Pre-miRNA Classification via Combinatorial Feature Mining and Boosting

Ling Zhong, Jason T. L. Wang, Dongrong Wen
Department of Computer Science
New Jersey Institute of Technology
Newark, NJ 07102, USA
Email: {lz25, wangj, dw39}@njit.edu

Bruce A. Shapiro
Center for Cancer Research Nanobiology Program
National Cancer Institute
Frederick, MD 21702, USA
Email: shapirbr@mail.nih.gov

***Abstract*—MicroRNAs (miRNAs) are non-coding RNAs with approximately 22 nucleotides (nt) that are derived from precursor molecules. These precursor molecules or pre-miRNAs often fold into stem-loop hairpin structures. However, a large number of sequences with pre-miRNA-like hairpins can be found in genomes. It is a challenge to distinguish the real pre-miRNAs from other hairpin sequences with similar stem-loops (referred to as pseudo pre-miRNAs). Several computational methods have been developed to tackle this challenge. In this paper we propose a new method, called MirID, for identifying and classifying microRNA precursors. We collect 74 features from the sequences and secondary structures of pre-miRNAs; some of these features are taken from our previous studies on non-coding RNA prediction while others were suggested in the literature. We develop a combinatorial feature mining algorithm to identify suitable feature sets. These feature sets are then used to train support vector machines to obtain classification models, based on which classifier ensemble is constructed. Finally we use a boosting algorithm to further enhance the accuracy of the classifier ensemble. Experimental results on a variety of species demonstrate the good performance of the proposed method, and its superiority over existing tools.**

***Keywords*-miRNA precursor; ensemble method; support vector machine; AdaBoost**

I. Introduction

MicroRNAs (miRNAs) are non-coding RNAs (ncRNAs) of approximately 22 nucleotides that are known to regulate post-transcriptional expression of protein-coding genes [2]. They are derived from pre-miRNAs that often fold into stem-loop hairpin structures. These characteristic stem-loop structures are highly conserved in different species. One challenging research problem is to distinguish pre-miRNAs from other sequences with similar stem-loop structures (referred to as pseudo pre-miRNAs). In this paper we present a novel combinatorial feature mining method for pre-miRNA classification. Our method, named MirID, identifies and classifies an input RNA sequence as a pre-miRNA or not. Experimental results demonstrate the effectiveness of MirID.

II. Materials and Methods

A. Datasets

We collected real pre-miRNAs and pseudo pre-miRNAs from eleven species. These RNA sequences were evenly divided into training data and test data. Table I presents a summary of the data. The first column of Table I shows a species or organism name. The second column of Table I shows the number of training sequences followed by the number of test sequences with respect to the organism's real pre-miRNAs. The third column of Table I shows the number of training sequences followed by the number of test sequences with respect to the organism's pseudo pre-miRNAs. As an example, referring to *Arabidopsis thaliana* in Table I, its training set contains 66 real pre-miRNAs and 114 pseudo pre-miRNAs; its test set contains 67 real pre-miRNAs and 114 pseudo pre-miRNAs.

Table I
Summary of datasets

Species	Real pre-miRNA	Pseudo pre-miRNA
Arabidopsis thaliana	66, 67	114, 114
Caenorhabditis briggsae	66, 67	400, 400
Caenorhabditis elegans	84, 85	288, 289
Danio rerio	170, 170	468, 468
Drosophila melanogaster	81, 82	370, 370
Drosophila pseudoobscura	98, 99	115, 116
Epstein barr virus	12, 13	30, 31
Gallus gallus	241, 241	336, 336
Mus musculus	315, 315	780, 781
Oryza sativa	172, 172	358, 359
Rattus norvegicus	193, 193	250, 250

The real pre-miRNAs were downloaded from miRBase available at http://www.mirbase.org/ [4]. We used RNAfold [3] to predict the secondary structures of all the RNA sequences. The lengths of the real pre-miRNAs in the dataset ranged from 60 to 120 nt. The pseudo pre-miRNAs used in this study were collected from GenBank (http://www.ncbi.nlm.nih.gov/genbank/). Following [7], we searched for the protein-coding regions of the genome sequences of the species in Table I, and divided the regions into shorter sequences, each of them having 100 nucleotides. The pseudo pre-miRNAs were chosen from those 100-nucleotide sequences. The criteria used in choosing the pseudo pre-miRNAs are: (i) they must contain at least 18 base pairs, including Watson-Crick and GU wobble base pairs, on the stem region of the stem-loop structure, and (ii) their secondary structures have a maximum of -15 kcal/mol free energy without multiple hairpin loops [4]. These criteria

ensure that the stem-loop structures of the pseudo pre-miRNAs are similar to those of the real pre-miRNAs.

B. Feature Pool

In designing our pre-miRNA classification method, we examined multiple features extracted from a pre-miRNA sequence and its secondary structure. Some of these features were taken from our previous studies on ncRNA prediction while others were suggested in the literature [5], [7], [9]. These features included the sequence length, the number of base pairs, GC content, the ratio between the number of base pairs and the sequence length, the number of nucleotides contained in the hairpin loop (i.e., the loop size), the free energy of the sequence's secondary structure obtained from RNAfold [3], the number of bulge loops, and the size of the largest bulge loop in the secondary structure.

In addition, we considered the features described in [9]. These features included the difference of the lengths of the two tails in the secondary structure where a tail represented the strand of unpaired bases in the $5'$ or $3'$ end of the structure, the number of tails, and the length of the larger tail. Besides, several combined features were considered. They included the length difference of two tails plus the larger tail length, the size of the hairpin loop plus the larger tail length, the size of the hairpin loop plus the largest bulge size, the ratio between the larger tail length and the sequence length, the ratio between the size of the hairpin loop and the sequence length, the ratio between the largest bulge size and the sequence length, the ratio between the largest bulge size and the number of base pairs, the normalized free energy [6], which is the minimum free energy of the sequence's secondary structure divided by the sequence length, and the ratio between the normalized free energy and the GC content.

The next set of features included the triplets of structure-sequence elements described in [7]. Here we used the dot-bracket notation [3] to represent an RNA secondary structure. A triplet is composed of three contiguous structure elements (bases or base pairs) that correspond to three contiguous nucleotides along with the middle nucleotide. There are 32 triplets, and hence 32 such features in total.

Finally we considered the symmetric and asymmetric loops defined in [5]. We refer to the portion of the sequence from the $5'$ end to the hairpin loop as the left arm, and the portion of the sequence from the hairpin loop to the $3'$ end as the right arm. In a symmetric (internal) loop, the number of nucleotides in the left arm equals the number of nucleotides in the right arm. In an asymmetric (internal) loop, the number of nucleotides in the left arm is different from the number of nucleotides in the right arm. Features related to these loops included the size of each loop, the average size of the loops, and the average distance between the loops. Other features included the proportion of A/C/G/U in the stem, and the proportion of A-U/C-G/G-U base pairs in the stem. Totally, there are 74 features in the feature pool.

C. Combinatorial Feature Mining

MirID adopts a novel feature mining algorithm for pre-miRNA classification. Initially the algorithm randomly generates N feature sets from the feature pool. (The default value of N is 100.) Each feature set contains between 1 and 150 features, randomly chosen with replacement from the feature pool. Some features may repeatedly occur in a feature set (thus a bagging approach is used here). Duplicate features have more weights than the other features in the feature set. The numbers 1 and 150 are chosen, to ensure that there are enough feature sets containing duplicate features. We then build a SVM model based on each feature set using training sequences, and apply the classification model to test sequences to calculate the accuracy of the model. The SVM used in this study is the LIBSVM package downloaded from http://www.csie.ntu.edu.tw/∼cjlin/libsvm/. We use the polynomial kernel provided in the LIBSVM package. The polynomial kernel achieves the best performance among all kernel functions included in the package.

Then, we remove the SVM models whose accuracies are less than a user-determined threshold t. (The default value of t is 0.8.) The feature sets used to build those removed SVM models are also eliminated from further consideration. We construct a classifier ensemble from the remaining SVM models. The ensemble works by taking the majority vote from the individual SVM models used to build the classifier ensemble. This ensemble will be refined through several iterations until its accuracy cannot be enhanced further. In each iteration, the user-determined threshold t is incremented by a $step$ value, so that more accurate SVM models are used to construct a (hopefully) better classifier ensemble in the next iteration. (The default value of $step$ is 0.005.)

It is likely that different combinations of remaining features may yield an even better classifier. Our algorithm then performs pairwise *merge* and *split* operations on the set S_b of feature sets used to build the best classifier ensemble obtained so far. In doing so, MirID takes four steps: (1) picks each pair of feature sets s_1 and s_2 in S_b; (2) merges s_1 and s_2 into a single feature set s_3 with, say p, features; (3) randomly generates a number q, $q < p$; (4) randomly assigns q features in s_3 into a set s'_1 and assigns the remaining $p-q$ features into another set s'_2. Thus, these four steps take two feature sets s_1 and s_2 in S_b as input and produce two new feature sets s'_1 and s'_2 as output.

These pairwise merge and split operations are applied to the feature sets used to build the best classifier ensemble obtained so far, to generate new feature sets. The new feature sets are then used to build new SVM models. Accurate new SVM models, whose accuracies are greater than or equal to the newly computed threshold t, are then used to build a new classifier ensemble. This procedure is repeated several times

Procedure: Combinatorial Feature Mining
Input: Features extracted from pre-miRNA sequences of a species.
Output: A classifier ensemble and its component SVM models (feature sets).

```
1.  acc := 0, acc_b := −∞, t := 0.8, step := 0.005, N := 100, S := ∅, S_r := ∅, S_b := ∅;
2.  randomly generate N feature sets and S := {N feature sets};
3.  while acc_b < acc do
4.      acc_b := acc;
5.      acc := 0;
6.      t := 0.8;
7.      build a SVM model for each feature set in S;
8.      remove SVM models/feature sets with accuracy < t from S and
        S_r := {remaining feature sets};
9.      build a classifier ensemble based on the feature sets in S_r;
10.     while accuracy of the classifier ensemble > acc do
11.         acc := accuracy of the classifier ensemble;
12.         S := S_r;
13.         t := t + step;
14.         remove SVM models/feature sets with accuracy < t from S and
            S_r := {remaining feature sets};
15.         build a classifier ensemble based on the feature sets in S_r;
16.     end while
17.     if acc_b < acc then
18.         S_b := S;
19.         perform merge/split on feature sets in S_b to generate new feature sets and
            S := {new feature sets};
20.     end if
21. end while
22. return S_b and the classifier ensemble constructed based on S_b;
```

Figure 1. Algorithm for combinatorial feature mining.

to obtain a best classifier ensemble. Figure 1 summarizes our feature mining algorithm.

D. Boosting

The performance of a classification algorithm can be further enhanced through boosting. We apply AdaBoost [1] to the classifier ensemble produced by our feature mining algorithm. Specifically, we treat the classifier ensemble as a weak classifier and continue refining it into a strong classifier through an iterative procedure. Let X be a set of sequences $x_1, x_2, \ldots, x_m$ where x_i, $1 \leq i \leq m$, is associated with a label y_i such that

$$y_i = \begin{cases} +1 & \text{if } x_i \text{ is a real pre-miRNA} \\ -1 & \text{if } x_i \text{ is a pseudo pre-miRNA} \end{cases}$$

The AdaBoost algorithm works with K iterations. (The default value of K is 20.) In iteration k, $1 \leq k \leq K$, the algorithm updates a weight function W_k as explained below, which will be used in selecting training sequences in iteration $k+1$. Initially, every sequence has an equal weight, i.e. $W_0(x_i) = 1/m$, $1 \leq i \leq m$. In iteration k, the algorithm samples 1/3 sequences with replacement from X based on the weight function W_{k-1} to form a training set X_k. The set X_k is then used to train a weak classifier H_k, which classifies each sequence x_i as either a real pre-miRNA or a pseudo pre-miRNA. That is,

$$H_k(x_i) = \begin{cases} +1 & H_k \text{ classifies } x_i \text{ as a real pre-miRNA} \\ -1 & H_k \text{ classifies } x_i \text{ as a pseudo pre-miRNA} \end{cases}$$

Let $E_k = \{x_i | H_k(x_i) \neq y_i\}$. The error rate ϵ_k of H_k is:

$$\epsilon_k = \sum_{x_i \in E_k} W_{k-1}(x_i) \tag{1}$$

Let

$$\alpha_k = \frac{1}{2} ln(\frac{1-\epsilon_k}{\epsilon_k}) \tag{2}$$

The algorithm updates W_k for each sequence x_i, $1 \leq i \leq m$, as follows:

$$\begin{aligned} W_k(x_i) &= \begin{cases} \frac{W_{k-1}(x_i)}{Z_k} \times e^{-\alpha_k} & \text{if } H_k(x_i) = y_i \\ \frac{W_{k-1}(x_i)}{Z_k} \times e^{\alpha_k} & \text{if } H_k(x_i) \neq y_i \end{cases} \\ &= \frac{W_{k-1}(x_i) exp(-\alpha_k y_i H_k(x_i))}{Z_k} \end{aligned} \tag{3}$$

where Z_k is a normalization factor chosen such that W_k is normally distributed. Thus, the sequences causing classification errors in iteration k will have a greater probability

of being selected as training sequences for constructing the weak classifier H_{k+1} in iteration $k+1$. Using this technique, each weak classifier should have greater accuracy than its predecessor. The final, strong classifier H combines the vote of each individual weak classifier H_k, $1 \leq k \leq K$, where the weight of each weak classifier's vote is a function of its accuracy. Specifically, for an unlabeled test sequence x, $H(x)$ is calculated as follows:

$$H(x) = sign(\sum_{k=1}^{K} \alpha_k H_k(x)) \quad (4)$$

The function *sign* indicates that if the sum inside the parentheses is greater than or equal to zero, then H classifies x as positive (i.e. a real pre-miRNA); otherwise H classifies x as negative (i.e. a pseudo pre-miRNA).

III. Experiments and Results

We compared our method with two closely related methods, PMirP [8] and TripletSVM [7]. Like our method, both PMirP and TripletSVM were implemented using support vector machines. PMirP adopted a hybrid coding scheme, combining features such as free bases, base pairs, minimum free energy of secondary structure, among others. TripletSVM used triplets of structure-sequence elements, which also were included in our feature pool.

The performance measure used here is *accuracy*, defined as follows. A method is said to classify a test sequence correctly if the sequence is a real pre-miRNA (pseudo pre-miRNA, respectively) and the method indicates that the sequence is indeed a real pre-miRNA (pseudo pre-miRNA, respectively). A method is said to classify a test sequence incorrectly if the sequence is a real pre-miRNA (pseudo pre-miRNA, respectively) but the method mistakenly indicates that the sequence is a pseudo pre-miRNA (real pre-miRNA, respectively). For each species, the accuracy of a method is defined as the number of correctly classified test sequences of that species divided by the total number of test sequences of that species. Since our feature mining procedure is a randomized algorithm, we ran MirID thirty times and calculated the average.

Table II shows the accuracies of the three methods on the species taken from Table I. These species were used to pretrain PMirP and TripletSVM, and available from their web servers. For each species, the highest accuracy yielded by a tool is in bold. It can be seen that MirID achieves better performance than the related methods.

Table II
Comparison of three pre-miRNA classification methods

Species	TripletSVM	PMirP	MirID
Arabidopsis thaliana	0.92	0.96	**0.98**
Caenorhabditis briggsae	0.96	0.97	**1.00**
Caenorhabditis elegans	0.86	0.86	**0.93**
Danio rerio	0.67	0.83	**1.00**
Drosophila melanogaster	0.92	0.96	**1.00**
Drosophila pseudoobscura	0.90	0.92	**1.00**
Epstein barr virus	**1.00**	0.80	**1.00**
Gallus gallus	0.85	**1.00**	**1.00**
Mus musculus	0.94	0.94	**1.00**
Oryza sativa	0.95	**1.00**	**1.00**
Rattus norvegicus	0.80	0.92	**1.00**

IV. Conclusion

In this paper we present a new method (MirID) for pre-miRNA classification. Experimental results showed that MirID outperforms two closely related methods, PMirP and TripletSVM. Since all the three methods were implemented using support vector machines with similar features, we conclude that the superiority of our method is due to its feature mining and boosting algorithms. Future work includes extending these algorithms for classifying other RNA structures.

References

[1] E. Bindewald and B. A. Shapiro, "RNA secondary structure prediction from sequence alignments using a network of k-nearest neighbor classifiers," *RNA*, vol. 12, no. 3, pp. 342-352, 2006.

[2] R. S. Bindra, J. T. L. Wang, and P. S. Bagga, "Bioinformatics methods for studying microRNA and ARE-mediated regulation of post-transcriptional gene expression," *International Journal of Knowledge Discovery in Bioinformatics*, vol. 1, no. 3, pp. 97-112, 2010.

[3] I. L. Hofacker, "Vienna RNA secondary structure server," *Nucleic Acids Res.*, vol. 31, pp. 3429-3431, 2003.

[4] A. Kozomara and S. Griffiths-Jones, "miRBase: integrating microRNA annotation and deep sequencing data," *Nucleic Acids Res.*, vol. 39, pp. D152-D157, 2011.

[5] A. Sewer, N. Paul, P. Landgraf, A. Aravin, S. Pfeffer, M. J. Brownstein, T. Tuschl, E. van Nimwegen, and M. Zavolan, "Identification of clustered microRNAs using an *ab initio* prediction method," *BMC Bioinformatics*, vol. 6, no. 267, 2005.

[6] J. Spirollari, J. T. L. Wang, K. Zhang, V. Bellofatto, Y. Park, and B. A. Shapiro, "Predicting consensus structures for RNA alignments via pseudo-energy minimization," *Bioinformatics and Biology Insights*, vol. 3, pp. 51-69, 2009.

[7] 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*, vol. 6, no. 310, 2005.

[8] D. Zhao, Y. Wang, D. Luo, X. Shi, L. Wang, D. Xu, J. Yu, and Y. Liang, "PMirP: a pre-microRNA prediction method based on structure-sequence hybrid features," *Artificial Intelligence in Medicine*, vol. 49, pp. 127-132, 2010.

[9] Y. Zheng, W. Hsu, M. L. Lee, and L. Wong, "Exploring essential attributes for detecting microRNA precursors from background sequences," Lecture Notes in Computer Science, Springer, vol. 4316, pp. 131-145, 2006.