Chapter 2
MiRNA Recognition with the *yasMiR* System: The Quest for Further Improvements

Daniel Pasailă, Andrei Sucilă, Irina Mohorianu, Ştefan Panţiru, and Liviu Ciortuz

Abstract The paper "Using Base Pairing Probabilities for MiRNA Recognition" by Daniel Pasailă, Irina Mohorianu, and Liviu Ciortuz, that has been published in Proceedings of the International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC) 2008, IEEE Computer Society, pp. 519–525, has introduced a new SVM for microRNA identification, whose novelty is twofolded: first, many of its features incorporate the base-pairing probabilities provided by McCaskill's algorithm, and second the classification performance is improved using a certain similarity ("profile"-based) measure between the training and test microRNAs and a set of carefully chosen ("pivot") RNA sequences. Comparisons with some of the best existing SVMs for microRNA identification proved that our SVM obtains truly competitive results.

Here we add several significant extensions to the work reported in Daniel Pasailă et al. Proceedings of the International (SYNASC) 2008, pp. 519–525: testing this classifier on a more recent version of miRBase (12.0), evaluating the effect of using probabilistic patterns instead of non-probabilistic ones, analysing the discriminative power of different categories of features we used, and automatically searching for good pivot RNA sequences, which are critical for classification in our approach.

1 *yasMiR*: The Approach and Main Results

MicroRNAs (henceforth abbreviated miRNAs) are short RNA molecules that play important gene regulatory roles [1]. In the paper [3],[1] we proposed a miRNA recognition system based on a support vector machine [8], which was subsequently named *yasMiR*.[2] It was mainly built upon features using the base-pair binding probabilities

[1] Daniel Pasailă and Liviu Ciortuz are joint first authors of both paper [3] and the present paper.

[2] ***yasMiR*** is an abbreviation for ***y**et **a**nother **S**VM for **mi**RNA **r**ecognition*.

L. Ciortuz (✉)
Department of Computer Science, "Alexandru Ioan Cuza" University of Iaşi, Iaşi, Romania
e-mail: ciortuz@info.uaic.ro

H.R. Arabnia and Q.-N. Tran (eds.), *Software Tools and Algorithms for Biological Systems*, Advances in Experimental Medicine and Biology 696,
DOI 10.1007/978-1-4419-7046-6_2,

Table 1 Categories of features for *yasMiR* SVM. The *rightmost column* gives the number of features in the respective (sub)category

A	– Alignment scores against pivot sequences, where n is the number of pivots used	n
B	– The probabilistic mean for the number of occurrences for each triplet pattern	32
C	– The mean base-pairing distance	1
	– The overall non base-pairing probability	1
	– The non-pairing probability for each nucleotide	4
	– The sum of pairing probabilities for each pair of nucleotides a and b	10
	– The folding minimum free energy (MFE)	1
	– Dinucleotide frequencies	16
	– The average frequency for each nucleotide	4

provided by McCaskill's algorithm [6], supplemented with some other simple features. *yasMiR*'s features are summarized in Table 1: profile similarity scores against "pivot" RNA sequences (A), means of probabilistic triplet patterns (B), and finally other probabilistic and non-probabilistic features (C).

The two remaining parts of this section will summarize the results that we have obtained when comparing *yasMiR* to Triplet-SVM and miPred (previously detailed in [3]) and when testing this classifier on miRBase 12.0. Section 2 will present the effect of using probabilistic patterns instead of non-probabilistic ones in *yasMiR*, and then it will analyse the discriminative power of different categories of features we used. Section 3 will detail our work on the automatic search for good pivot RNA sequences, which are critical for miRNA classification in our approach.

1.1 Comparisons with Triplet-SVM and miPred

We compared *yasMiR* first to the Triplet-SVM classifier [2], after having trained our SVM on the same dataset as Triplet-SVM. The training set included 163 human pre-miRNAs from miRBase registry version 5.0 and 168 pseudo pre-miRNA like hairpins as negative examples. A fivefold cross-validation accuracy of 96.07% was obtained on this training set. On the test datasets created by the authors of Triplet-SVM, our SVM obtained significantly higher prediction results.

Then we made comparative tests with the miPred classifier [7], the best SVM-based miRNA classifier up to our knowledge. Here, the training set included 200 human pre-miRNAs from miRBase version 8.2 as positive examples, and 400 pseudo pre-miRNA hairpins as negative examples. We obtained at fivefold cross-validation an accuracy of 93.66% on this training set, compared to miPred's 93.50%. Running the same tests as miPred, our SVM obtained similar and sometimes significantly better specificity than miPred. Compared to miPred, one of the advantages of our approach is that it makes no use of so-called normalized features which are based on sequence shuffling; in turn it enables the feature computation in our approach to be much less time consuming.

We also checked whether the Random Forests machine learning algorithm is able to obtain comparable results to SVM (as suggested by MiPred, another SVM for miRNA recognition) when using our set of features. While on many test datasets that we used the answer was positive, the overall conclusion is that RF is not a good enough candidate to replace SVM for pre-miRNA identification using our set of features.

1.2 Results on miRBase 12.0

We have also tested our SVM on sequences from miRBase 12.0 (released in October 2008). For the training set, this time all 678 human miRNAs from miRBase 11.0 were used as positive examples, and also 1,256 sequences from the CODING dataset as negative examples. The testing set includes 3,651 positive examples from miRBase 12.0 and 7,198 negative examples from the CODING dataset. The set of 3,651 positives was obtained by removing the positive training sequences from miRBase 12.0, and using the clustering algorithm presented in the supplementary material of the miPred paper [7] for the removal of similar sequences. First, all the sequences were sorted in decreasing length order, and the first one became the representative of the first cluster. Then, each of the remaining sequences was compared with the existing representatives and added into a cluster if the similarity measure with any representative is above 90%. The remaining set of 3,651 sequences is the final set of representatives, using the above algorithm on miRBase 12.0 (after the training positives have been removed). The BLAST system was used for sequence comparison. On this dataset, the yasMiR system obtained 89.64% sensitivity and 97.37% specificity, with the resulting accuracy of 94.77%.

2 *yasMiR*'s Features Analysis

Since *yasMiR* uses features which are a probabilistic (McCaskill) version of the features used by Triplet-SVM, one would question whether our design decision is indeed justified. Therefore, we made a test in which we replaced the features related to the probabilistic triplet patterns with those taken from the Triplet-SVM package. We used the same procedure as for the comparative test between *yasMiR* and Triplet-SVM. The results we obtained for *yasMiR* (Table 2, first column) are usually slightly (and even significantly) better than the ones we obtained with non-probabilistic features computed for triplet patterns (Table 2, second column). This is especially true for the TE-C (human) and CONSERVED-HAIRPIN datasets.

To further analyse *yasMiR*'s set of features, we also investigated what prediction results are obtained when removing each one of the different categories of features defined for our system (see Table 1).

Table 2 Prediction accuracy (%) results obtained by *yasMiR* on the Triplet-SVM datasets when the features for probabilistic triplet patterns were replaced with their non-probabilistic (Triplet-SVM) counterpart

Test	*yasMiR*	Using non-probabilistic triplet patterns
TE-C: Human pre-miRNAs	100	96.67 (29/30)
TE-C: Pseudo pre-miRNAs	96.2	95.9 (959/1,000)
UPDATED	94.9	94.9 (37/39)
CROSS-SPECIES	95.2	95.87 (557/581)
CONSERVED-HAIRPIN	94.23	93.09 (2,275/2,444)

Table 3 Prediction results for *yasMiR* on miPred datasets (column 1), when removing one category (A, B or C) of its features (columns 2–4). *Bold faces* designate values which are better than those in column 1

	yasMiR			$B \cup C$			$A \cup C$			$A \cup B$		
Test	Sen.	Acc.	Spec.	Sen.	Acc.	Spec.	Sen.	Acc.	Spec.	Acc.	Sen.	Spec.
TE-H	87.80	93.77	96.74	83.73	93.22	**97.96**	89.43	**94.30**	**96.74**	81.30	91.32	96.34
IE-NH	90.35	94.11	95.99	88.58	92.64	94.68	**93.32**	**94.26**	94.73	84.04	92.26	**96.37**
IE-NC		82.95			78.94			59.84			**91.77**	
IE-M		100			100			6.45			100	

Using the same datasets as miPred, we investigated the effect on accuracy, sensitivity and specificity when removing one of the three categories A, B, or C. It can be easily seen in Table 3 that the prediction results with the complete feature set (found in column 1 of Table 3) are in many cases significantly better than those that have been obtained when a category of features is removed. This is especially true for the IE-NC and IE-M datasets. Going into more details, one can see the following facts:

- Retracting the category A of attributes (see column 2 in Table 3) slightly improves the specificity on TE-H (from 96.74% to 97.96%) at the significant cost of sensitivity (from 87.80% down to 83.73%)
- Retracting the category B of attributes (see column 3 in Table 3) slightly improves some of the statistics we obtained previously for *yasMiR* on TE-H and IE-NH but drastically affects the performance on IE-NC (from 82.95% down to 59.84%) and especially on IE-M (from 100% down to 6.45%)
- Retracting the category C of attributes (see column 4 in Table 3) improves the specificity on IE-NC (from 82.95% up to 91.77%) and on IE-NH (from 95.99% to 96.37%), but significantly affects the sensitivity on TE-H (from 87.80% down to 81.30%) and IE-NH (from 90.35% down to 84.04%).

The above analysis implies that each of these categories of features has its own contribution towards the overall good classification results produced by *yasMiR*.

It is also interesting to note that the categories A and C of attributes are more suitable for the TE-H and IE-NH datasets, while B is indispensable for the IE-NC and IE-M datasets. These facts suggest that there are slightly specialized contributions of these categories of features towards discriminating among different categories of RNA sequences.

For expressing the quality of the ith feature we used the $F1$ and $F2$ scores, defined by the following expressions:

$$F1 = \frac{|\mu_i^+ - \mu_i^-|}{|\sigma_i^+ + \sigma_i^-|}, \quad F2 = \frac{(\mu_i^+ - \bar{\mu}_i)^2 + (\mu_i^- - \bar{\mu}_i)^2}{(\sigma_i^+)^2 + (\sigma_i^-)^2},$$

where μ_i^+/μ_i^-, σ_i^+/σ_i^- denote the means and standard deviations of the positive and negative training datasets for the ith feature. After sorting the features in descending order according to the $F1$ and $F2$ scores, we identified the first three features, and they proved to be the same for both sorting measures:

- Feature D: the overall non base-pairing probability ($F1 = 1.21$ and $F2 = 1.64$)
- Feature E: the folding minimum free energy ($F1 = 0.95$ and $F2 = 0.99$)
- Feature F: the probabilistic feature corresponding to the triplet pattern "…" with the nucleotide C on the middle position ($F1 = 0.93$ and $F2 = 0.90$)

The effects on *yasMiR* when each of these three features is removed are shown in Table 4. It is interesting to note that removal of features D and E has a big impact on the IE-NC and IE-M datasets, while feature F seems to be only slightly affecting the result on the IE-NC dataset. Our opinion is that this last feature is made almost redundant by other features.

We therefore tried feature selection applying the Kolmogorov–Smirnov filter [5] for redundancy elimination on the full set of *yasMiR*'s 169 features including the 100 randomly chosen pivots used so far. The Kolmogorov–Smirnov filtering procedure goes as follows: first we rank and sort the features according to the Symmetrical Uncertainty (SU) score which is a normalized version of the mutual information statistics, and then, starting from the top ranking feature that has not yet been filtered, we eliminate all features of lower rank which are redundant to it, according to the Kolmogorov–Smirnov test, up to a certain confidence level.

Using a 0.95 confidence level, the number of features gets reduced to 144 – remarkably, all but one of the 26 eliminated features are pivots – most of the classification statistics on the miPred's test datasets get improved, as shown in Table 5. At 0.90 confidence, things do not go so well, and unfortunately a 5.55% specificity/accuracy loss is reported on the IE-NC dataset (from 82.95% down to 77.20%). However, it is worth noting that this time 31 pivots got eliminated, together with six non-pivot features.

Table 4 Prediction results for *yasMiR* on miPred datasets when removing one of the features D, E or F. *Bold faces* designate values which are better than those in the first column of Table 3

	$A \cup B \cup C \setminus \{D\}$			$A \cup B \cup C \setminus \{E\}$			$A \cup B \cup C \setminus \{F\}$		
Test	Sen.	Acc.	Spec.	Sen.	Acc.	Spec.	Sen.	Acc.	Spec.
TE-H	86.17	93.76	97.56	**86.17**	93.49	**97.15**	**87.80**	93.49	96.34
IE-NH	**91.24**	**94.99**	**96.87**	**90.45**	**94.40**	**96.37**	**90.45**	**94.14**	95.98
IE-NC		67.68			61.95			79.74	
IE-M		19.35			22.58			100	

Table 5 Prediction results of *yasMiR* on miPred's test datasets using 144 features and respectively 132 features selected from the whole set of 169 features via Kolmogorov–Smirnov redundancy filtering. *Bold faces* designate values that are better than those in the first column of Table 3

	0.95 Confidence			0.90 Confidence		
Test	Sensitivity	Accuracy	Specificity	Sensitivity	Accuracy	Specificity
TE-H	**87.80**	**94.30**	**97.50**	**87.80**	**93.76**	**96.74**
IE-NH	90.14	94.07	**96.03**	**91.08**	93.39	94.55
IE-NC		**83.28**			77.20	
IE-M		**100**			**100**	

3 Automatically Choosing the Pivots

Until now we performed several runs with *yasMiR* using different sets of randomly generated pivots, and we retained the results for the set of pivots that produced the best overall results on the Triplet-SVM and the miPred test datasets. However, one could ask whether we could get better results by automatically selecting (or improving) the set of pivots.

3.1 Using Clustering

Here we report on using choosing "representative" pivots among a pool of candidates, using clustering and the Euclidean distance between the vectors associated with pivots. For each candidate pivot, its vector was obtained by computing the profile similarity measure between the pivot and each of the sequences in the training set (e.g. TR-H).

The left column in Table 6 shows the results we obtained for 200 pivots automatically selected from a pool of 2,000 randomly generated sequences. The *k*-means clustering algorithm was used to get those 2,000 sequences grouped into 50 clusters, and then we randomly selected four pivots from each cluster. The results show that the obtained specificity for *yasMiR* SVM's is slightly lower than that obtained with the manually chosen pivots on miRBase 8.2 (on TE-H: from 96.74% to 95.53%, and on IE-NH: from 95.99% to 94.97%), while the sensitivity decreased significantly (TE-H: from 87.80% to 85.37%, and IE-NH: from 90.35% to 83.58%). Remarkably, the specificity/accuracy of *yasMiR* SVM was dramatically improved for IE-NC (from 82.95% to 93.61%, while miPred reported only 68.68%), and for IE-M the specificity/accuracy was kept at 100%.

These results make us conclude that automatically searching for better pivots is worth further working on. In the following section, we will use the Kolmogorov–Smirnov filter for searching among a large pool of randomly generated pivots.

Table 6 *Left column*: prediction results of *yasMiR* on miPred's test datasets using 200 pivots selected via clustering from a pool of 2,000 randomly generated pivots. *Right column*: prediction results of *yasMiR* on miPred's test datasets using the best 13 pivots selected via Kolmogorov–Smirnov filtering from 10,000 randomly generated pivots. *Bold faces* designate values which are better than those in the first column of Table 3

	SVM			KS		
Test	Sensitivity	Accuracy	Specificity	Sensitivity	Accuracy	Specificity
TE-H	85.37	92.14	95.53	85.37	92.53	**96.74**
IE-NH	83.58	91.17	94.97	86.24	91.35	93.90
IE-NC		**93.61**			**87.44**	
IE-M		**100**			**100**	

3.2 Using the Kolmogorov–Smirnov Filter

The probabilistic alignment scores to pivots used in describing sequences lead to a distance-based description. It is clear that the pivots need not be chosen from positive or negative examples, but at a correct distance from members of these classes. As such, we have tried to implement a non-linear feature selection algorithm to choose a better set of pivots. Such a method is the Kolmogorov–Smirnov filter, which has been reported to work well in conjunction with SVM's. We have implemented such a procedure following directions from [5].

The Kolmogorov–Smirnov filter is divided into two parts. The first part is concerned with ranking the features according to a mutual information measure, and the second part recursively eliminates redundant features. We used a confidence level of 95% for determining whether two features were redundant.

The right column in Table 6 shows that when using the best 13 features selected by the Kolmogorov–Smirnov filter from a large pool of randomly generated pivots, the results obtained by *yasMiR* on the miPred datasets are comparable with those reported in the previous section. On the TE-H dataset, we got a better specificity (96.74%) compared to the one produced via clusterization, while on the IE-NH dataset, the sensitivity improved (from 83.58% to 86.24%) but it still remained significantly lower than the one obtained with hand-chosen pivots (90.35%). On IE-NC, the specificity/accuracy is now at midway between the one obtained via clusterization (93.61%) and the original one, produced by hand-chosen pivots (82.95%). On IE-M, the specificity/accuracy stayed at 100%.

Finally, we would suggest that this method would be best used in conjunction with another feature selection method, where the initial bulk of features would be removed by the Kolmogorov–Smirnov filter, and the final features would be selected by the other, more complex method.

4 Conclusions and Further Work

We proved that the base-pairing probabilities provided by McCaskill's algorithm combined with some other, simple statistical measures make an SVM classifier achieve high pre-miRNA prediction accuracy rates, comparable to the best published results up to our knowledge.

We plan to make direct comparisons with a quite recent kNN-based classifier for non-coding RNAs [9]. Its results seem to be very competitive, due to the use of certain topological features. We will see whether those features could be generalized using again the base-pairing probabilities computed by McCaskill's algorithm. If so, we will check whether adopting them into *yasMiR*'s feature set will make it further improve the quality of pre-miRNA prediction.

It will also be interesting to see whether a more recent work for identifying miRNAs [10], which also used the Triplet-SVM patterns but replaced the automate classifier with a ranking algorithm, will improve its results when replacing the simple triplet features with their enhanced counterpart McCaskill's probabilities.

Acknowledgments LC thanks Dr. Mihaela Zavolan from Biozentrum, University of Basel, Dr. Hélène Touzet from the University of Lille, and Dr. Marti Tammi from the National University of Singapore for useful discussions on miRNA identification.

Availability The source code of our system and the datasets we used can be found at the address www.info.uaic.ro/~ciortuz/yasmir. The technical report [4] offers a unified view on the past [3] and present work on *yasMiR*.

References

1. Andrew Fire, Siqun Xu, Mary Montgomery, Steven Kostas, Samuel Driver, and Craig Mello. Potent and specific genetic interference by double-stranded RNA in Caenorhabditis elegans. *Nature*, 391(6669):806–811, 1998
2. Chenghai Xue, Fei Li, Tao He, Guoping Liu, Yanda Li, and Xuegong Zhang. Classification of real and pseudo microRNA precursors using local structure-sequence features and support vector machine. *BMC Bioinformatics*, 6(310), 2005
3. Daniel Pasailă, Irina Mohorianu, and Liviu Ciortuz. Using base pairing probabilities for MiRNA recognition. In *SYNASC '08: Proceedings of the 2008 10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, pages 519–525, 2008
4. Daniel Pasailă, Irina Mohorianu, Andrei Sucilă, Ştefan Panţiru, and Liviu Ciortuz. Yet another SVM for miRNA recognition: yasMiR, 2010. Technical Report TR-10-01, Faculty of Computer Science, University of Iasi, Romania
5. Jacek Biesiada and Wlodzislaw Duch. Feature selection for high-dimensional data: A Kolmogorov–Smirnov correlation-based filter. *Computer Recognition Systems*, 30:95–103, 2005
6. John S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structures. *Biopolymers*, 29:1105–1119, 1990
7. Kwang Loong Stanley Ng and Santosh Mishra. De novo SVM classification of precursor microRNAs from genomic pseudo hairpins using global and intrinsic folding measures. *Bioinformatics*, 23(11):1321–1330, 2007

8. Nello Cristianini and John Shawe-Taylor. *An introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, New York, NY, USA, 2000
9. Wenjie Shu, Xiaochen Bo, Zhiqiang Zheng, and Shengqi Wang. A novel representation of RNA secondary structure based on element-contact graphs. *BMC Bioinformatics*, 9(1):188, 2008
10. Yunpen Xu, Xuefeng Zhou, and Weixiong Zhang. MicroRNA prediction with a novel ranking algorithm based on random walks. *Bioinformatics*, 24(13), 2008

http://www.springer.com/978-1-4419-7045-9

Software Tools and Algorithms for Biological Systems
Arabnia, H.R.; Tran, Q.-N. (Eds.)
2011, XLIV, 776 p., Hardcover
ISBN: 978-1-4419-7045-9