University “Alexandru Ioan Cuza” of Iaşi
Faculty of Computer Science

TECHNICAL REPORT

Yet Another SVM for MiRNA Recognition: yasMiR

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

TR 10-01, January 2010

ISSN 1224-9327

Universitatea “Alexandru Ioan Cuza” Iaşi
Facultatea de Informatică
Str. Berthelot 16, 700483-Iaşi, Romania
Tel. +40-232-201090, email: bibl@infoiasi.ro

Yet Another SVM for MiRNA Recognition: yasMiR

Daniel Pasailă, Irina Mohorianu, Andrei Sucilă, Ştefan Panţiru, Liviu Ciortuz
Department of Computer Science, “Al. I. Cuza” University of Iaşi
Iaşi, Romania
{daniel.pasaila, irina.mohorianu, andrei.sucila, spantiru, ciortuz}@info.uaic.ro

Abstract

We designed a new SVM for microRNA identification, whose novelty is two-folded: firstly many of its features incorporate the base-pairing probabilities provided by McCaskill’s algorithm, and secondly the classification performance is improved by 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 prove that our SVM obtains truly competitive results.

Topical keywords: bioinformatics, molecular sequence classification, tools and methods for computational biology.

Contact author: Liviu Ciortuz (ciortuz@info.uaic.ro).

Availability: The source code of our system and the datasets we used can be found at the address www.info.uaic.ro/∼ciortuz/yasmir.

Results:

We compared our approach to the Triplet-SVM classifier, after training 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 5-fold 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, 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 5-fold 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 normalised 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.

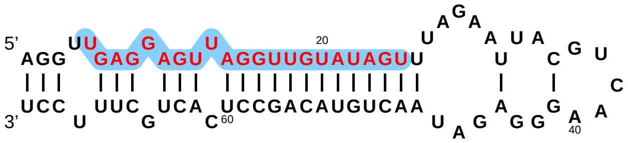
Note: This is a revised and substantially extended version of the paper “Using Base Pairing Probabilities for MiRNA Recognition” by Daniel Pasailă, Irina Mohorianu, and Liviu Ciortuz, that has been published in Proceedings of The International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC) 2008 , IEEE Computer Society, pp. 519-525 [25]. The contributions added here are: the evaluation of the effect of using probabilistic patterns instead of non-probabilistic ones, the analysis of the discriminative power of different categories of features we used, testing our classifier on a more recent version of miRBase (12.0), and the automatic search for good pivot RNA sequences, which are critical for classification in our approach.

1 Introduction

MicroRNAs (miRNAs) are short RNA molecules that play important gene regulatory roles [11]. It is well known that most miRNA precursors (pre-miRNAs) fold as hairpins, however many other RNA sequences in different genomes have a similar structure. Several methods have been proposed for miRNA recognition, among which support vector machines (SVMs) are generally seen as the best ones [9] [10]. Most of these SVMs for miRNA identification rely on the accuracy of the best RNA secondary structure provided by one of the available prediction programs, for instance *RNAfold* from the RNA Vienna package [16]. We will describe another approach, also using the SVM classifier, for which most of the features are computed using the base-pair binding probabilities provided by McCaskill's algorithm [20], based on thermodynamics principles. Such an approach seems promising because it does not rely on a single, predicted secondary structure. We prove this claim through direct comparisons with two other SVMs, namely Triplet-SVM [31] and miPred [24], the last of which has reported best results for pre-miRNA identification up to our knowledge.

The plan of this paper is as follows: Section 2 presents the biological background of the miRNAs and introduces the reader to previous work in the area of identifying new pre-miRNAs using machine learning techniques, especially support vector machines. Section 3 defines the features that we will use for building up a new SVM, while Section 4 will give the main results we obtained on different test datasets, and will compare them to (some of) the best results available in the literature. Section 5 will analyse the contribution of different categories of features that we employ towards discriminating between different classes of RNA sequences. Section 6 reports the results that we obtained when trying to find out whether another classifier, namely Random Forests, is capable of delivering better results than SVM when using the features presented in Section 3. The same section also documents our effort towards automatic identification of the special ("pivot") features that we use for miRNA discrimination. Section 7 draws the conclusions of our work and sketches some improvements that we plan to do in the future.

a.



```
AGGUUGAGGUAGUAGGUUGUAUAGUUUAGAAUUACAUCAAGGGAGAUAACUGUACAGCCUCCUAGCUUUCCU
(((..(((.(((.(((((((((((((.....(..(.....)..)...))))))))))))).))).))).)))
ppp..ppp.ppp.ppppppppppppp.....p..p.....p..p...ppppppppppppp.ppp.ppp.ppp
```

b.

i :	1	2	3	6	7	8	9	10	11	12	14
$PF[i, 0]$:	.54	.98	1	.96	.99	1	.01	1	1	.99	.99

15	16	17	18	19	20	21	22	23	24	25	26
1	1	1	1	1	1	1	1	1	1	.92	.87

27	28	29	30	31	32	33	34	35	36	37	38
.17	.22	.10	.01	.06	.56	.32	.01	.50	.22	.32	.31

i :	33	34	35	37	38	39	40	41	42	43	44	45
$PF[i, 1]$:	.01	.01	.08	.01	.01	.01	.04	.46	.14	.26	.47	.31

46	47	48	49	50	51	52	53	54	55	56	57	58
.33	.51	.94	.99	1	1	1	1	1	1	1	1	1

59	60	62	63	64	65	66	67	68	69	70	71	72
1	.99	.99	1	.99	.01	1	1	.96	.01	.92	1	.60

Figure 1. *a*. The human pre-miRNA *hsa-let-7a-2* and its stem-loop structure. The mature miRNA is shaded. *b*. The two tables give the non-null components of the "profile" arrays $PF[i, 0]$ and respectively $PF[i, 1]$ computed for the *hsa-let-7a-2* pre-miRNA using base-pairing probabilities (see Section 3.2).

2 Background

MicroRNAs (miRNAs) are non-coding RNA molecules that regulate gene expression at post-transcriptional level. First, miRNAs are transcribed from DNA as *primary miRNAs*. Then the Microprocessor complex, containing the nuclease Drosha, when interacting with a primary miRNAs cuts it down to a short hairpin, or stem-loop structure, that is called *precursor miRNA* (pre-miRNA), and has $70-100$ nucleotides. Later, pre-miRNAs are processed to *mature miRNAs* (21-23 nucleotides) in the cytoplasm, by interaction with the Dicer enzyme. Figure 1 (part *a*) illustrates the structure of human precursory miRNA *hsa-let-7a-2*, that has been proved to be a good indicator in cases of adenocarcinoma, a malignancy of the mucous glands in the lungs.

Bioinformatics methods can successfully be used for identifying of new microRNA genes in genomes. The miRNA identification problem is usually defined over pre-miRNAs because their length is larger than that of mature miRNAs, and therefore more information can be extracted from their sequences. Because pre-miRNAs usually have a stem-loop structure, but many other RNA sequences in different genomes have a similar structure, the real challenge is to differentiate real pre-miRNAs from other hairpin-shaped RNA sequences, the latter ones being usually called pseudo pre-miRNAs.

The first bioinformatics attempts to miRNA identification used sequence alignment systems like BLASTN [2]. Because pre-miRNAs often have non-conserved sequences, and instead they tend to conserve their secondary structure, the approach based on sequence comparison is not very promising. Therefore the focus turned on using machine learning (ML) techniques, with a clear preference toward support vector machines, a very good classification tool.

Some of the precursors of ML-based systems for miRNA identification were: miRScan [19] that worked on the *C. elegans* and *H. sapiens* genomes, miRseeker [18] on *D. melanogaster*, and miRfinder [5] on *A. thaliana* and *O. sativa*.

For classification based on ML techniques, a feature vector is extracted from the RNA sequence. The selected features are usually statistical, structural, topological and thermodynamical. Since 2005 an impressive number of SVM-systems have been built, aiming to get better and better results in recognizing miRNAs. The first two of these systems, miR-*abela* [28] and Triplet-SVM [31] proved very inspiring. MiR-*abela*'s authors have shown that their SVM-based predictions were really valuable to biologists: it turned out through laboratory work that about 30% of the proposed candidates were real pre-miRNAs. Triplet-SVM instead was remarkable due to its simplicity: the features employed are patterns over words of 3 consecutive nucleotides in the pre-miRNA sequence. These patterns gather informations from the first and secondary structure levels of the sequence.

Two other systems were basically derived from Triplet-SVM's approach: MiPred [24], and miREncoding [33]. MiPred added a couple of thermodynamical features (minimum free energy MFE, and the so-called P-value [12]), and then succeeded to get better results by replacing SVM with Random Forests, an ensemble learning technique using decision trees. MiREncoding added several new features and tried to improve SVM's classification performances by using DFL, a feature selection algorithm.

Another SVM, RNAmicro [15], tried to explore similarities provided by multiple alignments of related miRNAs. The paper [14] describes an SVM called Microprocessor that identifies the Drosha cutting site in the extended primary miRNA sequence, and then uses informations regarding this site to improve the performance of another SVM in charge with pre-miRNA recognition.

Finally, an SVM called miPred [24] produced what seems to be the best results up to date, by making extensive use of thermodynamical features.[1] The miPred system uses so-called normalised features, which are computed on a large number of shuffled versions of the given pre-miRNAs. However, this approach is not very welcome by biologists due to its lack of biological meaning. At the same time, working with normalised features is computationally very time consuming.[2] One of our aims when we started this work was to produce results comparable to those of miPred, without using normalised features.

All SVMs for miRNA recognition use an RNA secondary structure prediction program, and then compute different features based on the model predicted by this program. As stated in [21], this approach is limited by the secondary structure prediction accuracy. Therefore, a classification system relying on a probabilistic model that takes into account all possible secondary structures of a given RNA is expected to deliver better results compared to those obtained when using features based on a single predicted structure. The present work follows this lead.

Before concluding this section we mention several non-SVM machine learning systems for miRNA identification: BayesMIRfinder [32] was based on the naive Bayes classifier, and proMIR [22] used a Hidden Markov Model. A more recent paper [29] reports on using the k-NN clustering algorithm to learn how to distinguish between different categories of non-coding RNAs, while another one [30] introduces MiRank, a system that uses a ranking algorithm based on random walks, a stochastic process defined on weighted finite state graphs.

3 Our SVM: yasMiR

We propose a novel support vector machine, henceforth called yasMiR, built mainly upon features using the base-pair binding probabilities provided by McCaskill's algorithm [20], supplemented with some other, simple features. The subsection 3.1 will give the formal definition of base-pairing probabilities as introduced in [12], while the subsequent subsections will present our SVM's features. There will be several categories of features as summarized in Table 1: profile similarity scores against "pivot" RNA sequences (subsection 3.2), means of probabilistic triplet patterns (subsection 3.3), and finally other probabilistic and non-probabilistic features (subsections 3.4 and 3.5).

3.1 Base-pairing probabilities

Given an RNA sequence, the probability that the nucleotides on positions i and j form a base-pair is defined as follows:

$$p_{ij} = \sum_{S_\alpha \in \mathcal{S}} P(S_\alpha)\, \delta_{ij}^{\alpha}$$

where $\mathcal{S}$ is the set of all possible secondary structures for the given sequence, and δ_{ij}^{α} is 1 if the nucleotides i and j form a base-pair in the structure S_α and 0 otherwise. The probability of the structure $S_\alpha \in \mathcal{S}$ follows a Boltzmann distribution:

$$P(S_\alpha) = \frac{e^{-MFE_\alpha \,/\, (R \cdot T)}}{Z}$$

where

MFE_α is the folding minimum free energy of S_α,

$Z = \sum_{S_\alpha \in \mathcal{S}} e^{-MFE_\alpha \,/\, (R \cdot T)}$,

R = 8.31451 J mol$^{-1}K^{-1}$ (a molar gas constant), and

[1] The reader should not confuse the two miRNA identification systems that have very similar names: MiPred, cited above, and miPred which has been introduced before.

[2] Supplementary materials published on the web for miPred [24] says that it uses 10,000 shuffled versions for each (real or pseudo) pre-miRNA. It is therefore expected that computing the features for our SVM, when using 100 pivots (see Section 3.2) will be around 100 times faster. Real time comparisons are even more compelling: computing miPred features using only 100 shufflings took about 130 times more time using the miPred's source code (mostly written in Perl), than computing yasMiR features (in C) for the same set of pre-miRNAs.

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

Table 1. Categories of features for yasMiR SVM. The rightmost column gives the number of features in the respective subcategory.

T = 310.15K (37° C).

The probabilities p_{ij} are efficiently computed using McCaskill's algorithm [20].

3.2 A base-pairing profile similarity measure, and related features

We use the idea introduced in [21] for computing a similarity measure for two RNA sequences based on their pattern of base-pairing formation. To compute this similarity score, two steps are needed: first, a base-pair *profile* is calculated for each of the two RNA sequences, and then the *similarity score* for the two resulting profiles is obtained by adapting the global alignment algorithm Needleman-Wunsch [23], using a modified match score and zero gap penalty.

Given a pre-miRNA sequence of length L and the base pairing (McCaskill) probabilities, we compute for every nucleotide i three probabilities: the first is the probability of i forming a base pairing upstream, the second $-$ downstream, and the third $-$ for not forming a base pairing at all. Thus, we obtain a *profile* for the given sequence, under the form of an $L \times 3$ matrix defined as follows:

$$PF[i,0] = \sum_{j>i} p_{ij}$$

$$PF[i,1] = \sum_{j<i} p_{ij}$$

$$PF[i,2] = 1 - PF[i,0] - PF[i,1]$$

As an exemplification, Figure 1b shows the base-pairing profile of *hsa-let-7a-2*. The global alignment of two such profiles is calculated using the Needleman-Wunsch algorithm. We use zero gap penalties, and as match score the inner product of the corresponding two columns in the profiles of the given RNA sequences. Here is the recurrence relation:

$$S[i,j] = \mathit{max} \begin{cases} S[i-1,j] \\ S[i,j-1] \\ S[i-1,j-1] + \sum_{k=0}^{2} PF[i,k] \cdot PF[j,k]. \end{cases}$$

The algorithm computes the best alignment score of the profiles computed for the given pair of RNA sequences.

We will now show how this similarity measure will be used to compute a number of *profile-based features* for our SVM. First, we will construct a set of RNA sequences that we call *pivot sequences*. Then, the alignment scores of a given (training or testing) pre-miRNA with each one of the pivot sequences will be included in the pre-miRNA's feature vector. We conjecture that the way in which the pre-miRNA base-pairing profiles align to the profiles of pivot sequences can be successfully used as a discriminative factor in classifying real vs. pseudo pre-miRNAs. In the developing phase of our system, we used pseudo-miRNAs and pre-miRNAs as pivots, but then we saw that the prediction accuracy didn't significantly change — it even slightly improved — when we used randomly generated sequences. Also, we noticed that about $50-200$ pivot sequences were needed to achieve best performance. The length of the used pivot sequences seemed to affect the classification results. We noticed that in practice sequences of 45-65 nucleotides were most appropriate.

3.3 Local contiguous structure-sequence probabilistic features

The Triplet-SVM [31] classifier used quite successfully a set of 32 local sequence features for pre-miRNA identification. It employed the *RNAfold* function for the secondary structure prediction. Then features were computed by counting certain patterns on triplets of nucleotides in the given pre-miRNA sequence. For yasMiR we also used the patterns proposed there, but instead of only relying on the structure predicted by *RNAfold*, we worked with probabilities provided by McCaskill's algorithm.

In the secondary structure of RNAs, each nucleotide is either paired or unpaired. Let $\mathit{PNP}[i] = \mathit{PF}[i,2]$ store the probability that base on position i is unpaired. For any 3 consecutive nucleotides there are $8 = 2^3$ possible structure patterns: 'ppp', 'pp.', 'p.p', '.pp', 'p..', '.p.', '..p', and '...'. Here, 'p' denotes a paired nucleotide, and '.' an unpaired one. The reader is referred to Figure 1a for the exemplification of such an annotation of RNA secondary structure. Further on, if we consider the middle nucleotide (A, C, G or

U) in a triplet, there will be $32 = 8 \times 4$ possible combinations. Given a pre-miRNA, we will compute the probability of every such combination occurring inside the sequence.

First, we compute a two-dimensional matrix $Pt[2..(L-1), 1..8]$ where $Pt[i, j]$ stores the probability that the triplet centered on the i-th nucleotide has the pattern j. Making an obvious independence assumption, $Pt(i, j)$ can be easily computed by multiplying the probabilities that correspond to the three positions inside that pattern. For example, the probability computed for the pattern 'p.p' for some i is $(1-PNP[i-1]) \cdot PNP[i] \cdot (1-PNP[i+1])$.

After having computed the matrix *Pt*, it is easy to calculate the two-dimensional matrix $Pn[1..4, 1..8]$ where $Pn[a, j]$ denotes the probability that nucleotide a appears in the middle position of occurrences of pattern j, inside the given sequence $S[1..L]$:

$$Pn[a, j] = \left(\sum_{S[i]=a} Pt[i, j] \right) / L.$$

The $Pn[a, j]$ values are included in the feature vector we associate to a given pre-miRNA sequence. These 32 features are a natural generalisation to the structure-sequence features defined for Triplet-SVM, now using base-pair binding (McCaskill) probabilities.

3.4 Other features using base-pairing probabilities

The *overall non base-pairing probability* was included in the yasMiR's feature vector. This value is given by:

$$\sum_{i=1}^{L} PNP[i]/L.$$

The output of the *mean_bp_dist* function in the Vienna RNA package was also used as a feature. This value represents the mean base pair distance in the equilibrium state of a given RNA, which constitutes a measure of its structural diversity. It is also computed using the probabilities obtained by McCaskill's algorithm.

We also computed the non base-pairing probability for every nucleotide $a \in \{A, C, G, U\}$ in the following way:

$$\sum_{S[i]=a} PNP[i]/cnt(a).$$

where $cnt(a)$ denotes the number of nucleotides of type a in the sequence S.

For every pair of nucleotides a and b we computed the sum of the base-pair probabilities for all the corresponding positions in the sequence. There are in total 10 such combinations, since (a, b) and (a, b) count as only one pair and the case $a = b$ is allowed. We used the following formula:

$$\sum_{S[i]=a, S[j]=b} p_{ij}.$$

3.5 Other features

As features not based on McCaskill's probabilities we first added the folding *minimum free energy*. This was obtained using the *fold* function in the Vienna RNA package, which is based on Zuker's algorithm [34]. Then, the *average dinucleotide frequencies* (16 combinations) were also included in the feature vector. Finally, we added the *average frequencies* of A, C, G and U in the current sequence, calculated as $cnt(a)/L$, for each nucleotide a.

4 Datasets and Main Results

The first and main objective of this section is to evaluate the set of features presented in the previous section, by comparing the results it provides when using the SVM classifier with the results reported in the literature for the miRNA identification systems Triplet-SVM and miPred. The second goal of this section is to evaluate our features by performing different analyses on them.

As SVM implementation, we used the libSVM package [8] version 2.84. The values of the penalty parameter C and the RBF kernel parameter γ were selected using the grid search implemented by a Python script provided with libSVM. The scaling of feature values was performed using the default parameters (-1, 1).

4.1 Comparison with Triplet-SVM

To train the Triplet-SVM classifier [31], its authors have built a dataset called TR-C. As positive examples, 163 pre-miRNAs have been randomly selected from the 193 human pre-miRNAs in miRBase version 5.0. As negative examples, 168 pre-miRNA-like hairpins with a similar stem-loop structure to real pre-miRNAs have been randomly selected from CODING, a set of 8494 sequences chosen by Triplet-SVM's authors from the NCBI RefSeq database [26]. There are no multiple loops in these sequences.

For the test phase, the authors of Triplet-SVM built four datasets:

– The TE-C dataset included the 30 remaining human pre-miRNAs from miRBase version 5.0, and 1000 pseudo pre-miRNAs randomly selected from the CODING set, excluding those already allocated to the TR-C training set.

– The UPDATED dataset was made of 39 human pre-miRNAs, reported after the release of miRBase 5.0 and up to the time when Triplet-SVM was completed.

– The CROSS-SPECIES dataset consists of 581 pre-miRNAs from 11 species in miRBase 5.0, different from human.

Test	yasMiR accuracy(%)	Triplet-SVM acc.(%)
TE-C: Human pre-miRNAs	**100.0** (30/30)	93.3
TE-C: Pseudo pre-miRNAs	**96.2** (962/1000)	88.1
UPDATED	**94.9** (37/39)	92.3
CROSS-SPECIES	**95.2** (553/581)	90.9
CONSERVED-HAIRPIN	**94.23** (2303/2444)	89.0

Table 2. Comparison of yasMiR with Triplet-SVM. The results for Triplet-SVM are taken from [31]. In parenthesis: the ratio of correctly classified instances. For positive datasets (TE-C human, UPDATED and CROSS-SPECIES), accuracy coincides with sensitivity. For negative datasets (TE-C pseudo and CONSERVED-HAIRPIN), accuracy coincides with specificity.

− The CONSERVED-HAIRPIN dataset was built by extracting 2444 hairpins from the human chromosome 19, between positions 56000001 and 57000000, obtained from the UCSC database (hg17, May 2004) [17]. Of all these hairpins, 3 are real pre-miRNAs, while the others are pseudo pre-miRNAs.

On the TR-C training dataset, when doing 5-fold cross-validation yasMiR obtained a prediction accuracy of 96.07% following the grid parameter search, compared to 93.50% reported for Triplet-SVM. For the profile, we included 100 pivots, which are randomly generated RNA sequences of 45-65 nucleotides. Table 2 shows the results we obtained on the above four test datasets, compared to Triplet-SVM, after both SVMs were trained on the same dataset, TR-C. One can see that yasMiR has a better accuracy/specificity/sensitivity, namely 2.6%−8.1% higher than Triplet-SVM on all four test datasets. Detailed comparisons on the different species in the CROSS-SPECIES dataset are shown in Table 3. These good results encouraged us to do further comparisons, this time with miPred SVM.

4.2 Comparison with miPred

For miPred [24], the training set (called TR-H) included 200 human pre-miRNAs randomly selected from miRBase 8.2, and 400 pseudo-miRNAs from the CODING set, built by Triplet-SVM's authors.

In order to test their classifier, the authors of miPred built four datasets: TE-H, IE-NH, IE-NC and IE-M:

− TE-H and IE-NH were designed similarly to the datasets TE-C and respectively CROSS-SPECIES used for testing Triplet-SVM: TE-H included the 123 human pre-miRNAs remaining from miRBase 8.2 after 200 such pre-miRNAs have been allocated for training (TR-H), while IE-NH contains 1918 pre-miRNAs from 40 non-human species from miRBAse 8.2. Both datasets included twice more negative examples than positives, randomly selected from the CODING set.

Test	yasMiR accuracy(%)	Triplet-SVM accuracy(%)
Mus musculusi	**97.2** (35/36)	94.4
Rattus norvegicus	**80.0** (20/25)	80.0
Callus Gallus	**100.0** (13/13)	84.6
Dnio Rerio	**83.3** (5/6)	66.7
Caenorhabditis briggsae	**100.0** (73/73)	95.9
Caenorhabditis elegans	**93.6** (103/110)	86.4
Drosophila pseudoobscura	**93.0** (66/71)	90.1
Drosophila melanogaster	**97.2** (69/71)	91.5
Oryza sativa	**95.8** (92/96)	94.8
Arabidopsis thaliana	**96.0** (72/75)	92.0
Epstein Barr Virus	**100.0** (5/5)	**100.0**
Total	**95.2** (553/581)	90.9

Table 3. Detailed comparison of yasMiR with Triplet-SVM: accuracy on the CROSS-SPECIES dataset. The results for Triplet-SVM are taken from [31]. In parenthesis: the ratio of correctly classified instances. Here accuracy coincides with specificity.

− IE-NC consists of 12387 non-coding RNAs (other than miRNAs) from the Rfam 7.0 database [13], and IE-M is made of 31 messenger RNAs selected from GenBank [3].

We recreated these five datasets according to the above specifications made by the authors of miPred, since they did not provide the datasets themselves.[3]

In order to ensure fair comparisons, we re-trained yasMiR on the TR-H dataset, similarly to miPred, and then we ran it on the above four test datasets. Table 4 shows comparative results with miPred and Triplet-SVM. We used the same set of 100 randomly generated pivots as we have previously done for the comparison with Triplet-SVM. Our SVM not only outperformed again Triplet-SVM on all above mentioned test datasets, but it also definitely outperformed miPred on the IE-NC and IE-M datasets (82.95% vs. 68.68%, and respectively 100% vs. 87.09% accuracy/specificity). On IE-NH yasMiR loses respectively 1.53%/1.73%/1.43% in accuracy/sensitivity/specificity compared to miPred. On the IE-NH dataset, the accuracy of yasMiR is slightly better than that of miPred (93.77% vs. 93.50%) while its specificity is 1.23% lower (down to 96.74% from 97.97%). Note that Triplet-SVM misclassified all 31 instances in the IE-M set,

[3] Because we re-created the miPred's train and test datasets, it is possible that there will be slight differences between the results published in [24] and those that would be obtained by running miPred and Triplet-SVM on the re-created datasets.

Test	yasMiR			miPred			Triplet-SVM		
	accuracy(%)	se.(%)	sp.(%)	accuracy(%)	se.(%)	sp.(%)	accuracy(%)	se.(%)	sp.(%)
TE-H	**93.77**	**87.80**	96.74	93.50	84.55	**97.97**	87.96	73.15	93.57
IE-NH	94.11	90.35	95.99	**95.64**	**92.08**	**97.42**	86.15	86.15	96.27
IE-NC	**82.95**			68.68			78.37		
IE-M	**100**			87.09			0		

Table 4. Comparison of yasMiR with miPred and Triplet-SVM. The results for miPred and Triplet-SVM are taken from [24]. Only accuracy is given for IE-NC and IE-M since these datasets are made only of pseudo miRNAs; in such a case, specificity is equal to accuracy, and sensitivity is undefined.

while yasMiR correctly classifies them all.

Our conclusion so far is that yasMiR is a serious contender not only for Triplet-SVM, but also for miPred.

4.3 yasMiR 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 1256 sequences from the CODING dataset as negative examples. The testing set includes 3651 positive examples from miRBase 12.0 and 7198 negative examples from the CODING dataset. The set of 3651 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 [24] 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 3651 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%.

Test	**using non-probabilistic triplet patterns**	**yasMiR**
TE-C: Human pre-miRNAs	96.67 (29/30)	100
TE-C: Pseudo pre-miRNAs	95.9 (959/1000)	96.2
UPDATED	94.9 (37/39)	94.9
CROSS-SPECIES	95.87 (557/581)	95.2
CONSERVED-HAIRPIN	93.09 (2275/2444)	94.23

Table 5. 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. The right column results are from Table 2.

Test	$B \cup C$			$A \cup C$			$A \cup B$		
	accuracy(%)	se.(%)	sp.(%)	accuracy(%)	se.(%)	sp.(%)	accuracy(%)	se.(%)	sp.(%)
TE-H	93.22	83.73	**97.96**	**94.30**	89.43	**96.74**	91.32	81.30	96.34
IE-NH	92.64	88.58	94.68	**94.26**	**93.32**	94.73	92.26	84.04	**96.37**
IE-NC	78.94			59.84			**91.77**		
IE-M	100			6.45			100		

Table 6. Prediction results for yasMiR on miPred datasets when removing one category (A, B or C) of its features. Bold faces were used to designate better values than those in Table 4, the leftmost numerical column.

5 Feature analysis

Since yasMiR uses features which are a probabilistic (McCaskill) version of the features employed 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) are usually slightly (and even significantly) better than the ones we obtained with non-probabilistic features computed for triplet patterns (Table 5). 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):

– Category A: profile alignment scores with the randomly chosen pivot sequences.

– Category B : probabilistic means of the number of occurrences of triplet patterns.

– Category C: other probabilistic and non-probabilistic features.

Using the same datasets as miPred [24], we investigated the effect on accuracy, sensitivity and specificity when removing one of the three categories A, B, or C. The reader should compare the first numerical column in Table 4 with Table 6. It can be easily seen that the prediction results with the complete feature set 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 1 in Table 6) 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 2 in Table 6) 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 3 in Table 6) 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 imply 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 specialised contributions of these categories of features towards discriminating among different categories of RNA sequences.

For expressing the quality of the i-th feature we used the $F1$ and $F2$ scores, defined by

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

where μ_i^+/μ_i^-, σ_i^+/σ_i^- denote the means and standard deviations of the positive and negative training datasets for the i-th 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 $F2 = 1.64$),

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

Table 7. Prediction results for yasMiR on miPred datasets when removing one of the features D, E or F. Bold faces were used to designate better values than those in Table 4, the leftmost numerical column.

	0.95 confidence			**0.90 confidence**		
Test	accuracy(%)			accuracy(%)		
		se.(%)	sp.(%)		se.(%)	sp.(%)
TE-H	**94.30**	**87.80**	**97.50**	**93.76**	**87.80**	**96.74**
IE-NH	94.07	90.14	**96.03**	93.39	**91.08**	94.55
IE-NC	**83.28**			77.20		
IE-M	**100**			**100**		

Table 8. 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 were used to designate better values than those in the leftmost numerical column of Table 4.

– 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 7. 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 for redundancy elimination [4] 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 normalised 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. (For more details see Section 6.2.2.)

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 the Table 8. At 0.90 confidence, things don't 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 6 non-pivot features.

6 Searching for Further Improvements

In this section we will firstly present the results we got when replacing the SVM classifier in our system with Random Forests, another classifier which has was reported to give better results than SVM on certain tasks. Secondly, here we will report on our efforts to automatically select a set of good pivot RNA sequences.

6.1 Random Forests vs. SVM

The MiPred system [24] got better results for miRNA identification when using the Random Forests (RF) classifier [7] instead of SVM, with the same set of features, namely the Triplet-SVM features plus the folding minimum free energy and the P-value [12]. We wanted to see whether the same is true for yasMiR's set of features. This subsection briefly presents Random Forests (RF), and then it reports on the tests we did using RF as classifier for our miRNA identification problem.

Random Forests is an *ensemble learning* algorithm that was derived from *bagging*, also devised by Leo Breiman [6]. Like *boosting* [27] too, these two techniques use certain strategies for aggregating some simpler classification algorithms. In the sequel we will consider that the aggregated classifiers are *decision trees*.

In the *bagging* approach, whose name comes from *b*ootstrap *agg*regat*ing*, each tree is independently constructed using a bootstrap sample (i.e. sampling with replacing) from the training dataset. Classification of a test instance is done by taking a simple majority vote among the decision trees.

The Random Forests algorithm extends bagging with and additional layer of randomness, namely the random feature selection: while in standard decision trees each node is split using the best split among all variables, in RF each node is split using the best among a subset of features randomly chosen at that node. Thus, RF uses (only) two parameters: the number of variables in the random subset at each node, and the number of trees in the forest.

Test	**RF**		**SVM**
	without feature sel.	with feature sel.	with feature sel.
TE-C	96.45	96.41	95.34
UPDATED	94.87	94.87	94.87
CROSS-SPECIES	93.70	93.21	**95.53**
CONSERVED-HAIRPIN	93.30	93.65	91.90

Table 9. Comparing the predictive accuracy(%) of RF and SVM on test datasets from Triplet-SVM, using yasMiR features. Feature selection was based on the *importance* function found in the package R. 99 features have been selected. Bold faces were used to designate better values than those given in the leftmost numerical column of Table 4.

Although RF is a somehow counter-intuitive strategy, it proved to be robust against overfitting, and it produced some good results when compared to other machine learning techniques including SVMs, neural networks, discriminate analysis, etc. As implementation for RF, we used the *randomForest* (version 4.5-25) package for the R language [1].

Table 9 shows the accuracy results that we obtained when we ran the RF and classifier on TR-C, which was the Triplet-SVM's training set, and we did comparisons on its test sets: TE-C, UPDATED, CROSS-SPECIES, and CONSERVED-HAIRPIN. We used the features described in Section 3. Profile similarities were computed on the same set of 100 pivots as before. RF produced results which are slightly below those obtained by yasMiR SVM (see Table 2).

We then used the *importance* function from the R package to select the best features following the analysis of the decision trees produced by RF on the full set of features. By using the 99 best features RF only got a slight improvement on the CONSERVED-HAIRPIN dataset.

Running SVM using the same set of 99 selected features provided a better result only on the CROSS-SPECIES dataset: 95.53% vs. the 95.2% accuracy/sensitivity obtained by yasMiR when using the full set of features.

We also performed a similar comparison between RF and SVM on the test datasets designed by miPred's authors, after having had both classifiers trained on miPred's training dataset, TR-H. Profile similarities were computed on the same 100 pivots as before. Table 10 shows that unfortunately RF did not produce better results than the yasMiR SVM described in Section 3 on any of these test datasets (see yasMiR's results in Table 4). Instead, on the IE-NC and IE-M datasets, RF registered heavy losses of accuracy/specificity: 64.79% vs. 82.95%, and respectively

Test	**RF**		**SVM**
	without feature sel.	with feature sel.	with feature sel.
TE-H	92.41	91.84	91.06
IE-NH	94.03	93.46	92.2
IE-NC	64.79	63.87	78.86
IE-M	22.00	23.87	90.32

Table 10. Comparing the predictive accuracy(%) of RF and SVM on test datasets from miPred, using yasMiR features. Feature selection was again based on the *importance* function. 19 features have been selected.

Test	**SVM**		**RF**	
	accuracy(%)		accuracy(%)	
	se.(%)	sp.(%)	se.(%)	sp.(%)
TE-H	92.14		91.59	
	85.37	95.53	81.82	96.47
IE-NH	91.17		93.98	
	83.58	94.97	89.94	96.00
IE-NC	**93.61**		64.20	
IE-M	**100**		18.77	

Table 11. Prediction results of yasMiR (both SVM and RF variants) on miPred's test datasets using 200 pivots selected via clustering from a pool of 2000 randomly generated pivots. Bold faces were used to designate better values than those given in the leftmost numerical column of Table 4.

22.45% vs. 100%. The *importance* RF-supported feature selection function did not help RF to get any significant improvement. However, one can see that the SVM classifier, when using the same set of 19 best features selected via the *importance* function, produces similar results to RF on TE-H and IN-NH, but much better results on IE-NC and IE-M. That means that SVM's generalisation power on the negative instances that have been used by miPred for training is much better than that of RF.

The conclusion of this subsection is that RF seems to be a not good enough candidate to replace SVM for pre-miRNA identification using the set of features presented in Section 3.

6.2 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.

6.2.1 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 to 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).

Table 11 shows the results we obtained for 200 pivots automatically selected from a pool of 2000 randomly generated sequences. The k-means clustering algorithm was used to get those 2000 sequences grouped into 50 clusters, and then we randomly selected 4 pivots from each cluster.[4] 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%).[5] 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%. The same table shows that using these 200 pivots, RF provided only slightly different accuracies on TE-H and IE-NH, but performed very badly on IE-NC (from 93.61% down to 64.20% specificity/accuracy) and IE-M (from 100% down to 18.77% specificity/accuracy).

These results make us conclude that automatically searching for better pivots is worth further working, and again that SVM is a machine learning technique undoubtedly better suited than RF for our task and data.

6.2.2 Using the Kolmogorov-Smirnov filter

In this subsection we will use the Kolmogorov-Smirnov filter for searching among a large pool of randomly generated pivots.

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. We decided to try a non-linear feature selection algorithm in order to search for 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

[4] The libSVM's parameters used here were $C = 4$ and $\gamma = 0.5$.

[5] See Table 4 for correlation.

from [4]. Here we will briefly explain how this filter works, and then we will discuss the results we got when using this filter for automatically selecting better pivots.

The Kolmogorov-Smirnov filter [4] 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. For the first step, recall that the Shannon entropy for a random variable, X, is

$$H(X) = -\sum_{i=1} P(x_i) \log P(x_i)$$

while the joint Shannon entropy of two variables X and C is given by the formula

$$H(X,C) = -\sum_{i=1} P(xi,cj) \log P(xi,cj).$$

The mutual information of X and C is then defined as:

$$MI(X,C) = H(X) + H(C) - H(X,C)$$

The ranking criterion used by the Kolmogorov-Smirnov filter is based on MI as it is just a normalised version of it. Hence we define the Symmetrical Uncertainty coefficient SU as:

$$SU(X,C) = 2 * \frac{MI(X,C)}{H(X) + H(C)}$$

For our particular problem, the observed values of X are chosen as the scores obtained by aligning every sequence to a fixed pivot, discretised into a number of bins (we chose 100 bins), while the observed values of C are the class labels.

The second part of the filter is *redundancy elimination*, which is based on a theorem by Kolmogorov and Smirnov that shows when two underlying one-dimensional probability distributions differ from one another. For a random variable, X, observed through n samples, the empirical distribution function, F_n is :

$$F_n(x) = \frac{1}{n} \sum_{i=1}^{n} I_{X_i \leq x}$$

where $I_{X_i \leq x}$ is the indicator function.

The Kolmogorov-Smirnov statistic for two variables is

$$D_{n,n'} = \sup_x |F_n(x) - F'_{n'}(x)|$$

We will say that the values observed have not been generated by the same distribution, with a confidence level of α if

$$\sqrt{\frac{nn'}{n+n'}} D_{n,n'} > K_\alpha,$$

where the K_α constant is obtained from the Kolmogorov-Smirnov distribution. Recall that the Kolmogorov-Smirnov distribution is the distribution of the random variable $K = \sup_{t \in [0,1]} |B(t)|$, where $B(t)$ is the Brownian bridge. The cumulative distribution of K is

$$Pr(K \leq x) = \frac{\sqrt{2\pi}}{x} \sum_{i=1}^{\infty} e^{\frac{-(2i-1)^2 \pi^2}{8x^2}}$$

and K_α is found from the equation $Pr(K \leq K_\alpha) = 1 - \alpha$.

Test	**SVM**		
	acc.(%)	se.(%)	sp.(%)
TE-H	92.53	85.37	**96.74**
IE-NH	91.35	86.24	93.90
IE-NC	**87.44**		
IE-M	**100**		

Table 12. Prediction results of yasMiR on miPred's test datasets using the best 13 pivots selected from the 10000 randomly generated pivots. Bold faces were used to designate better values than those in Table 4, the leftmost numerical column.

We used a confidence level of 95% for determining whether two features were redundant.

When we applied the Kolmogorov-Smirnov filter to our problem — selecting good representatives among a large pool of randomly generated pivots —, we tried larger and larger sets of pivots. We began with an initial set of 5000 pivots, and end up with a set of 10000. As stopping criteria for the filter, we used a joint condition: halt when either 100 pivots have been selected (and the remaining ones will be subsequently discarded), or no more redundancies could be found. When selecting from the first set of 5000 pivots, only 72 non redundant sequences were found, so we chose the last remaining 28 from possibly redundant ones. When selecting from the set of 10000, we found a number of 134 non redundant pivots.

The Table 12 shows that when using the best 13 features selected by the Kolmogorov-Smirnov filter, the results obtained by yasMiR on the miPred datasets are comparable with those reported in the previous subsection. 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 remained at 100%.

We suggest that this method would be best used in conjunction with another feature selection method, were 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.

7 Conclusions and Further Work

We proved that the base pairing probabilities provided by McCaskill's algorithm combined with some other, simple statistical measures make a 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 [29]. Its results seem to be very competitive, due to the use of certain topological features. We will see whether those features could be generalized by 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 an even more recent work for identifying miRNAs [30], 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 obtained by using McCaskill's probabilities.

Authors' contributions: Liviu Ciortuz explored the existing scientific documentation for miRNA recognition (in particular, SVMs designed for this task), identified possible ways in which we could contribute to this research area, designed the overall "philosophy" of the experiments, coordinated the whole project/work and shaped the group's publications. Daniel Pasailă identified the use of McCaskill probabilities as a valuable way to design a new SVM for miRNA recognition, came up with the idea of using "pivot" sequences to improve the classification, designed the whole set of features for yasMiR SVM and implemented their computation, programmed many of the experiments reported for yasMiR (namely the comparisons with Triplet-SVM and miPred, the test on miRBase 12.0, the feature analysis — except for redundancy tests —, the search for better pivots using clustering), and documented the experiments, for both publication and web versions. Irina Mohorianu run the RF experiments and also the PCA experiments, documented them, and helped with running additional yasMiR experiments. Andrei Sucilă implemented the Kolmogorov-Smirnov filter, documented it, and applied it to the data for both identifying redundant features and finding better pivots. Ştefan Panţiru fine-tuned some of the tests, wrote scripts and prepared the web page for yasMiR data and the implemented code.

Acknowledgments: LC thanks Dr. Mihaela Zavolan from Biozentrum, Basel, for introducing him to the problem of miRNA identification. He is also indebted to Dr. Hélène Touzet form University of Lille and Dr. Marti Tammi from the National University of Singapore for useful discussions on this subject. During the first stages of this work, LC was partially supported by the CEEX grant "ForMol" from the Romanian Ministry of Education and Research.

References

[1] http://www.r-project.org/.

[2] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

[3] D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and D. L. Wheeler. GenBank. *Nucleic Acids Research*, 33(Database-Issue):34–38, 2005.

[4] J. Biesiada and W. Duch. Feature selection for high-dimensional data: A Kolmogorov-Smirnov correlation-based filter. *Computer Recognition Systems*, 30:95–103, 2005.

[5] E. Bonnet, J. Wuyts, P. Rouzé, and Y. Van de Peer. Detection of 91 potential conserved plant microRNAs in Arabidopsis thaliana and Oryza sativa identifies important target genes. *Proc Natl Acad Sci U S A*, 101(31):11511–11516, 2004.

[6] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[7] L. Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

[8] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[9] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[10] N. Cristianini and J. Shawe-Taylor. *An introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, New York, NY, USA, 2000.

[11] A. Fire, S. Xu, M. Montgomery, S. Kostas, S. Driver, and C. Mello. Potent and specific genetic interference by double-stranded RNA in Caenorhabditis elegans. *Nature*, 391(6669):806–811, 1998.

[12] E. Freyhult, P. P. Gardner, and V. Moulton. A comparison of RNA folding measures. *BMC Bioinformatics*, 6:241, 2005.

[13] S. Griffiths-Jones, S. Moxon, M. Marshall, A. Khanna, S. R. Eddy, and A. Bateman. Rfam: annotating non-coding RNAs in complete genomes. *Nucleic Acids Research*, 33:D121, 2005.

[14] S. Helvik, O. J. Snøve, and P. Sætrom. Reliable prediction of Drosha processing sites improves microRNA gene prediction. *Bioinformatics*, 23(2):142–149, 2007.

[15] J. Hertel and P. F. Stadler. Hairpins in a haystack: recognizing microRNA precursors in comparative genomics data. *Bioinformatics*, 22(14):e197–e202, July 2006.

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

[17] D. Karolchik, R. Baertsch, M. Diekhans, T. Furey, A. Hinrichs, Y. Lu, K. Roskin, M. Schwartz, C. Sugnet, D. Thomas, R. Weber, D. Haussler, and W. Kent. The UCSC Genome Browser Database. *Nucleic Acids Res*, 31(1):51–54, 2003.

[18] E. C. Lai, P. Tomancak, R. W. Williams, and G. M. Rubin. Computational identification of Drosophila microRNA genes. *Genome Biology*, 4(7), 2003.

[19] L. P. Lim, N. C. Lau, E. G. Weinstein, A. Abdelhakim, S. Yekta, M. W. Rhoades, C. B. Burge, and D. P. Bartel. The microRNAs of Caenorhabditis elegans. *Genes Dev*, 17(8):991–1008, 2003.

[20] J. S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structures. *Biopolymers*, 29:1105–1119, 1990.

[21] L. M. C. Meireles. Evaluation of a kernel function for recognizing microRNAs, 2006. Project report, School of Computer Science, CMU.

[22] J.-W. Nam, K.-R. Shin, J. Han, Y. Lee, V. N. Kim, and B.-T. Zhang. Human microRNA prediction through a probabilistic co-learning model of sequence and structure. *Nucleic Acids Research*, 33(11):3570–3581, 2005.

[23] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970.

[24] K. L. S. Ng and S. Mishra. De novo SVM classification of precursor microRNAs from genomic pseudo hairpins using global and intrinsic folding measures. *Bioinformatics*, 23(11):1321–1330, 2007.

[25] D. Pasailă, I. Mohorianu, and L. 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.

[26] K. D. Pruitt and D. R. Maglott. RefSeq and LocusLink: NCBI gene-centered resources. *Nucleic Acids Research*, 29(1):137–140, 2001.

[27] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, 1998.

[28] 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*, 6:267, 2005.

[29] W. Shu, X. Bo, Z. Zheng, and S. Wang. A novel representation of RNA secondary structure based on element-contact graphs. *BMC Bioinformatics*, 9(1):188, 2008.

[30] Y. Xu, X. Zhou, and W. Zhang. MicroRNA prediction with a novel ranking algorithm based on random walks. *Bioinformatics*, 24(13), 2008.

[31] C. Xue, F. Li, T. He, G. 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(310), 2005.

[32] M. Yousef, M. Nebozhyn, H. Shatkay, S. Kanterakis, L. Showe, and M. Showe. Combining multi-species genomic data for microRNA identification using a Naive Bayes classifier machine learning for identification of microRNA genes. *Bioinformatics*, 22(11):1325–1334, 2006.

[33] Y. Zheng, W. Hsu, M.-L. Lee, and L. Wong. Exploring essential attributes for detecting microRNA precursors from background sequences. In M. M. Dalkilic, S. Kim, and J. Yang, editors, *2006 VDMB Workshop on Data Mining in Bioinformatics*, volume 4316 of *Lecture Notes in Computer Science*, pages 131–145. Springer, 2006.

[34] M. Zuker and P. Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, 9(1):133–148, 1981.