Front. Comput. Sci., 2022, 16(6): 166904
https://doi.org/10.1007/s11704-021-1118-x

REVIEW ARTICLE

String kernels construction and fusion: a survey with bioinformatics application

Ren QI[1], Fei GUO[1], Quan ZOU (✉)[2,3]

1 College of Intelligence and Computing, Tianjin University, Tianjin 300350, China
2 Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu 610056, China
3 Hainan Key Laboratory for Computational Science and Application, Hainan Normal University, Haikou 571158, China

Abstract The kernel method, especially the kernel-fusion method, is widely used in social networks, computer vision, bioinformatics, and other applications. It deals effectively with nonlinear classification problems, which can map linearly inseparable biological sequence data from low to high-dimensional space for more accurate differentiation, enabling the use of kernel methods to predict the structure and function of sequences. Therefore, the kernel method is significant in the solution of bioinformatics problems. Various kernels applied in bioinformatics are explained clearly, which can help readers to select proper kernels to distinguish tasks. Mass biological sequence data occur in practical applications. Research of the use of machine learning methods to obtain knowledge, and how to explore the structure and function of biological methods for theoretical prediction, have always been emphasized in bioinformatics. The kernel method has gradually become an important learning algorithm that is widely used in gene expression and biological sequence prediction. This review focuses on the requirements of classification tasks of biological sequence data. It studies kernel methods and optimization algorithms, including methods of constructing kernel matrices based on the characteristics of biological sequences and kernel fusion methods existing in a multiple kernel learning framework.

Keywords multiple kernel learning, kernel fusion methods, support vector machines, biological sequences analysis

1 Introduction

In many bioinformatics tasks, it is a fundamental issue to define or learn a kernel to distinguish samples. The kernel method is based on statistical learning theory and kernel trick. In statistical learning theory, learning is considered to be a process of selecting an optimal function from a given set of functions, which is a statistical learning problem that minimizes risk functional [1]. The kernel function maps linearly inseparable data to high dimensional feature space then constructs a linear classifier in feature space [2,3]. In a learning algorithm with inner product operations, the inner product operations can be replaced by kernel functions.

Mercer mathematically gave the positive definite kernel function and the reproducing kernel Hilbert space theory and proved the necessary and sufficient conditions for the judgment of a positive definite kernel function [4]. Vapnik et al. used the kernel method to construct a new learning algorithm - Support Vector Machine (SVM) [5]. Meanwhile, the SVM algorithm greatly promotes the research and application of the kernel method. Schölkopf et al. further extended the kernel method to any learning algorithms containing the inner product operation and introduced algorithms such as kernel principal component analysis [2]. Mika et al. applied the kernel method to the linear Fisher decision criterion and formed the kernel Fisher decision criterion [6]. Cristianini et al. transformed a measure of similarity between kernels into a measure of the fitness of a given kernel [7].

Nowadays, the application of kernel methods has become one of the focuses of computer intelligence research. In the current circumstance, the research on the classification of kernel methods mainly focuses on the fields of the linear support vector machine, the nonlinear support vector machine, and multiple kernel learning algorithms. Classification based on the kernel method is an important branch that has developed in recent years and has obtained plentiful and substantial achievements. Subsequently, the kernel method was gradually applied to various research fields such as time series [8], Gene selection [9,10], disease detection [11], image classification [12], and image retrieval [13]. It can be said that ultimately, the kernel method has shown potentiality in its application.

In the past two decades, bioinformatics researchers explored the relationship between kernel methods and biology and the applicability of kernel methods to biological sequence analysis, kernel methods have exerted major influences in the development of biological sequence analysis. Leslie et al. introduced the mismatch kernel into protein classification.

Received March 15, 2021; accepted June 30, 2021

E-mail: zouquan@nclab.net

Experiments showed that mismatch also performed well in remote homology detection. The mismatch kernel belongs to classes of string kernels and it is a good kernel method for biological sequences [14]. Meanwhile, Tsuda and Noble showed that computing the diffusion kernel could achieve the same effect as maximizing von Neumann entropy, and they successfully applied the kernel method to the classification of protein functional prediction from metabolic and protein-protein interaction networks [15].

To improve the performance of classification or clustering tasks, many researchers have proposed some feature extraction methods, such as k-skip, k-gram, k-mer, PseAAC [16]. However, there are still problems such as overfitting and feature redundancy. In many cases, feature dimension reduction methods are required. Kernel techniques realize spatial transformation and generate new embeddings. Instead of getting the inner product of two vectors after feature extraction, the sequence kernel directly calculates the similarity of samples of raw data. The sequence kernel can fully retain and utilize the information of the data, avoiding the loss of information that may occur in feature extraction, and making the method more explanatory.

Following Tsuda's work, Kato et al. proposed a new supervised graph inference method, which was based on the kernel method. They used the proposed method to conduct experiments on various types of biological data such as amino acid sequences, and the method was successfully applied to gene expression and phylogenetic profiles [10]. In 2005, Swamidass proposed methods based on the kernel to address problems related to data coming from molecular biology. For example, Swamidass firstly focused on kernel analysis of small molecular sequences, found the rules, and then used the obtained information to predict the mutagenicity and anti-cancer activity of these sequences [17]. Ben-Hur and Noble have also successfully applied an unweighted sum of kernels in the prediction of protein-protein interactions [18]. Song et al. introduce a kernel-reweighted logistic regression method (KELLER), which is based on logistic regression. The method assigns weights to different kernels for the dynamic interactions between genes based on their time series [8].

As researchers have studied the problem in more depth, we have found that on some issues, a single kernel cannot be handled well. In bioinformatics, most of our samples are heterogeneous, and samples are characterized by multimodality and multi-view. For example, a gene fragment has a fixed physicochemical property, and besides, the gene fragment has some statistical and coding information from a sequence point of view. More broadly, we can also obtain homologous information from an evolutionary perspective. Therefore, each sample is represented by data from multiple modalities. In practice, whether the various modal data are directly stacked into a long feature vector or an extremely complex kernel is constructed, the information cannot be fully utilized.

Recently, other authors have provided a broader view of kernel methods. In particular, several research groups have proposed multiple kernel learning (MKL) methods [19−24]. Lewis et al. describe topics in a heterogeneous biological sequence where kernel methods have provided valuable solutions [25]. These methods obtain the SVM solutions to classification problems in multiple sets of data from different sources with different structures. Meanwhile, some researchers have developed many methods to obtain the weight of a single kernel corresponding to a single data type. Lanckriet et al. formulate the problem using semi-definite programming [19], whereas some authors formulate the problem using semi-infinite linear programming, such as the work of Rätsch et al. [26]. More recently, a generalized kernel learning was proposed, compared with the traditional MKL algorithm, generalized multiple kernel learning (GMKL) which is based on optimization schemes and regularizes subject to mild constraints [27]. However, the projected gradient descent GMKL optimizer is efficient. Following Varma's work, Jain et al. develop a Spectral Project Gradient descent optimizer which can take quick steps when far away from the optimum [28].

As a sample can be represented with different modalities, multi-modal and multiple kernel learning are developed [29]. MKL is non-linear learning, which is proposed as an alternative to cross-validation, feature selection, metric learning, and ensemble methods. It is a technically sound way of combining features, and different data formats can be used in the same formulation. MKL has good learning bounds, and it can combine features and train the classifier simultaneously. There are two key components in MKL, i.e., kernel construction and kernel fusion method. For the kernel construction method, one is to obtain different kernel matrices by selecting different kernel functions. The other method uses the same kernel function, but adds some information to the characteristics of the original data and then uses the same kernel function to perform operations. The kernel can be predefined, or generated by an intermediate process, or constructed by side information. Additionally, the kernel construction method and kernel fusion method are jointly learned to make good use of the complementary property of data.

Biological sequence analysis has greatly benefited from various kernel methods, and these MKL approaches have been successfully used in various bioinformatics tasks. For example, Lanckriet et al. apply the MKL method to predict yeast protein functional [21], and Borgwardt et al. create a protein function system that using graph kernels to predict the functional class membership of enzymes and non-enzymes [30]. In 2006, a protein subcellular localization system was explored by Zien and Ong [31], using a class of protein sequence kernels. Damoulas and Girolami show a probabilistic multi-class multi-kernel learning to solve the multi-features and multi-classes problem [32]. This method is based on the framework of naive Bayes. It combines multiple string kernels to achieve protein folding recognition and remote homology detection and classification tasks in protein datasets.

In 2007, Vert et al. explored a new pairwise kernel, which is similar to metric learning. The goal of the pairwise kernel is to make the distance between similar samples more closer while pushing away the different sample pairs and then use SVM to infer the relationship between sample pairs [33]. They propose

two types of pairwise kernels in their work. One is a direct inference based on the similarity between nodes connected by edges called Tensor Product Pairwise Kernel (TPPK), and the other is an indirect inference based on the similarity between two pairs of nodes called Metric Learning Pairwise Kernel (MLPK). The proposed kernel for pairs can be used by most SVM tools and inference pairwise relationships.

The MKL learning method has become a powerful tool to solve the problem of bioinformatics because it can integrate various forms of biomedical information in the form of multiple kernels. In this context, the focus of this study is on the application of kernel methods in biological sequence analysis, concentrating on those kernel methods with non-linear data at the same time. It also addresses the construction of kernels and kernel fusion methods which are commonly used in bioinformatics.

The rest structure of this review is organized as follows. In Section 2, we briefly introduce the property of SVM and kernel, then we describe the methods of constructing kernels in bioinformatics, such as pairwise kernels, sequence kernels, and non-sequence kernels. In this part, we outline multiple kernel learning and focus on common kernel fusion methods in biological data, and analyze the pros and cons of these fusion methods. In Section 3, we describe the application of kernel fusion in protein sequences, gene sequences, and binary networks (drug targeting, cancer detection, etc.). In the final section, we discuss the existing multiple kernel fusion methods and summarize existing available multiple kernel learning fusion methods into a table. Finally, we further discuss and look forward to the work that has not yet been done.

2 Methods

2.1 Kernel definitions and properties and SVM

Statistical Learning Theory was proposed by Vapnik [5], and it was a theoretical framework based on statistical analysis and functional analysis. The theory is constantly improved in practical applications, and gradually formed a series of learning algorithms represented by SVM [34]. Aronszajn first proposed the theory of reproducing kernels and kernel methods in 1950 [35]. At the end of the 20th century, Vapnik et al. transformed a nonlinear problem into a linear algorithm through the kernel function [36].

The kernel method has gradually become an important learning algorithm. The SVM makes the statistical learning theory more than a purely theoretical analysis tool, and it can be applied to the derivation of the actual algorithm. The idea of kernel methods is to map the input vector into a feature space by a given kernel function, and then classify samples in the high dimensional space. The goal of the training process is to find the hyperplane with the maximum margin. Theoretical analysis shows that the maximum margin of hyperplane generalization is better if can better distinguish newly added samples, and the algorithm is more robust [34].

For the given n train samples $\{(x_1,y_1),(x_2,y_2),...,(x_i,y_i),...,(x_n,y_n)\}$, where x_i is a sample vector and y_i is the corresponding label, SVM map sample x by function $\phi(\cdot)$. By optimizing the solution, the optimal discriminant function is obtained in the high-dimensional feature space [34]:

$$f(x) = \langle \mathrm{u}, \phi(x) \rangle + b, \tag{1}$$

where the u and b are calculated as follows [34].

$$\begin{aligned} &\min_{u,b} \frac{1}{2}\|u\|^2 + P\sum_{i=1}^{N}\xi_i, \\ &\text{s.t. } y_i(\langle u, \phi(x)\rangle + b) \geqslant 1 + \xi_i, \xi_i \geqslant 0, \ \forall i, \end{aligned} \tag{2}$$

P is the training error penalty coefficient, and it is used to balance the training error and generalization ability of models, and ξ_i is slack variable. To avoid the curse of dimensionality induced by directly calculating the feature map of the sample, the minimization optimization problem in Eq. (2) is generally used to be converted into the Lagrange dual problem. The above problem Eq. (2) is generally called the original problem of SVM, and its corresponding dual problem is represented in [34]:

$$\begin{aligned} &\max_{\alpha} \sum_{i=1}^{n}\alpha - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \left\langle \phi(x_i), \phi(x_j) \right\rangle, \\ &\text{s.t. } \sum_{i=1}^{n}\alpha_i y_i = 0, 0 \leqslant \alpha_i \leqslant C, \forall i, \end{aligned} \tag{3}$$

where α_i is a nonnegative dual variable. $\langle \phi(x_i), \phi(x_j) \rangle$ is the inner product of the vector, which can be directly obtained by the kernel function, i.e. $\langle \phi(x_i), \phi(x_j) \rangle = K(x_i, x_j)$. $K(\cdot,\cdot)$ is a certain kernel function, as we are known that the essence of the kernel is the inner product, and the inner product is the similarity.

Given a positive definite kernel K, there exists a corresponding Hilbert space $\mathcal{H}$. Properties of functions in Hilbert space are determined by the kernel. Then we introduce several common kernels. The Gaussian kernel function is the most commonly used kernel function by researchers. If K is a Gaussian kernel, the functions in Hilbert space $\mathcal{H}$ are smooth. Usually, the smoothness assumption is sensible for most datasets we want to tackle. The Gaussian kernel function's definition is shown in the following formula:

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle = \exp\left(\frac{-\left\| x_i - x_j \right\|^2}{2\sigma^2} \right). \tag{4}$$

Besides, there are linear kernels, polynomial kernels, sigmoid kernels, and so on, and we will not describe them in detail. The advantage of the kernel method is that it does not need to explicitly calculate the nonlinear transformation process of the model, nor does it need to know the specific mapping relationship to directly obtain the inner product between samples.

The dual optimization problem is a convex quadratic programming problem, and the global optimal solution can be obtained by the optimization solution [37]. After obtaining the optimal solution α^* of the dual problem (3) by the optimization solution, the optimal solution w^*, b^* of the original problem can be directly obtained by the Lagrangian formula (C is a constant):

$$w^* = \sum_{i=1}^{n} \alpha_i^* y_i \phi(x_i),$$

$$b^* = y_i - \sum_{t=1}^{n} \alpha_t^* y_t K(x_t, x_i), \exists C \geqslant \alpha_i^* \geqslant 0. \quad (5)$$

Therefore, the optimal discriminant function of the model is:

$$f(x) = \text{sign}\left(\sum_{t=1}^{n} \alpha_t^* y_t K(x_t, x_i) + b^* \right). \quad (6)$$

2.2 Methods for constructing kernels in bioinformatics

For decades, many outstanding researchers have contributed a lot to the kernel method, and have put forward a lot of effective theory and applied practices. Kernel functions are considered as similarity functions. They are often used in kernel methods. Using kernel functions is a versatile and elegant way because it replaces the raw data by using kernel matrices, unifying the format of heterogeneous data. The data for many bioinformatics tasks is represented by a variety of formats, as these data are from different sources and collection methods. Generally, biological data can contain binary vectors, continuous data, discrete sequence, graph data, etc. We need to build different types of kernels to handle different types of data so that the data preserves its meaning in the MKL algorithms. In this section, we introduce several methods for building kernels commonly used in bioinformatics. Figure 1 shows the process of heterogeneous data to matrix data by kernel functions. Each K means a kernel function, which can be any appropriate kernel, such as a linear kernel or a Gaussian kernel. Biological data can be converted to a matrix by the kernel function to carry out the following analysis.

● Sequence kernels

Sequence kernel is a kind of string kernel in bioinformatics, and sequences can be proteins or gene sequences. A protein or gene sequence is considered as a string defined on an alphabet with a certain length of 20 amino acids. The sequence kernel utilizes the sequence features to better preserve the original valid information, and it directly acts on the original data,

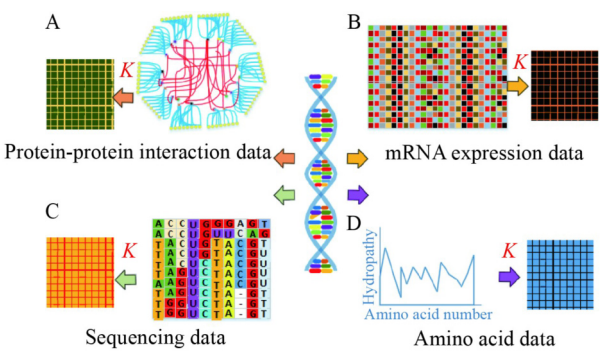


Fig. 1 The process of heterogeneous data to matrix data by kernel functions. A: Protein-protein interaction data can be converted to a distance matrix by a kernel function; B: mRNA expression data can be converted to a distance matrix by a kernel function; C: Sequencing data can be converted to a distance matrix by a kernel function; D: Hydrophobicity data can be converted to a distance matrix by a kernel function

which can effectively avoid the problem of feature redundancy. With the development of the kernel method, the application scenarios and data types are also constantly complicated, and some kernel methods for strings have emerged, such as the spectrum kernel [38], mismatch kernel [14], and local alignment kernel [39]. In their specific topics, the performance of the sequence kernel for sequence data is superior to the traditional methods. In this section, we describe several sequence kernels that are commonly used in biological sequence analysis.

The spectrum kernel is the most commonly used string kernel. All combinations of characters of finite length constitute the input space. The length of a string after the mapping function is the number of times the string appears in the sequence [38]. The spectrum kernel can be used for the classification of proteins with the constraint that no spaces are allowed between the strings, i.e., the similarity measurement length is defined as a common sub-region that must be identical. The spectrum kernel has a problem with high time cost. In general, the number of sequences we want to analyze is relatively small, but the combination of characters in the sequence causes the number of sequences to grow exponentially, so the input space is not directly related to the number of sequences. It depends on the number of subsequences constructed, which directly leads to the sparsity of the input space, we can use tree nodes structure to represent the sequence to avoid space waste to the greatest extent.

The weighted degree kernel is also a kernel method base on k-mers, the main difference from other methods is that the weighted kernel takes into account the positional information of the motif. It is an efficient method to compute sequence similarities [40]. The definition of the weighted degree kernel is the count of the k-mers in the two sequences co-occurrences at the corresponding position. The kernel is weighted by β_k:

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{k=1}^{d} \beta_k \sum_{l=1}^{L-k+1} I(u_{k,l}(\boldsymbol{x}_i) = u_{k,l}(\boldsymbol{x}_j)). \quad (7)$$

The weighted degree kernel is obtained by summing the number of sequences satisfying the motif conditions having the same position and the same length, and as shown in the above formula, the sum of the numbers of the same motifs of length L is gradually explored from i to $L-k+1$. The weighted summation operation is then performed on the sum of -mer. For a detailed explanation of specific parameters, please refer to [40].

Another recognized sequence kernel is the motif kernel [41], and motifs can be extracted using the eMOTIF method [42,43]. Suppose that $\mathcal{M}$ is a set of motifs, the definition of the motif kernel is, as usual, the difference is in the mapping function $\phi(x)$. where $\phi(x) = (\phi_m(x))_{m \in \mathcal{M}}$. $\phi_m(x)$ mean the frequency or times of the motif occurrences. The motif kernel is used for remote homology detection, they set samples which have the same remote homology as positive samples, and other samples in the database as negative samples.

The mismatch kernel belongs to the string kernel, which is proposed by Leslie et al. in 2003 [14] and successfully applied to protein classification. It is consisting of expressing a

sequence by a set of fixed-length (typically three to six amino acids long) blocks. The mismatch kernel between two sequences is the inner product between the fix-length blocks.

These string kernels play an important role in sequence analysis, and they have been successfully applied to protein remote homology detection [39,44], gene identification [45,46], and drug design [47,48]. We summarize the existing string kernel method and its application in Table 1.

● Pairwise kernels

Sample imbalance problem exist in many fields. If positive samples accounts for a large proportion in data, even if all negative samples were predicted as positive samples, the prediction function will still have high accuracy, but the actual prediction effect is very poor. To solve the problem, some researchers proposed pairwise kernels.

Ben-Hur and Noble proposed using a pairwise kernel to predict the interaction between two proteins [18]. Because predictive interactions require the similarity of a pair of protein samples rather than the similarity between individual samples, a pairwise kernel function is also needed to predict whether the two samples will interact.

The pairwise kernel between two pairs of protein samples (P_1, P_2) and (P'_1, P'_2) is defined as follows [18]:

$$K((P_1,P_2),(P'_1,P'_2)) = K'(P_1,P'_1)K'(P_2,P'_2) + K'(P_1,P'_2)K'(P_2,P'_1). \quad (8)$$

$K(\cdot,\cdot)$ can be any kernel that operates on individual genes or proteins. However, in fact, P_1 may be similar to P'_1, or it may be similar to P'_2. Suppose P_1, P_2 are represented p_1, p_2 with components $p_i^{(1)}$, $p_i^{(2)}$, we form the vector P_{12} with $p_i^{(1)}p_i^{(2)} + p_i^{(2)}p_i^{(1)}$. We can replace the above kernel definition with an explicit representation [18]:

$$K((P_1,P_2),(P'_1,P'_2)) = K'(p_{12},p'_{12}), \quad (9)$$

where p_{12} is the detailed representation of the pair (P_1, P_2) [18].

Vert et al. introduce two pairwise kernels, one is a direct inference based on the similarity between nodes connected by edges called TPPK [33]:

$$K_{\mathrm{TPPK}}((x_1,x_2),(x_3,x_4)) = K(x_1,x_3)K(x_2,x_4) + K(x_1,x_4)K(x_2,x_3). \quad (10)$$

And the other is an indirect inference based on the similarity between two pairs of nodes called MLPK. The kernel function is defined as follows [33]:

$$K_{\mathrm{MLPK}}((x_1,x_2),(x_3,x_4)) = (K(x_1,x_3) - K(x_1,x_4) - K(x_2,x_3) + K(x_2,x_4))^2. \quad (11)$$

Above two formulas, (x_1, x_2) and (x_3, x_4) represent two nodes, and x_i means a coordinate. Then in 2018, Cichonska proposed the PairwiseMKL method [55], which is successfully applied to drug bioactivity prediction. In pairwise kernel learning, training data is expressed as (x_d, x_c, y), where x_d represent drug and x_c represent cancer cell line, y is its associated response label. The block matrix is as follows [55]:

$$K = K_d \otimes K_c = \begin{pmatrix} k_d(x_{d_1},x_{d_1})K_c & k_d(x_{d_1},x_{d_2})K_c & \cdots & k_d(x_{d_1},x_{d_{nd}})K_c \\ k_d(x_{d_2},x_{d_1})K_c & k_d(x_{d_2},x_{d_2})K_c & \cdots & k_d(x_{d_2},x_{d_{nd}})K_c \\ \vdots & \vdots & & \vdots \\ k_d(x_{d_{nd}},x_{d_1})K_c & k_d(x_{d_{nd}},x_{d_2})K_c & \cdots & k_d(x_{d_{nd}},x_{d_{nd}})K_c \end{pmatrix}. \quad (12)$$

K is calculated by the drug kernel K_d and cell line kernel K_c[55]. The kernel weights assigned to different kernels represent the prediction capabilities of corresponding data sources.

Another famous pairwise kernel is that SVM-Pairwise kernel [56], Liao and Noble test it as the best method. The kernel is defined as an RBF kernel on the feature vectors $U^{\mathrm{pw}}(x)$ normalized to unit length, that is:

Table 1 Summary of existing string kernel types and applications

String kernels	Ref	Year	Applications	Literatures and Refs.
Spectrum kernel	[38]	2001	siRNA functionality prediction	Prediction of siRNA functionality using generalized string kernel and support vector machine [49]
			Protein homology detection	Semi-Supervised Abstraction-Augmented String Kernel for Multi-Level Bio-Relation Extraction [50]
			Protein classification	The spectrum kernel: A string kernel for SVM protein classification [38]
Mismatch kernel	[14]	2002	Protein classification	Mismatch string kernel for discriminative protein classification [14]
				Mismatch String Kernels for SVM Protein Classification [51]
				Semi-supervised protein classification using cluster kernels [52]
				Scalable Algorithms for String Kernels with Inexact Matching [53]
				Fast String Kernels using Inexact Matching for Protein Sequences [54]
Local alignment kernel	[39]	2004	Protein homology detection	Protein homology detection using string alignment kernels [39]
Weighted degree kernel	[40]	2004	Drug design	Combining Structure and Sequence Information Allows Automated Prediction of Substrate Specificities within Enzyme Families [48]
				Efficient peptide–MHC-I binding prediction for alleles with few known binders [47]
Motif kernel	[41]	2003	Gene identification	Improving the Caenorhabditis elegans Genome Annotation Using Machine Learning [45]
				mGene: Accurate SVM-Based Gene Finding with an Application to Nematode Genomes [46]
			Protein homology detection	Remote homology detection: a motif based approach [41]
Profiled-based string kernel	[44]	2005	Protein homology detection	Profile-based string kernels for remote homology detection and motif extraction [44]

$$k^{pw}(x,y)=\exp\left(-\frac{1}{2\sigma^2}\left(\frac{U^{pw}(x)}{\|U^{pw}(x)\|}-\frac{U^{pw}(y)}{\|U^{pw}(y)\|}\right)^2\right). \quad (13)$$

The specific parameters, such as σ in the formula are described in [56].

● Other kernels

In addition to pairwise kernels, sequence kernels, there are some other types of kernels, such as non-sequence kernels. The kernel is suitable for the non-sequence data: $K((x,y),(x',y'))=K'(s(x,y),s(x',y'))$. Such as Lempel-Ziv-Welch-Kernel (LZW-Kernel), Local Alignment Kernel (LAK kernel), Fisher kernel.

LZW-Kernel is a fast kernel utilizing variable length code blocks from LZW compressors [57]. The kernel properties of positive, symmetric, self-similarity, free of alignment and can be used directly in SVM classification problems. LZW-Kernel is a new convolution kernel, and it is based on code words identified with LZW universal text compressor. It is a one-step algorithm for backgrounds in big data applications. The convolution kernel function-LZW kernel is defined as follows [57]:

$$K_L(x_i,x_j)=\frac{\widetilde{K}_L(x_i,x_j)}{\sqrt{\widetilde{K}_L(x_i,x_i)\widetilde{K}_L(x_j,x_j)}}$$
$$=\exp\left\{\gamma\sum_{x_{id}\in D(x)\cap D(y)}\omega_d-\frac{1}{2}\gamma\left(\sum_{x_{id}\in D(x_i)}\omega_d+\sum_{y_{id}\in D(x_j)}\omega_d\right)\right\}. \quad (14)$$

LZW-Kernel can be applied in any method that contains kernel methods, such as SW [58] and BLAST [59]. SW and BLAST are sequence alignment methods.

Vert et al. proposed a family of convolution kernels called a local alignment kernel (LAK) [60] adapted to protein sequences. LAK can be considered as the kernelization SW. It can perform classification tasks of any kind of data which makes it popular.

The Fisher kernel combines Hidden Markov Models with SVM, which is proposed by Jaakkola [61]. It is a kernel function that is based on data features and edge information, etc. The gradient of the Loglikelihood of any sequence x ∈ X under a profile HMM probabilistic model. It allows detection in the case of partial information defects. The method is used to compare the homology correlation between proteins and has obtained good results in biological data experiments.

2.3 Multiple kernel Learning

Although SVM has a solid theoretical foundation and is widely used in various practical fields, selecting kernel function and debugging hyperparameters of SVM are difficult. It is a tough task to effectively integrate heterogeneous data from multiple different data sources. To solve these problems, Lanckriet et al. proposed Multiple Kernel Learning (MKL) [19]. Relative to single kernel SVM, multiple kernel learning can combine multiple kernels (or kernel matrices) in various forms, and it can achieve automatically select kernel functions and hyperparameters [62,63].

The core of multiple kernel learning is the fusion strategy of multiple kernel matrices. There are currently three different combinations, linear, nonlinear, and sample-based combination [62]. In multiple kernel learning algorithms, linear combinations are the most used. For a given set of basis kernel $\{K_m\}$ and the corresponding feature mapping function $\{\phi_m\}$, linear multiple kernel learning obtains the optimal linear combination of each base kernel through an optimization solution.

$$K=\sum_{m=1}^{M}d_mK_m, \quad (15)$$

where d_m is the weight parameter corresponding to the base kernel K_m. According to the difference of the weighting coefficient constraint of the base kernel, the linear combination method can be further subdivided into ordinary linear combination ($d_m\in\mathbb{R}$), cone combination ($d_m\in\mathbb{R}_+$) and convex combination ($d_m\in\mathbb{R}_+$ and $\sum_{m=1}^{M}d_m=1$). The kernel weighting coefficient d_m is defined as non-negative (a cone combination or a convex combination) to improve the interpretability of the discriminant function. The larger kernel weight coefficient, the greater importance of the kernel matrix. Non-negative weight coefficients combine the feature vector corresponding to each kernel matrix by mathematical operation. The multiple feature map $\phi(x)$ is composed of a set of feature maps $\{\sqrt{d_m}\phi_m(x),m=1,\cdots,M\}$, and its expression is described in[19]:

$$\phi(x)=\begin{pmatrix}\sqrt{d_1}\phi_1(x)\\ \sqrt{d_2}\phi_2(x)\\ \vdots\\ \sqrt{d_M}\phi_M(x)\end{pmatrix}. \quad (16)$$

Correspondingly, the inner product in the combined feature space equivalent to a linear weighted combination of kernel matrices [19]:

$$<\phi(x_i),\phi(x_j)>=\begin{pmatrix}\sqrt{d_1}\phi_1(x)\\ \sqrt{d_2}\phi_2(x)\\ \vdots\\ \sqrt{d_M}\phi_M(x)\end{pmatrix}^{\mathrm{T}}\begin{pmatrix}\sqrt{d_1}\phi_1(x)\\ \sqrt{d_2}\phi_2(x)\\ \vdots\\ \sqrt{d_M}\phi_M(x)\end{pmatrix}$$
$$=\sum_{m=1}^{M}d_mK_m(x_i,x_j). \quad (17)$$

Therefore, the corresponding multiple kernel learning discriminant function is:

$$f_{w,b,d}(x)=\sum_{m=1}^{M}<w_m,\sqrt{d_M}\phi_k(m)>+b. \quad (18)$$

Multiple kernel learning not only automatically selects kernel functions and hyperparameters, but also effectively fuses data from multiple different data sources. Generally, data from different data sources have different similarity metric (or distance metric) methods and require different kernel functions for processing. For example, images, sounds,

and text in video data need to be processed using different kernel functions. It is precisely because of these advantages of multiple kernels learning that it has received close attention from many researchers and has developed rapidly. However, multiple learning faces the problem of slow training, and it is difficult to extend to the processing of large-scale data. Therefore, fast and efficient training algorithms have always been a focus of the multiple kernel learning research [63].

2.4 Common kernel fusion methods in bioinformatics

However, a single-kernel-based classifier can only treat all features together, rather than treating them differently to take advantage of their characteristics. Suppose that there are M classes kernels, we get a series of kernel matrices $\{K_i\}_{i=1}^{m}$. Some researchers use the weighted linear fusion method to obtain the final kernel matrix $K = \sum_{i=1}^{m} d_i K_i,\ d_i \geqslant 0$, and use the optimization method to obtain the weighted coefficients of fusion. The weight reflects the role of the corresponding kernel in the process of prediction. Training weights, that is, find the best feature fusion method. Researchers need to pay more attention to finding cost-effective methods for fusing these different kernels computed from biological data. In this part, we introduce several combination methods for fusing kernels in bioinformatics. The first kind of model uses one kernel to combine all kinds of data. These kernels are combined by a weighted linear fusion method. The final kernel matrix K is obtained by $K = \sum_{i=1}^{m} \beta_i K_1(M_i),\ \beta_i \geqslant 0$. m means the number of datasets (Fig. 2A). The second model uses one kernel to combine several kinds of data. In this case, some types of data corresponding to the same kernel function, and some data corresponding to a unique kernel function. The final kernel matrix K is obtained by $K = \sum_{i=1}^{g} \beta_i K_i(M_i), \beta_i \geqslant 0$, and g means the number of kernel functions (Fig. 2B). In the third model, a type of data corresponds to a type of kernel function K_i. The final kernel matrix K is obtained by $K = \sum_{i=1}^{m} \beta_i K_i(M_i),\ \beta_i \geqslant 0$ means of the weighted coefficients of fusion (Fig. 2C). We show the framework of multiple kernel fusion in Fig. 2.

- Linear combination

Suppose there are M classes kernels, we get a series of kernel matrices $\{K_m\}_{m=1}^{M}$. Some researchers use the weighted linear fusion method to obtain the final kernel matrix $K = \sum_{m=1}^{M} \beta_m K_m$, $\beta_m \geqslant 0$, and use the optimization method to obtain the weighted coefficients of fusion, that is, find the best feature fusion method.

For improving the efficiency of kernel data fusion through the convex combination of kernel matrices, Lanckriet et al. used conic combinations for the SVM and showed the optimization problem converted to a QCQP [19]. Though the QCQP problem can be solved using the YAMIP toolbox, it can solve this problem only for a small number of data points and kernels.

Based on Lanckriet's work, Bach et al. [64] proposed a novel dual formulation called SKM of the QCQP as a second-order cone programming problem and solved the MY-regularized SKM by using sequential minimal optimization techniques. SKM is closely related to SVM, and the only difference is the choice of the norm of the inverse margin. SKM is just the multiple kernel problem proposed by Lanckriet.

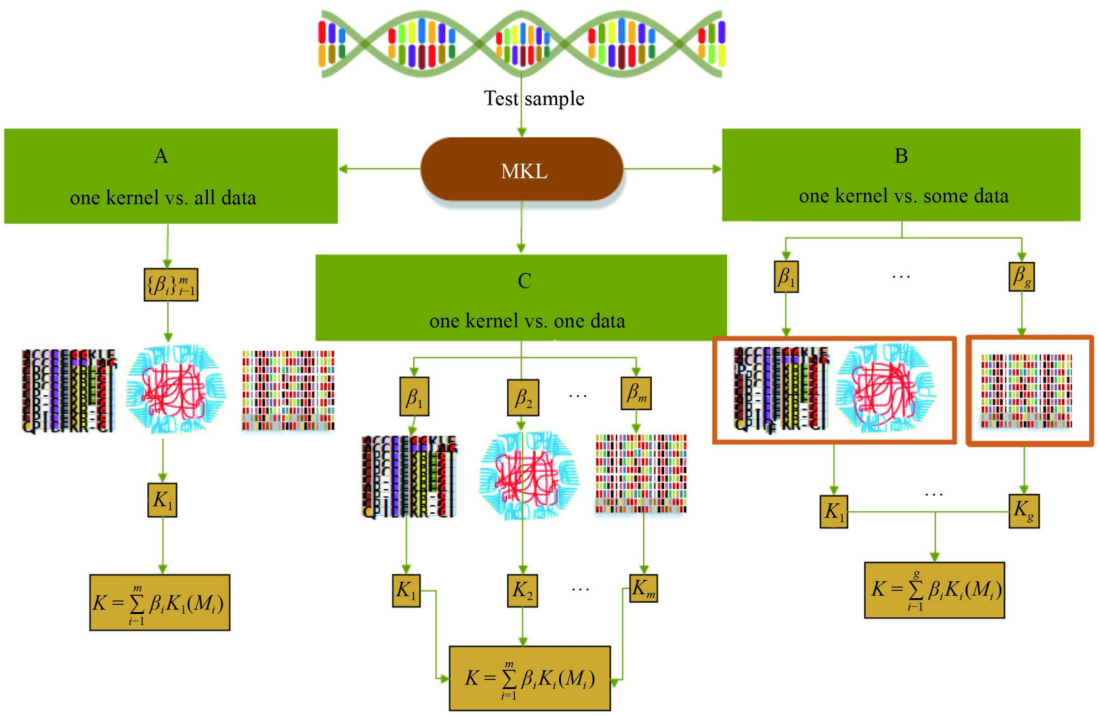


Fig. 2 The framework of multiple kernel fusion. A: One kernel vs. all data. All kinds of data use a kernel function, and combine these kernels by a weighted linear fusion method; B: One kernel vs. some data. In this case, some types of data corresponding to the same kernel function, and some data corresponding to a unique kernel function; C: One kernel vs. one data. In this case, a type of data corresponds to a type of kernel function

Sonnenburg et al. showed that the conic combination of kernel matrices for classification can be rewritten as a semi-infinite linear program [65], and the solution of the program can be obtained by SVM. To solve the large scale MKL problem, $\phi(x)$ is typically very sparse, though the feature space can be very high dimensional.

In short, the methods of linear combination in multiple kernel learning methods are to solve the problem of quadratic equations. By adding constraints and adding regulation terms and constraints to the objective function, the objective function is optimized to obtain the largest segmentation interval to achieve the best classification effect.

● Nonlinear combination

At present, many multiple kernel learning methods are devoted to the linear combination. However, these linear combination methods have limitations in some applications. For example, the linear combination methods cannot explain the physical meaning of the distance between two points on the Riemannian manifold appropriately. There is another type of combination method is geometric kernel data fusion, which begins at geodesic convexity. Geodesic convexity is the generalization of ordinary convexity to manifolds and metric spaces [66,67]. On Riemannian manifolds, geodesics are curves with zero acceleration, and they locally minimize the Riemannian distance between two points. The mean corresponding to Riemannian distance on all SPD matrices $P(n)$ is GM. The unique solution of GM $g(A_1,...,A_k)$ satisfied the non-linear matrix equation $\sum_{i=1}^{n}\log(K_i^{-1}K)=0$. The geodesic curve of K_i and K_j on the SPD manifold is $K_i\#_t K_j=(K_i^{-1/2}K_jK_i^{-1/2})^tK_i^{1/2}$.

Zakeri et al. proposed a kernel fusion method based on geometric mean using the geometric property of the SPD matrix [68]. Traditional linear MKL fusion methods have limitations in applications and may result in losing some useful edge information. The development of sequencing makes the number of biological sequences grow rapidly, while the process of protein mining structural information is relatively slow. But we can still get the structural information of the protein through the existing sequence information and machine learning methods. This work uses two models, is Karcher-KF and AGH-KF. Actually, the Karcher mean is obtained by searching the minimizer of an optimization problem, which is given as follows:

$$g(A_1,...,A_k)=\min_{X\in P_n}\sum_{i=1}^{k}\|\log(A_i^{-1/2}XA_i^{-1/2})\|_F^2. \quad (19)$$

AGH mean is considered as an approximation to the Karcher mean [69], and AGH means has a lower computing complexity compared with Karcher mean.

The GM method is a quite reasonable and effective way to fuse the SPD matrix. It uses the concept of Riemann manifold, putting two matrices as two points on the manifold. This type of fusion is more reasonable than the linear convex combination of Euclidean space. This method has achieved good results in protein folding recognition and homology detection. The matrix calculation of the GM method is invertible, which greatly simplifies the formula derivation of the algorithm. Besides geometric kernels, there are some custom fusion functions. In Wang's work [70], he combines multiple kernels by a kernel function rather than the common-used SVM predictor. The kernel function is defined as $K(Dr_APr_A,Dr_BPr_B)=S_{comp}(Dr_A,Dr_B)\times S_{geno}(Pr_A,Pr_B)$[70]. The symbol $S_{chem}(Dr_A,Dr_B)$ represent the structure similarity between drug Dr_A and Dr_B, the pharmacological similarity between drug Dr_A and Dr_B are represented by $S_{phar}(Dr_A,Dr_B)$ and $S_{ther}(Dr_A,Dr_B)$, and symbol $S_{geno}(Pr_A,Pr_B)$ is the sequence similarity between protein Pr_A and Pr_B. S_{comp} can be any one of $S_{chem}(Dr_A,Dr_B)$, $S_{phar}(Dr_A,Dr_B)$ and $S_{ther}(Dr_A,Dr_B)$ or their combination. Chem, Phar, Ther denotes the condition of $S_{comp}=S_{chem}$, $S_{comp}=S_{phar}$, $S_{comp}=S_{ther}$, and *ChemPhar* represents the condition of $S_{comp}=\max(S_{chem},S_{phar})$ and so on. Taken together, in the defined kernel function, two compound-protein pairs are similar only if the corresponding compounds and proteins are simultaneously judged to be similar by different types of data sources. We show linear and non-linear kernel fusion in Fig. 3.

3 Application and results

There are many classification and clustering problems in bioinformatics that draw on kernel fusion methods. The sample is first mapped to a point in the feature space through feature selection, and then an advanced machine learning algorithm is used to fit a separating hyperplane or a plurality of cluster centers of maximum margin. Multiple kernel learning methods play an important role in the prediction of anticancer drug response and drug-targeted interaction, the inference of protein-protein interaction, and the binding affinity between protein and peptide or mRNA and miRNA. This section will briefly describe some kernel fusion methods that are applied to protein, DNA, RNA, and binary networks.

3.1 Application of kernel fusion in protein sequence-related problems

Multiple kernel learning expresses the similarity of multiple features from different sources with different kernel functions and learns to fuse into a global similarity measure, which can effectively improve the classification effect of multiple feature fusion. In bioinformatics, protein secondary structure is the basis for tertiary structure prediction and protein function

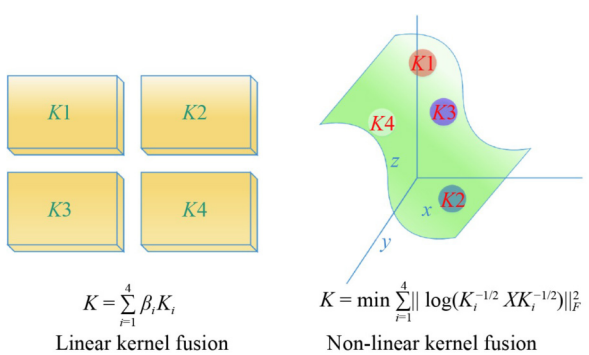


Fig. 3 Linear and non-linear multiple kernel fusion frameworks. Linear kernel fusion means that kernels are combined linearly. Non-linear kernel fusion means that kernels are combined nonlinearly

prediction and is the first step in many protein types of research. And protein homology detection is a core issue in bioinformatics. It is necessary to detect sequence similarity between proteins because sequence similarity means homology, and homology means functional similarity. In recent years, multiple kernel learning methods have played an important role in bioinformatics and have shown good performance in many applications [71−73].

Borgwardt K M's work [30] proposed to combine the random walk graph kernel, type kernel, length kernel, and node labels kernel into a protein graph kernel, which contains a similar sequence, structure, and interaction partners or phylogenetic profiles. The graph kernel measures the structure, sequence, and chemical similarity of two proteins. They first used a graphical kernel to determine the structural and sequence similarity between proteins. If they were similar, the similarity was measured by comparing the physicochemical properties of the protein. Combining different similarity measures into one graph kernel to distinguish the protein is enzymes or not. Besides, the removal of structural edges from the graphics kernel significantly reduces prediction accuracy, illustrating the need for structural information.

Liao and Noble [56] proposed a pairwise kernel, that is the SVM-Pairwise kernel, for remote protein homology detection. Ben and Noble [18] presented several kernels, that is the pairwise kernel, sequence kernel, and non-sequence kernel, to predict protein-protein interactions. They adopted summing methods to combine these kernels, considering the pairwise kernel's dimensionality. Though they using $K_p(\sum_i K_i)$ to mix features among different types of kernels. Also, they demonstrated that the SVM-Pairwise kernel is not only applicable to sequence data.

Filatov et al. present LZW-Kernel [57] for protein sequences classification. LZW-Kernel is a new convolution kernel, and it is based on code words identified with LZW universal text compressor. It is a one-step algorithm for backgrounds in big data applications. Their results show that LZW-based methods were the fastest in all experiments. It can also be applied in remote protein homology detection. Leslie et al. proposed the mismatch kernel in 2003 [14] and successfully applied it to protein sequence classification. It is consisting of expressing a sequence by a set of fixed-length (typically three to six amino acids long) blocks. The mismatch kernel between two sequences is the inner product between the fix-length blocks.

Since the structure corresponding to the amino acid is closely related to the surrounding amino acids, based on the primary structure of a protein, several features can be extracted, such as Position-Specific Scoring Matrix (PSSM), amino acid physicochemical properties, and folding tendency score. PSI-Blast is a multiple sequence alignment technique that obtains PSSM by comparing the target protein sequence to the protein in the database. Retention scores for all amino acids at each position are stored in the PSSM, and the amino acids with high evolutionary retention receive higher scores, whereas the lower scores are lower.

It is very meaningful to study proteins in bioinformatics. Effective classification of proteins, as well as a more detailed understanding of protein structure and finding the same remote for different proteins to explore their common property will help us better understand proteins. The multiple kernel learning method plays a powerful role in this process, and it solves many problems that could not be solved by projecting complex linear indivisible problems into feature space. For multiple kernel learning, finding optimal and accelerate fusion methods are the focus of research.

There are also more kernel fusion methods for bioinformatics, for example, multiplication, power, exponentiation, polynomial [74], and convex combinations [75], and some of them show better performance than linear combination in protein fold prediction. And more recent work proposed Nuclear-norm-Constrained MKL [76] which present outstanding results on protein fold prediction tasks.

3.2 Application of kernel fusion in DNA and RNA sequence-related problems of kernel fusion in protein sequence-related problems

The establishment of a functional RNA sequence modeling and analysis method based on random context-free grammar successfully simulated a typical RNA secondary structure. The RNA secondary structure was constructed for predicting the structure of RNA sequences. In fact, RNA structure is complex and non-linear and these traditional models do not adequately distinguish between member sequences and non-member sequences from the genomic sequence. Multiple kernel learning can effectively improve the classification effect of multiple feature fusion, Therefore, some researchers proposed to expresses the similarity of multiple features from different sources with different kernel functions. Some researchers construct some new kernels to predict RNA structure and have made a series of achievements [77].

Sakakibara et al. [78] proposed a new kernel function, the stem kernel, which uses SVMs to identify and detect functional RNA sequences. The stem kernel belongs to the string sequence kernel, which measures the similarity from the perspective of the RNA secondary structure. The stem kernel is used to distinguish between RNA family members and non-member members using SVMs. Also, the stem kernel can detect a remote homologous RNA family from the perspective of the secondary structure.

Karklin et al. proposed the marginalized kernel on RNA sequences which is represented as a labeled dual graph, but the marginalized kernel is not robust since the structural prediction is sensitive to boundaries. Navarin and Costa [79] proposed a kernel-based prediction system in 2017. They use a graphical encoding to preserve assumptions and extend to large scale data with representation and induction. The method is used for functional annotation of non-coding RNA and it improves the most advanced prediction methods.

Recently Brayet et al. [80] use multiple kernel fusion and SVM to predict piRNA. The work uses the Gaussian kernel to process the raw data. The major innovation is that the data is preprocessed before using the Gaussian kernel function. For example, the first column is added with uridine base information in the feature matrix of the sequence.

In this work, the researchers defined three kernels. For the

first kernel, each sequence is represented by a vector, which indicates the frequency of 32 k-mers (two 4-mers and thirty 5-mers) appearance. According to the presence of uridine base in the sequence, the information is marked as a feature in the first position of the sequence. Then the new 33-dimensional matrix is calculated by Gaussian kernel function. The second kernel is a new matrix generated from four-dimensional eigenvectors, which represent the distance from the sequence to the centromere and sub-telomeric regions. If the sequence is in the telomere or centromere region or not on the analyzed strand or chromosome, the value of the distance is infinite. When the sequence appears at different positions of the gene, four distances each take a minimum value as the distance value. Then use the Gaussian kernel to calculate. The third kernel takes the information of the position of the piRNA cluster on the chromosome into account. Neighbors are the k sequences closest to the target sequence in the training set, and then constructing a $(k+1)\times(k+1)$ matrix, including the distance of all sequences. These distances are then calculated by the Gaussian kernel function. After generating all the kernel matrices, the authors used LIBSVM [81] and SPG-GMKL [28] methods for kernel fusion experiments. The available code will be concluded in the last part.

3.3 Kernel fusion based on the application of bipartite networks

The enormous scale of data makes the whole biological activity analysis of pharmaceutical compounds difficult in practice, so it is necessary to use information technology simulation drug synthesis and prediction. According to the computer simulation of drug development and drug combination, the mechanical labor of the actual operation can be avoided, and the error caused by the manual operation can be reduced. The use of kernel methods to synthesize pharmaceutical compounds and conduct activity analysis and functional prediction, screening and analyzing mature models, and then implementing them is an efficient and feasible strategy. Kernel-based approaches have performed well in many applications, including inferring the effect of drugs on cancer cells [82] and elucidating drugs through drug-protein binding predictions [83]. The multiple kernel learning methods can integrate multiple physical and chemical properties and other data sources from a clinical record or genomics into the clinical operable prediction model, which is of great significance for clinical practice.

Wang et al. [70] collect multiple source information, such as drug pharmacological and therapeutic effects and structures. They give a kernel function for data fusion rather than the common SVM predictor. In the kernel function, the similarity of the corresponding compounds and protein is determined by the information of the multiple sources that make up the data. If the composite protein is similar only in one data type and judged to be different from other information sources, the composite protein is still not defined as similar. Only when multiple data sources are supported at the same time can they be defined as similar.

In 2012, Mehmet Gönen proposed a KBMF2K method [84] based on Bayesian formula to predict drug-targeted interaction networks. It could use a kernel component to express targets. Then he extended KBMF2K to the KBMF2MKL [85] in 2014. The KBMF2MKL decomposes the drug-target matrix and then multiplying the drug projected in this subspace by the target kernels. The kernels that are projected for each subspace are then assigned weights, and the resulting matrix is used for prediction. In 2016, Nascimento et al. proposed the KronRLS-MKL [86] for drug-target interaction prediction. The method uses bipartite networks to solve the drug-target interaction task.

After Nascimento's work, Cristianini et al. proposed a computational-experimental approach and PairwiseMKL [55] in 2017 and 2018 separately. The former studied kinase inhibitors and used a sophisticated kernel-based regression algorithm as a predictive model for clinical validation and prediction of drug-targeted interactions. The later inputs pairwise kernel and the data that constitutes the pairwise kernel is heterogeneous. PairwiseMKL integrates these different structures into the algorithm model through the MKL method. The kernel weights assigned to different kernels represent the prediction capabilities of corresponding data sources. The results show that gene expression, subsequent gene mutation, and methylation patterns contribute most to two nuclei eventually used to predict anticancer drug responses in genomics view. The PairwiseMKL method performs well in predicting the anticancer efficacy of drugs and target profiles of anticancer compounds.

4 Discussion and conclusions

In the study of bioinformatics, protein sequences and nucleic acid sequences are the main goals of the research. We need to mine useful information in these data, such as protein structure and function information and category information of nucleic acid sequences. Studying the above information plays an important role in disease prevention and drug development. Amino acids are always present in the form of a string, and their amino hydroxyl groups, as well as the side chains, bind to each other and fold in space to form different proteins. There are also some interactions between proteins and proteins. For example, if the depression of one protein is combined with the protrusion of another protein, the two proteins will not work, which is the principle of the drug. Let the drug combine with the broken protein so that the broken protein does not work. Therefore, we need to study the interaction between proteins. The protein consists of 20 different amino acids. We can think of them as strings of 20 different characters, then extract features according to the sequence, generate feature vectors, and then the set of proteins will become a matrix. One of the main challenges facing bioinformatics is the heterogeneity of information structure in logical data. As we are known proteins can be divided into seven categories. These seven categories can be divided into many layers, many families, according to some criteria such as function and structure. A total of more than 1,000 layers, which contain nearly 2,000 superfamilies, close to 4,000 families. Due to the diversity of data sources and the differences in data collection methods, there is a large amount of mixed heterogeneous data with a diversity of structure and

Table 2 Summary of 12 existing multiple kernel fusion methods in the literature

Methods	Year	URL	Ref.
Shogun(MKL-SIP)	2009	http://shogun-toolbox.org/	[87]
MKL-SMO	2010	http://research.microsoft.com/enus/um/people/manik/codus/code/SMO-MKL/download.html	[88]
SimpleMKL	2008	http://asi.insa-rouen.fr/enseignants/~arakoto/code/mklindex.html	[89]
Multi-Label MKL	2010	http://www.cse.msu.edu/~bucakser/software.html	[90]
Localized MKL	2008	http://users.ics.aalto.fi/gonen/	[91]
SPG-GMKL	2012	http://www.cs.cornell.edu/~ashesh/pubs/code/SPG-GMKL/download.html	[28]
GMKL	2009	http://manikvarma.org/code/GMKL/download.html	[27]
Geometric Mean	2014	http://people.cs.kuleuven.be/~raf.vandebril/homepage/software/geomean.php?menu=5/	[68]
pairwise MKL	2018	https://github.com/aalto-ics-kepaco	[55]
KronRLS-MKL	2016	https://www.cin.ufpe.br/~acan/kronrlsmkl/	[86]
KBMF2K	2012	http://users.ics.aalto.fi/gonen/kbmf2k	[84]
KBMF2MKL	2013	http://research.ics.aalto.fi/mi/software/kbmf/	[85]
SW	2010	http://morrislab.med.utoronto.ca/sara/SW/	[92]
TRAM	2013	http://lamda.nju.edu.cn/files/TRAM.zip	[93]
MKL-Sum	2009	http://www.public.asu.edu/~ltang9/code/mkl-multiple-label/	[94]
MKL-SA	2010	http://www.cse.msu.edu/bucakser/ML-MKL-SA.rar	[95]

numerical types. For example, there is a wide range of data in the biomedical field, including time series, stream data, and text types. Different types of data contain different spatial structure information. Some data are continuous, such as the parameters that describe the physical state of the patient: blood pressure, body temperature, blood sugar, and other medical indicators; some are discrete, such as gender, nationality, etc.

Multiple kernel learning can automatically select kernel functions and hyperparameters, and it can realize effectively fuses data from multiple different data sources. It is precisely because of these advantages of multiple kernels learning that it has received close attention from many researchers and has developed rapidly. The existing kernel fusion algorithms have been introduced and analyzed in the body and are not explained in detail in this section. Multiple learning faces the problem that it is difficult to extend to the processing of large-scale data. Therefore, fast and efficient training algorithms have always been the focus of multiple kernel learning research. We summarize the source code that has been publicly available in recent years in Table 2.

Although researchers have proposed and defined many types of kernel functions, in general, the more specially designed kernel functions, the worse their generalization performance. In comparison, such kernel functions are generally only specific. The data set performed well and was suspected of overfitting. In bioinformatics, the research object is very complex. For example, there are obvious hierarchical classification problems in protein classification, which has always been a challenge for researchers. Besides, the uncertainty of the enzyme data set, multi-label, sample imbalance, and other issues. The structure of the data set is complicated and the difference is large. In the face of these problems and challenges, we believe that the nuclear function will have more room for development.

Acknowledgements The work was supported by the National Natural Science Foundation of China (Grant Nos. 61922020, 61771331, 61902259).

References

1. Vapnik V N. An overview of statistical learning theory. IEEE Transactions on Neural Networks, 1999, 10(5): 988–999
2. Schölkopf B, Smola A, Müller K R. Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation, 1998, 10(5): 1299–1319
3. Nello C, John S T. An introduction to support vector machines and other kernel-based learning methods. Cambridge university press, 2000
4. Mercer J. XVI. Functions of positive and negative type, and their connection the theory of integral equations. Philosophical Transactions of the Royal Society of London, 1909, 209(441−458): 415–446
5. Vapnik V. Statistical learning theory. New York: Wiley, 1998
6. Mika S, Ratsch G, Weston J, Scholkopf B, Mullers K R. Fisher discriminant analysis with kernels. In: Proceedings of the 1999 IEEE Signal Processing Society Workshop. 1999, 41−48
7. Cristianini N, John S T, Elisseeff A, Kandola J S. On kernel-target alignment. Advances in Neural Information Processing Systems, 2002, 367−373
8. Song L, Kolar M, Xing E P. KELLER: estimating time-varying interactions between genes. Bioinformatics, 2009, 25(12): i128–i136
9. Song L, Bedo J, Borgwardt K M, Gretton A, Smola A. Gene selection via the BAHSIC family of algorithms. Bioinformatics, 2007, 23(13): i490–i498
10. Kato T, Tsuda K, Asai K. Selective integration of multiple biological data for supervised network inference. Bioinformatics, 2005, 21(10): 2488–2495
11. Donini M, Monteiro J M, Pontil M, Shawe-Taylor J, Mourao-Miranda J. A multimodal multiple kernel learning approach to Alzheimer's disease detection. In: Proceedings of the 26th IEEE International workshop on Machine Learning for Signal Processing. 2016, 1−6
12. Gu Y, Liu T, Jia X, Benediktsson J A, Chanussot J. Nonlinear multiple kernel learning with multiple-structure-element extended morphological profiles for hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing, 2016, 54(6): 3235–3247
13. Han L, Yue Z, Guo X. Image segmentation implementation based on FPGA and SVM. In: Proceedings of the 3rd International Conference on Control, Automation and Robotics. 2017, 405−409
14. Leslie C S, Eskin E, Cohen A, Weston J, Noble W S. Mismatch string kernels for discriminative protein classification. Bioinformatics, 2004, 20(4): 467–476
15. Tsuda K, Noble W S. Learning kernels from biological networks by maximizing entropy. Bioinformatics, 2004, 20(suppl_1): i326–i333
16. Chou K C. Pseudo amino acid composition and its applications in

bioinformatics, proteomics and system biology. Current Proteomics, 2009, 6(4): 262–274
17. Swamidass S J, Chen J, Bruand J, Phung P, Ralaivola L, Baldi P. Kernels for small molecules and the prediction of mutagenicity, toxicity and anti-cancer activity. Bioinformatics, 2005, 21(suppl_1): i359–i368
18. Asa B H, Noble W S. Kernel methods for predicting protein–protein interactions. Bioinformatics, 2005, 21(suppl_1): i38–i46
19. Lanckriet G R, Cristianini N, Bartlett P, Ghaoui L E, Jordan M I. Learning the kernel matrix with semidefinite programming. Journal of Machine Learning Research, 2004, 5(Jan): 27–72
20. Lanckriet G R, Tijl D B, Cristianini N, Jordan M I, Noble W S. A statistical framework for genomic data fusion. Bioinformatics, 2004, 20(16): 2626–2635
21. Lanckriet G R, Deng M, Cristianini N, Jordan M I, Noble W S. Kernel-based data fusion and its application to protein function prediction in yeast. Biocomputing: World Scientific, 2003
22. Bach F R, Thibaux R, Jordan M I. Computing regularization paths for learning multiple kernels. Advances in Neural Information Processing Systems, 2005, 73−80
23. Sonnenburg S, Rätsch G, Schäfer C. A general and efficient multiple kernel learning algorithm. Advances in Neural Information Processing Systems, 2006, 1273−1280
24. Jebara T. Multi-task feature and kernel selection for SVMs. In: Proceedings of the 21th International Conference on Machine Learning. 2004, 55
25. Lewis D P, Jebara T, Noble W S. Support vector machine learning from heterogeneous data: an empirical analysis using protein sequence and structure. Bioinformatics, 2006, 22(22): 2753–2760
26. Rätsch G, Sonnenburg S, Schäfer C. Learning interpretable SVMs for biological sequence classification. BMC Bioinformatics, 2006, 7(1): S9
27. Varma M, Babu B R. More generality in efficient multiple kernel learning. In: Proceedings of the 26th Annual International Conference on Machine Learning. 2009, 1065−1072
28. Jain A, Vishwanathan S V, Varma M. SPF-GMKL: generalized multiple kernel learning with a million kernels. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2012, 750−758
29. Wu P, Hoi S C, Zhao P, Miao C, Liu Z-Y. Online multi-modal distance metric learning with application to image retrieval. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(2): 454–467
30. Borgwardt K M, Ong C S, Schönauer S, Vishwanathan S, Smola A J, Kriegel H-P. Protein function prediction via graph kernels. Bioinformatics, 2005, 21(suppl_1), i47−i56
31. Zien A, Ong C S. An automated combination of sequence motif kernels for predicting protein subcellular localization, 2006
32. Damoulas T, Girolami M A. Probabilistic multi-class multi-kernel learning: on protein fold recognition and remote homology detection. Bioinformatics, 2008, 24(10): 1264–1270
33. Vert J P, Qiu J, Noble W S. A new pairwise kernel for biological network inference with support vector machines. BMC Bioinformatics, 2007, 8(10): 1–10
34. Vapnik V. The nature of statistical learning theory. Springer science & business media, 2013
35. Aronszajn N. Theory of reproducing kernels. Transactions of the American Mathematical Society, 1950, 68(3): 337–404
36. Boser B E, Guyon I M, Vapnik V N. A training algorithm for optimal margin classifiers. In: Proceedings of the 5th annual workshop on Computational learning theory. 1992, 144−152
37. Boyd S, Vandenberghe L. Convex optimization. Cambridge university press, 2004
38. Leslie C, Eskin E, Noble W S. The spectrum kernel: A string kernel for SVM protein classification. Biocomputing 2002: World Scientific, 2001
39. Saigo H, Vert J P, Ueda N, Akutsu T. Protein homology detection using string alignment kernels. Bioinformatics, 2004, 20(11): 1682–1689
40. Rätsch G, Sonnenburg S. Accurate splice site prediction for caenorhabditis elegans. Computational Molecular Biology, 2004, 277–298
41. Asa B H, Brutlag D. Remote homology detection: a motif based approach. Bioinformatics, 2003, 19(suppl_1): i26–i33
42. Nevill M, Craig G, Wu T D, Brutlag D L. Highly specific protein sequence motifs for genome analysis. In: Proceedings of the National Academy of Sciences. 1998, 95(11): 5865–5871
43. Huang J Y, Brutlag D L. The EMOTIF database. Nucleic Acids Research, 2001, 29(1): 202–204
44. Kuang R, WANG K, Wang K, Siddiqi M, Freund Y, Leslie C. Profile-based string kernels for remote homology detection and motif extraction. Journal of Bioinformatics and Computational Biology, 2005, 3(3): 527–550
45. Rätsch G, Sonnenburg S, Srinivasan J, Witte H, Müller K R, Sommer R J, Schölkopf B. Improving the Caenorhabditis elegans genome annotation using machine learning. PLoS Computational Biology, 2007, 3(2): e20
46. Schweikert G, Zien A, Zeller G, Behr J, Dieterich C, Ong C S, Philips P, De Bona F, Hartmann L, Bohlen A. mGene: accurate SVM-based gene finding with an application to nematode genomes. Genome Research, 2009, 19(11): 2133–2143
47. Jacob L, Vert J P. Efficient peptide–MHC-I binding prediction for alleles with few known binders. Bioinformatics, 2007, 24(3): 358–366
48. Röttig M, Rausch C, Kohlbacher O. Combining structure and sequence information allows automated prediction of substrate specificities within enzyme families. PLoS Computational Biology, 2010, 6(1): e1000636
49. Teramoto R, Aoki M, Kimura T, Kanaoka M. Prediction of siRNA functionality using generalized string kernel and support vector machine. FEBS letters, 2005, 579(13): 2878–2882
50. Kuksa P, Qi Y, Bai B, Collobert R, Weston J, Pavlovic V, Ning X. Semi-supervised abstraction-augmented string kernel for multi-level bio-relation extraction. Joint European Conference on Machine Learning and Knowledge Discovery in Databases: Springer, 2010, 128−144
51. Leslie C, Eskin E, Weston J, Noble W S. Mismatch string kernels for SVM protein classification. Advances in Neural Information Processing Systems, 2003, 1441−1448
52. Weston J, Leslie C, Ie E, Zhou D, Elisseeff A, Noble W S. Semi-supervised protein classification using cluster kernels. Bioinformatics, 2005, 21(15): 3241–3247
53. Kuksa P, Huang P H, Pavlovic V. Scalable algorithms for string kernels with inexact matching. Advances in Neural Information Processing Systems, 2008, 21, 881−888
54. Leslie C, Kuang R, Bennett K. Fast string kernels using inexact matching for protein sequences. Journal of Machine Learning Research, 2004, 5(9)
55. Cichonska A, Pahikkala T, Szedmak S, Julkunen H, Airola A, Heinonen M, Aittokallio T, Rousu J. Learning with multiple pairwise kernels for drug bioactivity prediction. Bioinformatics, 2018, 34(13): i509–i518
56. Liao L, Noble W S. Combining pairwise sequence similarity and

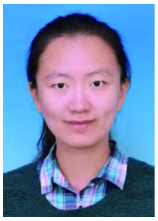
support vector machines for remote protein homology detection. In: Proceedings of the Sixth Annual International Conference on Computational Biology. 2002, 225−232

57. Filatov G, Bauwens B, Attila K F. LZW-Kernel: fast kernel utilizing variable length code blocks from LZW compressors for protein sequence classification. Bioinformatics, 2018, 34(19): 3281–3288
58. Smith T F, Waterman M S. Identification of common molecular subsequences. Journal of Molecular Biology, 1981, 147(1): 195–197
59. Altschul S F, Gish W, Miller W, Myers E W, Lipman D J. Basic local alignment search tool. Journal of Molecular Biology, 1990, 215(3): 403–410
60. Vert J P, Saigo H, Akutsu T. Local alignment kernels for biological sequences. Kernel Methods in Computational Biology, 2004, 131−154
61. Jaakkola T, Diekhans M, Haussler D. A discriminative framework for detecting remote protein homologies. Journal of Computational Biology, 2000, 7(1−2), 95−114
62. Gönen M, Alpaydın E. Multiple kernel learning algorithms. Journal of Machine Learning Research, 2011, 12: 2211–2268
63. Bucak S S, Jin R, Jain A K. Multiple kernel learning for visual object recognition: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, 36(7): 1354–1369
64. Bach F R, Lanckriet G R, Jordan M I. Multiple kernel learning, conic duality, and the SMO algorithm. In: Proceedings of the 21th International Conference on Machine Learning. 2004, 6
65. Sonnenburg S, Rätsch G, Schäfer C, Schölkopf B. Large scale multiple kernel learning. Journal of Machine Learning Research, 2006, 7: 1531–1565
66. Papadopoulos A. Metric spaces, convexity and nonpositive curvature. European Mathematical Society, 2005
67. Rapcsak T. Geodesic convexity in nonlinear optimization. Journal of Optimization Theory and Applications, 1991, 69(1): 169–183
68. Zakeri P, Jeuris B, Vandebril R, Moreau Y. Protein fold recognition using geometric kernel data fusion. Bioinformatics, 2014, 30(13): 1850−1857
69. Jeuris B, Vandebril R, Vandereycken B. A survey and comparison of contemporary algorithms for computing the matrix geometric mean. Electronic Transactions on Numerical Analysis, 2012, 39(ARTICLE): 379–402
70. Wang Y C, Zhang C H, Deng N Y, Wang Y. Kernel-based data fusion improves the drug–protein interaction prediction. Computational Biology and Chemistry, 2011, 35(6): 353–362
71. Yu G, Rangwala H, Domeniconi C, Zhang G, Zhang Z. Protein Function Prediction by Integrating Multiple Kernels. In: Proceedings of Twenty-Third International Joint Conference on Artificial Intelligence. 2013
72. Yu G, Rangwala H, Domeniconi C, Zhang G, Zhang Z. Predicting Protein Function using Multiple Kernels. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2015, 12(1): 219–233
73. Yu G, Fu G, Wang J, Zhu H. Predicting Protein Function via Semantic Integration of Multiple Networks. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2016, 13(2): 220–232
74. Cortes Corinna M M, Afshin Rostamizadeh. Learning non-linear combinations of kernels. Advances in neural information processing systems, 2009
75. Kloft M, Brefeld U, Sonnenburg S, Zien A. Non-sparse regularization and efficient training with multiple kernels. 2010, arXiv preprint arXiv: 1003.0079 2010, 186: 189−190
76. Eli M, Kisilev P. Nuc-mkl: A convex approach to non linear multiple kernel learning. Artificial Intelligence and Statistics, 2016, 610−619
77. Wilson C M, Li K, Yu X, Kuan P F, Wang X. Multiple-kernel learning for genomic data mining and prediction. BMC Bioinformatics, 2019, 20(1): 1–7
78. Sakakibara Y, Popendorf K, Ogawa N, Asai K, Sato K. Stem kernels for RNA sequence analyses. Journal of Bioinformatics and Computational Biology, 2007, 5(05): 1103–1122
79. Navarin N, Costa F. An efficient graph kernel method for non-coding RNA functional prediction. Bioinformatics, 2017, 33(17): 2642–2650
80. Brayet J, Zehraoui F, Laurence J L, Israeli D, Tahi F. Towards a piRNA prediction using multiple kernel fusion and support vector machine. Bioinformatics, 2014, 30(17): i364–i370
81. Chang C C, Lin C J. LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2011, 2(3): 27
82. Costello J C, Heiser L M, Georgii E, Gönen M, Menden M P, Wang N J, Bansal M, Hintsanen P, Khan S A, Mpindi J P. A community effort to assess and improve drug sensitivity prediction algorithms. Nature Biotechnology, 2014, 32(12): 1202
83. Cichonska A, Ravikumar B, Parri E, Timonen S, Pahikkala T, Airola A, Wennerberg K, Rousu J, Aittokallio T. Computational-experimental approach to drug-target interaction mapping: A case study on kinase inhibitors. PLoS Computational Biology, 2017, 13(8): e1005678
84. Gönen M. Predicting drug–target interactions from chemical and genomic kernels using Bayesian matrix factorization. Bioinformatics, 2012, 28(18): 2304–2310
85. Gönen M, Khan S, Kaski S. Kernelized Bayesian matrix factorization. International Conference on Machine Learning, 2013, 864−872
86. Nascimento A C, Prudêncio R B, Costa I G. A multiple kernel learning algorithm for drug-target interaction prediction. BMC Bioinformatics, 2016, 17(1): 46
87. Kloft M, Brefeld U, Laskov P, Müller K-R, Zien A, Sonnenburg S. Efficient and accurate lp-norm multiple kernel learning. Advances in Neural Information Processing Systems, 2009, 997−1005
88. Sun Z, Ampornpunt N, Varma M, Vishwanathan S. Multiple kernel learning and the SMO algorithm. Advances in Neural Information Processing Systems, 2010, 2361−2369
89. Rakotomamonjy A, Bach F R, Canu S, Grandvalet Y. SimpleMKL. Journal of Machine Learning Research, 2008, 9: 2491–2521
90. Bucak S, Jin R, Jain A. Multi-label multiple kernel learning by stochastic approximation: Application to visual object recognition. Advances in Neural Information Processing Systems, 2010, 325−333
91. Gönen M, Alpaydin E. Localized multiple kernel learning. In: Proceedings of the 25th International Conference on Machine Learning. 2008, 352−359
92. Mostafavi S, Morris Q. Fast integration of heterogeneous data sources for predict-ing gene function with limited annotation. Bioinformatics, 2010, 26(14): 1759–1765
93. Kong X, Ng M K, Zhou Z-H. Trans-ductive multi-label learning via label set propagation. IEEE Transactions on Knowledge and Data Engineering, 2013, 25(3): 704–719
94. Tang L, Chen J, Ye J. On multiplekernel learning with multiple labels. In: Proceedings of Twenty-First International Joint Conference on Artificial Intelligence. 2009, 1255–1260
95. Bucak S, Jin R, Jain A. Multi-label multiple kernel learning by stochastic approximation:Application to visual object recognition.

Advances in Neural Information Processing Systems, 2010, 24: 325–333

Ren Qi received the MS degrees in computer technology from the College of Intelligence and Computing, Tianjin University, China in 2018 and she is a doctoral student in College of Intelligence and Computing from Tianjin University, China in 2018.

Her research interests include machine learning, metric learning, and bioinformatics. Several related works have been published by IJCAI, Journal of software.



Fei Guo received the BSc and the PhD degrees in School of Computer Science and Technology, Shandong University, China in 2007 and 2012, respectively. She is currently an assistant professor in the College of Intelligence and Computing, Tianjin University, China.

Her research interests include bioinformatics, algorithms, machine learning. Several related works have been published by Briefings in Bioinformatics, Bioinformatics, IEEE/ACM Transactions on Computational Biology and Bioinformatics, etc. She is the PC member of APBC-2016, APBC-2017, and reviewer of BMC Bioinformatics, BMC System Biology, IEEE TCBB, PLoS ONE, etc.

Quan Zou (M'13-SM'17) received the BSc, MSc and the PhD degrees in computer science from Harbin Institute of Technology, China in 2004, 2007 and 2009, respectively. He worked in Xiamen University and Tianjin University, China, from 2009 to 2018 as an assistant professor, associate professor and professor. He is currently a professor in the Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, China.

His research is in the areas of bioinformatics, machine learning and parallel computing. Several related works have been published by Science, Briefings in Bioinformatics, Bioinformatics, IEEE/ACM Transactions on Computational Biology and Bioinformatics, etc. Google scholar showed that his more than 100 papers have been cited more than 5000 times. He is the editor-in-chief of Current Bioinformatics, associate editor of IEEE Access, and the editor board member of Computers in Biology and Medicine, Genes, Scientific Reports, etc. He was selected as one of the Clarivate Analytics Highly Cited Researchers in 2018 and 2021.