.

[23] T. Kin, K. Yamada, G. Terai, H. Okida, Y. Yoshinari, Y. Ono, A. Kojima, Y. Kimura, T. Komori, and K. Asai. fRNAdb: a platform for mining/annotating functional RNA candidates from non-coding RNA sequences. *Nucleic acids research*, 35(suppl_1):D145–D148, 2006.

[24] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[26] A. Krogh and J. A. Hertz. A simple weight decay can improve generalization. In J. E. Moody, S. J. Hanson, and R. P. Lippmann, editors, *Advances in Neural Information Processing Systems 4*, pages 950–957. Morgan-Kaufmann, 1992.

[27] J. Kukačka, V. Golkov, and D. Cremers. Regularization for deep learning: A taxonomy. *arXiv preprint arXiv:1710.10686*, 2017.

[28] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[29] L. Lestrade and M. J. Weber. snoRNA-LBME-db, a comprehensive database of human H/ACA and C/D box snoRNAs. *Nucleic acids research*, 34(suppl_1):D158–D162, 2006.

[30] J. V. Maizel and R. P. Lenk. Enhanced graphic matrix analysis of nucleic acid and protein sequences. *Proceedings of the National Academy of Sciences*, 78(12):7665–7669, 1981.

[31] A. Mathelier and A. Carbone. MIReNA: finding microRNAs with high accuracy and no learning at genome scale and from deep sequencing data. *Bioinformatics*, 26(18):2226–2234, 2010.

[32] A. Mathuriya, D. A. Bader, C. E. Heitsch, and S. C. Harvey. GTfold: a scalable multicore code for RNA secondary structure prediction. In *Proceedings of the 2009 ACM symposium on Applied Computing*, pages 981–988. ACM, 2009.

[33] J. Mendenhall and J. Meiler. Improving quantitative structure–activity relationship models using artificial neural networks trained with dropout. *Journal of computer-aided molecular design*, 30(2):177–189, 2016.

[34] K. L. S. Ng and S. K. Mishra. De novo SVM classification of precursor microRNAs from genomic pseudo hairpins using global and intrinsic folding measures. *BMC bioinformatics*, 23(11), 2007.

[35] S. Park, S. Min, H.-S. Choi, and S. Yoon. Deep recurrent neural network-based identification of precursor micrornas. In *Advances in Neural Information Processing Systems*, pages 2891–2900, 2017.

[36] S. Pfeffer, M. Zavolan, F. A. Grässer, M. Chien, J. J. Russo, J. Ju, B. John, A. J. Enright, D. Marks, C. Sander, et al. Identification of virus-encoded microRNAs. *Science*, 304(5671):734–736, 2004.

[37] M. E. Rahman, R. Islam, S. Islam, S. I. Mondal, and M. R. Amin. MiRANN: A reliable approach for improved classification of precursor microRNA using artificial neural network model. *Genomics*, 99(4):189–194, 2012.

[38] S. Singh, R. G. Benton, A. Singh, and A. Singh. Machine learning techniques in exploring microRNA gene discovery, targets, and functions. In *Bioinformatics in MicroRNA Research*, pages 211–224. Springer, 2017.

[39] M. Spencer, J. Eickholt, and J. Cheng. A deep learning network approach to ab initio protein secondary structure prediction. *IEEE/ACM transactions on computational biology and bioinformatics (TCBB)*, 12(1):103–112, 2015.

[40] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[41] R. Staden. An interactive graphics program for comparing and aligning nucleic acid and amino acid sequences. *Nucleic Acids Research*, 10(9):2951–2961, 1982.

[42] S. Tempel, B. Zerath, F. Zehraoui, F. Tahi, et al. miRBoost: boosting support vector machines for microRNA precursor classification. *RNA*, 2015.

[43] J. Thomas, S. Thomas, and L. Sael. DP-miRNA: An improved prediction of precursor microRNA using deep learning model. In *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 96–99, Feb 2017.

[44] V. D. T. Tran, S. Tempel, B. Zerath, F. Zehraoui, and F. Tahi. miRBoost: boosting support vector machines for microRNA precursor classification. *RNA*, 21(5):775, 2015.

[45] K. U. Tüfekci, M. G. Öner, R. L. J. Meuwissen, and Ş. Genç. The role of microRNAs in human diseases. In *miRNomics: MicroRNA Biology and Computational Analysis*, pages 33–50. Springer, 2014.

[46] C. Xue, F. Li, T. He, G.-P. Liu, Y. Li, and X. Zhang. Classification of real and pseudo microRNA precursors using local structure-sequence features and support vector machine. *BMC bioinformatics*, 6(1), 2005.