Mathematical Biosciences and Engineering

MBE, 20(7): 13149–13170.
DOI: 10.3934/mbe.2023586
Received: 22 March 2023
Revised: 10 May 2023
Accepted: 17 May 2023
Published: 06 June 2023

http://www.aimspress.com/journal/MBE

Research article

Identification of DNA-binding protein based multiple kernel model

Yuqing Qian[1,†], Tingting Shang[1,†], Fei Guo[2], Chunliang Wang[3], Zhiming Cui[1], Yijie Ding[4,*] and Hongjie Wu[1,*]

[1] College of Electronic and Information Engineering, Suzhou University of Science and Technology, Suzhou, China
[2] School of Computer Science and Engineering, Central South University, Changsha, China
[3] The Second Affiliated Hospital of Soochow University, Suzhou, China
[4] Yangtze Delta Region Institute (Quzhou), University of Electronic Science and Technology of China, Quzhou, China

† These two authors contributed equally.

* **Correspondence:** Email: wuxi_dyj@163.com, Hongjie.wu@qq.com.

Abstract: DNA-binding proteins (DBPs) play a critical role in the development of drugs for treating genetic diseases and in DNA biology research. It is essential for predicting DNA-binding proteins more accurately and efficiently. In this paper, a Laplacian Local Kernel Alignment-based Restricted Kernel Machine (LapLKA-RKM) is proposed to predict DBPs. In detail, we first extract features from the protein sequence using six methods. Second, the Radial Basis Function (RBF) kernel function is utilized to construct pre-defined kernel metrics. Then, these metrics are combined linearly by weights calculated by LapLKA. Finally, the fused kernel is input to RKM for training and prediction. Independent tests and leave-one-out cross-validation were used to validate the performance of our method on a small dataset and two large datasets. Importantly, we built an online platform to represent our model, which is now freely accessible via http://8.130.69.121:8082/.

Keywords: DNA-binding proteins; multiple kernel learning; local kernel alignment; restricted kernel machine

1. Introduction

Many biological processes are carried out by the DBPs, such as specific nucleotide sequence recognition, transcription and DNA replication. Therefore, identification of DBPs has become an import subject of biology. The protein can be identified by various experimental techniques, such as

ChIP-chip [1,2] and filter binding assays [3]. However, with the development of high-throughput sequencing technology, protein sequence databases have increased unprecedentedly. Proteins whose structure and function are unknown are on the rise. A rapid and accurate method for identifying and characterizing DBPs based on their protein sequence is highly desired. Computer prediction methods have been widely applied to various biological problems [4–11].

The existing prediction methods are broadly divided into two groups. The first group is model based prediction methods. These methods that borrow prior information across sequences to predict DBPs, including amino acid composition [12,13], evolutionary information [14,15] and physicochemical [16] character. For example, Rahman et al. [17] presented a predictor named DPP-PseACC. They used Chou's PseAAC [18] to extract features from amino acid composition and Random Forest (RF) model to reduce the dimension of feature vector. Then, they applied Support Vector Machine [19] (SVM) with linear kernel to train prediction model. Similarly, StackPDB take three steps to predict DBPs, including feature extraction, feature selection and model construction. StackPDB extract protein sequence features from amino acid and composition and evolutionary information. Evolutionary information can be represented by the position specific scoring matrix (PSSM), which is generated by PSI-BLAST [20] program. In StackPDB method, PsePSSM, PSSM-TPC, EDT and RPT are used to extract PSSM. They then used extreme gradient boosting-recursive feature elimination to select the best features. Finally, the excellent feature subset is fed into the stacked ensemble classifier, which composes XGBoost, SVM and LightGBM. From the previous study [21–25], we can see that the protein sequence can be described by different representations, such as amino acid composition and PSSM. Because fusion methods can exploit information from all representations to effectively improve the model performance, some fusion techniques are performed in identification of DBPs.

For example, CKA-MKL [26], HSIC-MKL [27], HKAM-MKL [28] and MLapSVM-LBS [29]. CKA-MKL, HSIC-MKL and HKAM-MKL are the Multiple Kernel Learning (MKL), which is popular early fusion techniques. MKL aims to learn optimal kernel weights. The optimal kernel is linear combined by multiple base kernels based on the related weight. CKA-MKL maximizes the cosine similarity score between the optimal and the ideal kernel. In addition, CKA-MKL introduce the Laplacian term about weights into objective function to avoid extreme situations. CKA-MKL only considers the global kernel alignment and ignore the difference information between local samples. Therefore, HKAKM-MKL both maximize the score of local and global kernel alignment. CKA-MKL and HKAM-MKL both use SVM as a classifier. HSIC-MKL maximizes the value of independence between trained samples and labels in Reproducing Kernel Hilbert Space (RKHS). Then, the optimal kernel was input into a hypergraph based Laplacian SVM, which is the extension of SVM. CKA-MKL only considers the global manner. Furthermore, HKAM-MKL both consider global and local manners. HKAM-MKL is therefore superior to CKA-MKL in predicting DBPs. Different from the above MKL methods, MLapSVM-LBS fuses multiple information during training progress. MLapSVM-LBS uses the multiple local behavior similarity graph as the regularization term. Because the objective function of MLapSVM-LBS is non-convex, an alternation algorithm is employed. The advantage of MLapSVM-LBS is that the multiple information is fused during the training phase while allowing for some degree of freedom to model the views differently.

There are several methods for predicting protein sequences that are based on structural information. Using structural alignment and statistical potential, Gao et al. [4] proposed the DBD-Hunter. DBD-Threader was subsequently proposed by Gao et al. [30] for the prediction of DBPs. The DBD-Threader uses a template library consisting of DNA-protein complex structures, while its classification relies only on the sequence of the target protein. When the structure of a candidate protein is known, structure-based predictors can be used. Therefore, predictors that rely solely on structural information about proteins are limited in their application.

The second group is deep learning-based prediction methods. Deep learning-based methods are designed by capture the hidden representation of protein sequence. For example, Du et al. [31] reported a deep learning-based method called MsDBP. MsDBP only relies on the primary sequence, without human-crafted feature selection. Lu et al. [32] proposed a predictor that contains parallel long and short-term memory (LSTM) and convolutional neural networks (CNN). In Lu's work, the input of LSTM and CNN is sequence and PSSM, respectively. The spatial structure of a protein contains richer information compared with protein sequences. Therefore, Lu et al. [33] further constructed a graph convolutional network based on the contact map, which is generated by Pconsc4 [34]. Yan et al. [35] employed the transfer learning to construct data sets and build a deep learning neural network with attention mechanisms to detect DBPs. Because of their nature, most deep learning-based methods [33,36,37] are not suitable for small datasets.

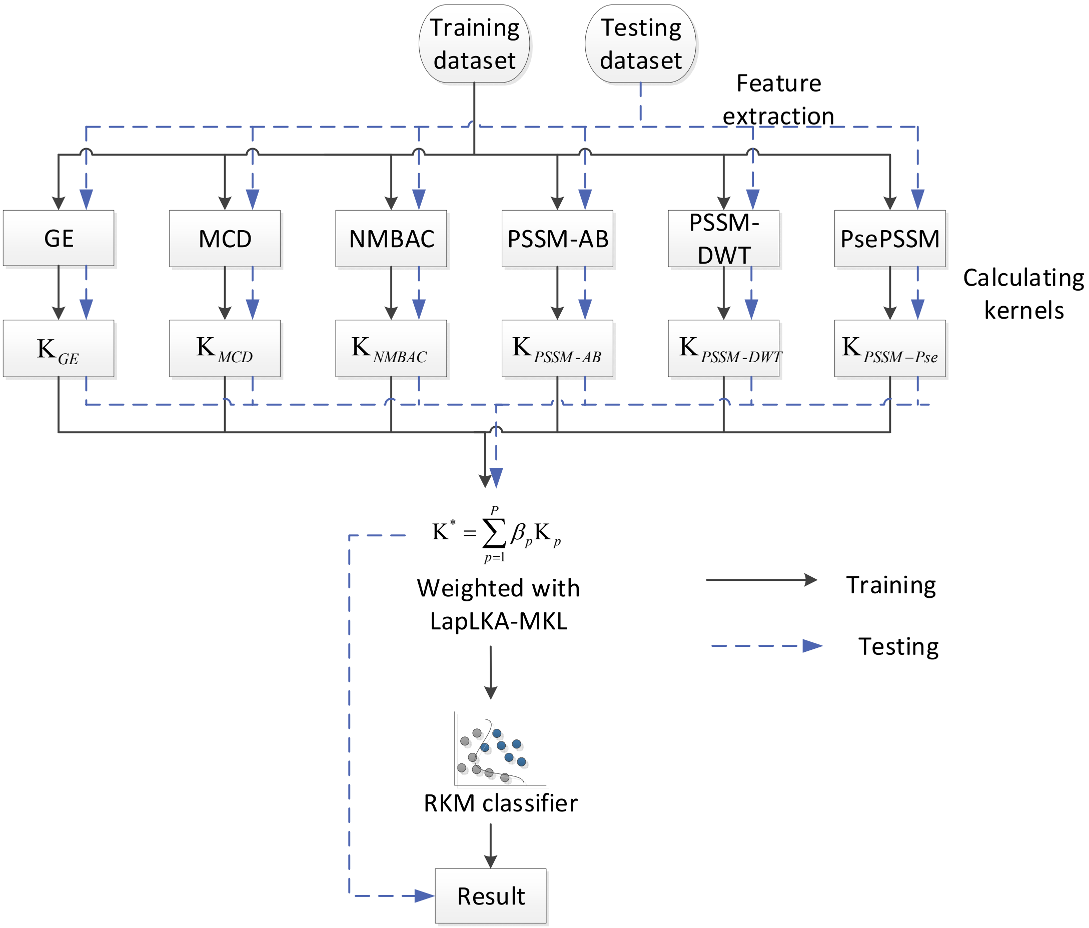


Figure 1. A workflow of the LapLKA-RKM.

Inspired by a series of recent publications [26,27,38–46], we propose a predictor for detecting DBPs. This predictor was called LapLKA-RKM, which needs the following three steps: 1) represent the protein sequence with a set of feature vectors, including Global Encoding (GE), Multi-scale Continuous and Discontinuous descriptor (MCD), Normalized Moreau-Broto Auto Correlation (NMBAC), PSSM-based Discrete Wavelet Transform (PSSM-DWT), PSSM-based Average Blocks (PSSM-AB) and PSSM-Pse; 2) fuse these features by LapLKA (this progress can be seen as selection of features); 3) RKM was developed to make the prediction. A brief architecture of LapLKA-RKM is shown in Figure 1. We conducted LOOCV and independent testing on PDB1075 and PDB2272, respectively. The prediction accuracy indicate that our methods is an effectively tool for DBPs detection.

The contributions of our methods include: 1) we propose a MKL algorithm, called LapLKA, which can outperform other MKL methods in handling multiple kernels; 2) we extend the RKM to a multiple kernel setting by weighting shared hidden features.

2. Methods

2.1. Datasets and experiment setup

Three protein datasets with different sizes were adopted in our study to test the ability of LapLKA-RKM in predicting DBPs. These datasets collected from the PDB, UniProt and Swiss-Prot database, namely PDB1075 [12], PDB14189 [31] and PDB2272 [31].

The dataset construction rules are as follows:

$$N = N^{+} \cup N^{-} \tag{1}$$

where N is the number of total samples, N^{+} is the number of DBPs samples and N^{-} is the number of non-DBPs samples. We present a brief summary of the three datasets in Table 1. Sequences with sequence similarity greater than 25%, 25%, 40% in PDB1075, PDB2272 and PDB14189 were removed, respectively.

Leave-one-out cross-validation (LOOCV) and independent testing are conducted to show the ability of predictor. We conduct the LOOCV and 10-CV on PDB1075, because PDB1075 is a small dataset and its running time is acceptable. To show the robustness of generalization and ability of big dataset of models, we take PDB14189 dataset as training set and PDB2272 as test set.

Table 1. A summary of three datasets used in this study.

Datasets	N^{+}	N^{-}	N
PDB1075	525	550	1075
PDB14189	7129	7060	14,189
PDB2272	1153	1119	2272

2.2. Feature extraction

A total of six sequence-based features are extracted from proteins, including GE [47], MCD [48], NMBAC [49], PSSM-DWT [50], PSSM-AB [51] and PSSM-Pse [13,52–54]. Where GE and MCD extract feature vectors from the amino acid composition of sequences. NMBAC describes the six physicochemical properties of amino acids, namely Polarizability, Polarity, Solvent Accessible Surface Area, Hydrophobicity, Net Charge Index of Side Chains and Volume of Side Chains. PSSM-AB, PSSM-DWT and PSSM-Pse consider proteins' evolutionary information, which can be represented by the position specificity score matrix (PSSM). PSSM is generated by PSI-BLAST [20]. The optimal parameters of NMBAC and PSSM-Pse were implemented by previous study [26]. In the related literature, these features are described in detail.

RKM is a kind of kernel methods [55–58]. It maps data points from the input space to the feature space. The mapping is determined implicitly by a kernel function. Therefore, we need to construct kernel metrics as input to RKM. The kernel function mainly includes Linear Function, Polynomial Function and Radial Basis Function. Like other methods [27,38,59–61], RBF is employed to construct kernels and its formula is defined as:

$$K_{ij} = K(x_i, x_j) = exp\left(-\gamma \left\|x_i - x_j\right\|^2\right), i, j = 1,2, \cdots, N \tag{2}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are sample points, γ is the kernel bandwidth. Then, a predefined kernel set $\mathbf{K}$ is obtained:

$$\mathbf{K} = \{\mathbf{K}_{GE}, \mathbf{K}_{MCD}, \mathbf{K}_{NMBAC}, \mathbf{K}_{PSSM-AB}, \mathbf{K}_{PSSM-DWT}, \mathbf{K}_{PSSM-Pse}\} \tag{3}$$

2.3. Laplacian Local Kernel Alignment Algorithm

Laplacian Local Kernel Alignment (LapLKA) is a kind of supervised Multiple Kernel Learning (MKL). As we all know, an appropriate kernel matrix is very important to the success of any kernel method [62]. However, choosing an appropriate kernel matrix is difficult for biological applications. In terms of protein sequence, it can be described by different kernel matrixes. To address this limitation, MKL is proposed [39]. MKL aims to combine a set of predefined kernels by linear weight and the optimal kernel accurately represent a set of protein sequences. Let P as the number of predefined kernels, $\mathbf{K} = \{\mathbf{K}_1, \ldots, \mathbf{K}_P\}$ as the kernel set. The optimal kernel $\mathbf{K}^*$ is the linear combination of the kernel set:

$$\mathbf{K}^* = \sum_{p=1}^{P} \beta_p \mathbf{K}_p \tag{4}$$

where β_p is the kernel mixture weight. Usually, the L_1-norm is imposed to constraint the structure of $\boldsymbol{\beta}$:

$$\|\boldsymbol{\beta}\|_1 = \sum_{p=1}^{P} |\beta_p| = 1 \tag{5}$$

The main goal of the LapLKA algorithm is to determine the values of $\boldsymbol{\beta}$. There are two parts to LapLKA's learning strategy: local kernel and the inner relationship of global kernels. In previous studies [63–65], the score of kernel alignment is calculated only in global or local manner. Global manner aims to maximize the alignment score between the whole optimal kernel and the ideal kernel. Global manner may ignore the difference between similar samples. Contrary the global manner, local manner only considers the sub kernel, which is constructed by a set of similar samples. In the global manner, whole samples will be missed. For this reason, we propose LapLKA, which integrated local kernel alignments and the global kernel alignments.

First, we define the function of kernel alignment as follow:

$$A(\mathbf{P}, \mathbf{Q}) = \frac{\langle \mathbf{P}, \mathbf{Q} \rangle_F}{\|\mathbf{P}\|_F \|\mathbf{Q}\|_F} \tag{6}$$

where $\mathbf{P}$ and $\mathbf{Q}$ are positive defined matrix, $\langle \cdot, \cdot \rangle_F$ and $\|\cdot\|_F$ are the Frobenius inner product and Frobenius norm, respectively. The value of kernel alignment is the cosine similarity between two kernels.

For the local manner, we maximize the alignment score between the local kernel and the ideal kernel. The local kernel is constructed by each sample and its neighbors. We select the index of the k samples neighbor samples that are nearest to each sample. We choose the Euclidean distance in the input space as the evaluation of sample similarity. Then, we select the sample's neighbor samples based the similarity. The set of neighbors of samples of x_i is $N_k(x_i)$. The local kernel about x_c can be represented as:

$$\mathbf{K}_c^i = [\mathbf{K}_c(u,v)]_{k\times k}, x_u, x_v \in N_k(x_i) \tag{7}$$

We maximize the average of all local kernel alignment scores. There, the objective function of local manner can be presented as follows:

$$\arg\max_{\beta} \frac{1}{N}\sum_{i=1}^{N} A\left(\sum_{p=1}^{P} \beta_p \mathbf{K}_p^{(i)}, \mathbf{K}_Y^{(i)}\right) \tag{8}$$

where $K_Y = YY^T$ is ideal kernel, $K_Y^{(i)}$ is calculated by the label of related samples.

The global kernel alignment information is introduced into Eq (8) by the Laplacian regular term:

$$\begin{aligned}\sum_{i,j}^{P}(\beta_i-\beta_j)^2\mathbf{W}_{ij} &= \sum_{i,j}^{P}(\beta_i^2+\beta_j^2-2\beta_i\beta_j)\mathbf{W}_{ij}\\ &= \sum_{i,j}^{P}\beta_i^2\mathbf{D}_{ii}+\sum_{i,j}^{P}\beta_j^2\mathbf{D}_{jj}-2\sum_{i,j}^{P}\beta_i\beta_j\mathbf{W}_{ij}\\ &= 2\boldsymbol{\beta}^T\mathbf{L}\boldsymbol{\beta}\end{aligned} \tag{9}$$

where $\mathbf{W}\in\mathbf{R}^{P\times P}$ is the global kernel alignment matrix, $\mathbf{W}_{ij}$ represents the value of kernel alignment $A\left(\mathbf{K}^i,\mathbf{K}^j\right)$. Equations (8) and (9) are integrated as follow:

$$\begin{aligned}&\arg\max_{\beta} \frac{1}{N}\sum_{i=1}^{N} A\left(\sum_{p=1}^{P} \beta_p \mathbf{K}_p^{(i)}, \mathbf{K}_Y^{(i)}\right) - 2\lambda\boldsymbol{\beta}^T\mathbf{L}\boldsymbol{\beta}\\ &s.t.\sum\nolimits_{i=1}^{P}\beta_p = 1\end{aligned} \tag{10}$$

To optimize Eq (10), we introduce the auxiliary variable τ_i into Eq (10). τ_i is defined as:

$$\tau_i = \frac{\sqrt{\boldsymbol{\beta}^T\mathbf{M}^{(i)}\boldsymbol{\beta}}}{\sqrt{\boldsymbol{\beta}^T\mathbf{M}\boldsymbol{\beta}}} \tag{11}$$

where $M_{ij} = tr\left(K_i^T K_j\right)$, $M_{ij}^{(l)} = tr\left(K_i^{(l)^T} K_j^{(l)}\right)$. Therefore, Equation (10) can be rewrite as:

$$\begin{aligned}&\arg\max_{\beta}\frac{\boldsymbol{\beta}^T\mathbf{Q}}{\sqrt{\boldsymbol{\beta}^T\mathbf{M}\boldsymbol{\beta}}} - 2\lambda\boldsymbol{\beta}^T\mathbf{L}\boldsymbol{\beta}\\ &s.t.\sum\nolimits_{i=1}^{P}\beta_p = 1\end{aligned} \tag{12}$$

From [66–72], Equation (12) is equivalent to the following Quadratic Programming problem:

$$\begin{aligned}&\arg\max_{\beta}\boldsymbol{\beta}^T\mathbf{M}\boldsymbol{\beta} - \boldsymbol{\beta}^T(2\mathbf{Q}+4\lambda\mathbf{L})\\ &s.t.\sum\nolimits_{i=1}^{P}\beta_p = 1\end{aligned} \tag{13}$$

We employ CVX package [73] to optimization Eq (13).

2.4. Restricted kernel machine

Restricted Kernel Machine (RKM) classification model is a kind of kernel method [8]. It was proposed by Suykens [56]. The objective function of RKM is closely similar to the Least Squares Support Vector Machine (LS-SVM) [74] model. SVM is also a kernel method and most methods [15,23,26,28,75] select SVM as classification. However, we choose RKM as classification. The reason is that, RKM is easily extend to deep framework, called Deep RKM [56]. Deep RKM can produce good results and we will use it throughout the rest paper.

$\left\{\left(x_i, y_i\right)\right\}_{i=1}^{N}$ denotes as the training data, where $x_i \in R^d$ is the i -th input pattern and $y_i \in \{-1,1\}$ the related sample label. It is well known that the objective function of LS-SVM is:

$$\arg\min_{w,b} \frac{\eta}{2} w^T w + \sum_{i=1}^{N} e_i^2 \\ s.t. e_i = 1 - (\varphi(x_i)^T w + b) y_i \tag{14}$$

We formulate a lower bound on the function Eq (14), and then the objective function of RKM classification is obtained:

$$\arg\min_{w,b} \frac{\eta}{2} w^T w + \sum_{i=1}^{N} (1 - (\varphi(x_i)^T w + b) y_i) h_i - \frac{\mu}{2} \sum_{i=1}^{N} h_i^2 \tag{15}$$

where b is a bias term, η and μ are hyperparameters and h_i is a hidden feature. The map function $\varphi(\cdot)$ maps x from the input space into a reproducing kernel Hilbert space. Hidden features are obtained by an internal pairing of $e^T h$, where e is the classification loss.

The stationary points of the objective function Eq (15) in the primal formulation are characterized by:

$$\begin{cases} 1 = \varphi(x_i)^T w + b + \lambda h_i, i = 1, \cdots, N \\ w = \frac{1}{\eta} \sum_{i=1}^{N} \varphi(x_i) y_i h_i , i = 1, \cdots, N \\ \sum_{i=1}^{N} y_i h_i = 0 \end{cases} \tag{16}$$

By eliminating the weights w , the linear formulation is obtained:

$$\begin{bmatrix} \frac{1}{\eta} \mathbf{K} + \mu \mathbf{I}_N & 1_N \\ 1_N^T & 0 \end{bmatrix} \begin{bmatrix} y \odot h \\ b \end{bmatrix} = \begin{bmatrix} y \\ 0 \end{bmatrix} \tag{17}$$

where $\mathbf{I}_N$ and 1_N are the identity matrix and a one column vector, $\odot$ is the element-wise product.

In this paper, we mainly focus on the RKM-based MKL formulations. The final linear system of RKM-based MKL is given by:

$$\begin{bmatrix} \frac{1}{\eta}\sum_{i=1}^{P} \beta_p \mathbf{K}_p + \mu \mathbf{I}_N & 1_N \\ 1_N^T & 0 \end{bmatrix} \begin{bmatrix} y \odot h \\ b \end{bmatrix} = \begin{bmatrix} y \\ 0 \end{bmatrix} \tag{18}$$

The linear system Eq (18) can be solved based on the training data. The variables h and bias term b are used to construct the classifier. For a test data point x_t, the final decision function is:

$$f(x_t) = \mathbf{sign}\left(\frac{1}{\eta}\sum_{p=1}^{P} \beta_p \sum_{i=1}^{N} y_i h_i K_p(x_i, x_t) + b\right) \tag{19}$$

3. Results and discussion

3.1. Evaluation measurements

Because the identification of DBPs is the binary classification problem. The following parameters are employed to measure the performance of predictor:

$$ACC = \frac{TP + FN}{TN + TP + FP + FN} \times 100\% \tag{20}$$

$$SP = \frac{TP}{TN+FP} \times 100\% \tag{21}$$

$$SN = \frac{TP}{TP+FN} \times 100\% \tag{22}$$

$$MCC = \frac{TN \times TP - FN \times FP}{\sqrt{(TP+FP)(TN+FN)(TP+FN)(TN+FP)}} \times 100\% \tag{23}$$

Here, TP is the number of DBPs that are predicted to be non-DBPs; FN is the number of non-DBPs that are predicted to be DBPs; FP is the number of DBPs that are predicted to be non-DBPs, and TN is the number of non-DBPs that are predicted to be non-DBP. In addition, ROC curve [76,77] and PR curve are also used to evaluate classification performance.

3.2. Parameters selection

We tune parameters for best performance by 5-fold cross-validation (5-CV) and grid searching on PDB1075. First, we try to find the optimal kernel bandwidth for six types of kernels. The optimal kernel bandwidth is obtained from its single kernel RKM and set the range from 2^{-5} to 2^{5} with step 2^{1}. The results are shown in Table 2. Then, we select the parameters λ, η and μ from 2^{-5} to 2^{5} with step 2^{1}, k from 10 to 50 with step 5. λ and k are parameters of LapLKA. λ weighs the relationship between the local manner and the global manner, and k is the number of neighbors for samples. η and μ are regularization parameters in RKM objective function.

To demonstrate parameters sensitivity of LapLKA, we study the variation of performance according to change of λ and k with fixed parameters of RKM. Figure 2 shows the ACC variation with λ and k on PDB1075. We can see that our method is not sensitive to λ and k, especially k. Similarity, we study parameters sensitivity of RKM with fixed parameters of LapLKA. The ACC variation of η and μ is shown in Figure 3. We can observe that η and μ are both sensitivity parameters. When $\lambda = 2^5$ and $\mu = 2^5$, the ACC score is the lowest. With λ and μ decreases gradually, the predicted performance of 5-CV is increase. It is still an open problem that the sensitively of the model to hyperparameters. Finally, we set k, λ, μ and η to be 15, 2, 0.125 and 2, respectively.

Table 2. The optimal parameters for single kernel RKM.

Parameters	NMBAC	GE	MCD	PSSM-AB	PSSM-DWT	PSSM-Pse
γ	2^{1}	2^{0}	2^{0}	2^{0}	2^{0}	2^{0}
η	2^{0}	2^{0}	2^{0}	2^{1}	2^{-1}	2^{-1}
μ	2^{-1}	2^{-1}	2^{-3}	2^{0}	2^{-4}	2^{-1}

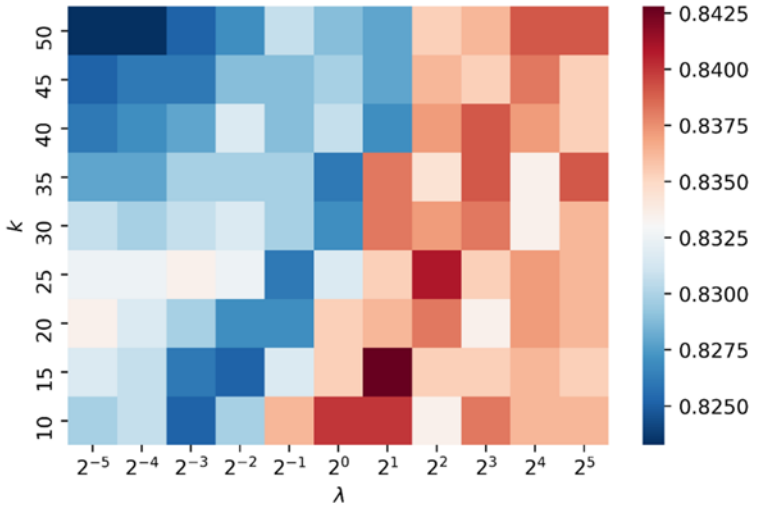


Figure 2. Effect of k and λ on ACC with fixed $\mu = 0.125$ and $\eta = 2$ via 5-CV on PDB1075.

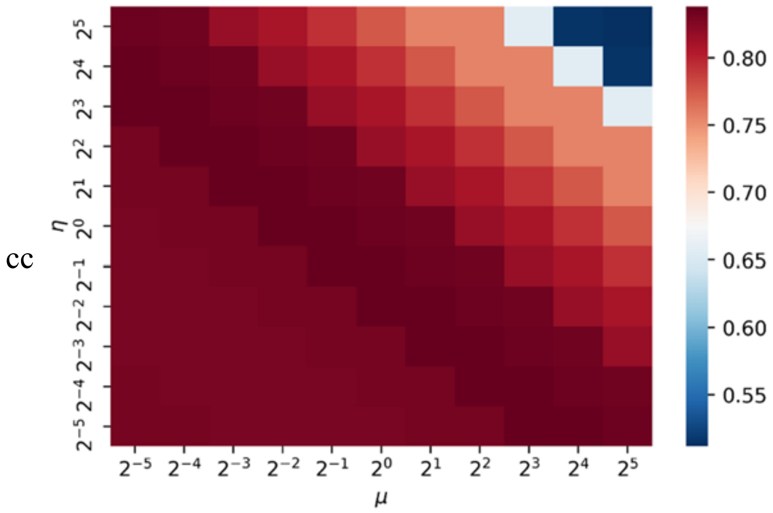


Figure 3. Effect of μ and η on ACC with fixed $k = 15$ and $\lambda = 2$ via 5-CV on PDB1075.

In our method, there are four hyperparameters: k, λ, η, μ. Here, k is the parameter in the local multi-kernel, λ weighs the relationship between the global kernel and the local kernel, and η, μ is the RKM positive real regularization constant.

3.3. Compared with single kernel

To analyze the performance of these kernels, we evaluate different kernels in two experiments, as shown in Tables 3 and 4 and Figure 4.

Table 3. Compared with single kernel on PDB1075 (LOOCV).

Kernel type	ACC (%)	SN (%)	SP (%)	MCC	AUC
NMBAC	73.49	76.57	70.55	0.4716	0.818
GE	70.98	71.81	70.18	0.4198	0.786
MCD	68.19	76.76	60.00	0.3723	0.762
PSSM-AB	76.37	80.76	72.18	0.5307	0.840
PSSM-DWT	74.51	75.05	74.00	0.4903	0.828
PSSM-Pse	77.95	82.67	73.45	0.5628	0.864
Weighted with LapLKA	**85.77**	**89.90**	**81.82**	**0.7185**	**0.926**

Table 4. Compared with single kernel on PDB2272 (independent test).

Kernel type	ACC (%)	SN (%)	SP (%)	MCC	AUC
NMBAC	68.31	72.77	63.72	0.3665	0.7512
GE	65.36	69.04	61.57	0.3070	0.7225
MCD	71.79	79.36	63.99	0.4391	0.7853
PSSM-AB	77.33	88.99	**65.33**	0.5601	0.8656
PSSM-DWT	71.74	92.11	50.76	0.4723	0.8377
PSSM-Pse	75.53	91.33	59.25	0.5354	0.8608
Weighted with LapLKA	**79.53**	**96.62**	61.93	**0.6264**	**0.9303**

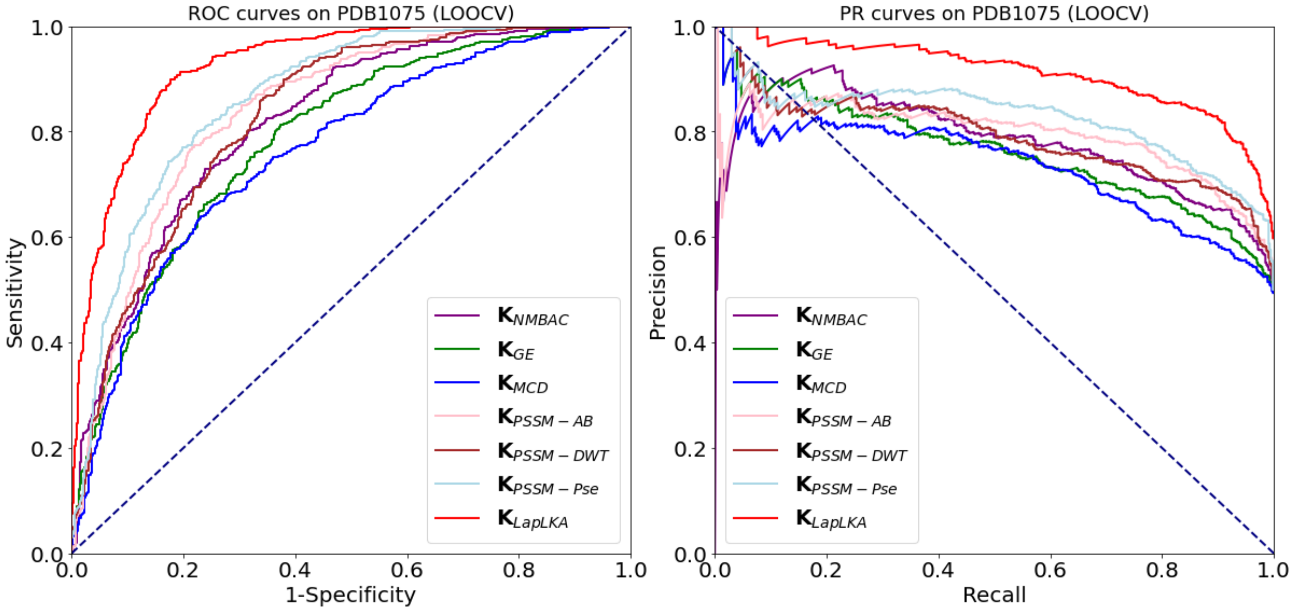


Figure 4. The ROC and PR curves of different kernels (LOOCV).

Results of LOOCV on PDB1075 are listed in Table 3 and Figure 4. Because LapLKA is a linear combination of six types of kernels, LapLKA performs much better than the single kernel. In addition, the average scores of ACC, SN, SP, MCC and AUC with kernels using PSSM information (PSSM-AB, PSSM-DWT, PSSM-Pse) are 76.28%, 79.49%, 73.21%, 0.5279 and 0.8439, respectively. The kernels using AAC information (GE, MCD) perform worst, its average score of metrices is ACC:69.58%, SN:74.28%, SP:65.09%, MCC:0.3960 and AUC:0.7743. We can observe that the model using PSSM information is better than other information. Thus, PSSM is an excellent feature extraction method that contains the evolutionary relationship with other sequences.

Table 5. The running time of different kernels on PDB2272 (independent test).

Kernel type	Sec
NMBAC	38.73
GE	38.91
MCD	42.04
PSSM-AB	38.76
PSSM-DWT	44.62
PSSM-Pse	38.68
Weighted with LapLKA	162.32

Results of independent test on PDB2272 are list in Table 4. Table 4 shows a same trend with Table 3. LapLKA achieves best performance and the model using PSSM information is better than other information. In addition, PSSM-AB achieves highest SP (65.33%) and second highest ACC (77.33%), MCC (0.5601) and AUC (0.8656). The advantage of LapLKA is also reflected on PDB2272. The improvement in ACC, SN, MCC and AUC are 2.2% (PSSM-AB), 4.51% (PSSM-DWT), 0.0663 (MCC) and 0.0647 (AUC), respectively.

The running time of RKM with different kernels is also evaluated. In Table 5, the results are presented. RKM with multiple kernels is implemented in Matlab. It runs on an Intel i7-10750H CPU with 16 GB RAM. As we can see, our method is the most time-consuming. This can be explained by looking at the time complexity of RKM with single kernel and RKM with LapLKA-MKL. In RKM with single kernel, the time complexity of the training phase is largely influenced by calculating kernel matrices ($O\left(N^2d\right)$) and solving linear problems ($O\left(N^3\right)$). Three steps are involved in RKM with LapLKA-MKL: calculate the kernel matrices, MKL and solve a linear problem. These steps have a time complexity of $O\left(PN^2\overline{d}\right)$, $O\left(N^3\right)$ and $O\left(N^3\right)$.

3.4. Compared with baseline methods

Compared with single kernels, LapLKA achieves an obvious advantage. As a further demonstration of LapLKA's fusion capabilities, we compare it with BSV, FC, Comm and MV. Other MKL algorithms are also evaluated, including CKA, HSIC and FKL. In addition, we compare our method with other well-known classifiers. Other classifiers are fed multiple features concatenated for fair comparison. Details of the baseline methods are as follows:

- Best Single Kernel with RKM (BSK-RKM): The results of applying RKM in the best performance.
- Feature Concatenation with RKM (FC-RKM): Multiple features are concatenated and RKM is used to do classification.
- Feature Concatenation with Xtreme gradient boosting (FC-XGBoost): Multiple features are concatenated and XGBoost is used to do classification. The XGBoost [78] algorithm is a kind of ensemble learning model, which produces a strong model by assembling decision trees.
- Feature Concatenation with Random Forest (FC-RF): Multiple features are concatenated and RF is used to do classification. RF [79] is a classification algorithm combining ensemble tree-structed classifiers.
- Feature Concatenation with K Nearest Neighbors (FC-KNN): Multiple features are concatenated

and KNN [80] is used to do classification. KNN is an algorithm for classification, which assigns a class label to a new data point based on the k nearest neighbors in the feature space.

• Committee RKM with RKM (Comm-RKM): Each kernel was input to RKM classification separately and taking the average of multiple RKM results as the final prediction result.

• Multi-View RKM classification [55] (MV-RKM): MV-RKM is an extension of the RKM Classification by assuming shared hidden nodes over all different features. The linear system of MV-RKM is:

$$\begin{bmatrix} \frac{1}{\eta}\sum_{i=1}^{P}\mathbf{K}_p + \mu\mathbf{I}_N & P_N \\ 1_N^T & 0 \end{bmatrix}\begin{bmatrix} y\odot h \\ b \end{bmatrix} = \begin{bmatrix} Py \\ 0 \end{bmatrix} \tag{24}$$

where P_N is a column vector where each element equals P. From Eq (24), we can observe that MV-RKM can be seen as the MKL with mean weighted based RKM.

• Centered Kernel Alignment [26] with RKM (CKA-RKM): CKA is a kind of MKL algorithm. CKA estimates the optimal weights of kernels by maximizing the cosine similarity between the optimal kernel and ideal kernel. Different from LapLKA, CKA only consider the global manner.

• Hilbert Schmidt Independence Criterion [81] with RKM (HSIC-RKM): HSIC is a kind of MKL algorithm. HSIC optimize the kernel weight by maximize the dependence between the optimal kernel and ideal kernel. The advantage of HSIC is simple calculation and fast convergence.

• Fast Kernel Learning [82] with RKM (FKL-RKM): FKL also is a kind of MKL algorithm. FKL find fusing weight by minimize the Euclidean distance between the optimal kernel and ideal kernel. Since the objective function of FKL is quadratic programming, it is fast and effective at solving kernel weights.

The hyperparameters of these fusion methods are detected by the 5-CV and the grid search on PDB1075.

Table 6. Performance compared with other baseline methods on PDB1075 (LOOCV).

Basline method	ACC (%)	SN (%)	SP (%)	MCC	AUC
BSV-RKM	77.95	82.67	73.45	0.5628	0.864
FC-RKM	81.77	85.71	78.00	0.6382	0.898
FC-XGBoost	73.52	76.93	70.34	0.4730	0.807
FC-RF	75.18	65.58	84.39	0.5091	0.837
FC-KNN	75.68	79.03	72.34	0.5140	0.829
Comm-RKM	81.86	87.05	76.91	0.6418	0.899
MV-RKM	83.35	88.76	78.18	0.6720	0.922
CKA-RKM	82.51	84.19	80.91	0.6509	0.910
HSIC-RKM	83.72	88.00	79.64	0.6777	0.916
FKL-RKM	84.09	89.14	79.27	0.6863	0.921
LapLKA-RKM	**85.77**	**89.90**	**81.82**	**0.7185**	**0.926**

Table 6 and Figure 5 show all baseline methods and LapLKA on the PDB1075 by LOOCV. Table 7 shows comparison between each baseline methods on the PDB2272 by independent test. We can see: 1) LapLKA has the best performance no matter LOOCV on PDB1075 or independent test on big dataset. This indicates that LapLKA can obtain the best optimal kernel for classification by effectively combing the multiple kernels. 2) MV, CKA, HSIC and KTA perform better than typical fusion methods (BSV, FC and Comm) on PDB1075 by LOOCV. However, these MKL methods (MV, CKA, HSIC and KTA) are slightly inferior to typical fusion methods.

A good prediction method should have good generalization capabilities. In light of this, we report the uncertainties of our method and baseline methods by 10-CV on PDB1075. The results are shown in Figure 6. According to the boxplot, our method is likely to produce similar results for different cross-validation splits. Additionally, our method produces the highest mean ACC. Furthermore, we report statistical tests of the differences under 10-CV on PDB1075. Table 7 demonstrates that, our method has statistically significant improvement over other baseline methods (P-value < 0.05, by t-test, in terms of ACC, for BSV-RKM, FC-RKM, FC-XGBoost, FC-RF, FC-KNN and CKA-RKM).

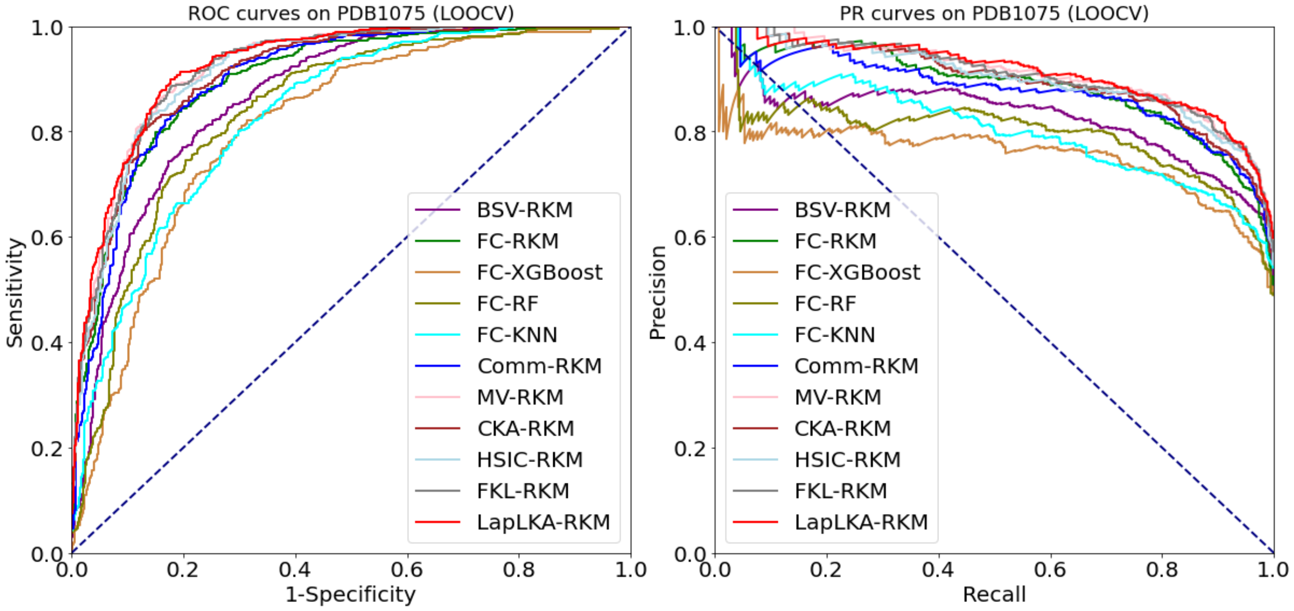


Figure 5. The ROC and PR curves of different baseline methods.

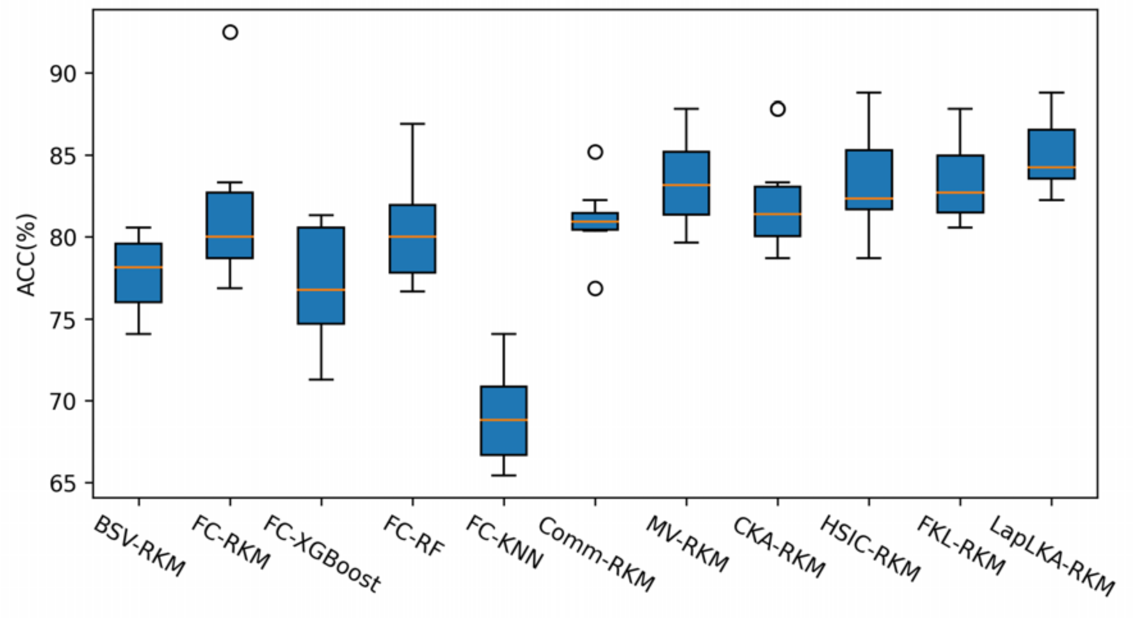


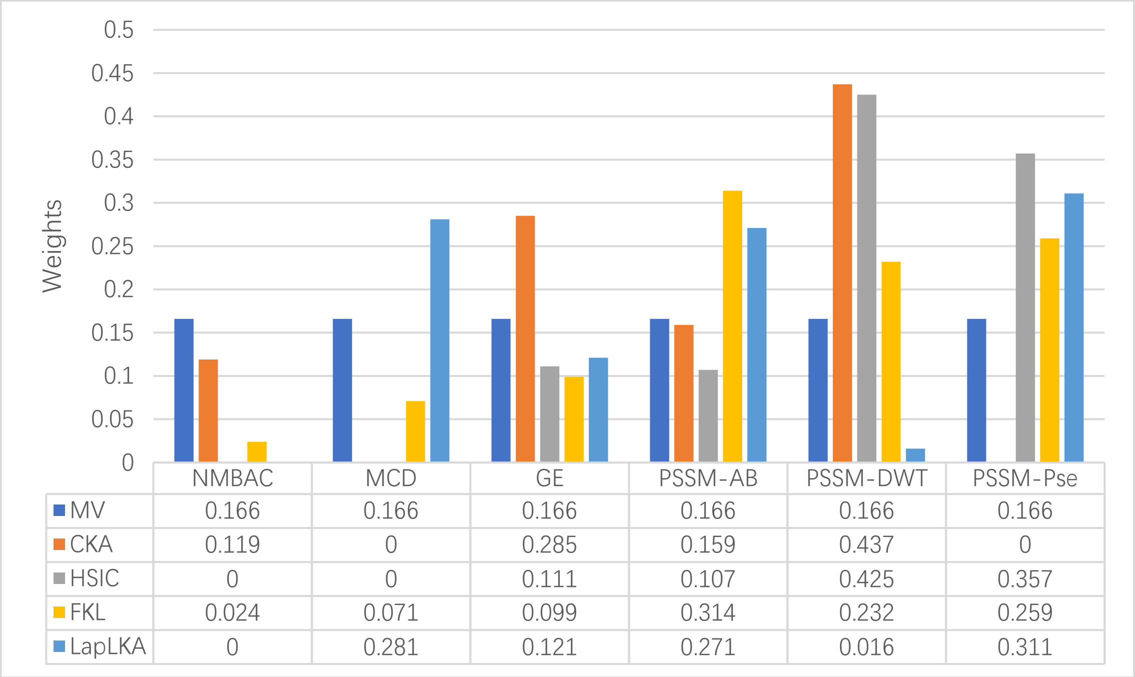
Figure 6. ACC of different baseline methods on PDB1075 (10-CV).

Table 7. The statistics of different baseline methods on PDB1075 (10-CV).

Basline method	*P* value
BSV-RKM	1.09E-6
FC-RKM	2.88E-2
FC-XGBoost	1.95E-5
FC-RF	1.15E-3
FC-KNN	5.81E-11
Comm-RKM	4.85E-4
MV-RKM	1.75E-1
CKA-RKM	4.08E-2
HSIC-RKM	1.57E-1
FKL-RKM	1.50E-1

Table 8. Performance compared with other baseline methods on PDB2272 (independent test).

Basline method	ACC (%)	SN (%)	SP (%)	MCC	AUC
BSV-RKM	77.33	88.99	65.33	0.5601	0.8656
FC-RKM	74.25	92.63	55.32	0.5183	0.8871
FC-XGBoost	73.62	55.92	91.35	0.5067	0.8243
FC-RF	76.34	60.24	**92.46**	0.5561	0.8340
FC-KNN	74.56	78.45	70.91	0.4968	0.8224
Comm-RKM	77.11	90.63	63.18	0.5609	0.8647
MV-RKM	75.18	93.50	56.30	0.5381	0.8855
CKA-RKM	75.09	92.80	56.84	0.5336	0.8717
HSIC-RKM	75.79	93.15	57.91	0.5472	0.8774
FKL-RKM	75.48	92.97	57.46	0.5412	0.8836
LapLKA-RKM	**79.53**	**96.62**	61.93	**0.6264**	**0.9303**



	NMBAC	MCD	GE	PSSM-AB	PSSM-DWT	PSSM-Pse
■ MV	0.166	0.166	0.166	0.166	0.166	0.166
■ CKA	0.119	0	0.285	0.159	0.437	0
■ HSIC	0	0	0.111	0.107	0.425	0.357
■ FKL	0.024	0.071	0.099	0.314	0.232	0.259
■ LapLKA	0	0.281	0.121	0.271	0.016	0.311

Figure 7. The weights of kernels obtained by different MKL on the PDB14189.

In addition, the weight of each kernel (with MV, CKA, HSIC, KTA and LapLKA) on PDB14189 is shown in Figure 7. In HSIC and LapLKA approaches, the weights of PSSM-Pse are the largest and NMBAC is close to 0. Additionally, the weights of kernels using AAC usually lower than kernels using PSSM. For example, the sum of weights of PSSM kernels is 0.598, and the weights of AAC kernels is 0.281 in LapLKA. The analysis of performance of single kernel demonstrates that, the model using PSSM information is better than other information. Therefore, we can draw the conclusion that LapLKA could set low weights to noise kernels.

3.5. Compared with other existing methods

Table 9. Performance comparison with other existing methods on PDB1075 (LOOCV).

Method	ACC (%)	SN (%)	SP (%)	MCC
iDNA-Prot [83]	75.40	83.81	64.73	0.50
iDNA-Prot|dis [84]	77.30	79.40	75.27	0.54
PseDNA-Pro [12]	76.55	79.61	73.63	0.53
iDNAPro-PseAAC [13]	76.55	75.62	77.45	0.53
Local-DPP [85]	79.10	84.80	73.60	0.59
MKSVM-HKA [86]	81.30	82.29	80.36	0.63
FKRR-MVSF [87]	83.26	85.17	80.91	0.67
CKA with SVM [26]	84.19	85.91	82.55	0.68
MK-FSVM-SVDD [88]	82.23	81.90	82.55	0.65
UMAP-DBP [89]	82.97	82.83	83.72	0.67
HKAM-MKM [28]	84.28	80.00	88.76	0.69
MV-H-RKM [38]	84.65	87.24	**93.64**	0.69
LapLKA-RKM	**85.77**	**89.90**	81.82	**0.72**

Table 10. Performance comparison with other existing methods on PDB2272 (independent test).

Method	ACC (%)	SN (%)	SP (%)	MCC
DPP-PseACC [17]	58.1	56.6	59.6	0.163
PseDNA-Pro [12]	61.8	75.3	48.1	0.243
MsDBP [31]	64.3	70.7	63.2	0.340
MKL-HSIC with H-LapSVM [27]	69.4	72.1	56.1	0.401
MLapSVM-LBS [29]	71.2	71.6	**70.8**	0.424
DBP-CNN [37]	67.9	69.0	66.8	0.358
Deep Transfer Learning [35]	74.2	-	-	-
PDBP-Fusion [36]	77.8	73.3	66.9	0.567
GCN-method [33]	78.5	70.7	64.2	0.400
HKAM-MKM [28]	78.4	91.5	62.4	0.596
LapLKA-RKM	**79.5**	**96.6**	61.9	**0.626**

Here, we compare our approach with other existing methods on PDB1075 by LOOCV and PDB2272 by independent test, as shown in Tables 9 and 10, respectively. It can be observed that high ACC of 85.77% (PDB1075 by LOOCV), 79.5% (PDB2272 by independent test). On PDB1075, our method got 1.12%, 2.26% and 0.03 improvement in ACC, SN and MCC over the second bet MV-H-RKM, respectively. MV-H-RKM enforce the structure consistency between input feature and the hidden node by the hypergraph regularization term. Therefore, MV-H-RKM also achieves the good performance. However, MV-H-RKM couple multiple features by means of hidden vector, which is same as MV-RKM. This means MV-H-RKM cannot filter noise features. HKAM-MKM achieves good performance with ACC (84.28%) and MCC (0.69). Similar our method, HKAM-MKM both consider the local and global kernel alignment and propose a hybrid kernel alignment model. Difference our method, the optimal kernel is input to SVM.

4. Conclusions

In this paper, we developed an approach called LapLKA-RKM, a machine learning based predictor for DBPs. Our method contains three steps: feature extraction, feature fusion and classifier construction. We apply six different feature extraction methods (MCD, GE, NMBAC, PSSM-AB, PSSM-DWT and PSSM-Pse) to represent the protein sequences. Then, we utilize LapLKA-MKL to combine multiple predefined kernels. Finally, we employ RKM as a predictive classifier.

Compared with other baseline methods and existing DBPs predictor, our method achieves the best accuracy on different datasets by LOOCV and independent test. On the LOOCV of PDB1075, LapLKA-RKM achieves the highest ACC, SN, MCC and AUC of 85.77%, 89.90%, 81.82%, 0.72 and 0.9258, respectively. Further, our method was tested on PDB2272 via independent test and also achieves better performance with ACC (79.5%), SN (96.6%), MCC (0.626) and AUC (0.9303). The results demonstrated that our method is an accurate tool for identification of DBPs. We also built an online platform to represent our model. We hope the simple to use web interface will lead to wide adoption of our method.

Acknowledgments

This work is supported in part by the Postgraduate Research and Practice Innovation Program of Jiangsu Province (No. KYCX22_3276), National Natural Science Foundation of China (NSFC 62172076, 62073231, U22A2038), Zhejiang Provincial Natural Science Foundation of China (Grant No. LY23F020003), the Municipal Government of Quzhou (Grant No. 2022D006), the Municipal Government of Suzhou (Grant No. 2022SS03, SZFCXK202144).

Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Availability of data and materials

The data set and source code is obtained from https://figshare.com/articles/dataset/LapRKM-RKM_zip/22578496.

References

1. M. J. Buck, J. D. Lieb, ChIP-chip: considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments, *Genomics*, **83** (2004), 349–360. https://doi.org/10.1016/j.ygeno.2003.11.004
2. F. Cui, S. Li, Z. Zhang, M. Sui, C. Cao, A. E. Hesham, et al., DeepMC-iNABP: Deep learning for multiclass identification and classification of nucleic acid-binding proteins, *Comput. Struct. Biotechnol. J.*, **20** (2022), 2020–2028. https://doi.org/10.1016/j.csbj.2022.04.029
3. F. Cajone, M. Salina, A. Benelli-Zazzera, 4-Hydroxynonenal induces a DNA-binding protein similar to the heat-shock factor, *Biochem. J.*, **262** (1989), 977–979. https://doi.org/10.1042/bj2620977

4. M. Gao, S. Jeffrey, DBD-Hunter: a knowledge-based method for the prediction of DNA-protein interactions, *Nucleic Acids Res.*, **36** (2008), 3978–3992. https://doi.org/10.1093/nar/gkn332
5. Y. Fang, Y. Guo, Y. Feng, M. Li, Predicting DNA-binding proteins: approached from Chou's pseudo amino acid composition and other specific sequence features, *Amino Acids*, **34** (2008), 103–109. https://doi.org/10.1007/s00726-007-0568-2
6. C. Cao, L. Mak, G. Jin, P. Gordon, K. Ye, Q. Long, PRESM: personalized reference editor for somatic mutation discovery in cancer genomics, *Bioinformatics*, **35** (2019), 1445–1452. https://doi.org/10.1093/bioinformatics/bty812
7. C. Cao, M. Greenberg, Q. Long, WgLink: reconstructing whole-genome viral haplotypes using L0+ L1-regularization, *Bioinformatics*, **37** (2021), 2744–2746. https://doi.org/10.1093/bioinformatics/btab076
8. C. Cao, J. He, L. Mak, D. Perera, D. Kwok, J. Wang, et al., Reconstruction of microbial haplotypes by integration of statistical and physical linkage in scaffolding, *Mol. Biol. Evol.*, **38** (2021), 2660–2672. https://doi.org/10.1093/molbev/msab037
9. Z. Zhang, F. Cui, W. Su, L. Dou, A. Xu, C. Cao, et al., webSCST: an interactive web application for single-cell RNA-sequencing data and spatial transcriptomic data integration, *Bioinformatics*, **38** (2022), 3488–3489. https://doi.org/10.1093/bioinformatics/btac350
10. Z. Zhang, F. Cui, C. Wang, L. Zhao, Q. Zou, Goals and approaches for each processing step for single-cell RNA sequencing data, *Briefing Bioinf.*, **22** (2021), bbaa314. https://doi.org/10.1093/bib/bbaa314
11. F. Cui, Z. Zhang, C. Cao, Q. Zou, D. Chen, X. Su, Protein–DNA/RNA interactions: Machine intelligence tools and approaches in the era of artificial intelligence and big data, *Proteomics*, **22** (2022), 2100197. https://doi.org/10.1002/pmic.202100197
12. B. Liu, J. Xu, S. Fan, R. Xu, J. Zhou, X. Wang, PseDNA-Pro: DNA-binding protein identification by combining Chou's PseAAC and physicochemical distance transformation, *Mol. Inf.*, **34** (2015), 8–17. https://doi.org/10.1002/minf.201400025
13. B. Liu, S. Wang, X. Wang, DNA binding protein identification by combining pseudo amino acid composition and profile-based protein representation, *Sci. Rep.*, **5** (2015), 15479. https://doi.org/10.1038/srep15479
14. Y. Ding, J. Tang, F. Guo, Identification of protein-ligand binding sites by sequence information and ensemble classifier, *J. Chem. Inf. Model.*, **57** (2017), 3149–3161. https://doi.org/10.1021/acs.jcim.7b00307
15. Y. Ding, J. Tang, F. Guo, Human protein subcellular localization identification via fuzzy model on Kernelized Neighborhood Representation, *Appl. Soft Comput.*, **96** (2020), 106596. https://doi.org/10.1016/j.asoc.2020.106596
16. G. Nimrod, M. Schushan, A. Szilágyi, C. Leslie, N. Ben-Tal, iDBPs: a web server for the identification of DNA binding proteins, *Bioinformatics*, **26** (2010), 692–693. https://doi.org/10.1093/bioinformatics/btq019
17. M. S. Rahman, S. Shatabda, S. Saha, M. Kaykobad, M. S. Rahman, DPP-PseAAC: a DNA-binding protein prediction model using Chou's general PseAAC, *J. Theor. Biol.*, **452** (2018), 22–34. https://doi.org/10.1016/j.jtbi.2018.05.006
18. K. C. Chou, Some remarks on protein attribute prediction and pseudo amino acid composition, *J. Theor. Biol.*, **273** (2011), 236–247. https://doi.org/10.1016/j.jtbi.2010.12.024
19. C. Cortes, V. Vapnik, Support-vector networks, *Mach. Learn.*, **20** (1995), 273–297. https://doi.org/10.1007/BF00994018

20. S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, et al., Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, *Nucleic Acids Res.*, **25** (1997), 3389–3402. https://doi.org/10.1093/nar/25.17.3389
21. Y. Ding, J. Tang, F. Guo, Identification of protein–ligand binding sites by sequence information and ensemble classifier, *J. Chem. Inf. Model.*, **57** (2017), 3149–3161. https://doi.org/10.1021/acs.jcim.7b00307
22. F. Guo, Y. Ding, Z. Li, J. Tang, Identification of protein-protein interactions by detecting correlated mutation at the interface, *J. Chem. Inf. Model.*, **55** (2015), 2042–2049. https://doi.org/10.1021/acs.jcim.5b00320
23. Y. Ding, J. Tang, F. Guo, Protein crystallization identification via fuzzy model on linear neighborhood representation, *IEEE/ACM Trans. Comput. Biol. Bioinf.*, **18** (2019), 1986–1995. https://doi.org/10.1109/TCBB.2019.2954826
24. M. Wang, J. Yang, G. Liu, Z. Xu, K. Chou, Weighted-support vector machines for predicting membrane protein types based on pseudo-amino acid composition, *Protein Eng. Des. Sel.*, **17** (2004), 509–516. https://doi.org/10.1093/protein/gzh061
25. M. Hayat, A. Khan, Predicting membrane protein types by fusing composite protein sequence features into pseudo amino acid composition, *J. Theor. Biol.*, **271** (2010), 10–17. https://doi.org/10.1016/j.jtbi.2010.11.017
26. Y. Qian, L. Jiang, Y. Ding, J. Tang, F. Guo, A sequence-based multiple kernel model for identifying DNA-binding proteins, *BMC Bioinf.*, **22** (2021), 1–18. https://doi.org/10.1186/s12859-020-03875-x
27. Y. Qian, H. Meng, W. Lu, Z. Liao, Y. Ding, H. Wu, Identification of DNA-binding proteins via hypergraph based laplacian support vector machine, *Curr. Bioinf.*, **17** (2022), 108–117. https://doi.org/10.2174/1574893616666210806091922
28. S. Zhao, Y. Ding, X. Liu, X. Su, HKAM-MKM: a hybrid kernel alignment maximization-based multiple kernel model for identifying DNA-binding proteins, *Comput. Biol. Med.*, **145** (2022), 105395. https://doi.org/10.1016/j.compbiomed.2022.105395
29. M. Sun, P. Tiwari, Y. Qian, Y. Ding, Q. Zou, MLapSVM-LBS: Predicting DNA-binding proteins via a multiple Laplacian regularized support vector machine with local behavior similarity, *Knowledge-Based Syst.*, **250** (2022), 109174. https://doi.org/10.1016/j.knosys.2022.109174
30. M. Gao, J. Skolnick, A threading-based method for the prediction of DNA-binding proteins with application to the human genome, *PLoS Comput. Biol.*, **5** (2009), e1000567. https://doi.org/10.1371/journal.pcbi.1000567
31. X. Du, Y. Diao, H. Liu, S. Li, MsDBP: Exploring DNA-binding Proteins by Integrating Multi-scale Sequence Information via Chou's 5-steps Rule, *J. Proteome Res.*, **18** (2019). https://doi.org/10.1021/acs.jproteome.9b00226
32. W. Lu, X. Chen, Y. Zhang, H. Wu, Y. Ding, J. Shen, et al., Research on DNA-binding protein identification method based on LSTM-CNN feature fusion, *Comput. Math. Methods Med.*, **2022** (2022). https://doi.org/10.1155/2022/9705275
33. W. Lu, N. Zhou, Y. Ding, H. Wu, Y. Zhang, Q. Fu, et al., Application of DNA-binding protein prediction based on graph convolutional network and contact map, *Biomed Res. Int.*, **2022** (2022). https://doi.org/10.1155/2022/9044793
34. M. Michel, D. Menéndez Hurtado, A. Elofsson, PconsC4: fast, accurate and hassle-free contact predictions, *Bioinformatics*, **35** (2019), 2677–2679. https://doi.org/10.1093/bioinformatics/bty1036

35. J. Yan, T. Jiang, J. Liu, Y. Lu, S. Guan, H. Li, et al., DNA-binding protein prediction based on deep transfer learning, *Math. Biosci. Eng.*, **19** (2022), 7719–7736. https://doi.org/10.3934/mbe.2022362
36. G. Li, X. Du, X. Li, L. Zou, G. Zhang, Z. Wu, Prediction of DNA binding proteins using local features and long-term dependencies with primary sequences based on deep learning, *PeerJ*, **9** (2021), e11262. https://doi.org/10.7717/peerj.11262
37. O. Barukab, F. Ali, W. Alghamdi, Y. Bassam, S. A. Khan, DBP-CNN: Deep learning-based prediction of DNA-binding proteins by coupling discrete cosine transform with two-dimensional convolutional neural network, *Expert Syst. Appl.*, **197** (2022), 116729. https://doi.org/10.1016/j.eswa.2022.116729
38. S. Guan, Y. Qian, T. Jiang, Y. Ding, M. Jiang, H. Wu, MV-H-RKM: A Multiple View-based Hypergraph Regularized Restricted Kernel Machine for predicting DNA-binding proteins, *IEEE/ACM Trans. Comput. Biol. Bioinf.*, **20** (2022), 1246–1256. https://doi.org/10.1109/TCBB.2022.3183191
39. Y. Ding, J. Tang, F. Guo, Identification of drug-target interactions via Dual Laplacian Regularized Least Squares with Multiple Kernel Fusion, *Knowledge-Based Syst.*, **204** (2020), 106254. https://doi.org/10.1016/j.knosys.2020.106254
40. Y. Ding, J. Tang, F. Guo, Identification of drug-side effect association via semisupervised model and multiple kernel learning, *IEEE J. Biomed. Health Inf.*, **23** (2018), 2619–2632. https://doi.org/10.1109/JBHI.2018.2883834
41. H. Yang, Y. Ding, J. Tang, F. Guo, Drug-disease associations prediction via multiple kernel-based dual graph regularized least squares, *Appl. Soft Comput.*, **112** (2021), 107811. https://doi.org/10.1016/j.asoc.2021.107811
42. X. Guo, P. Tiwari, Q. Zou, Y. Ding, Subspace projection-based weighted echo state networks for predicting therapeutic peptides, *Knowledge-Based Syst.*, **263** (2023), 110307. https://doi.org/10.1016/j.knosys.2023.110307
43. C. Cao, J. Wang, D. Kwok, F. Cui, Z. Zhang, D. Zhao, et al., webTWAS: a resource for disease candidate susceptibility genes identified by transcriptome-wide association study, *Nucleic Acids Res.*, **50** (2022), D1123–D1130. https://doi.org/10.1093/nar/gkab957
44. C. Cao, B. Ding, Q. Li, D. Kwok, J. Wu, Q. Long, Power analysis of transcriptome-wide association study: Implications for practical protocol choice, *PLos Genet.*, **17** (2021), e1009405. https://doi.org/10.1371/journal.pgen.1009405
45. Z. Zhang, F. Cui, C. Cao, Q. Wang, Q. Zou, Single-cell RNA analysis reveals the potential risk of organ-specific cell types vulnerable to SARS-CoV-2 infections, *Comput. Biol. Med.*, **140** (2022), 105092. https://doi.org/10.1016/j.compbiomed.2021.105092
46. F. Cui, Z. Zhang, Q. Zou, Sequence representation approaches for sequence-based protein prediction tasks that use deep learning, *Briefings Funct. Genomics*, **20** (2021), 61–73. https://doi.org/10.1093/bfgp/elaa030
47. Y. Cai, S. L. Lin, Support vector machines for predicting rRNA-, RNA-, and DNA-binding proteins from amino acid sequence, *Biochim. Biophys. Acta, Proteins Proteomics*, **1648** (2003), 127–133. https://doi.org/10.1016/S1570-9639(03)00112-2
48. Z. You, L. Zhu, C. Zheng, H. Yu, S. Deng, Z. Ji, Prediction of protein-protein interactions from amino acid sequences using a novel multi-scale continuous and discontinuous feature set, *BMC Bioinf.*, **15** (2014). https://doi.org/10.1186/1471-2105-15-S15-S9
49. Z. P. Feng, C. T. Zhang, Prediction of membrane protein types based on the hydrophobic index of amino acids, *J. Protein Chem.*, **19** (2000), 269–275. https://doi.org/10.1023/A:1007091128394

50. L. Nanni, S. Brahnam, A. Lumini, Wavelet images and Chou's pseudo amino acid composition for protein classification, *Amino Acids*, **43** (2012), 657–665. https://doi.org/10.1007/s00726-011-1114-9
51. J. Jeong, X. Lin, X. Chen, On position-specific scoring matrix for protein function prediction, *IEEE/ACM Trans. Comput. Biol. Bioinf.*, **8** (2011), 308–315. https://doi.org/10.1109/TCBB.2010.93
52. K. C. Chou, H. B. Shen, MemType-2L: A Web server for predicting membrane proteins and their types by incorporating evolution information through Pse-PSSM, *Biochem. Biophys. Res. Commun.*, **360** (2007), 339–345. https://doi.org/10.1016/j.bbrc.2007.06.027
53. R. Xu, J. Zhou, H. Wang, Y. He, X. Wang, B. Liu, Identifying DNA-binding proteins by combining support vector machine and PSSM distance transformation, *BMC Syst. Biol.*, **9** (2015), S10. https://doi.org/10.1186/1752-0509-9-S1-S10
54. B. Liu, F. Liu, X. Wang, J. Chen, L. Fang, C. K. Chen, Pse-in-One: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences, *Nucleic Acids Res.*, **43** (2015), W65–W71. https://doi.org/10.1093/nar/gkv458
55. L. Houthuys, J. Suykens, Tensor-based restricted kernel machines for multi-view classification, *Inf. Fusion*, **68** (2021), 54–66. https://doi.org/10.1016/j.inffus.2020.10.022
56. J. Suykens, Deep restricted kernel machines using conjugate feature duality, *Neural Comput.*, **29** (2017), 2123–2163. https://doi.org/10.1162/neco_a_00984
57. Y. Ding, W. He, J. Tang, Q. Zou, F. Guo, Laplacian regularized sparse representation based classifier for identifying DNA N4-methylcytosine sites via L2, 1/2-matrix norm, *IEEE/ACM Trans. Comput. Biol. Bioinf.*, **20** (2023), 500–511. https://doi.org/10.1109/TCBB.2021.3133309
58. C. Ai, P. Tiwari, H. Yang, Y. Ding, J. Tang, F. Guo, Identification of DNA N4-methylcytosine sites via multi-view kernel sparse representation model, *IEEE Trans. Artif. Intell.*, **2022** (2022), 1–10. https://doi.org/10.1109/TAI.2022.3187060
59. Y. Qian, Y. Ding, Q. Zou, F. Guo, Multi-view kernel sparse representation for identification of membrane protein types, *IEEE/ACM Trans. Comput. Biol. Bioinf.*, **20** (2022), 1234–1245. https://doi.org/10.1109/TCBB.2022.3191325
60. Y. Ding, P. Tiwari, Q. Zou, F. Guo, H. M. Pandey, C-loss based higher order fuzzy inference systems for identifying DNA N4-methylcytosine sites, *IEEE Trans. Fuzzy Syst.*, **30** (2022), 4754–4765. https://doi.org/10.1109/TFUZZ.2022.3159103
61. Y. Ding, P. Tiwari, F. Guo, Q. Zou, Shared subspace-based radial basis function neural network for identifying ncRNAs subcellular localization, *Neural Networks*, **156** (2022), 170–178. https://doi.org/10.1016/j.neunet.2022.09.026
62. T. Wang, L. Zhang, W. Hu, Bridging deep and multiple kernel learning: A review, *Inf. Fusion*, **67** (2021), 3–13. https://doi.org/10.1016/j.inffus.2020.10.002
63. Y. Ding, J. Tang, F. Guo, Identification of drug-side effect association via semi-supervised model and multiple kernel learning, *IEEE J. Biomed. Health Inf.*, **23** (2018), 2619–2632. https://doi.org/10.1109/JBHI.2018.2883834
64. Y. Qian, Y. Ding, Q. Zou, F. Guo, Identification of drug-side effect association via restricted Boltzmann machines with penalized term, *Briefings Bioinf.*, **23** (2022), bbac458. https://doi.org/10.1093/bib/bbac458
65. Y. Ding, J. Tang, F. Guo, Identification of drug-side effect association via multiple information integration with centered kernel alignment, *Neurocomputing*, **325** (2019), 211–224. https://doi.org/10.1016/j.neucom.2018.10.028

66. Y. Wang, X. Liu, Y. Dou, Q. Lv, Y. Lu, Multiple kernel learning with hybrid kernel alignment maximization, *Pattern Recognit.*, **70** (2017), 104–111. https://doi.org/10.1016/j.patcog.2017.05.005
67. J. O. Agushaka, A. E. Ezugwu, L. Abualigah, Dwarf mongoose optimization algorithm, *Comput. Methods Appl. Mech. Eng.*, **391** (2022), 114570. https://doi.org/10.1016/j.cma.2022.114570
68. L. Abualigah, D. Yousri, M. A. Elaziz, A. A. Ewees, M. A. A. Al-Qaness, A. H. Gandomi, Aquila optimizer: a novel meta-heuristic optimization algorithm, *Comput. Ind. Eng.*, **157** (2021), 107250. https://doi.org/10.1016/j.cie.2021.107250
69. L. Abualigah, M. A. Elaziz, P. Sumari, Z. W. Geem, A. H. Gandomi, Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer, *Expert Syst. Appl.*, **191** (2022), 116158. https://doi.org/10.1016/j.eswa.2021.116158
70. O. N. Oyelade, A. E. Ezugwu, T. I. A. Mohamed, L. Abualigah, Ebola optimization search algorithm: A new nature-inspired metaheuristic optimization algorithm, *IEEE Access*, **10** (2022), 16150–16177. https://doi.org/10.1109/ACCESS.2022.3147821
71. L. Abualigah, A. Diabat, S. Mirjalili, M. A. Elaziz, A. H. Gandomi, The arithmetic optimization algorithm, *Comput. Methods Appl. Mech. Eng.*, **376** (2021), 113609. https://doi.org/10.1016/j.cma.2020.113609
72. L. Abualigah, A. Diabat, P. Sumari, A. H. Gandomi, Applications, deployments, and integration of internet of drones (IoD): a review, *IEEE Sens. J.*, **21** (2021), 25532–25546. https://doi.org/10.1109/JSEN.2021.3114266
73. M. Grant, S. Boyd, Y. Ye, *CVX: Matlab Software For Disciplined Convex Programming*, 2011.
74. L. Houthuys, Z. Karevan, J. Suykens, Multi-view LS-SVM regression for black-box temperature prediction in weather forecasting, in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017. https://doi.org/10.1109/IJCNN.2017.7965975
75. L. Cheng, Y. Hu, J. Sun, M. Zhou, Q. Jiang, DincRNA: a comprehensive web-based bioinformatics toolkit for exploring disease associations and ncRNA function, *Bioinformatics*, **34** (2018), 1953–1956. https://doi.org/10.1093/bioinformatics/bty002
76. N. Q. K. Le, Q. Ho, V. Nguyen, J. Chang, BERT-Promoter: An improved sequence-based predictor of DNA promoter using BERT pre-trained model and SHAP feature selection, *Comput. Biol. Chem.*, **99** (2022), 107732. https://doi.org/10.1016/j.compbiolchem.2022.107732
77. N. Q. K. Le, D. T. Do, Q. A. Le, A sequence-based prediction of Kruppel-like factors proteins using XGBoost and optimized features, *Gene*, **787** (2021), 145643. https://doi.org/10.1016/j.gene.2021.145643
78. T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (2016), 785–794. https://doi.org/10.1145/2939672.2939785
79. L. Breiman, Random forests, *Mach. Learn.*, **45** (2001), 5–32. https://doi.org/10.1023/A:1010933404324
80. G. Guo, H. Wang, D. Bell, Y. Bi, K. Greer, KNN model-based approach in classification, in *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE. OTM 2003. Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, **2888** (2003). https://doi.org/10.1007/978-3-540-39964-3_62
81. H. Wang, Y. Ding, J. Tang, F. Guo, Identification of membrane protein types via multivariate information fusion with Hilbert-Schmidt Independence Criterion, *Neurocomputing*, **383** (2020), 257–269. https://doi.org/10.1016/j.neucom.2019.11.103

82. J. He, S. Chang, L. Xie, Fast kernel learning for spatial pyramid matching, in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2008. https://doi.org/10.1109/CVPR.2008.4587636
83. W. Lin, J. Fang, X. Xiao, K. Chou, iDNA-Prot: Identification of DNA binding proteins using random forest with grey model, *PLOS ONE*, **6** (2011), e24756. https://doi.org/10.1371/journal.pone.0024756
84. B. Liu, J. Xu, X. Lan, R. Xu, J. Zhou, X. Wang, et al., iDNA-Prot|dis: Identifying DNA-binding proteins by incorporating amino acid distance-pairs and reduced alphabet profile into the general pseudo amino acid composition, *PLOS ONE*, **9** (2014), e106691. https://doi.org/10.1371/journal.pone.0106691
85. L. Wei, J. Tang, Z. Quan, Local-DPP: An improved DNA-binding protein prediction method by exploring local evolutionary information, *Inf. Sci.*, **384** (2016), 135–144. https://doi.org/10.1016/j.ins.2016.06.026
86. Y. Ding, F. Chen, X. Guo, J. Tang, H. Wu, Identification of DNA-binding proteins by multiple kernel support vector machine and sequence information, *Curr. Proteomics*, **17** (2020), 302–310. https://doi.org/10.2174/1570164616666190417100509
87. Y. Zou, Y. Ding, J. Tang, F. Guo, L. Peng, FKRR-MVSF: A fuzzy kernel ridge regression model for identifying DNA-binding proteins by multi-view sequence features via Chou's five-step rule, *Int. J. Mol. Sci.*, **20** (2019), 4175. https://doi.org/10.3390/ijms20174175
88. Y. Zou, H. Wu, X. Guo, L. Peng, Y. Ding, J. Tang, et al., MK-FSVM-SVDD: a multiple kernel-based fuzzy SVM model for predicting DNA-binding proteins via support vector data description, *Curr. Bioinf.*, **16** (2021), 274–283. https://doi.org/10.2174/1574893615999200607173829
89. J. Wang, S. Zhang, H. Qiao, J. Wang, UMAP-DBP: an improved DNA-binding proteins prediction method based on uniform manifold approximation and projection, *Protein J.*, **40** (2021), 562–575. https://doi.org/10.1007/s10930-021-10011-y