Methods 69 (2014) 220–229

Contents lists available at ScienceDirect

Methods

journal homepage: www.elsevier.com/locate/ymeth

Ensemble learning can significantly improve human microRNA target prediction

Seunghak Yu [a,b], Juho Kim [a], Hyeyoung Min [c], Sungroh Yoon [a,d,*]

[a] Department of Electrical and Computer Engineering, Seoul National University, Seoul 151-744, Republic of Korea
[b] Department of IT Convergence, Korea University, Seoul 156-713, Republic of Korea
[c] RNA Biopharmacy Laboratory, College of Pharmacy, Chung-Ang University, Seoul 156-756, Republic of Korea
[d] Bioinformatics Institute, Interdisciplinary Program in Bioinformatics, Seoul National University, Seoul 151-747, Republic of Korea

ARTICLE INFO

Article history:
Received 19 March 2014
Revised 16 July 2014
Accepted 18 July 2014
Available online 1 August 2014

Keywords:
MicroRNA
Target prediction
Sequence analysis
Algorithm
Machine learning

ABSTRACT

MicroRNAs (miRNAs) regulate the function of their target genes by down-regulating gene expression, participating in various biological processes. Since the discovery of the first miRNA, computational tools have been essential to predict targets of given miRNAs that can be biologically verified. The precise mechanism underlying miRNA–mRNA interaction has not yet been elucidated completely, and it is still difficult to predict miRNA targets computationally in a robust fashion, despite the large number of *in silico* prediction methodologies in existence. Because of this limitation, different target prediction tools often report different and (occasionally conflicting) sets of targets. Therefore, we propose a novel target prediction methodology called *stacking-based miRNA interaction learner ensemble* (SMILE) that employs the concept of stacked generalization (stacking), which is a type of ensemble learning that integrates the outcomes of individual prediction tools with the aim of surpassing the performance of the individual tools. We tested the proposed SMILE method on human miRNA–mRNA interaction data derived from public databases. In our experiments, SMILE improved the accuracy of the target prediction significantly in terms of the area under the receiver operating characteristic curve. Any new target prediction tool can easily be incorporated into the proposed methodology as a component learner, and we anticipate that SMILE will provide a flexible and effective framework for elucidating *in vivo* miRNA–mRNA interaction.
© 2014 Elsevier Inc. All rights reserved.

1. Introduction

MicroRNAs (miRNAs) are small non-protein coding RNAs that regulate gene function by down-regulating the expression of their target gene(s) [1,2]. As central players in post-transcriptional gene regulation, miRNAs are known to be involved in many biological processes and diseases [3,4], and miRNA research continues to rapidly increase. MicroRNAs exert their function through binding to target sites present in the 3′ untranslated region (UTR) of their cognate mRNAs. For a complete understanding of miRNA function, it is thus critical to identify the target mRNAs which a miRNA binds to and functions through.

However, the targets of most miRNAs remain unknown because of the lack of robust bioinformatics methods that predict potential mRNA targets precisely and the lack of non-laborious experimental verification systems. Since the base-pairings between miRNAs and their target mRNAs include bulges and non-canonical base pairs and are thus not perfect, it is difficult to predict miRNA targets with high accuracy. Efforts have been made to identify the molecular mechanism underlying miRNA–mRNA interaction, but the exact mechanism by which miRNAs select their targets and mediate translational repression has not yet been completely elucidated [5].

Nevertheless, studies have suggested that sequence complementarity, target site accessibility, and evolutionary conservation are important factors for target recognition [6]. The first-generation of miRNA target prediction tools focused on utilizing these factors to predict novel miRNA–target pairs. Examples include TargetScan [1], miRanda [7], RNAhybrid [8], DIANA-microT [9], PicTar [10], and PITA [11]. These first-generation tools significantly contributed to the field by reporting a large number of putative targets, some of which were later confirmed by experimental studies, thus expanding the library of known the miRNA–target interactions. However, these tools mostly consider only subsets of known factors that affect the binding of miRNAs to their targets, and they

* Corresponding author at: Department of Electrical and Computer Engineering, Seoul National University, Seoul 151-744, Republic of Korea.
E-mail address: sryoon@snu.ac.kr (S. Yoon).

http://dx.doi.org/10.1016/j.ymeth.2014.07.008
1046-2023/© 2014 Elsevier Inc. All rights reserved.

often show unsatisfactory performance in terms of the rates of false positives and false negatives.

As the number of experimentally verified miRNA targets increased, second-generation tools emerged that commonly assumed the existence of 'training' data. These tools are based on machine-learning approaches that employ classifiers computationally trained using experimentally verified miRNA–target pairs. Examples include miRTarget2 [12], TargetBoost [13], TargetSpy [14], and TargetMiner [15]. These methods learn (occasionally subtle but critical) features appearing in real miRNA–target pairs and non-pairs and apply them to the prediction of unknown pairs, thus boosting the accuracy of prediction compared to the first-generation techniques. Recently, hybrid approaches that consider structural features and machine-learning features together have been proposed, *e.g.*, miREE [16], NBmiRTar [17], and DIANA-microT-ANN [18]. The performance of the second-generation methods relies on the quantity and quality of training data and the learning algorithm used, and these methods often exhibit limited abilities to reduce generalization error in the prediction of unknown interactions.

Various prediction tools have been proposed in the first and second generation, and the third-generation prediction methods collect the outcomes from different tools and combine them to obtain better results than individual tools can deliver. Some tools also integrate miRNA and mRNA expression data for target prediction, given miRNA expression and mRNA expression are inversely correlated. Examples of the third-generation methods include MiRror [19], MMIA [20], MAGIA2 [21], imiRTP [22], miRecords [23], miRGator ver2.0 [24], miRNAMap 2.0 [25], and miRGen [26]. Most of these tools do not account for the strengths and weaknesses of the unique features of each algorithm; rather, they merely incorporate the presence or absence of detection and target scores independently. Since each prediction method intrinsically possesses numerous false positives, the simple integration of multiple methods may result in increased false positives, making accurate target prediction more difficult.

To address the limitations of the existing approaches, here we propose a novel miRNA target prediction algorithm called *stacking-based microRNA interaction learner ensemble* (SMILE). We introduce this approach because existing prediction tools often produce inconsistent results even when given identical inputs [27–29] and the resulting diverse outputs from multiple prediction models can reduce the prediction error (see Eq. (1) in Section 2). The proposed SMILE method can be classified as a third-generation approach and works in two stages: each prediction tool is considered as an individual prediction model (1st stage), and the multiple models are then combined through ensemble learning (2nd stage). There exist multiple approaches for ensemble learning as described in Section 2, and our method is based on the idea of stacked generalization (stacking) [30,31], which summarizes the outcomes of individual learners for each object as a feature vector and classifies the feature vectors using a second-level learner. Using this idea, we can implicitly consider the features related to miRNA–mRNA interactions that individual tools miss, thus achieving improved prediction performance.

2. Background

2.1. Key idea behind ensemble learning

Ensemble learning consists of a set of prediction models and a method to combine these models. In ensemble learning, we train multiple models and combine the outputs from individual models to reduce generalization error. Depending on how the diversity of individual models is addressed, ensemble learning can be categorized into implicit and explicit methods [32]. An implicit method utilizes random subsets of training data when creating each individual model. Examples include *bagging*, which selects samples randomly and *random forests*, which selects samples and features randomly [33]. An explicit method obtains different individual models using a measurement. An example is *boosting*, which utilizes the error from one stage to the next, thus gradually increasing accuracy.

To reduce generalization error effectively, the diversity of the prediction models used in the first stage is important. Specifically, the outputs from individual prediction models had better be negatively correlated [34]. For example, in the case of a linear combiner, the classification error of a simple averaging ensemble denoted by E_{ave} is given in [34] as

$$E_{ave} = E_{add} \times \frac{1 + \delta \times (T-1)}{T} \tag{1}$$

where E_{add} is the added classification error of individual models, T is the number of prediction models used, and δ is the correlation coefficient among the outputs from individual models. If the individual models are equal ($\delta = 1$), then the ensemble error becomes identical to the individual error (*i.e.*, $E_{ave} = E_{add}$). If all of the models are uncorrelated ($\delta = 0$), the ensemble error becomes the average individual error (*i.e.*, $E_{ave} = E_{add}/T$). If the models are negatively correlated ($\delta < 0$), then the ensemble error becomes smaller than the average individual error (*i.e.*, $E_{ave} < E_{add}/T$).

2.2. Performance evaluation metrics

In this work, we formulate the problem of predicting miRNA–mRNA pairs as an instance of a binary classification task. To facilitate later discussion, we review the performance statistics widely used to evaluate different methodologies.

In the binary classification setting, evaluation metrics are based on the notion of true and false positives (TP and FP), and true and false negatives (TN and FN). TP (TN) refers to a positive (negative) instance that is correctly classified as positive (negative). FP (FN) is a negative (positive) instance that is incorrectly classified as positive (negative).

Based on these classifications, widely used performance metrics are defined [35]:

$$\text{sensitivity} = \text{true positive rate} = \text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

$$\text{specificity} = \text{true negative rate} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{3}$$

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4}$$

$$\text{positive predictive value(PPV)} = \text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5}$$

$$\text{negative predictive value(NPV)} = \frac{\text{TN}}{\text{TN} + \text{FN}} \tag{6}$$

$$\text{F-measure} = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{7}$$

2.3. Overview of the proposed approach

Fig. 1 shows the overview of our approach, which largely consists of three steps. First, we select six existing miRNA target prediction tools and let them produce the prediction results individually. Second, we preprocess the outcomes from the individual tools and then generate positive and negative training samples using a public database of experimentally verified miRNA–target pairs and various statistics obtained from the first stage results. Lastly, we train a binary classifier by using the

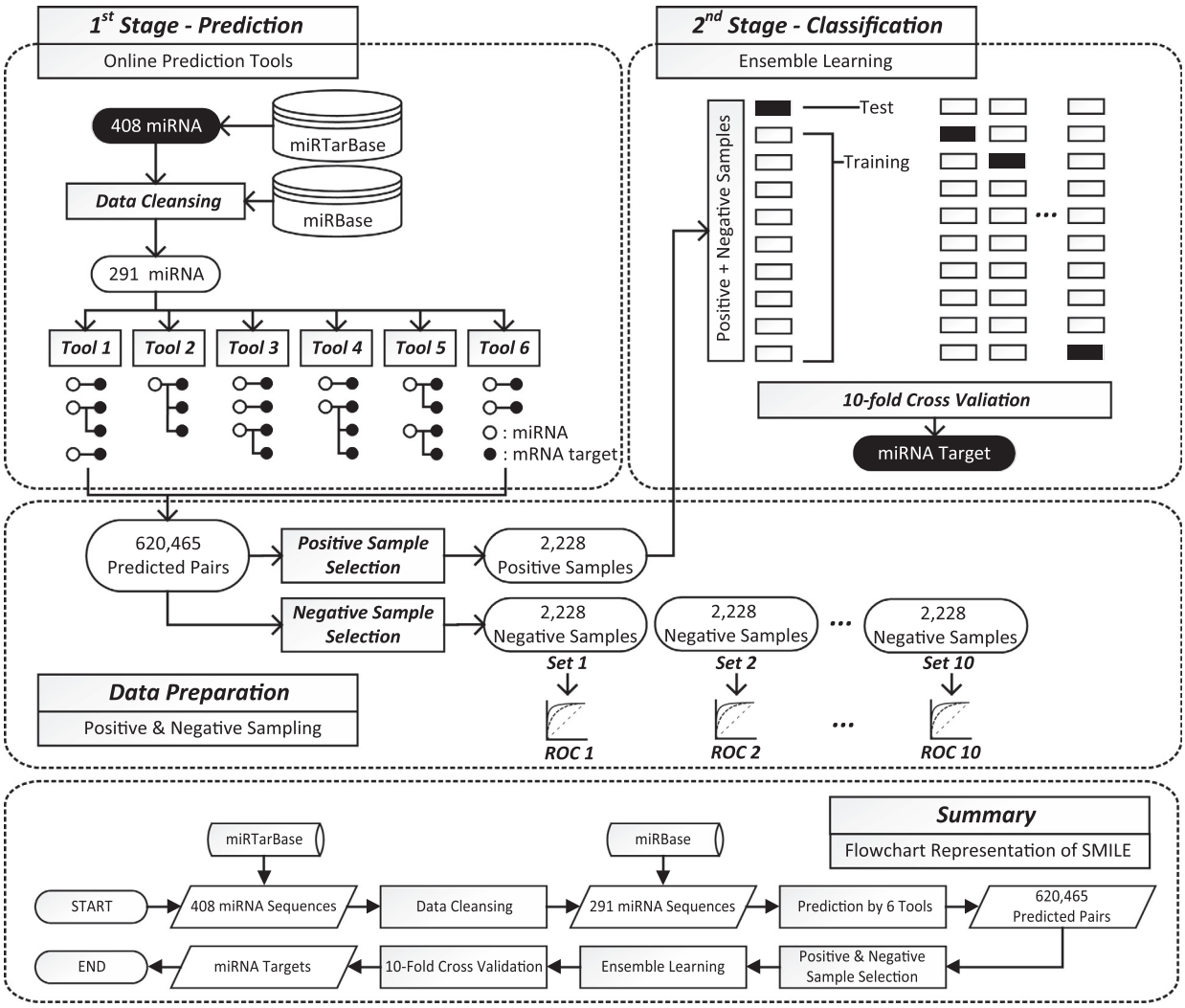

Fig. 1. Overview of the proposed method. For reference, we also include a flowchart representation of the proposed SMILE approach in the bottom pane.

training samples created in the second step and select the best model via cross-validation. More details of each step will follow next. For reference, we also include a flowchart representation of the proposed SMILE approach in the bottom pane of Fig. 1.

3. First-stage prediction by individual tools

3.1. Selection of tools and data preparation

In the first step of our approach, we predict miRNA targets using the following six tools: TargetScan [1], miRanda [7], DIANA-microT [9], PITA [11], miRTarget2 [12], and PicTar [10]. We selected these tools because they can process a large number of queries conveniently and provide prediction results via online interfaces and/or downloadable files.

For data preparation, we utilize miRTarBase [36], a public database that provides 408 human miRNAs with 3588 experimentally verified miRNA–mRNA pairs. Out of the 408 miRNAs, we select 291 (71.32%) by removing redundancy with reference to miRBase [37]. We let each of the above tools predict the target(s) of each of the 291 miRNAs using the default parameters. Note that every tool used reports a summary score that indicates the certainty of prediction. Since different summary scores have different ranges, we normalize all of the scores to make them ranged between 0 and 1.

For the input 291 miRNAs, the six tools produce 620,465 miRNA–mRNA pairs in total (Table 1). In the table, row *A* lists the numbers of miRNA–mRNA pairs predicted by the six tools. Each number in row *B* corresponds to the number of pairs detected by only a single tool. Row *C* lists the numbers of predicted pairs that are listed in miRTarBase. For instance, TargetScan predicts 149,979 pairs (row *A*); among these pairs, 62,907 pairs (row *B*) are detected only by TargetScan; among the 149,979 pairs, 1623 pairs are listed in miRTarBase. Note that, for the total counts in the last column of rows *A* and *C*, we count those pairs that are detected by multiple tools only once.

TargetScan is the tool that can retrieve the largest number of experimentally verified pairs stored in miRTarBase, followed by PITA. Nonetheless, they still fail to predict many of the verified pairs. Even by using six popular tools, only 2228 (62.62%) out of 3558 experimentally verified pairs are computationally predictable. This outcome clearly demonstrates a need for further improvements in miRNA target prediction.

3.2. Checking suitability for ensemble learning

As explained in Eq. (1), an ensemble approach works better when the outcomes from individual models are not correlated. In this work, the six prediction tools correspond to the individual models. To determine how the prediction results from the six tools

Table 1
Statistics on prediction of miRNA–mRNA pairs for 291 human miRNAs by six tools. Row *A* lists the numbers of miRNA–mRNA pairs predicted by the six tools. Each number in row *B* corresponds to the number of pairs detected by only a single tool. Row *C* lists the numbers of predicted pairs that are listed in miRTarBase.

	TargetScan	miRanda	DIANA-microT	PITA	miRTarget2	PicTar	total
A	149,979	12,234	10,012	449,477	122,426	29,115	620,465†
B	62,907	3,217	1,766	387,329	53,399	297	508,915
C	1,623	229	268	1,070	975	913	2,228†

A: the number of predicted pairs.
B: the number of pairs predicted only by a particular single tool.
C: the number of predicted pairs that are listed in miRTarBase.
†: pairs detected by multiple tools are counted only once.

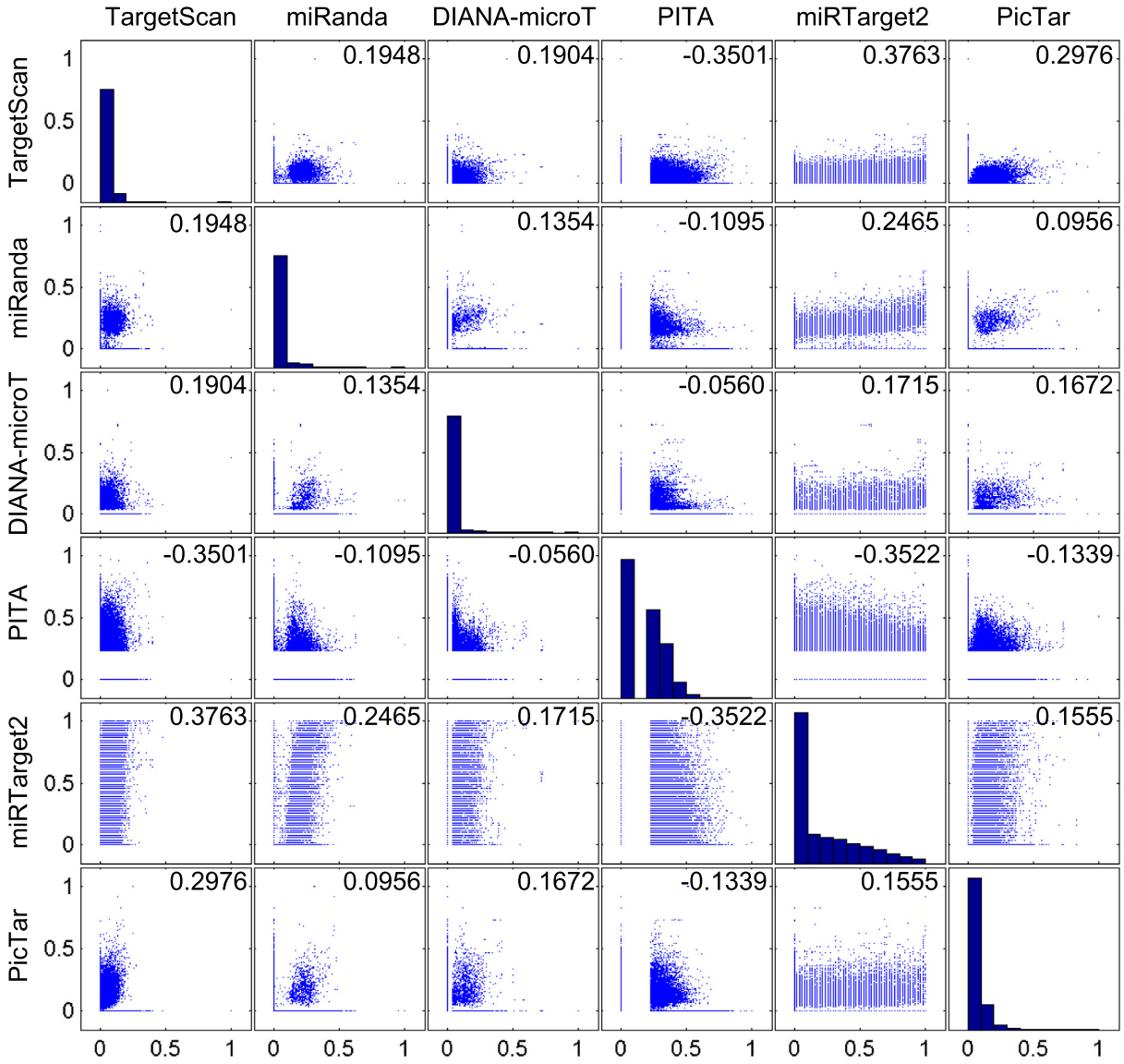

Fig. 2. Scatter-plot matrix of scores from different prediction tools. For the correlation analysis, we normalize the scores from each tool so that they range between [0,1]. The subplot drawn at location (i,j), $i \neq j$ is the scatter plot showing the correlation between prediction tools i and j, whereas the diagonal plot at location (i,i) is the histogram of score distributions for tool i. The number shown in each of the off-diagonal plots at location (i,j) denotes the correlation between tools i and j.

are related, we use a scatter-plot matrix showing the information on pairwise correlation between tools (Fig. 2). The correlation values range from −0.3522 to 0.3763, and no pair of tools shows a strong correlation in general. This result suggests that combining these tools has the potential to reduce generalization error compared with using them individually, according to the discussion in Section 2.1.

3.3. Comparison between experimentally verified and unverified pairs

Among the 620,465 pairs predicted by the six tools, 2228 are experimentally verified pairs stored in miRTarBase [36], and we call them *predicted and experimentally verified* (PEV) pairs, whereas the other 618,237 pairs are called *predicted but unverified* (PUV) pairs (Fig. 3).

Fig. 4 shows the score distributions of the PEV and PUV pairs for each of the six prediction tools. For all of the tools used, the average score of the PEV pairs is higher than that of the PUV pairs, and their variances are similar. Since the score distributions are not Gaussian, we perform the (one-sided) Wilcoxon rank-sum test [38] for the null hypothesis that the two distributions have the same mean, which is rejected at significance level of 0.05 for all of the prediction tools used.

Additionally, we can distinguish the PEV and PUV pairs in terms of the number of tools that can predict them, as listed in Table 2. This table shows how many of the six tools used can detect how many of the 618,237 PUV pairs and the 2228 PEV pairs. For instance, out of the total 618,237 PUV pairs, 508,915 pairs (82.32%) are detected by a single tool, and 77,680 pairs (12.56%) are predicted by two tools. According to this table, only 33.44% of PEV pairs are predicted by a single tool, and the remaining 66.56% are predicted by two or more tools. In contrast, most (82.32%) of PUV pairs are detected by a single tool, and only 17.68% are detected by multiple tools.

PEV and PUV pairs show different score distributions (Fig. 4) and prediction statistics (Table 2). We utilize these differences when preparing samples for the second-stage prediction, as elaborated in Section 4.2.

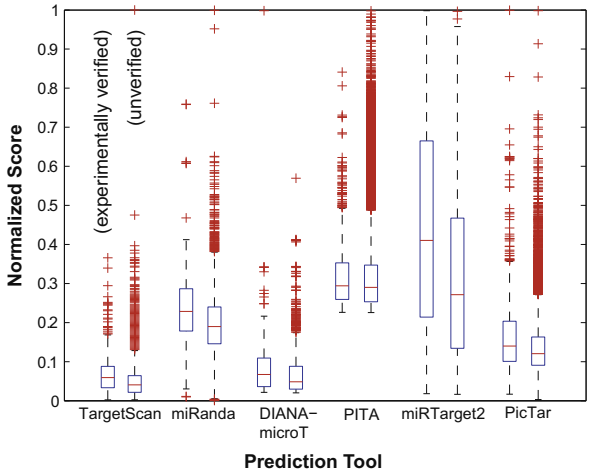

Fig. 4. Distribution of normalized scores. For each tool, the left box plot shows the score distribution of PEV pairs, whereas the right box plot shows that of PUV pairs. In each box plot, the line in the middle of a box indicates the median of the distribution presented, and the upper and lower boundaries of the box represent the values of the 75th and 25th percentiles, respectively. Symbol + indicates an outlier.

4. Second-stage classification by ensemble learning

4.1. Formulation of miRNA target prediction as a binary classification

We can formulate the task of miRNA target prediction as an instance of binary classification. Given an miRNA–mRNA pair, the objective is to classify it into either the 'positive' (true miRNA–mRNA interaction) class or the 'negative' (no interaction) class. To train a binary classifier, we need both positive and negative samples.

For positive samples, it is natural to use the 2228 PEV pairs. It is more challenging to prepare negative samples, because experimentalists are typically uninterested in collecting invalid miRNA–mRNA pairs and discard them [39]. Consequently, there is no database of 'experimentally verified' invalid pairs, to the best of the authors' knowledge.

4.2. Procedure to generate negative samples

Examples of negative-sample generation in the literature include using random sequences [13,17], deleting the target site from a target mRNA and using the remaining sequence [40], and collecting sequences from public databases, proteomics studies, and high-throughput cross-linking immunoprecipitation (CLIP) studies [16]. However, according to our experiments, applying these methods gives unsatisfactory training samples in that there are insufficient negative samples, or the differences between negative and positives samples are too salient, making classification a trivial task.

In this work, we utilize the differences in prediction statistics between experimentally verified and unverified miRNA–mRNA pairs to generate negative samples. More specifically, our approach is based on the following observations:

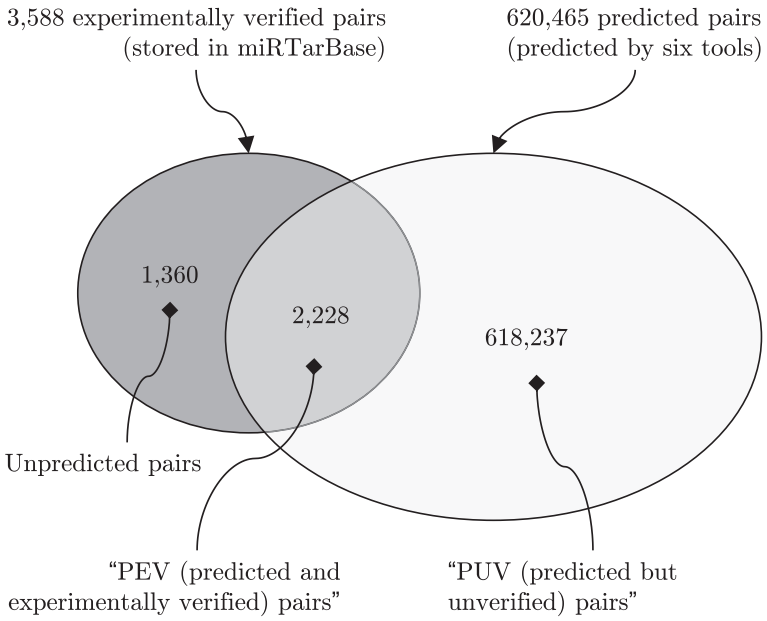

Fig. 3. Definitions: predicted and experimentally verified (PEV) pairs and predicted but unverified (PUV) pairs.

Table 2
The number of tools that predict PUV and PEV pairs. This table shows how many of the six tools used can detect how many of the 618,237 PUV pairs and the 2,228 PEV pairs.

	Predicted by						Total
	1 Tool	2 Tools	3 Tools	4 Tools	5 Tools	6 Tools	
# PUV pairs	508,915	77,680	24,269	6,007	1,141	225	618,237
(%)	82.32	12.56	3.93	0.97	0.18	0.04	100
# PEV pairs	745	651	447	260	100	25	2228
(%)	33.44	29.22	20.06	11.67	4.49	1.12	100

1. PEV pairs are likely to be predicted by multiple tools. According to Table 2, 66.56% of PEV pairs are detected by two or more tools. In contrast, only 17.68% of PUV pairs are detected by multiple tools. This suggests that relatively high number of false positives is likely to be contained in the remaining 82.32% of PUV pairs.
2. The average score of the PEV pairs is higher than that of the PUV pairs. After the normalization process to place the output scores of different tools in the same range, the average score of the PEV pairs is 29.53% higher than that of the PUV pairs on average (43.94% higher for TargetScan, 31.10% for miRanda, 35.26% for DIANA-mircroT, 2.01% for PITA, 37.73% for miRTarget2, and 27.11% for PicTar).
3. The minimum score of the PEV pairs is not smaller than that of the PUV pairs. For TargetScan, PITA, and miRTarget2, the minimum scores are identical, whereas the minimum score of PEV pairs is higher by 10% or more for miRanda, DIANA-microT, and PicTar.

Fig. 5 outlines our algorithm to label a PUV pair. The input is composed of three six-dimensional vectors and a scalar: (1) vector S_{PUV} in which $S_{PUV}(i)$ is the score of the input PUV pair tool i returns; (2) vector M_{PEV} in which $M_{PEV}(i)$ is the minimum score of a PEV pair tool i returns; (3) vector A_{PEV} in which $A_{PEV}(i)$ is the average score of all PEV pairs tool i returns; and (4) the scalar t, a threshold to determine the boundary between positive and negative samples. Note that in essence, the algorithm increases $S_{PUV}(i)$ by one if it is greater than the average score of all PEV pairs for tool i. If $S_{PUV}(i)$ is less than the minimum score of a PEV pair for tool i, then it is decreased by one. S_{tmp} stores the updated score of the input PUV pair. Note that the above updating operation gives the effect of separating scores according to the prediction strength (Fig. 6):

1. negative (strong): $S_{tmp}(i) = -1$
2. negative (weak): $-1 < S_{tmp}(i) < M_{PEV}(i) - 1$
3. positive (weak): $M_{PEV}(i) \leqslant S_{tmp}(i) \leqslant A_{PEV}(i)$
4. positive (strong): $1 + A_{PEV}(i) < S_{tmp}(i) \leqslant 2$

After this update is completed for all tools, the algorithm sums S_{tmp} element-wise and compares the result with the t threshold for labeling. The input PUV pair is labeled negative only if the sum is less than or equal to the threshold t.

Depending on the value of t, the algorithm gives different sets of negative samples. In our experiments, we set $t = -4$, which corresponds to the assumption that a negative sample has one of the following characteristics: (i) A negative sample is predicted by only one tool, and its score is less than the average score of all PEV pairs, or (ii) A negative sample is predicted by two or more tools, but the score from each tool is less than the minimum score of a PEV pair.

Using $t = -4$ gives 354,710 negative-labeled PUV pairs. Out of these pairs, we randomly sample 10 sets of 2228 pairs (the same as the number of positive samples) with replacement. For each

```
1: procedure LabelPUV(S_PUV, M_PEV, A_PEV, t)
2: // S_PUV ∈ ℝ^6                          ▷ score vector of input PUV pair
3: // M_PEV ∈ ℝ^6                          ▷ min-score vector for a PEV pair
4: // A_PEV ∈ ℝ^6                          ▷ avg-score vector for PEV pairs
5: // t ∈ ℝ                                ▷ threshold to decide positive/negative
6:     S_tmp ← S_PUV                       ▷ temporary score vector
7:     for i ← 1 to 6 do                   ▷ for each of 6 tools
8:         if S_PUV(i) > A_PEV(i) then
9:             S_tmp(i) ← S_tmp(i) + 1
10:        else if S_PUV(i) < M_PEV(i) then
11:            S_tmp(i) ← S_tmp(i) − 1
12:        end if
13:    end for
14:    if sum{S_tmp(:)} > t then
15:        return Positive
16:    else
17:        return Negative
18:    end if
19: end procedure
```

Fig. 5. Algorithm for labeling a PUV pair.

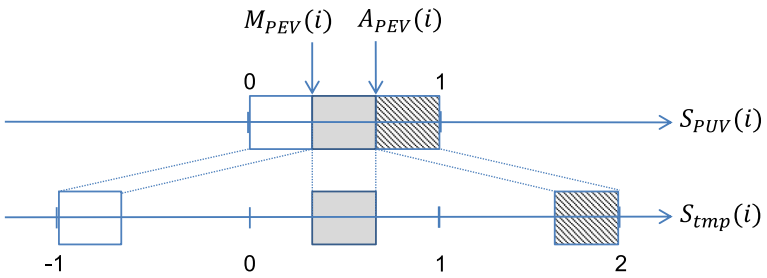

Fig. 6. The updating action in lines 7–13 of the algorithm in Fig. 5 gives the effect of separating the score of the input PUV pair into the four categories: strong negative ($S_{tmp}(i) = -1$), weak negative ($-1 < S_{tmp}(i) < M_{PEV}(i) - 1$), weak positive ($M_{PEV}(i) \leqslant S_{tmp}(i) \leqslant A_{PEV}(i)$), and strong positive ($1 + A_{PEV}(i) < S_{tmp}(i) \leqslant 2$).

set of negative samples, we train the second-stage classifier along with the 2228 positive samples. Section 5.2 presents experimental results regarding the effect of the parameters used to generate negative samples (*i.e.*, their relative quantity for positive samples and the threshold) on the prediction performance.

5. Results and discussion

5.1. Main results

Fig. 7 shows the receiver operating characteristic (ROC) curves of the proposed SMILE method and the six prediction tools used for comparison. This is one of the ten sets of ROC curves informally depicted in Fig. 1. A ROC curve is a plot of the true positive rate versus the false positive rate in binary classification and is frequently used to visualize and compare the performance of a binary classifier as its threshold for discrimination is varied. The area-under-the-curve (AUC) values of different classifiers are used for comparing their performance. An ideal binary classifier would have an AUC value of 1.0.

In our experimental results, the average AUC values for SMILE, TargetScan, miRanda, DIANA-microT, PITA, miRTarget2, and PicTar are 0.9206, 0.8187, 0.5472, 0.5581, 0.4596, 0.6727, and 0.7045, respectively. SMILE outperforms the alternatives in terms of the AUC by a minimum of 12.45% (with respect to TargetScan) and a maximum of 100.30% (with respect to PITA), with an average enhancement of 52.25%. This result confirms the effectiveness of the proposed approach as a binary classifier that discriminates between real miRNA–mRNA pairs and false ones.

Table 3 lists the prediction results from SMILE and miRNAMap2.0 [25], another integrative miRNA target predictor. To generate this table, we randomly sample 100 pairs from the 2228 PEV pairs and let the two tools each predict whether or not these 100 samples are valid. For this specific dataset, SMILE predicts 81 pairs, and miRNAMap2.0 returns 43 pairs as valid. Table 3 lists 10 results sampled from our experimental results. Note that for, a given miRNA–mRNA pair, SMILE not only reports the details of prediction but also explicitly suggests whether it is valid or not (indicated by symbols ◯ and × in the table, respectively) based on ensemble learning, whereas mirRNAMap2.0 only lists the prediction results from the individual prediction tools used without any integrative analysis or collective prediction recommendations. This result demonstrates that the proposed SMILE approach is not only effective in target prediction performance but also convenient for users in that the prediction result is explicitly provided as a binary label, leveraged by the ensemble-learning technique.

Recall that our approach consists of a two-stage classification. In the first stage, individual prediction algorithms predicts possible miRNA–mRNA pairs, and positive and negative samples are collected. In the second stage, by combining the results from the first stage, another round of classification is implemented for final, integrative prediction. Given this procedure, the factors that affect the performance of the proposed approach include the following: (1) the prediction results from the individual tools in the

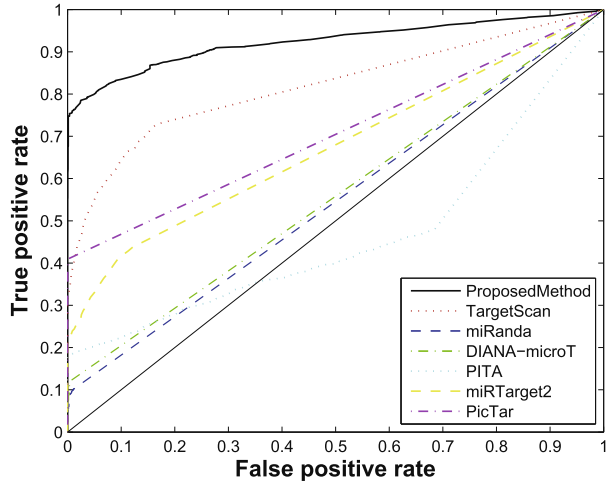

Fig. 7. Comparison of the proposed method with six existing tools in terms of receiver operating characteristic (ROC) curves. This is one of the ten sets of ROC curves informally depicted in Fig. 1. On average, our approach outperforms the others by 52.5% in terms of the area-under-the-curve(AUC). The AUC values of different classifiers are used for comparing their performance, and an ideal binary classifier would have an AUC value of 1.0.

Table 3
Result comparison with miRNAMap2.0 [25], another integrative tool for miRNA–mRNA interaction prediction. Symbol ◯ denotes correct prediction by SMILE while symbol × denotes no prediction by SMILE; symbol - indicates that no result is returned by miRNAMap2.0.

(miRNA, target)	SMILE	miRNAMap2.0			
		MFE	SM	TL	PC
(hsa-let-7d-5p, APP)	◯	−24.4	–	RNAhybrid	0.08
(hsa-miR-1, GAK)	◯	–	–	–	–
(hsa-miR-29a-3p, CDC42)	×	–	–	–	–
(hsa-miR-483–3p, SMAD4)	◯	–	–	–	–
(hsa-miR-204–5p, TCF12)	◯	−12.3	154	miRanda	0.04
		−19.1	–	TargetScan	0.04
(hsa-miR-15a-5p, WNT3A)	◯	–	–	–	–
(hsa-miR-34c-5p, MET)	◯	−24.6	–	RNAhybrid	−0.42
		−17.4	166	miRanda	−0.42
		−23.7	–	TargetScan	−0.42
(hsa-miR-182–5p, RARG)	◯	−13.3	158	miRanda	−0.23
		−18.4	–	TargetScan	−0.23
(hsa-miR-200c-3p, ZFPM2)	◯	–	–	–	–
(hsa-miR-1, HSPD1)	×	−17.6	–	RNAhybrid	0.06
		−17.1	–	TargetScan	0.06

Abbreviations: MFE, minimum free energy; SM, score returned by miRanda; TL, the prediction tool invoked by miRNAMap2.0; PC, Pearson's correlation coefficient in expression levels between miRNA and target.

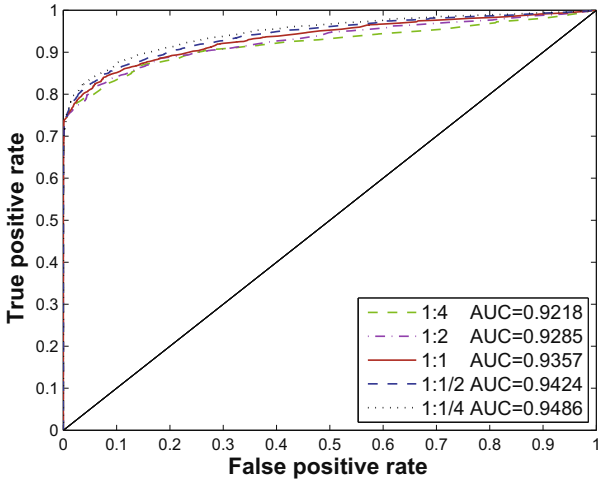

Fig. 8. Effects of the number of negative samples. # positive = 2,228. # positive: # negative = 1:4, 1:2, 1:1, 1:0.5, and 1:0.25. We observe that the effect of using different numbers of negative samples on the AUC is negligible, although using fewer negative samples than positive samples gives slightly higher AUC values.

first stage, (2) the selection of negative samples, and (3) the second-stage classification procedure that works based on the first-stage results. The remainder of this section will present the experimental results related to these factors in turn.

5.2. Effects of negative samples

As explained earlier, experimentally verified negative samples are rare when compared with experimentally verified positive samples. Therefore, it would be desirable to train the classifier even with a small number of negative samples. We analyze the effect of the number of negative samples on the second-stage classification.

To this end, when training the second-stage classifier, we fix the number of positive samples to 2228 but use five different numbers of negative samples: 1/4 and 1/2 of 2228 and a few multiples of 2228. Fig. 8 shows how the ROC curves are affected by the variation in the negatives samples.

We observe that the effect of using different numbers of negative samples on the AUC is negligible, although using fewer negative samples than positive samples gives slightly higher AUC values. This result suggests that we can prepare a relatively small number of negative samples and use them for the ensemble learning stage.

Another aspect of preparing negative samples is how to set the decision threshold t explained in Section 4.2. Recall that a higher threshold means stricter criteria for a predicted pair to be called a positive sample. In other words, as we increase the threshold, more predicted pairs will be called negative. For a low threshold, a predicted pair is called positive if at least one tool gives a score higher than the minimum. In contrast, for a high threshold, a predicted pair is called positive if almost all tools detect it.

We compare the performance of the six base learners in terms of the six statistics defined in Section 2: sensitivity, specificity, accuracy, positive predictive value (PPV), negative predictive value (NPV), and F-measure. Fig. 9 shows how these six different statistics vary as we change the threshold from −5 to +6. When the threshold is −5, only 0.04% of the 618,237 UV-pairs are labeled negative, whereas 99.98% are labeled negative if the threshold is +6. We generate these datasets so that they can cover not only common cases occurring in real data but also extreme cases that rarely appear in reality.

In Fig. 9, three tools (*i.e.*, miRanda, DIANA-microT, and PicTar) show higher sensitivity and NPV than the proposed method. According to the definition of these two metrics, they commonly have the number of false negatives (FN) in the denominator. That is, high sensitivity and NPV imply that FN is low. However, not only accurate classifiers but also those that classify most samples as positive can also have a low FN. To distinguish such classifiers from accurate ones, we need to consider specificity and PPV, whose

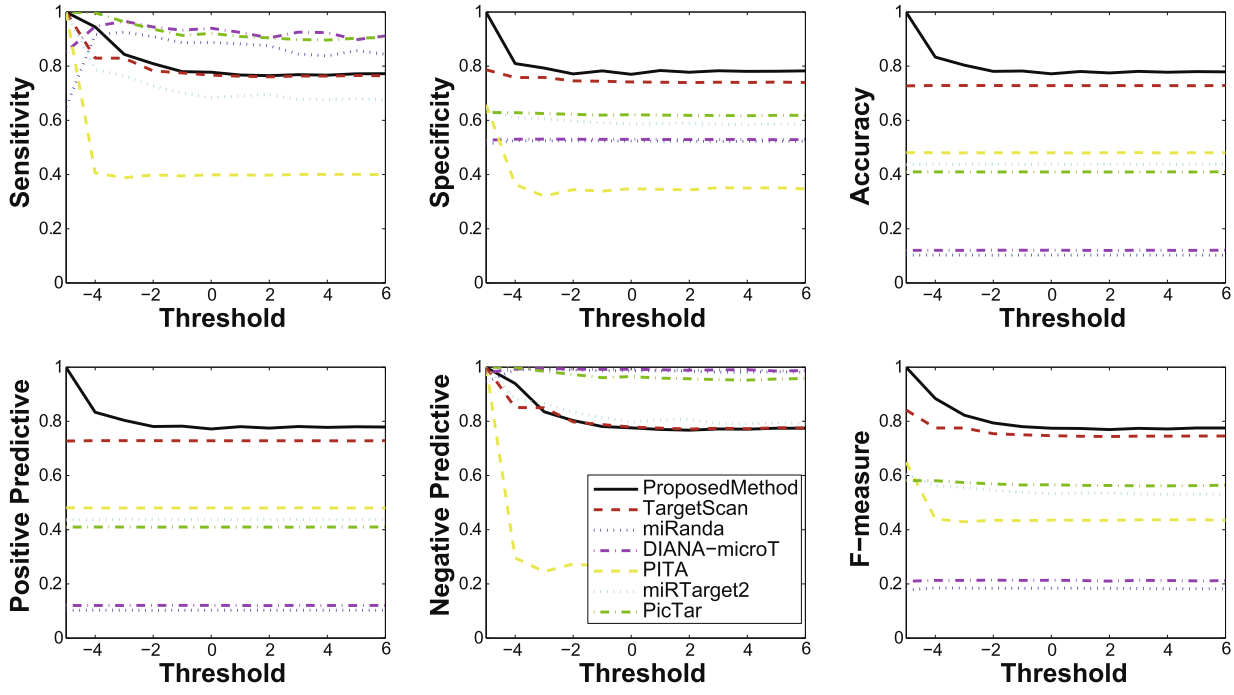

Fig. 9. Effects of the selection threshold on the performance of the second-stage classification. This figure shows how the six different statistics (sensitivity, specificity, accuracy, positive predictive value, negative predictive value, and F-measure) vary as we change the threshold from −5 to +6.

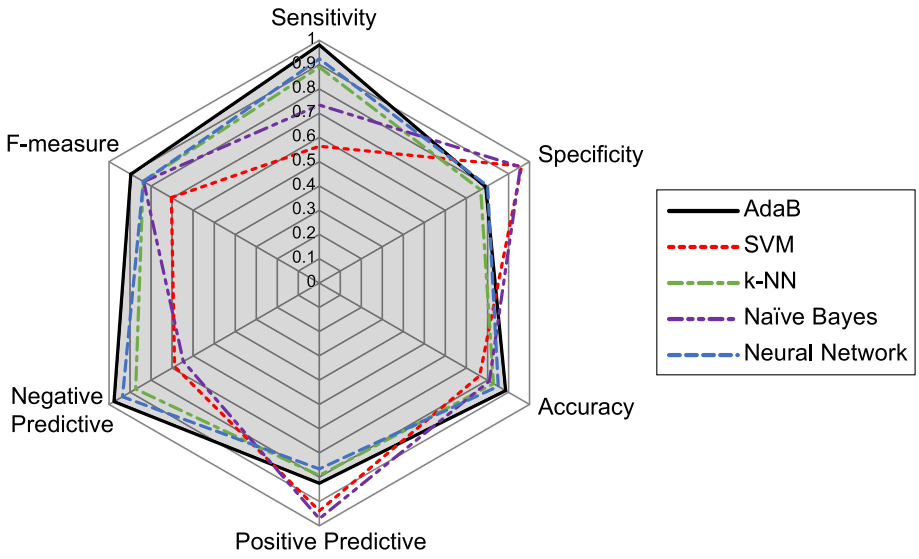

Fig. 10. Radar chart for comparison of classifier performance for the second-stage classification. Five types of classifiers (AdaBoost, *k*-nearest neighbor, artificial neural network, support vector machine, and naïve Bayes) are used. Refer to Table 4 for more details.

Table 4
Details of the statistics presented in Fig. 10.

	AdaBoost	SVM	k-NN	naïve Bayes	ANN
Sensitivity	0.9813	0.5636	0.8892	0.7334	0.9237
Specificity	0.7920	0.9627	0.7678	0.9569	0.8007
Accuracy	0.8867	0.7632	0.8285	0.8090	0.8512
PPV	0.8251	0.9387	0.7930	0.9709	0.7661
NPV	0.9766	0.6881	0.8734	0.6466	0.9360
F-measure	0.8963	0.7037	0.8382	0.8354	0.8369
AUC	0.9279	0.8954	0.8688	0.8922	0.9066

definitions contain the number of false positives (FP) in the numerator. As is evident in Fig. 9, the above mentioned three tools exhibit lower specificity and PPV than the proposed approach.

5.3. Comparison of classifier performance for the second-stage prediction

As the second-stage classifier, we consider five alternatives and compare their performance in terms of the aforementioned six statistics. The classifiers tested are AdaBoost, support vector machine (SVM), *k*-nearest neighbor classifier (*k*-NN), naïve Bayes classifier, and artificial neural network (ANN). More details for each classification method can be found in [41–43].

For SVM, the radial basis function (RBF) kernel gives the best result among the four types tried (linear, polynomial, sigmoid, and RBF). For ANN, varying the number of layers does not affect the accuracy significantly, and we use ANN with 10 neurons configured in a single layer. For *k*-NN, we sweep the value of *k* and find that $k = 5$ gives the best result. For AdaBoost, we use a decision tree as the component weak leaner and empirically choose 250 as the number of trees.

Fig. 10 presents the performance measurement result in a radar chart, while Table 4 lists more detailed statistics presented in the radar chart. Overall, we can group the classifiers into two according to their statistics pattern on the radar chart. The first group (AdaBoost, *k*-NN, and ANN) tends to outperform the second group (SVM and NB) in all statistics but specificity and PPV. In this work, we decide to choose AdaBoost as the second-stage predictor, since it gives us the highest AUC value and excellent performance in the most statistics measured. Other methodologies may be employed if different characteristics are desired depending on the user's need.

6. Conclusion

Since the discovery of miRNA, computational tools have been invaluable in uncovering the mechanism underlying the biology of miRNA–mRNA interactions [44], although there is still much room for improvements in terms of prediction accuracy. This challenge originates partly from the amount of exceptions and the degree of subtlety in miRNA–mRNA interactions that are difficult to model under an engineering framework, and partly from the lack of high-throughput biological methods that can screen out the invalid predictions at a reasonable cost and time.

Nevertheless, it is clear that positive loops between computational prediction and biological validation are already formed, leading to gradual and continuous advances in the field. In this context, employing ensemble learning techniques for predicting miRNA targets is appropriate. Ensemble learning provides a natural and powerful means to fuse different (often conflicting) discoveries from independent research activities. Any breakthrough from a novel methodology can be immediately and easily reflected in an ensemble framework such as the proposed SMILE. Consequently, we anticipate that an increasing number of approaches will adopt a mechanism to integrate the results from multiple tools with the aim of boosting the robustness and accuracy of miRNA target prediction.

The current version of SMILE combines the results obtained from running multiple tools on the human miRNA and mRNA data. Considering different but related species together with the human data may produce even more accurate result in target prediction, given that evolutionary conservation has played key roles in many state-of-the-art miRNA target prediction tools. Curating more instances of negative samples is also critical to further enhance the performance of the proposed approach. Alternatively, anomaly detectors [45] or one-class classifiers may be employed, considering that there is a significant imbalance in the number of positive and negative examples. We are planning to incorporate these revisions into future versions of SMILE to deliver even more accurate and robust prediction performance than currently possible. Accompanied by future advances in biological verification

schemes, we expect our approach to be an invaluable resource for miRNA research.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grants funded by the Korea government (Ministry of Science, ICT and Future Planning) [Nos. 2011-0009963, 2012-R1A2A4A01008475, 2011-0020128, and 2012-M3A9D1054622].

References

[1] B.P. Lewis, I.-h. Shih, M.W. Jones-Rhoades, D.P. Bartel, C.B. Burge, et al., Cell 115 (7) (2003) 787–798.
[2] B.P. Lewis, C.B. Burge, D.P. Bartel, Cell 120 (1) (2005) 15–20.
[3] J. Lu, G. Getz, E.A. Miska, E. Alvarez-Saavedra, J. Lamb, D. Peck, A. Sweet-Cordero, B.L. Ebert, R.H. Mak, A.A. Ferrando, et al., Nature 435 (7043) (2005) 834–838.
[4] G.A. Calin, C.M. Croce, Nat. Rev. Cancer 6 (11) (2006) 857–866.
[5] X. Dai, Z. Zhuang, P.X. Zhao, Briefings Bioinf. 12 (2) (2011) 115–121.
[6] A. Krek, D. Grün, M.N. Poy, R. Wolf, L. Rosenberg, E.J. Epstein, P. MacMenamin, I. da Piedade, K.C. Gunsalus, M. Stoffel, et al., Nat. Genet. 37 (5) (2005) 495–500.
[7] A.J. Enright, B. John, U. Gaul, T. Tuschl, C. Sander, D.S. Marks, et al., Genome Biol. 5 (1) (2004). R1–R1.
[8] M. Rehmsmeier, P. Steffen, M. Höchsmann, R. Giegerich, RNA 10 (10) (2004) 1507–1517.
[9] M. Maragkakis, P. Alexiou, G.L. Papadopoulos, M. Reczko, T. Dalamagas, G. Giannopoulos, G. Goumas, E. Koukis, K. Kourtis, V.A. Simossis, et al., BMC Bioinf. 10 (1) (2009) 295.
[10] X. Wang, I.M. El Naqa, Bioinformatics 24 (3) (2008) 325–332.
[11] M. Kertesz, N. Iovino, U. Unnerstall, U. Gaul, E. Segal, Nat. Genet. 39 (10) (2007) 1278–1284.
[12] D. Grün, Y.-L. Wang, D. Langenberger, K.C. Gunsalus, N. Rajewsky, PLoS Comput. Biol. 1 (1) (2005) e13.
[13] O. SaeTrom, O. SNØVE, P. SÆTROM, RNA 11 (7) (2005) 995–1003.
[14] M. Sturm, M. Hackenberg, D. Langenberger, D. Frishman, BMC Bioinf. 11 (1) (2010) 292.
[15] S. Bandyopadhyay, R. Mitra, Bioinformatics 25 (20) (2009) 2625–2631.
[16] P.H. Reyes-Herrera, E. Ficarra, A. Acquaviva, E. Macii, BMC Bioinf. 12 (1) (2011) 454.
[17] M. Yousef, S. Jung, A.V. Kossenkov, L.C. Showe, M.K. Showe, Bioinformatics 23 (22) (2007) 2987–2992.
[18] M. Reczko, M. Maragkakis, P. Alexiou, G.L. Papadopoulos, A.G. Hatzigeorgiou, Front. Genet., vol. 2.
[19] Y. Friedman, G. Naamati, M. Linial, Bioinformatics 26 (15) (2010) 1920–1921.
[20] S. Nam, M. Li, K. Choi, C. Balch, S. Kim, K.P. Nephew, Nucl. Acids Res. 37 (Suppl. 2) (2009) W356–W362.
[21] A. Bisognin, G. Sales, A. Coppe, S. Bortoluzzi, C. Romualdi, Nucl. Acids Res. 40 (W1) (2012) W13–W21.
[22] J. Ding, D. Li, U. Ohler, J. Guan, S. Zhou, BMC Genom. 13 (Suppl. 3) (2012) S3.
[23] F. Xiao, Z. Zuo, G. Cai, S. Kang, X. Gao, T. Li, Nucl. Acids Res. 37 (Suppl. 1) (2009) D105–D110.
[24] S. Cho, Y. Jun, S. Lee, H.-S. Choi, S. Jung, Y. Jang, C. Park, S. Kim, S. Lee, W. Kim, Nucl. Acids Res. 39 (Suppl. 1) (2011) D158–D162.
[25] S.-D. Hsu, C.-H. Chu, A.-P. Tsou, S.-J. Chen, H.-C. Chen, P.W.-C. Hsu, Y.-H. Wong, Y.-H. Chen, G.-H. Chen, H.-D. Huang, Nucl. Acids Res. 36 (Suppl. 1) (2008) D165–D169.
[26] P. Alexiou, T. Vergoulis, M. Gleditzsch, G. Prekas, T. Dalamagas, M. Megraw, I. Grosse, T. Sellis, A.G. Hatzigeorgiou, Nucl. Acids Res. 38 (Suppl. 1) (2010) D137–D141.
[27] P. Sethupathy, M. Megraw, A.G. Hatzigeorgiou, Nat. Methods 3 (11) (2006) 881–886.
[28] N. Rajewsky, Nat. Genet. 38 (2006) S8–S13.
[29] H. Min, S. Yoon, Exp. Mol. Med. 42 (4) (2010) 233–244.
[30] D.H. Wolpert, Neural Networks 5 (2) (1992) 241–259.
[31] G. Tzanis, C. Berberidis, I. Vlahavas, Comput. Biol. Med. 42 (1) (2012) 61–69.
[32] G. Brown, Ensemble Learn.
[33] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[34] K. Tumer, J. Ghosh, Connection Sci. 8 (3–4) (1996) 385–404.
[35] J. Han, M. Kamber, J. Pei, Data Mining: Concepts and Techniques, Morgan Kaufman, 2006.
[36] S.-D. Hsu, F.-M. Lin, W.-Y. Wu, C. Liang, W.-C. Huang, W.-L. Chan, W.-T. Tsai, G.-Z. Chen, C.-J. Lee, C.-M. Chiu, et al., Nucl. Acids Res. 39 (Suppl. 1) (2011) D163–D169.
[37] S. Griffiths-Jones, R.J. Grocock, S. Van Dongen, A. Bateman, A.J. Enright, Nucl. Acids Res. 34 (Suppl. 1) (2006) D140–D144.
[38] F. Wilcoxon, Biomet. Bull. 1 (6) (1945) 80–83.
[39] E. Ficarra et al., Genom. Proteom. Bioinform.
[40] S.-K. Kim, J.-W. Nam, J.-K. Rhee, W.-J. Lee, B.-T. Zhang, BMC Bioinf. 7 (1) (2006) 411.
[41] Y. Freund, R.E. Schapire, J. Comput. Syst. Sci. 55 (1) (1997) 119–139.
[42] C. Cortes, V. Vapnik, Mach. Learn. 20 (3) (1995) 273–297.
[43] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, New York, 2006.
[44] S. Yoon, G. De Micheli, Birth Defects Res. Part C: Embryo Tod.: Rev. 78 (2) (2006) 118–128.
[45] V. Chandola, A. Banerjee, V. Kumar, ACM Comput. Surv. (CSUR) 41 (3) (2009) 15.