Hindawi
Scientific Programming
Volume 2021, Article ID 9969282, 9 pages
https://doi.org/10.1155/2021/9969282

Research Article

Deep Multiple Kernel Learning for Prediction of MicroRNA Precursors

Hengyue Shi, Dong Wang, Peng Wu, Yi Cao, and Yuehui Chen

Information Science and Technology, University of Jinan, Jinan 250022, China

Correspondence should be addressed to Peng Wu; ise_wup@ujn.edu.cn

Received 12 March 2021; Revised 1 April 2021; Accepted 19 April 2021; Published 5 May 2021

Academic Editor: Yi-Zhang Jiang

MicroRNAs are a group of noncoding RNAs that are about 20–24 nucleotides in length. They are involved in the physiological processes of many diseases and regulate transcriptional and post-transcriptional gene expression. Therefore, the prediction of microRNAs is of great significance for basic biological research and disease treatment. MicroRNA precursors are the necessary stage of microRNA formation. RBF kernel support vector machines (RBF-SVMs) and shallow multiple kernel support vector machines (MK-SVMs) are often used in microRNA precursors prediction. However, the RBF-SVMs could not represent the richer sample features, and the MK-SVMs just use a simply convex combination of few base kernels. This paper proposed a localized multiple kernel learning model with a nonlinear synthetic kernel (LMKL-D). The nonlinear synthetic kernel was trained by a three-layer deep multiple kernel learning model. The LMKL-D model was tested on 2241 pre-microRNAs and 8494 pseudo hairpin sequences. The experiments showed that the LMKL-D model achieved 93.06% sensitivity, 99.27% specificity, and 98.03% accuracy on the test set. The results showed that the LMKL-D model can increase the complexity of kernels and better predict microRNA precursors. Our LMKL-D model can better predict microRNA precursors compared with the existing methods in specificity and accuracy. The LMKL-D model provides a reference for further validation of potential microRNA precursors.

1. Introduction

MicroRNAs are a class of highly conserved endogenous noncoding RNAs with a length of about 20–24 nucleotides. They are single stranded and regulate gene expression at the post-transcriptional or translational level by binding specifically to target messenger RNA [1, 2]. Studies have shown that some microRNAs can play a role by regulating cell proliferation, cell migration, invasion, and immune response [3], and at the same time, microRNAs can also play an important role in inflammatory response [4], neural development, and other processes [5, 6]. In organisms, microRNA is first transcribed by RNA polymerase II into long initial transcription, primary microRNA, which is then processed by Drosa enzyme into a precursor with a length of about 70 nucleotides, that is, pre-microRNA [3, 7]. Pre-microRNA is exported from the nucleus with the help of RanGTP/exportin 5 and then exported to be processed and matured by the Dicer enzyme in the cytoplasms [8, 9]. After being processed into mature microRNAs, microRNAs form RNA-induced silencing complex (RISC) in some way to affect the protein abundance of target genes by inhibiting translation or degrading the mRNAs of target genes.

MicroRNA precursors (pre-microRNAs) can fold into hairpin structures, which are considered the most important indicators of microRNA maturation [10]. Figure 1 shows a pre-microRNA sequence and its hairpin structure. However, there are a large number of nonprecursors with similar hairpin structures in many genomic regions, which are called pseudo hairpin sequences [11]. Accurately and effectively identifying microRNA precursors from a large number of candidate hairpin sequences is a challenging task [12].

The methods of finding new microRNAs mainly include biological experimental methods and computer prediction methods [13]. Biological experimental methods are more direct and highly reliable, but the expression level of microRNAs is relatively low. Some microRNAs are only

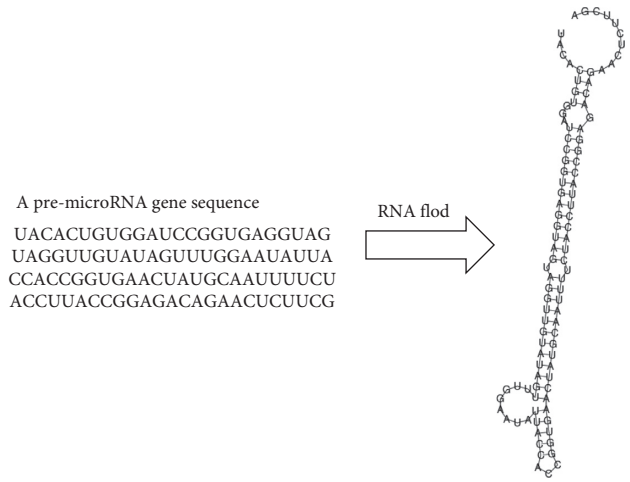

Figure 1: Structure of the pre-microRNA.

expressed under specific conditions, such as cell type and physiological state of the body. Moreover, due to the high cost and long experimental cycle, it is difficult to replicate microRNA expressed in a specific tissue and period. With the help of computer, the computer prediction method can identify new microRNAs more efficiently. The prediction method of microRNAs based on machine learning has been applied to bioinformatics, which can overcome various defects of biological experimental methods, prevent microRNAs from being affected by expression time, tissue specificity, or expression level, and provide reliable samples for subsequent biological experiments.

MicroRNA precursors have a unique hairpin structure and are easier to obtain than microRNAs. Thus, computational prediction methods use machine learning to mainly identify microRNA precursors from candidate hairpin sequences. The authors in [14] and [15] proposed a set of novel features and used a support vector machine (SVM) with only a RBF kernel to classify real and pseudo pre-microRNAs and proposed Triplet-SVM and PremipreD While different kernels have different characteristics, a RBF kernel could not adequately map the pre-microRNAs to the appropriate feature spaces. When the features of input data contain heterogeneous information [16, 17] or the data are nonflat in the high-dimensional feature space [18], it is not reasonable to use a single simple kernel to map all the input data. The authors in [19] used a random forest classifier to find shallow features and only got 91.29% accuracy and proposed MipIe. The authors in [20] adopted multiple kernel SVM with different weights, but only the shallow features are used and then LMKL-MiPred was proposed. It shows good accuracy but no deep features of pre-microRNAs were explored. The authors in [21] used a simple three-layer backpropagation neural network and proposed MiRANN. However, when there are limited candidate hairpin sequences, the three-layer backpropagation neural networks typically do not have a good generalization performance, and they can even increase the risk of over-fitting under some conditions.

Multiple kernel methods have been successful on small data sets. By mapping the samples into a high-dimensional reproducing kernel Hilbert space, they only use very few parameters to enable a classifier to learn a complex decision boundary. How to determine the basic kernel function is the difficulty and key problem of multiple kernel learning. The localized multiple kernel learning [22] uses different weights to combine simple basic kernel (linear kernel, polynomial kernel, and RBF kernel) but could not obtain the deep features of the samples. This paper presents a localized multiple kernel learning model with a nonlinear deep synthetic kernel (LMKL-D). The deep synthetic kernel was trained by a deep multiple kernel learning model with a tree structure [23]. We found that the neural networks are easy to obtain deep features by gradient descent. Thus, we adopt the gradient descent approach and use a deep multiple kernel learning model with a tree structure to get a nonlinear deep synthetic kernel. We combine kernels at each layer and then optimize over an estimate of the leave one out error [23]. Starting from some simple basic kernels, a deep synthetic kernel can be achieved after a learning process. We combined the deep synthetic kernel and other simple basic kernels by localized multiple kernel learning. The deep synthetic kernel was composed of complex combination of basic kernels. Thus, the LMKL-D model can take advantage of both the shallow and deep features of the input data. As a result, the LMKL-D model can represent more features and obtain better performance than existing SVM methods.

The rest of the paper is organized as follows. In Section 2, we introduce datasets selection and feature selection and then we provide the background about SVM, localized multiple kernel learning, and the multiple kernel learning methods. Kernels and model selection are also included in this section. In Section 3, we show the experimental results and comparisons with other methods. Finally, conclusions and future work are shown in Section 4.

2. Materials and Methods

2.1. Biologically Relevant Datasets. The LMKL-D model proposed in this paper should be able to correctly identify pre-microRNAs and pseudo hairpin sequences from the candidate hairpin sequences dataset. Thus, the candidate hairpin sequence datasets have two parts. One is the positive real pre-microRNAs sequences. We obtained a total of 4,028 annotated known pre-microRNA sequences spanned 45 species from miRBase 12 [24]. We removed sequences with homology greater than 90% from the original sequences, and finally 2,241 nonhomologous pre-microRNAs were selected as positive sequences. The other part is the negative pseudo hairpin sequences. The pseudo hairpin sequences were obtained from the UCSC refGene annotation list [25] and the human RefSeq gene [26]. For pseudo hairpin sequences, their sequence fragments have similar hairpin structures of pre-microRNAs and were not reported as pre-microRNAs. Finally, we selected 8,494 pseudo hairpin sequences from the protein coding region. These sequences must be guaranteed to be around 90 ribonucleotides, with a minimum of −15 kcal/mol and a maximum 18 kcal/mol free energy.

In order to select better model, we randomly selected seventy percent of the candidate hairpin sequences as the

training set and the remaining thirty percent as the test set. Thus, we randomly selected 1,500 pre-microRNAs and 6,000 pseudo pre-microRNAs as the training set. As for test set, 700 of the remaining positive real pre-microRNAs and 2,400 of the remaining negative pseudo hairpin sequences were randomly selected. Both training set and test set are normalized.

2.2. Feature Selection. There are many methods to select the pre-microRNAs features. Traditionally, sequence, secondary structure, and thermodynamic properties are considered. In this paper, we use the dinucleotide frequencies proposed in [27] to characterize sequence and secondary structure properties. Thermodynamic characteristics are also included. The LMKL-D model assumed that the hairpin structure of each sequence could be individually characterized as an eigenvector containing 29 global and intrinsic folding properties [27, 28]. Seventeen attributes are the A; C; G; U dinucleotide frequencies; and G + C ratio; three attributes are the folding measures, including the base pairing propensity, base pair distance, and Shannon entropy; three thermodynamic properties such as minimum free energy (MFE) of folding, MFE index 1 MFEI_1, and index 2 MFEI_2; one topological attribute; and five normalized variants of folding measures. The sequence properties and thermodynamic properties can be calculated by ViennaRNA Package 2.0 [29].

2.3. Kernels and Support Vector Machine. The kernels are the inner products of the mapping relationship. A kernel can be described by the dot product of its two basic mapping functions as follows [23]:

$$K(x, y) = \phi(x) \cdot \phi(y), \tag{1}$$

where $K(x, y)$ represents a kernel and $\phi(x)$ and $\phi(y)$ represent the mapping functions.

The mapping functions $\phi(x)$ and $\phi(y)$ are hard to find, but the dot product of the two mapping functions can be easily calculated by the kernel matrix [30]. We can use the characteristics of the kernels to construct a new kernel that can enhance the ability to represent richer features. Thus, the new kernel can map the input data from the low-dimensional linear indivisible feature space to a high-dimensional linearly separable feature space. Synthetic kernels can create different representations of the data using basic kernels.

Kernels are usually associated with SVMs. The basic principle of a single kernel SVM is, for a given dataset $x_i \in R^n (i = 1, \ldots, l)$ with corresponding labels $y_i (y_i = +1 \text{ or } -1)$, SVM finds the linear separable hyperplane with the maximum margin in the feature space induced by the mapping functions $\phi(x)$ and $\phi(y)$. Equation (2) is the SVM decision function [31] given as follows:

$$f(x) = \sum_{i=1}^{l} \alpha_i y_i K_\theta(x_i, x) + b, \tag{2}$$

where α_i are the coefficients to be learned and $K_\theta(x_i, x)$ is a kernel that depends on a set of parameters θ. Traditionally, parameters α_i are trained through maximizing the dual objective function as the following equation [31]:

$$\begin{aligned} &\max_{\alpha} \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j K_\theta\left(x_i, x_j\right) \\ &\alpha_i \geq 0, \quad i = 1, \ldots, l \\ &\text{s.t.} \quad \sum_{i=1}^{l} \alpha_i y_i = 0. \end{aligned} \tag{3}$$

2.4. Multiple Kernel Learning. Multiple kernel learning model is a kind of kernel-based learning model with more flexibility. Recent theories and applications have proved that using multiple kernels instead of single kernel can enhance the interpretability of the decision function and obtain better performance than the single kernel model [31, 32]. Multiple kernels can create different representations of the input data using basic kernels. When we combine multiple kernels within a kernel such as by taking their sum, we can obtain a new kernel that is different from each of them. Moreover, the new kernel has more complicated representations that could not be well represented by a single kernel [33, 34].

In the multiple kernel learning model, K_θ is considered as a linear convex combination of multiple basis kernels [31]:

$$\begin{aligned} &K_\theta = \sum_{i=1}^{m} \theta_i K_{i,} \\ &\text{s.t. } \theta_i \geq 0, \quad i = 1, \ldots, m, \\ &\sum_{i=1}^{m} \theta_i = 1, \end{aligned} \tag{4}$$

where K_i are the basic kernels and m is the total number of basic kernels.

2.5. Deep Multiple Kernel Learning. The traditional multiple kernel learning method is just a simple linear combination of a set of basic kernels and could not represent the deep features of the samples. Thus, we adopt a three-layer multiple kernel learning model to represent the deep features of the samples [23]. The complex combination of basic kernels still meets Mercer standards. A deep multiple kernel model is a n-layer multiple kernel model with m kernels at each layer:

$$K^{(n)} = \left\{\theta_1^{(n)} K_1^{(n)}\left(\theta_1^{(n-1)} K_1^{(n-1)} + \cdots\right) + \ldots \theta_m^{(n)} K_m^{(n)}(\cdots)\right\}, \tag{5}$$

where $K_m^{(n)}$ represents the m^{th} kernel at layer n with an associated weight parameter $\theta_m^{(n)}$ and $K^{(n)}$ represents the synthetic kernel at layer n. A deep multiple kernel learning model with n layers is shown in Figure 2.

Although the increased complexity of the kernels can increase the risk of over-fitting, Strobl et al. [23] proved that the upper bound of the generalization error for deep multiple kernels is significantly less than that for deep feedforward neural networks under some conditions.

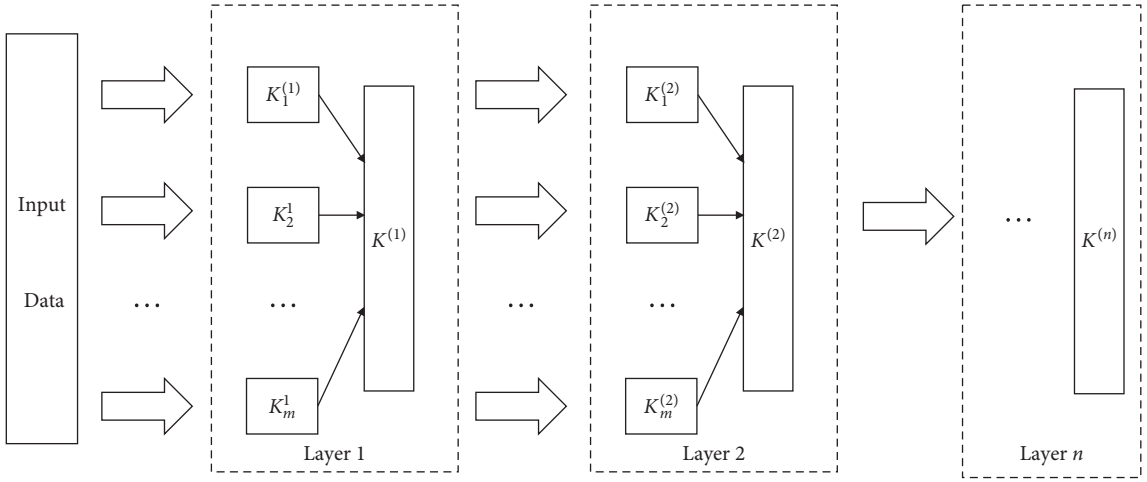

FIGURE 2: A deep multiple kernel model with n layers in this paper. The black solid lines represent the weights for each kernel, $\theta_m^{(n)}$. $K^{(n)}$ comes from the sum of each $K_m^{(n)}$ multiplied by its associated weight $\theta_m^{(n)}$ at layer n.

Leave one out error has shown better accuracy in multiple kernel learning [31]. To decide the weight parameter $\theta_m^{(n)}$, we adopted concept of the span of support vectors [35]. The main idea of SVMs is that mapping input data into a high-dimensional feature space where a hyperplane can separate the input data. The hyperplane can be constructed by maximizing the margin. It is well known that the error rate for SVM is bounded by $\sqrt{O(R^2/M^2)}$, where R is the radius of the smallest ball containing the training data in the feature space and M is the margin. The smaller the error, the better the SVM performance. However, traditional SVM methods only maximize M while R may still be very large. To address the above problem, we minimize R based on upper bounds of leave one out error. R can be shown in span. The span bound of the leave one out error can be shown as the following equation [31]:

$$L\left((x_1, y_1), \ldots, (x_l, y_l)\right) \leq \sum_{p=1}^{l} \phi\left(\alpha_p^0 S_p^2 - 1\right) =: T_{\text{span}}, \quad (6)$$

where L is leave one out error and S_p is the distance between the point $\phi_{K_\theta}(x_p)$ and the set Γ_p. Γ_p is the linear combination of support vectors mapping into feature space, $\Gamma_p = \left\{\sum_{i \neq p, a_i^0 > 0} \lambda_i \phi_{K_\theta}(x_i) \middle| \sum_{i \neq p} \lambda_i = 1\right\}$.

We use a contracting function $\phi(x) = (1 + \exp(-cx + d))^{-1}$ to smooth equation (6) and evaluate its performance. Here, c and d are nonnegative arguments and then a regularization term is added to prevent over-fitting [23]:

$$\overline{S_p^2} = \min_{\lambda, \sum \lambda_i = 1} \left\| \phi_{K_\theta}\left(x_p\right) - \eta \sum_{i \neq p}^{l} \lambda_i \phi_{K_\theta}(x_i) \right\| + \eta \sum_{i \neq p}^{l} \frac{1}{\alpha_i^0} \lambda_i^2. \quad (7)$$

Span is optimized using the gradient descent method. Now, we get the deep multiple kernel learning algorithm with the derivative of $(\partial T_{\text{span}}/\partial\theta)$. By using gradient descent, θ and α can be solved by fixing θ and solving for α and fixing α and solving for θ.

In the deep synthetic kernel proposed in this paper, the number of layers was set as three layers and each layer was set as 3 kernel functions. The kernel functions of the first layer were linear kernel, polynomial kernel, and Gaussian kernel.

2.6. Localized Multiple Kernel Learning. While a single kernel function has only one characteristic, multiple kernel learning (MKL) has more flexibility by choosing a combination of basic kernels. However, multiple kernel learning assigns the same weight to each kernel when combining the basic kernels. The localized multiple kernel learning (LMKL) algorithm uses a gating model to locally select the appropriate weight for each basic kernel. Compared with MKL, LMKL could select suitable weight for the datasets. Experimental results on bioinformatics datasets show that LMKL with the gating model has better accuracy than the model with single kernel [22]. Equation (8) gives a decision function for LMKL [22] as follows:

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i \sum_{m=1}^{P} \eta_m(x) K_m(x, x_i) \eta_m(x_i) + b, \quad (8)$$

where N is the number of samples, P is the number of basic kernels, b is the bias, K_m is the mth basic kernel, and $\eta_m(x)$ is the gating model. For input sample x, the gating model chooses feature space m as a function of input sample x. $\eta_m(x)$ can be learned from the sample datasets. The gating model $\eta_m(x)$ is defined as the following equation [22]:

$$\eta_m(x) = \frac{\exp\left(v_m, x + v_{m0}\right)}{\sum_{k=1}^{P} \exp\left(v_k, x + v_{k0}\right)}, \quad (9)$$

where v_m and v_{m0} are the parameters of the gating model and the softmax guarantees nonnegativity. By modifying equation (8), with selection function, we get the following optimization problem as the following equation [23]:

$$\max \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K_\eta\left(x_i, x_j\right)$$

$$0 \le \alpha_i \le C, \quad i = 1, \ldots, N \tag{10}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} \alpha_i y_i = 0,$$

where $K_\eta(x_i, x_j)$ is defined as equation (11); here, K_η is positive semidefinite [23].

$$K_\eta\left(x_i, x_j\right) = \sum_{m=1}^{P} \eta_m(x_i) K_m\left(x_i, x_j\right) \eta_m\left(x_j\right). \tag{11}$$

Derivatives of equation (10) are taken with respect to v_m and v_{m0}, and then we use gradient descent to train the gating model. We just need to fix $\eta_m(x)$ and then solve a canonical multiple kernel SVM dual problem first and then update the parameters of the gating model with gradient descent at each step.

2.7. Kernels Selection and Model Selection. Traditional multiple kernel learning methods only select a few simple basic kernels, such as linear kernel (K_L), polynomial kernel (K_P), and Gaussian (or RBF) kernel (K_G). In our proposed model, we selected three simple basic kernels, linear kernel, polynomial kernel, and Gaussian kernel and a deep synthetic kernel proposed above (K_D) as the localized multiple kernel learning combination. Finally, we got the LMKL-D model as shown in Figure 3. K_D was obtained by the deep multiple kernel learning (DMKL) model. The formulas for the three simple basic kernel functions are shown in the following equation [20]:

$$\begin{aligned} K_L\left(x_i, x_j\right) &= x_i, x_j, \\ K_P\left(x_i, x_j\right) &= \left(\alpha x_i, x_j + \beta\right)^q, \\ K_G\left(x_i, x_j\right) &= \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{s^2}\right). \end{aligned} \tag{12}$$

We used grid search to find the parameters of the simple basic kernels. The parameters with the highest accuracy were adopted. Finally, the parameter of the Gaussian kernel s was set to 1 and the polynomial kernel exponent q was set to 2, while α and β were both 1.

We used multiple kernel learning models to obtain K_D. Multiple kernel learning models often use linear kernels, Gaussian kernels, and polynomial kernels to map input data into feature spaces. Since the DMKL model should try to maximize the upper bound of the final kernel to increase its richness with each layer, we combined the linear kernel, polynomial kernel, and Gaussian kernel into one set of kernels. From [23], the number of layers for the DMKL was set to 3. K_D was trained on train set and tested on test set. We used leave one out validation and the minimum value of span to evaluate K_D. K_D with a minimum span value was adopted. To find better performance, the penalty parameter C was set in the range of $\{10^{-6}, 10^{-5}, \ldots, 10^{-1}, 1, 10\}$ and the learning rate was set in the range of $\{10^{-6}, 10^{-5}, \ldots, 10^{-1}, 1, 10\}$. After trained and tested, we got K_D. In the end, we chose four kernels, K_L, K_P, K_G, and the best K_D as the final basic kernels of localized multiple kernel learning.

For model selection, the dataset selection operations were repeated three times, and the average value of the results on test set was taken as the final performance of the model. Thus, for each training and test, the training set had 7,500 samples in total and the test set had 3,100 samples in total. For the DMKL model, we used LIBSVM [36] package to solve the SVM optimization problem. For localized multiple kernel learning, we used SMO to speed up the SVM optimization.

3. Results and Discussion

3.1. Comparison with Other Classification Methods. In order to evaluate the performance of the localized multiple kernel learning using the deep synthetic kernel (LMKL-D) model proposed in this paper, the performances of the LMKL-D model were measured by sensitivity (SE, the proportion of the positive examples correctly classified), specificity (SP, the proportion of the negative examples correctly classified), geometric mean (GM, the square root of the product SE and SP), and accuracy (ACC, the percentage of correctly classified instances). SE, SP, GM, and ACC are defined in equation (13) [20]. We compare the LMKL-D model with triplet-SVM [14], miPred [27], MiPred [37], and a three-layer backpropagation neural network (BPNN). The results are shown in Figure 4 and Table 1. Ultimately, the LMKL-D model obtained an accuracy rate of 98.03% on test set, while the triplet-SVM, miPred, MiPred, and BPNN (3 layers) on the test set obtained accuracy rate of 83.90%, 93.50%, 91.29%, and 95.18%, respectively.

$$\begin{aligned} \text{SE} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\ \text{SP} &= \frac{\text{TN}}{\text{TN} + \text{FP}}, \\ \text{GM} &= \sqrt{\text{SE} \times \text{SP}}, \\ \text{ACC} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \end{aligned} \tag{13}$$

As shown in Figure 4 and Table 1, LMKL-D has the best 99.27% SP, best 96.11% GM, and best 98.03% ACC, which means it can better distinguish real pre-microRNAs and pseudo hairpin sequences. Since there are a large number of gene sequences with hairpin structures to be identified, higher specificity can filter the pseudo hairpin sequences. For geometric mean, LMKL-D achieved the highest geometric mean (96.11%) among these methods. That means the LMKL-D model can achieve high performance while maintaining stability. The AUC (area under the curve) 0.9611 (we can see in Figure 5) indicates that the LMKL-D

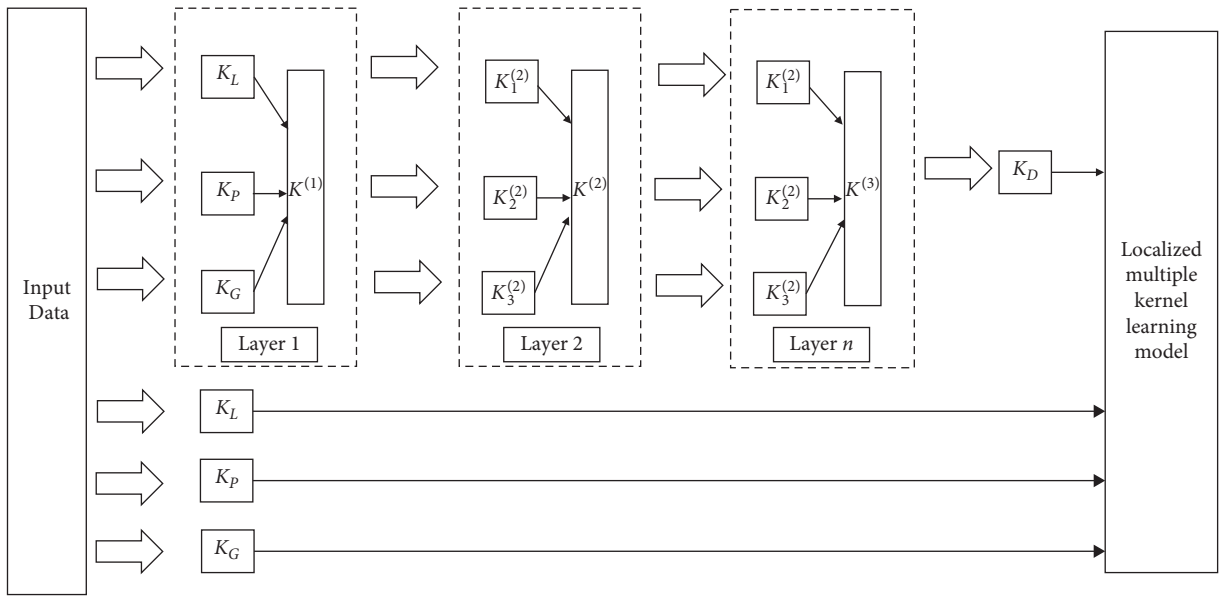

Figure 3: LMKL-D model proposed in this paper.

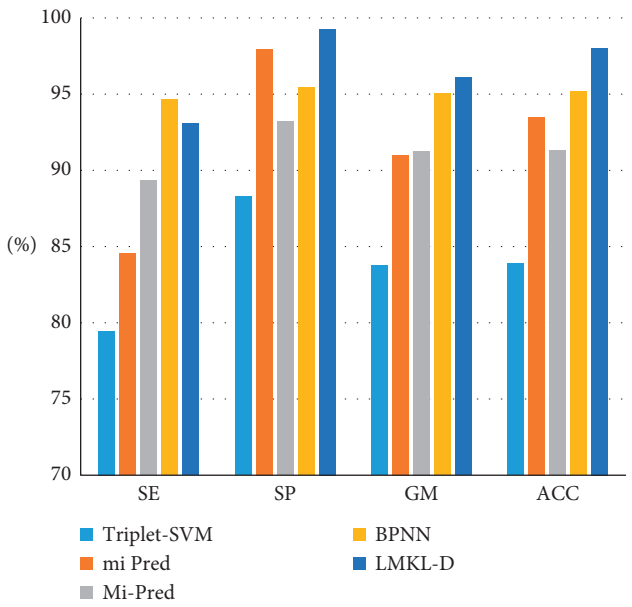

Figure 4: Comparison of prediction performance of LMKL-D with other existing methods. The BPNN was three layers here.

Table 1: Comparison of prediction performance of LMKL-D with other existing methods.

Methods	SE	SP	GM	ACC
Triplet-SVM (libSVM) [7]	79.47	88.30	83.77	83.90
miPred (libSVM) [19]	84.55	97.97	91.01	93.50
MiPred (random forest) [28]	89.35	93.21	91.26	91.29
BPNN (3 layer)	94.64	95.44	95.04	95.18
LMKL-D	93.06	99.27	96.11	98.03

can predict pre-microRNA accurately. Since the ratio of pre-microRNA sequences to pseudo hairpin sequences is about 1 to 4, the SE of the LMKL-D model might be lower. Next, we need find new methods to deal with class imbalances. These data prove that our proposed localized multiple kernel learning using the deep synthetic kernel model can increase classification accuracy with low risk of over-fitting and has a more accurate predictive ability and stability to identify the new microRNA precursors in many species.

3.2. Comparison with Localized Multiple Kernel Learning. In order to better evaluate our LMKL-D model, we also compared LMKL-D with LMKL. For basic kernels, the LMKL-D model used four basic kernels, K_L, K_P, K_G, and K_D. The LMKL model used three basic kernels, K_L, K_P, and K_G. K_L, K_P, and K_G of the two models adopted the same parameters. The penalty parameter C was fixed on 0.035. The results on test set are shown in Figure 6. The performance of the two models on training and test set is shown in Table 2.

From Figure 6 and Table 2, we can see that the LMKL-D model has 91.33% SE, 99.22% SP, 97.64% ACC, and 0.9535 AUC on training set, while the LMKL model obtained 87.47% SE, 99.60% SP, 97.17% ACC, and 0.9352 AUC. On test set, the LMKL-D model has 93.06% SE, 99.27% SP, 98.03% ACC, and 0.9611 AUC, while the LMKL model obtained 88.71% SE, 99.60% SP, 97.42% ACC, and 0.9407 AUC. On both the training set and the test set, the LMKL-D model has 91.83% SE, 99.23% SP, 97.75%, and 0.9574 AUC, while the LMKL model obtained 87.83% SE, 99.60% SP, 97.24% ACC, and 0.9383 AUC. Although LMKL-D acquires a little lower specificity than LMKL, on sensitivity, accuracy, and AUC, LMKL-D is always better than LMKL. On geometric mean, LMKL-D is always higher than LMKL. That means that LMKL-D is more stable than LMLK. We can draw a conclusion from Figure 6 and Table 2 that LMKL-D has better sensitivity, geometric mean, and accuracy than LMKL. For specificity, the two models have similar performance. The results show that in terms of correctly and stably identifying pre-microRNA, the LMKL-D is more

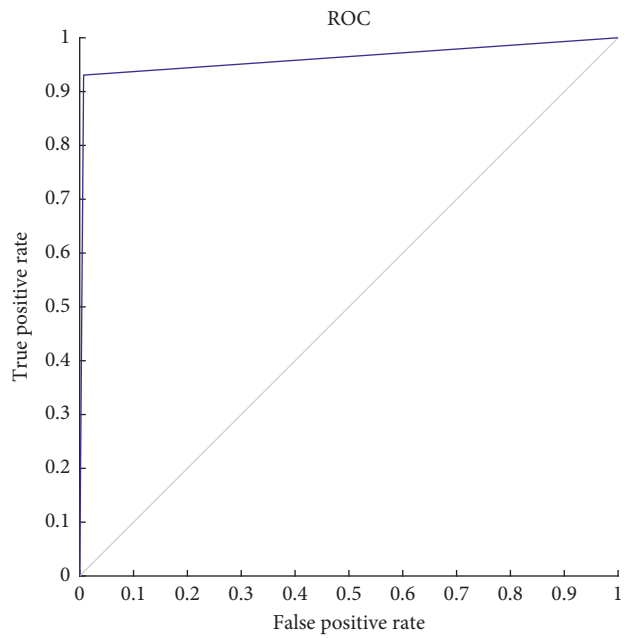

Figure 5: ROC curve of LMKL-D; AUC = 0.9611.

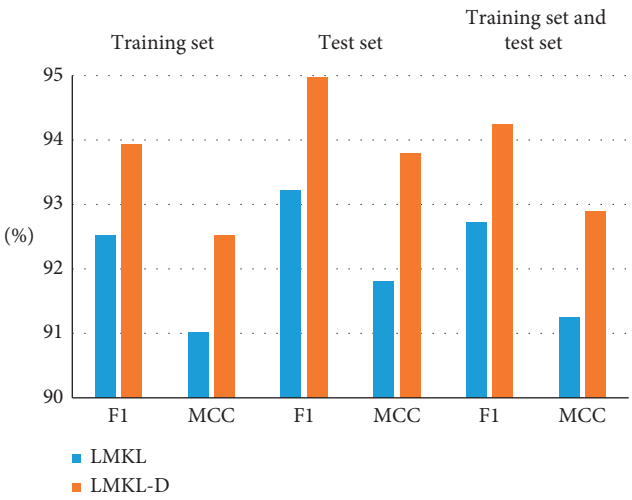

Figure 6: Comparison of LMKL-D and LMKL on F1 and MCC.

Table 2: Comparison of prediction performance of LMKL-D with LMKL.

Dataset	Methods	SE (%)	SP (%)	GM (%)	ACC (%)	AUC
Training set	LMKL	87.47	99.60	93.34	97.17	0.9352
	LMKL-D	91.33	99.22	95.19	97.64	0.9535
Test set	LMKL	88.71	99.60	94.00	97.42	0.9407
	LMKL-D	93.06	99.27	96.11	98.03	0.9611
Training set And test set	LMKL	87.83	99.60	93.53	97.24	0.9383
	LMKL-D	91.83	99.23	95.46	97.75	0.9574

efficient than LMKL. The experiments also show that the deep model can obtain more features than the LMKL model. The LMKL-D model obtained both deep and shallow features of the samples.

4. Conclusions

In this work, we have proposed a localized multiple kernel learning model with a three-layer deep synthetic kernel in improving the pre-microRNAs prediction accuracy of existing methods. The experiments show that our proposed model yielded comparable better predictive performances and is more stable than existing classifiers for identifying known pre-microRNAs. After being trained on hairpin sequences train set, the LMKL-D methods obtain 93.06% sensitivity, 99.27% specificity, 96.11% geometric mean, and 98.03% accuracy on test set. By applying deep architecture to localized multiple kernel learning, we found that the LMKL-D model is both useful and reliable, as demonstrated in the results above. The LMKL-D model was examined by comparing with the triplet-SVM, miPred, MiPred, and a three-layer backpropagation neural network. We also compare the LMKL-D model and LMKL model. Our results show a more efficient model compared with the multiple kernel learning model with simple basic kernels. The three-layer deep synthetic kernel can indeed increase the richness of kernels and represent deep features. On the other hand, the LMKL-D model could use both shallow and deep features. The number of pseudo hairpin sequences in nature is much larger than known pre-microRNAs. There are always more negative samples than positive samples. With the development of bioinformatics, it is still a challenging work to solve the problem of sample imbalance and explore more classification methods.

Data Availability

The known pre-microRNA sequences data can be downloaded from the miRBase website at http://www.mirbase.org; the UCSC refGene annotation list and the human RefSeq gene were available through https://www.ncbi.nlm.nih.gov/refseq/.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Funds (grant nos. 62072391, 61872419, and 61573166), Joint Funds of ShanDong Natural Science Funds (grant no. ZR2019LZH003), Taishan Scholars Program of Shandong Province, China (grant no. tsqn201812077), Shandong Provincial Natural Science Foundation (grant no. ZR2018LF005), University Innovation Team Project of Jinan (grant no. 2019GXRCO15), and Key Science and Technology

Innovation Project of Shandong Province (grant no. 2019JZZY010324).

References

[1] A. E. Erson-Bensan, "Introduction to microRNAs in biological systems," *Methods in Molecular Biology*, vol. 1107, pp. 1–14, 2014.

[2] D. Zhengren, Y. Xiuhong, W. Yuqiang, and W. yanjin, "Research progress of miRNA related to hyperlipidemia induced coronary arterial endothelial injury," *Chinese Journal of Arteriosclerosis*, vol. 28, no. 8, pp. 721–727, 2020.

[3] J. X. Wang Yuhan, "Research progress in biological function of microRNA-129," *Military Medical Sciences*, vol. 42, no. 5, pp. 384–387, 2018.

[4] L. Wei, M. Liao, Y. Gao, R. Ji, Z. He, and Q. Zou, "Improved and promising identification of human microRNAs by incorporating a high-quality negative set," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 11, no. 1, pp. 192–201, 2013.

[5] L. Jiang, J. Zhang, P. Xuan, and Q. Zou, "BP neural network could help improve pre-miRNA identification in various species," *BioMed Research International*, vol. 2016, Article ID 9565689, 2016.

[6] X. Zeng, L. Liu, L. Lü, and Q. Zoü, "Prediction of potential disease-associated microRNAs using structural perturbation method," *Bioinformatics*, vol. 34, no. 14, pp. 2425–2432, 2018.

[7] W. Bao, D.-S. Huang, and Y.-H. Chen, "Msit: malonylation sites identification tree," *Current Bioinformatics*, vol. 15, no. 1, pp. 59–67, 2020.

[8] W. Bao, B. Yang, D.-S. Huang et al., "IMKPse: identification of protein malonylation sites by the key features into general PseAAC," *IEEE Access*, vol. 7, pp. 54073–54083, 2019.

[9] R. C. Lee, R. L. Feinbaum, and V. Ambros, "The c. elegans heterochronic gene lin-4 encodes small RNAs with antisense complementarity to lin-14," *Cell*, vol. 75, no. 5, pp. 843–854, 1993.

[10] F. Huang, X. Yue, Z. Xiong, Z. Yu, S. Liu, and W. Zhang, "Tensor decomposition with relational constraints for predicting multiple types of microRNA-disease associations," *Briefings in Bioinformatics*, 2019.

[11] W. Zhang, Z. Li, W. Guo, W. Yang, and F. Huang, "A fast linear neighborhood similarity-based network link inference method to predict microRNA-disease associations," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 2, 2019.

[12] Y. Gong, Y. Niu, W. Zhang, and X. Li, "A network embedding-based multiple information integration method for the miRNA-disease association prediction," *BMC Bioinformatics*, vol. 20, no. 1, pp. 1–13, 2019.

[13] Y. Deng, X. Xu, Y. Qiu, J. Xia, W. Zhang, and S. Liu, "A multimodal deep learning framework for predicting drug-drug interaction events," *Bioinformatics*, vol. 36, no. 15, pp. 4316–4322, 2020.

[14] C. Xue, F. Li, T. He, G.-P. Liu, Y. Li, and X. Zhang, "Classification of real and pseudo microRNA precursors using local structure-sequence features and support vector machine," *BMC Bioinformatics*, vol. 6, no. 1, p. 310, 2005.

[15] S. G. Das, H. J. Chakraborty, and A. Datta, "PremipreD: precursor miRNA prediction by support vector machine approach," *Trends in Bioinformatics*, vol. 11, no. 1, pp. 17–24, 2018.

[16] S.-Y. Han, J. Zhou, Y.-H. Chen, Y.-F. Zhang, G.-Y. Tang, and L. Wang, "Active fault-tolerant control for discrete vehicle active suspension via reduced-order observer," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020.

[17] W. Bao, B. Yang, D. Li, Z. Li, Y. Zhou, and R. Bao, "Cmsenn: computational modification sites with ensemble neural network," *Chemometrics and Intelligent Laboratory Systems*, vol. 185, pp. 65–72, 2019.

[18] D. Zheng, J. Wang, and Y. Zhao, "Non-flat function estimation with a multi-scale support vector regression," *Neurocomputing*, vol. 70, no. 1–3, pp. 420–429, 2006.

[19] R. J. Peace, M. S. Hassani, and J. R. Green, "MipIe: NGs-based prediction of miRNA using integrated evidence," *Scientific Reports*, vol. 9, no. 1, p. 1548, 2019.

[20] H. Shi, W. Wang, P. Wu, and D. Wang, "Support vector machine based on localized multiple kernel learning in pre-microRNA classification," in *Proceedings of the 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, pp. 1–5, Istanbul, Turkey, June 2020.

[21] M. E. Rahman, R. Islam, S. Islam, S. I. Mondal, and M. R. Amin, "MiRANN: a reliable approach for improved classification of precursor microRNA using artificial neural network model," *Genomics*, vol. 99, no. 4, pp. 189–194, 2012.

[22] M. Gönen and E. Alpaydin, "Localized multiple kernel learning," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 352–359, Helsinki, Finland, July 2008.

[23] E. V. Strobl and S. Visweswaran, "Deep multiple kernel learning," in *Proceedings of the 2013 12th International Conference on Machine Learning and Applications*, vol. 1, pp. 414–417, Miami, FL, USA, December 2013.

[24] A. Kozomara and S. Griffithsjones, "MiRBase: annotating high confidence microRNAs using deep sequencing data," *Nucleic Acids Research*, vol. 42, no. 1, pp. 68–73, 2014.

[25] D. Karolchik, R. M. Kuhn, R. Baertsch et al., "The UCSC genome browser database: 2008 update," *Nucleic Acids Research*, vol. 36, pp. 773–779, 2007.

[26] D. Maglott, J. Ostell, K. D. Pruitt, and T. Tatusova, "Entrez gene: gene-centered information at NCBI," *Nucleic Acids Research*, vol. 33, pp. 26–31, 2004.

[27] K. L. S. Ng and S. K. Mishra, "De novo SVM classification of precursor microRNAs from genomic pseudo hairpins using global and intrinsic folding measures," *Bioinformatics*, vol. 23, no. 11, pp. 1321–1330, 2007.

[28] W. Bao, C.-A. Yuan, Y. Zhang et al., "Mutli-features prediction of protein translational modification sites," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 5, pp. 1453–1460, 2017.

[29] R. Lorenz, S. H. Bernhart, C. H. Z. Siederdissen et al., "ViennaRNA package 2.0," *Algorithms for Molecular Biology*, vol. 6, no. 1, p. 26, 2011.

[30] M. Gönen and E. Alpaydın, "Multiple kernel learning algorithms," *The Journal of Machine Learning Research*, vol. 12, pp. 2211–2268, 2011.

[31] H. Wang and S. Fuchun, "On multiple kernel learning methods," *Acta Automatica Sinica*, vol. 36, no. 8, pp. 1038–1050, 2010.

[32] Y. Liu, S. Liao, and Y. Hou, "Learning kernels with upper bounds of leave-one-out error," in *Proceedings of the 20th ACM International Conference on Information and knowledge management*, pp. 2205–2208, Glasgow, Scotland, October 2011.

[33] W.-J. Lee, S. Verzakov, and R. P. Duin, "Kernel combination versus classifier combination," in *Proceedings of the 7th*

International Workshop on Multiple Classifier Systems, pp. 22–31, Prague, Czech Republic, May 2007.

[34] W. Bao, D. Wang, and Y. Chen, "Classification of protein structure classes on flexible neutral tree," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 14, no. 5, pp. 1122–1133, 2016.

[35] O. Chapelle and V. Vapnik, "Model selection for support vector machines," *Advances in Neural Information Processing Systems*, vol. 12, pp. 230–236, 1999.

[36] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, 2011.

[37] P. Jiang, H. Wu, W. Wang, W. Ma, X. Sun, and Z. Lu, "MiPred: classification of real and pseudo microRNA precursors using random forest prediction model with combined features," *Nucleic Acids Research*, vol. 35, pp. 339–344, 2007.