frontiers | Frontiers in Genetics

ORIGINAL RESEARCH
published: 13 July 2022
doi: 10.3389/fgene.2022.960388

Identification of Vesicle Transport Proteins *via* Hypergraph Regularized K-Local Hyperplane Distance Nearest Neighbour Model

Rui Fan[1,2†], *Bing Suo*[3†] *and Yijie Ding*[2*]

[1]Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu, China, [2]Yangtze Delta Region Institute (Quzhou), University of Electronic Science and Technology of China, Quzhou, China, [3]Beidahuang Industry Group General Hospital, Harbin, China

The prediction of protein function is a common topic in the field of bioinformatics. In recent years, advances in machine learning have inspired a growing number of algorithms for predicting protein function. A large number of parameters and fairly complex neural networks are often used to improve the prediction performance, an approach that is time-consuming and costly. In this study, we leveraged traditional features and machine learning classifiers to boost the performance of vesicle transport protein identification and make the prediction process faster. We adopt the pseudo position-specific scoring matrix (PsePSSM) feature and our proposed new classifier hypergraph regularized k-local hyperplane distance nearest neighbour (HG-HKNN) to classify vesicular transport proteins. We address dataset imbalances with random undersampling. The results show that our strategy has an area under the receiver operating characteristic curve (AUC) of 0.870 and a Matthews correlation coefficient (MCC) of 0.53 on the benchmark dataset, outperforming all state-of-the-art methods on the same dataset, and other metrics of our model are also comparable to existing methods.

Keywords: transport proteins, protein function prediction, hypergraph learning, local hyperplane, membrane proteins

OPEN ACCESS

Edited by:
Zhibin Lv,
Sichuan university, China

Reviewed by:
Changli Feng,
Taishan University, China
Liangzhen Jiang,
Chengdu University, China

****Correspondence:***
Yijie Ding
wuxi_dyj@csj.uestc.edu.cn

†These authors have contributed equally to this work

Specialty section:
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

Received: *02 June 2022*
Accepted: *22 June 2022*
Published: *13 July 2022*

Citation:
Fan R, Suo B and Ding Y (2022) Identification of Vesicle Transport Proteins via Hypergraph Regularized K-Local Hyperplane Distance Nearest Neighbour Model. Front. Genet. 13:960388. doi: 10.3389/fgene.2022.960388

1 INTRODUCTION

Proteins are the basis of most life activities and perform important functions in different biochemical reactions. Proteins with different amino acid sequences and folding patterns have different functions. Understanding the factors that influence protein function has practical biological implications. Therefore, protein function prediction has been an important topic since the birth of bioinformatics. In recent years, machine learning-based protein function prediction methods have been widely used in many studies (Shen et al., 2019; Zhang J. et al., 2021; Zulfiqar et al., 2021; Ding et al., 2022b; Zhang et al., 2022), such as drug discovery (Ding et al., 2020c; Chen et al., 2021; Song et al., 2021; Xiong et al., 2021), protein gene ontology (Hong et al., 2020b; Zhang W. et al., 2021), DNA-binding proteins (Zou et al., 2021), enzyme proteins (Feehan et al., 2021; Jin et al., 2021), and protein subcellular localization (Ding et al., 2020b; Su et al., 2021; Wang et al., 2021; Zeng et al., 2022). In this study, we propose a novel method to identify vesicular transporters with machine learning.

Vesicular transport proteins are membrane proteins. The cell membrane separates the cell's internal environment from the outside and controls the transport of substances into and out of the

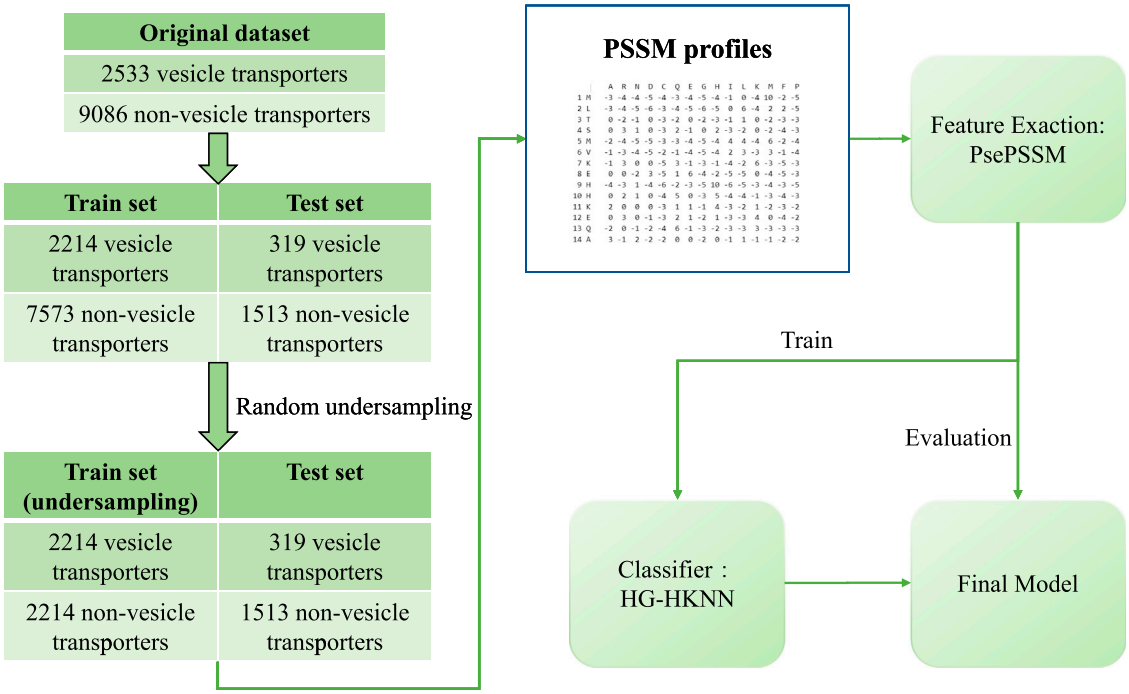


FIGURE 1 | Flowchart of our model.

cell. Different substances enter and leave cells in different ways, and the transport of macromolecular substances is called vesicular transport. In vesicular transport, cells first surround substances and form vesicles. Vesicles move within cells and release their contents through vesicle rupture or membrane fusion. The process of vesicle transport exists widely in life activities. Vesicular transport proteins play an important role in vesicle transport by regulating the interactions of specific molecules with the vesicle membrane. In biology, there have been many studies on vesicular transport proteins, such as (Cheret et al., 2021; Li et al., 2021; Fu T. et al., 2022). Many human diseases are associated with abnormal vesicle transport proteins, such as those described in (Buck et al., 2021; Mazere et al., 2021; Zhou et al., 2022).

With the development of protein sequencing technology, an increasing number of vesicle transport protein sequences have been discovered. The need to rapidly identify vesicle transporter protein sequences conflicts with traditional experimental techniques, which are costly and time-consuming. Therefore, it is imperative to develop a fast and efficient computational method. To date, there have been few studies on the computational identification of vesicle transport proteins.

Computational identification of protein, RNA and DNA sequences has similar steps, and their processes can be described as two steps of feature extraction and classification. In 2019, Le et al. proposed a method (Vesicular-GRU) to identify vesicle transporters using position-specific scoring matrix (PSSM) features and a neural network classifier based on a convolutional neural network (CNN) and gated recurrent unit (GRU) and released the dataset used in their study (Le et al., 2019). In 2020, Tao et al. (Tao et al., 2020) attempted to classify vesicular transport proteins with fewer feature dimensions. Their model used the composition part of the method of composition, transition, and distribution (CTDC) features and a support vector machine (SVM) classifier. After dimensionality reduction with the Maximum Relevance Maximum Distance (MRMD) method, they obtained a comparatively satisfactory accuracy with fewer feature dimensions on the Le et al. dataset.

In our study, we propose a new model to identify vesicular transporters using pseudo position-specific scoring matrix (PsePSSM) features and a classifier called hypergraph regularized k-local hyperplane distance nearest neighbour (HG-HKNN). The main contributions of our work are as follows: 1) a better identification model of vesicle transport protein, with fewer feature dimensions and better results than the state-of-the-art model; and 2) a classifier called HG-HKNN that combines hypergraph learning (Zhou et al., 2006; Ding et al., 2020a) with k-local hyperplane distance nearest neighbours (HKNN) (Vincent and Bengio, 2001; Liu et al., 2021). The flowchart of our study is illustrated in **Figure 1**.

2 MATERIALS AND METHODS

2.1 Dataset

The dataset we use to build and evaluate the model is the benchmark dataset released by Le et al. (Le et al., 2019). In the construction of the benchmark dataset, experimentally validated vesicular transport proteins were screened from the universal protein (UniProt) database (Consortium, 2019) and the gene ontology (GO) database (Consortium, 2004).

TABLE 1 | Details of the dataset used in our study.

	Original	Train Set	Train Set (RUS)	Test Set
Vesicular transport	2533	2214	2214	319
Non-vesicular transport	9086	7573	2214	1513

For the positive dataset, the authors collected protein sequences by searching the UniProt database for the keyword "vesicular transport" or the gene ontology term "vesicular transport". Likewise, for the negative dataset, the authors collected a set of universal protein (membrane protein) sequences and excluded vesicular transporters from them. Next, protein sequences annotated by biological experiments were selected in the original dataset, and all protein sequences that were not validated experimentally were filtered out. The authors then eliminated homologous sequences on the positive and negative datasets, respectively, with a 30% cut-off level by the basic local alignment search tool (BLAST) clustering (Johnson et al., 2008). The BLAST clustering ensures that any two sequences in the dataset have less than 30% pairwise sequence similarity. Finally, protein sequences with noncanonical amino acids (X, U, B, Z) were removed from the dataset.

The benchmark dataset contains 2533 vesicular transport proteins and 9086 non-vesicular transport proteins, and the dataset is divided into a training set and a test set. The training set consists of 2144 vesicular transporters and 7573 non-vesicular transporters, and the test set consists of 319 vesicular transporters and 1513 non-vesicular transporters. We perform random undersampling (RUS) on the training set to balance the proportions of positive and negative samples. In random undersampling, we randomly select a sample from the class with more samples in the training set to represent its class, and repeat until there are the same number of vesicular transport proteins and non-vesicular transport proteins in the training set. The randomly undersampled training set has 2214 positive samples and 2214 negative samples. The details of the dataset are listed in **Table 1**.

2.2 Feature Extraction

The feature type we use is PsePSSM (Chou and Shen, 2007), and the PSSM profile used to build PsePSSM is directly downloaded from the open-source data of Le et al. (Le et al., 2019). The authors of (Le et al., 2019) constructed these PSSM profiles by searching all sequences one by one in the non-redundant (NR) database with BLAST software. The PSSM matrix is an L*20 matrix similar to the following formula (Zhu et al., 2019). Each PSSM matrix corresponds to a protein sequence.

$$\boldsymbol{P}_{\mathbf{PSSM}} = \begin{bmatrix} \mathbb{E}_{1\to1} & \mathbb{E}_{1\to2} & \cdots & \mathbb{E}_{1\to20} \\ \mathbb{E}_{2\to1} & \mathbb{E}_{2\to2} & \cdots & \mathbb{E}_{2\to20} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbb{E}_{\boldsymbol{i}\to1} & \mathbb{E}_{\boldsymbol{i}\to2} & \cdots & \mathbb{E}_{\boldsymbol{i}\to20} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbb{E}_{\boldsymbol{L}\to1} & \mathbb{E}_{\boldsymbol{L}\to2} & \cdots & \mathbb{E}_{\boldsymbol{L}\to20} \end{bmatrix}. \tag{1}$$

In this formula, L is the length of the protein sequence. $\mathrm{E}_{i\to j}$ represents the relationship between the amino acid at position i of the protein sequence and the amino acid of type j in the homologous sequence. j is the amino acid type number ranging from 1 to 20. The PSSM matrix contains the position-specific frequency information of amino acids in the protein homologous sequences, which is used to decode the evolutionary information of proteins. Compared with other protein information (such as amino acid frequency and physicochemical properties), the PSSM matrix of proteins not only contains the information of the proteins in the dataset but also contains the motif information of the protein homologous sequences in the NR database. However, the dimension of the PSSM matrix is too large, so further PsePSSM feature extraction is required.

The PsePSSM feature we use is a $(\xi + 1)$*20 dimension feature, which can be calculated with this formula:

$$\boldsymbol{P}^{\xi}_{\mathrm{PsePSSM}} = \left[\bar{\mathbb{E}}_1 \cdots \bar{\mathbb{E}}_{20} G^1_1 \cdots G^1_{20} \cdots G^{\xi}_1 \cdots G^{\xi}_{20}\right]^{\boldsymbol{T}}. \tag{2}$$

where $\bar{\mathrm{E}}_j$ is the average value of each column of the PSSM matrix, and the calculation of G^{ξ}_j can be expressed by the following formula:

$$G^{\xi}_j = \frac{1}{L-\xi}\sum_{i=1}^{L-\xi}\left[\mathbb{E}_{i\to j} - \mathbb{E}_{(i+\xi)\to j}\right]^2 \quad \left(j = 1, 2, \cdots, 20; \xi < L\right). \tag{3}$$

G^{ξ}_j is the correlation factor obtained by coupling the ξ th-most contiguous PSSM scores along the protein chain with amino acid type j. Clearly, $\bar{\mathrm{E}}_j$ and G^0_j are the same. Note that the maximum value of ξ must be less than the length of the shortest protein sequence in the benchmark dataset. The value of ξ we choose is 6, so $\boldsymbol{P}^{\xi}_{PsePSSM}$ is a feature vector with 140 dimensions. When ξ increases, the evaluation metric first increases and then decreases and reaches the maximum value when ξ is 6.

2.3 Method for Classification

The hypergraph regularized k-local hyperplane distance nearest neighbour model (HG-HKNN) is a new classifier that combines the k-local hyperplane distance nearest neighbour algorithm (HKNN) and hypergraph learning.

2.3.1 HKNN

In the HKNN (Vincent and Bengio, 2001) workflow, multiple hyperplanes are constructed first, each hyperplane corresponds to a class in the training set, and the hyperplane is constructed by the k samples of the same class that is closest to the test sample. Then, the HKNN predicts the class of the test sample by comparing the distance between the test sample and the hyperplanes and assigns the test sample to the class corresponding to the nearest hyperplane (Ding et al., 2022c). **Figure 2** shows a sketch of an HKNN, where sample x obtains its class by comparing the distances to hyperplane 1 and hyperplane 2.

In class c, when x represents the test sample, the hyperplane can be expressed as the following formula:

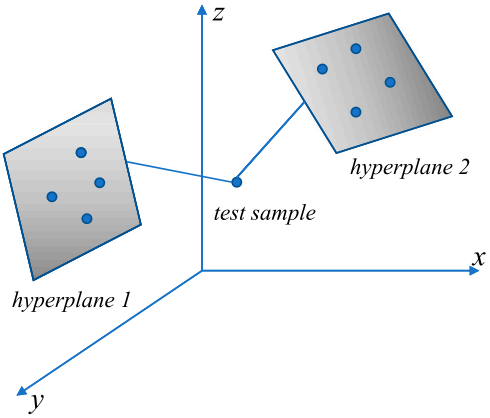


FIGURE 2 | Sketch of an HKNN.

$$LH_k^c(x) = \left\{ p^c \mid p^c = \bar{N}^c + \sum_{i=1}^{k} \alpha_i^c V_i^c, \alpha_{1\ldots k}^c \in R^k \right\}. \quad (4)$$

where k means that k nearest neighbour samples are taken to construct the hyperplane, and the i -th sample in class c can be expressed as N_i^c (i from 1 to k). Let $\bar{N}^c$ represent the centre of N_i^c, and let $V_i^c = N_i^c - \bar{N}^c$, where α_i^c is an undetermined parameter; then, p^c is a point on this hyperplane.

The mean squared distance of the test sample x to each hyperplane can be expressed as follows:

$$\left(LH_k^c(x)\right)^2 = \left\| x - \bar{N}^c - \sum_{i=1}^{k} \alpha_i^c V_i^c \right\|^2 + \lambda \sum_{i=1}^{k} \left(\alpha_i^c\right)^2. \quad (5)$$

where λ is the regularization parameter of α_i^c, which is used to reduce the complexity of the model. α^c is obtained by minimizing the distance. Finally, the classification result of the HKNN can be judged by the following formula:

$$c = argmin_c \left\| x - \bar{N}^c - \sum_{i=1}^{k} \alpha_i^c V_i^c \right\|^2. \quad (6)$$

HKNN has relatively good performance on unbalanced datasets because the same number of samples are selected in each class. However, since the distribution of samples cannot be fully expressed by a hyperplane, the performance of the HKNN is disturbed by the distribution of samples.

2.3.2 Hypergraph Learning

In machine learning, we can express the similarity between two samples by calculating the inner product of the features of the two samples to form a pairwise similarity matrix (Yang et al., 2020). However, the relationship between samples cannot simply be determined by pairwise similarity. Therefore, hypergraphs (Zhou et al., 2006) are proposed to express the relationship between three or more samples.

In a hypergraph, each hyperedge consists of multiple vertices. **Figure 3** is a hypergraph and its association matrix H. In our study, each hyperedge weights 1. When hyperedge e_j contains vertex v_i, then H_{ij} is 1; otherwise, it is 0.

Formally, the association matrix H, the degree of each hyperedge, and the degree of each vertex can be expressed as:

$$H(v,e) = \begin{cases} 1, & \text{if } v \in e \\ 0, & \text{if } v \notin e \end{cases}, \quad (7a)$$

$$\delta(e) = \sum_{v \in V} H(v,e), \quad (7b)$$

$$d(v) = \sum_{e \in E} H(v,e). \quad (7c)$$

The Laplacian matrix of a hypergraph association matrix H can be calculated as:

$$L_H = I - D_v^{-\frac{1}{2}} H A D_e^{-1} H^T D_v^{-\frac{1}{2}}. \quad (8)$$

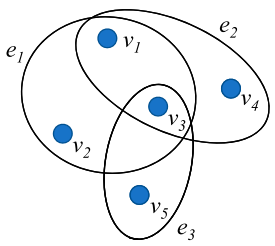
where D_v and D_e are the diagonal matrices formed by $d(v)$ and $\delta(e)$, respectively, and A is the same as the identity matrix I in our study. We construct the association matrix H with the k -nearest neighbour algorithm proposed by Zhou et al. (Zhou et al., 2006). Given a set of samples, we choose the k nearest neighbours of each sample and construct a hyperedge containing these k vertices. Finally, we construct N hyperedges for a dataset of N samples.

2.3.3 HG-HKNN

The HG-HKNN rewrites the mean squared distance from the test sample x to each hyperplane in the HKNN into the following form:

$$\left(LH_k^c(x)\right)^2 = \left\| \phi\left(\bar{x}\right) - \sum_{i=1}^{k} \alpha_i^c \phi\left(V_i^c\right) \right\|^2 + \lambda \sum_{i=1}^{k} \left(\alpha_i^c\right)^2 + \mu \sum_{p=1}^{k} \sum_{q=1}^{k} w_{p,q}^c \left(\alpha_p^c - \alpha_q^c\right)^2. \quad (9)$$

The kernel trick (Hofmann, 2006; Ding et al., 2019) is used to solve this problem, and the map ϕ maps the feature space to higher dimensions. $\bar{x} = x - \bar{N}$ is a simple rewrite. The third term in this formula is the Laplacian regularization term, which improves classification performance by smoothing the feature space (Ding et al., 2021). μ is the Laplacian regularization parameter, and $w_{p,q}^c$ is the similarity between the p -th nearest and the q -th nearest samples in the k samples in class c, which is



	v_1	v_2	v_3	v_4	v_5
e_1	1	1	1	0	0
e_2	1	0	1	1	0
e_3	0	0	1	0	1

FIGURE 3 | A hypergraph and its association matrix H.

calculated by the kernel function (Ding et al., 2022a). $K(x, y) = \phi(x), \phi(y)$ represents the kernel function, which is the radial basis function (RBF) in our study.

By minimizing the distance and making the partial derivative of $(LH_k^c(x))^2$ with respect to α^c zero, then the solution of α^c is obtained as follows:

$$\begin{aligned} \frac{\partial\left(\left(LH_k^c(x)\right)^2\right)}{\partial\alpha^c} &= 0, \\ \left(\phi(V^c)^T\phi(V^c) + \lambda I + \mu L\right)\alpha^c &= \phi(V^c)^T\phi\left(\bar{x}\right), \\ \alpha^c &= \left(\phi(V^c)^T\phi(V^c) + \lambda I + \mu L\right)^{-1}\phi(V^c)^T\phi\left(\bar{x}\right), \\ \alpha^c &= \left(K(V^c, V^c) + \lambda I + \mu L\right)^{-1}K\left(V^c, \bar{x}\right). \end{aligned} \tag{10}$$

We construct the hypergraph and use the Laplacian matrix of the hypergraph to replace the Laplacian matrix in the above formula:

$$\alpha^c = \left(K(V^c, V^c) + \lambda I + \mu L_H\right)^{-1}K\left(V^c, \bar{x}\right). \tag{11}$$

Note that the original Laplacian matrix contains pairwise similarities between samples, while our hypergraph Laplacian matrix contains more complex relationships between samples.

Now the distance from sample x to the c -th hyperplane can be expressed as follows:

$$\begin{aligned} distance_c &= \left\|\phi\left(\tilde{x}\right) - \sum_{i=1}^{k}\alpha_i^c\phi\left(V_i^c\right)^2\right\|, \\ &= \left(\phi(\bar{x}) - \phi(V^c)\alpha^c\right)^T\left(\phi(\bar{x}) - \phi(V^c)\alpha^c\right), \\ &= \left(K\left(\bar{x}, \bar{x}\right) - 2\left(\alpha^c\right)^T K\left(V^c, \bar{x}\right) + \left(\alpha^c\right)^T K(V^c, V^c)\alpha^c\right). \end{aligned} \tag{12}$$

Finally, we assign the test sample x to class c:

$$c = argmin_c\left(distance_c\right). \tag{13}$$

We define the prediction score as follows:

$$score_c = \frac{\sqrt{distance_c}}{\sum_{i=1}^{C}\sqrt{distance_i}}, i = 1, 2, \ldots, C. \tag{14}$$

The process of HG-HKNN is listed in Algorithm 1

Algorithm 1. Algorithm of HG-HKNN

Input: A test sample x, a training set N with C types of classes, five parameters k, λ, μ, γ and k_H;

Output: The prediction label and score of test sample x;

1. **for** $1 \le c \le C$ **do**
2. Getting k nearest neighbourhoods N_i^c (i from 1 to k) of sample x in c-th class;
3. Calculating $\bar{N}^c = \frac{1}{k}\sum_{i=1}^{k} N_i^c$, $V_i^c = N_i^c - \bar{N}^c$ and $\bar{x} = x - \bar{N}^c$;
4. Obtaining $K(\bar{x}, \bar{x})$, $K(V^c, \bar{x})$ and $K(V^c, V^c)$ with the radial basis function and its parameter γ;
5. Getting k_H nearest neighbours of each sample with kernel matrix $K(V^c, V^c)$,;
6. Constructing the hypergraph association matrix H, and obtaining the diagonal matrices D_v, D_e and A;
7. Calculating the Laplacian matrix of the hypergraph with $L_H = I - D_v^{-\frac{1}{2}}HAD_e^{-1}H^TD_v^{-\frac{1}{2}}$
8. Obtaining α^c with $\alpha^c = \left(K(V^c, V^c) + \lambda I + \mu L_H\right)^{-1}K(V^c, \bar{x})$;
9. Calculating $distance_c = (K(\bar{x}, \bar{x}) - 2(\alpha^c)^T K(V^c, \bar{x}) + (\alpha^c)^T K(V^c, V^c)\alpha^c)$;
10. **end for**
11. Assign sample x to class c with $c = argmin_c(distance_c)$;
12. Calculating prediction score for sample x with $score_c = \frac{\sqrt{distance_c}}{\sum_{i=1}^{C}\sqrt{distance_i}}$;
13. **Return** the label and score of test sample x.

3 RESULTS AND DISCUSSION

3.1 Evaluation

In this section, we will introduce the evaluation methods and metrics we use. We use positive to describe vesicular transport proteins and negative to describe non-vesicular transport proteins. We optimize the parameters with cross-validation (CV) on the training set and then evaluate our model on the test set.

Cross-validation sets aside a small portion of the dataset for validating the model, while the rest of the dataset is used for training the model (Zhang D. et al., 2021; Lv et al., 2021; Yang et al., 2021; Zheng et al., 2021; Li F. et al., 2022; Li X. et al., 2022). The leave-one-out cross-validation (LOOCV) is a classic cross-validation method (Qiu et al., 2021). LOOCV takes only one sample in the dataset at a time for validation and uses other samples in the dataset to train the model. Until all samples are left out once for validation, the leave-one-out method obtains statistical values for multiple results. However, the leave-one-out method is too time-consuming, so we adopted another cross-validation method: k -fold cross-validation (K-CV). K-CV divides the dataset into k subsets. Each time, one of the subsets is taken for validation, and the remaining $k-1$ subsets are used for training the model. In this way, k prediction results are obtained, and we take the average of these k results as the result of k -fold cross-validation.

The evaluation indicators we take include sensitivity, precision, specificity, accuracy (ACC), Matthews correlation coefficient (MCC), and area under the receiver operating characteristic curve (AUC), which have been widely used in previous studies (Hong et al., 2020a; Tang et al., 2020; Pan et al., 2022; Song et al., 2022).

$$sensitivity = \frac{TP}{TP + FN}, \tag{15a}$$

$$precision = \frac{TP}{TP + FP}, \tag{15b}$$

$$specificity = \frac{TN}{TN + FP}, \tag{15c}$$

$$ACC = \frac{TP + TN}{TP + FN + FP + FN}, \tag{15d}$$

$$MCC = \frac{1 - \left(\left(\frac{FN}{TP} + FN\right) + \left(\frac{FP}{TN} + FP\right)\right)}{\sqrt{\left(1 + \left(FP - \frac{FN}{TP} + FN\right)\right)\left(1 + \left(FN - \frac{FP}{TN} + FP\right)\right)}}. \tag{15e}$$

where TP, FP, TN, and FN represent true positives, false positives, true negatives, and false negatives, respectively. In addition, the AUC is obtained by integrating the receiver operating characteristic curve (ROC) (Fu J. et al., 2022). The ROC curve plots sensitivity and specificity at different classification thresholds (Tzeng et al., 2022). The more meaningful ones are AUC and precision since our test set is a class-imbalanced dataset. In our model, we perform 10-fold cross-validation on a training set of 4428 samples (2214 positive and 2214 negative). The binary

TABLE 2 | Details in parameter tuning of ***k***.

k	AUC	ACC	Precision	Specificity
200	0.8127	0.7256	0.7677	0.8035
350	0.8241	0.7319	0.7897	0.8311
500	0.8284	0.7362	0.7940	0.8338
650	0.8292	0.7398	0.7954	0.8333
800	0.8287	0.7425	0.7927	0.8265
950	0.8279	0.7437	0.7840	0.8134

TABLE 3 | Comparison of classification metrics among different kernels.

Kernel Type	AUC	MCC	ACC	Precision	Specificity
Linear	0.7618	0.3739	0.6719	0.7833	0.8686
Polynomial	0.8021	0.4664	0.7322	0.7519	0.7687
Laplacian	0.8243	0.5153	0.7575	0.7592	0.7597
RBF	0.8309	0.5099	0.7538	0.7760	0.7922

classification threshold is set to the default 0.5. Finally, the trained model is evaluated on the test set, which has 319 positive samples and 1513 negative samples.

3.2 Parameter Tuning

In this section, we describe the parameter tuning process for our model. Classification metrics are largely influenced by parameter tuning. The HG-HKNN has five parameters: k, λ, μ, γ, and k_H. k represents the number of neighbour samples selected when constructing the hyperplane. λ is the regularization parameter in L_2 regularization and μ is the Laplacian regularization parameter. γ is a parameter in the radial basis function. k_H is the number of neighbours used to construct the hypergraph.

We first adjust the k parameters among them. We set k, μ and γ to be 0.2, 0.2 and 0.2, respectively, and k_H to be 2. We perform 10-fold cross-validation for different values of k, and the best parameter k is determined to be 650; the details are shown in **Table 2**.

For λ, μ, γ and k_H, we adopt the grid search method for parameter tuning. The grid search method enumerates the possible values of each parameter, combines the possible values of all parameters into groups, and then trains the model with each group of parameters to obtain the best set of parameters. In our grid search, the possible values of λ, μ and γ are all 0.1, 0.2, 0.4, and 0.8, and the k_H values in the hypergraph range from 2 to 10. The best parameters for choosing λ, μ and γ are 0.4, 0.4 and 0.4, respectively. The best parameter k_H is 2, and the best AUC is 0.8309.

In our dataset, the dimension of features is much smaller than the number of samples, which is regarded as a sign that the dataset is linearly inseparable. On linearly inseparable datasets, the RBF kernel generally performs better than the linear or polynomial kernel. Formally, the Laplacian kernel is similar to the RBF kernel, and they usually have similar performance, but the Laplacian kernel function requires additional computational cost. We regard the type of kernel function used by HG-HKNN as an additional hyperparameter and conduct comparative experiments. The details of the experimental results are shown in **Table 3**. The results show that the RBF kernel has the best performance.

TABLE 4 | Comparison of classification metrics among different models.

Techniques	AUC	MCC	ACC	Precision	Specificity
KNN	0.7824	0.4189	0.7078	0.6886	0.6519
RF	0.8019	0.4576	0.7285	0.7267	0.7231
SVM	0.8091	0.4820	0.7405	0.7466	0.7502
HKNN	0.8203	0.4976	0.7484	0.7442	0.7371
OG-HKNN	0.8289	0.4944	0.7446	0.7843	0.8130
HG-HKNN	0.8309	0.5099	0.7538	0.7760	0.7922

3.3 Comparison With Traditional Machine Learning Methods

In the previous section, we have chosen the best parameters for our model. Our model is trained with traditional PsePSSM features, with nothing special in feature extraction. In this section, to highlight the effect of our proposed classifier HG-HKNN, we train some models with different traditional machine learning classifiers, the same training set, and the same PsePSSM feature extraction method. We perform 10-fold cross-validation on these models and compare the evaluation metrics of these models with ours. Note that the only difference between these models is the classifier.

We implement and train these models with the programming language's built-in library of functions. With the help of the parameter optimization function, we can automatically train the SVM model with the best evaluation metrics. After parameter tuning, the parameters in the other models are as follows: $K = 20$ in the k-nearest neighbour model(KNN), $ntrees = 60$ in the random forest model (RF), and $k = 30$ and $\lambda = 10$ in HKNN. **Table 4** shows the comparison of our model with other traditional machine learning models in 10-fold cross-validation.

Among them, the prediction effect of HKNN is better than that of the KNN algorithm. Intuitively explained in principle, although the classical K-nearest neighbour algorithm can fit the training samples well, it does not work well for the unseen samples located near the decision boundary. This is the overfitting problem of the KNN algorithm, and overfitting is more obvious in small data sets. HKNN constructs a hyperplane for k-nearest neighbour samples and then compares the distances between the test sample and the hyperplanes. The construction of the hyperplane can be analogous to adding more sample points to the k-nearest neighbours, which will reduce the interference of extreme samples on the decision boundary. Therefore, compared with KNN, the HKNN model has a smoother decision boundary, avoiding the disadvantage of overfitting in KNN.

Our proposed HG-HKNN model outperforms the other models on almost all metrics at the same level of comparison. By introducing Laplacian regularization in manifold learning, the HG-HKNN model incorporates local similarity information in the feature space into the construction process of the hyperplane.

TABLE 5 | Comparison of our model with other existing technologies.

Techniques	AUC	MCC	ACC	Sensitivity	Precision	Specificity
GRU	0.848	0.44	79.2	70.8	44.0	81.0
BLSTM	0.846	0.46	84.6	54.2	55.8	90.9
BLAST	0.82	0.43	83.6	54.1	52.8	89.8
Vesicular-GRU	0.861	0.52	82.3	79.2	48.7	82.9
HG-HKNN	0.870	0.53	84.1	72.1	53.2	86.7

Compared with the HKNN model, the HG-HKNN model not only reduces the disturbance of extreme samples to the decision boundary, but also preserves the local similarity information in the feature space. In the HG-HKNN model, we replace the ordinary graph with a hypergraph for Laplacian regularization. Hypergraph learning allows us to represent feature space local structures with more complex relationships than just pairwise similarity relationships. This further improves the performance of our HG-HKNN model. To highlight the effect of hypergraph learning, we add an ordinary graph regularized HKNN model (OG-HKNN) to our comparison, and the details are also listed in **Table 4**. The parameter tuning process of the OG-HKNN model is the same as that of the HG-HKNN. The best parameters for choosing λ, μ, γ and k are 0.2, 0.8, 0.4 and 350, respectively. The experimental results show that the AUC, MCC and ACC of the HG-HKNN model are better than the OG-HKNN model.

One disadvantage of our model is that HG-HKNN increases computation time and memory usage compared to HKNN. In terms of memory usage, the storage of hypergraphs, Laplacian matrices, and kernel matrices in HG-HKNN increases memory usage. In terms of operating efficiency, we conduct experiments on the test set with the same parameter $k = 20$, HKNN completes the computation in 362 milliseconds, while HG-HKNN completes the computation in 640 milliseconds. Such computational time cost is acceptable, especially considering the performance of HG-HKNN and time-consuming deep learning models in vesicle transporter identification.

3.4 Comparison With Previous Techniques

In this section, we aim to compare our model with previous techniques to highlight the performance of our proposed model on benchmark datasets. After optimizing the parameters with cross-validation, we obtain the optimal values of each parameter in HG-HKNN, where λ is 0.4, k is 650, γ is 0.4, μ is 0.4, and the value of k_H in the hypergraph part is 2. With these parameters, we no longer perform cross-validation on the training set but instead feed the entire training set into our model and then evaluate our final model on the test set. Among the metrics, the AUC is 87.0%, and the MCC is 0.53. Compared with the existing state-of-the-art Vesicular-GRU method with an AUC of 86.1% and MCC of 0.52, our model has higher AUC and MCC values, fewer feature dimensions (140 dimensions) and fewer parameters.

We compare our model with several other existing methods, among which the GRU model is a prediction method using traditional PSSM features and GRU and BLAST is a general-purpose protein prediction tool (Johnson et al., 2008). BLSTM is a commonly used prediction method in protein research (Li et al., 2020). The state-of-the-art method Vesicular-GRU (Le et al., 2019), a prediction method based on 1D CNN and GRU, is also listed in the comparison. The details of the comparison are shown in **Table 5**.

The meaning of the indicators has been described in the previous section. Experimental results show that our model achieves the best AUC and MCC metrics on this imbalanced benchmark dataset. Deep learning is involved in most of the methods in the comparison. The black box is an unavoidable problem for deep learning-based methods, and it is difficult to intuitively understand which factors lead to the predicted results. In deep learning models, researchers need to optimize a large number of parameters to improve the performance of the network, and these parameters are directly tuned through back-propagation of the prediction results, resulting in overfitting and the curse of dimensionality. The neural network in the Vesicular-GRU model has hundreds of thousands of parameters, which makes the Vesicular-GRU model a potential risk of overfitting on the training set. Our HG-HKNN has only five parameters, and the performance of our model is mainly attributable to hypergraph regularization and hyperplane rather than fitting to the parameters. Local hyperplane models have better performance on imbalanced datasets because the same number of samples are selected in each class. Like many biological sequence datasets, the vesicle transporter dataset is a typically imbalanced dataset, which is where the local hyperplane model excels. Furthermore, HG-HKNN applies kernel tricks to handle high-dimensional features, avoiding the curse of dimensionality. Although there is an increase in time and memory usage compared to HKNN, our model is faster relative to deep learning models trained with huge parameters via backpropagation. With only five parameters, our model avoids the black box, overfitting and curse of dimensionality problems in deep learning and makes predictions faster, and the performance of our model is equal to or higher than all the mentioned techniques, especially in terms of MCC and AUC.

4 CONCLUSION

In this study, we propose a novel approach for predicting vesicular transport proteins. The existing methods are typically performed with complex neural networks or by

extracting a large number of features. Our method classifies vesicular transport proteins with PsePSSM features and our proposed HG-HKNN model. We completed the prediction of vesicle transporters with only 140-dimensional features and 5 parameters with satisfactory results. Experimental results show that our method has the best AUC of 0.870 and MCC of 0.53 on the benchmark dataset and outperforms the state-of-the-art method (Vesicular-GRU) in ACC, MCC and AUC. Other metrics of our model are also comparable to other methods. A traditional machine learning computational model is used in our approach, avoiding some of the drawbacks of deep learning. Compared with another study (Tao et al., 2020) using traditional machine learning on the same dataset, their study achieved 72.2% accuracy and 0.34 MCC with 21-dimensional CTDC features after MRMD (He et al., 2020) dimensionality reduction, while our model achieves 84.1% accuracy and 0.53 MCC with 140-dimensional PsePSSM features. Furthermore, like CTDC features, the classical features we used imply that amino acids have a certain regularity in the arrangement of the protein sequence. Since PSSM matrix information is a commonly used motif representation, our study may help scholars to judge whether an unknown protein is a vesicle transporter.

The proposed method also has the following limitations: 1) In the case of large parameter k, the prediction takes a long time; 2) Our model uses the PsePSSM feature without incorporating sequence information for prediction; and 3) Feature selection and dimensionality reduction are not performed in our model. For the first limitation, parallel optimization can be used to solve the problem of computation time. For the second question, adding sequence features such as amino acid frequency or composition of k-spaced amino acid pairs (CKSAAP) to our model may further improve the prediction accuracy. For the third question, the dataset can be processed with feature selection and dimensionality reduction tools that remove redundant features. The results of this study can provide a basis for further studies in computational biology to identify vesicle transport proteins with classical features and traditional machine learning classifiers.

DATA AVAILABILITY STATEMENT

Our experimental code can be obtained from https://github.com/ferryvan/HG-HKNN, and the datasets used in this study can be found in (Le et al., 2019).

AUTHOR CONTRIBUTIONS

RF performed the experiment and wrote the manuscript; BS helped perform the experiment with constructive discussions; YD contributed to the conception of the study.

FUNDING

This work was supported by the Municipal Government of Quzhou under Grant Number 2020D003 and 2021D004.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fgene.2022.960388/full#supplementary-material

REFERENCES

Buck, S. A., Steinkellner, T., Aslanoglou, D., Villeneuve, M., Bhatte, S. H., Childers, V. C., et al. (2021). Vesicular Glutamate Transporter Modulates Sex Differences in Dopamine Neuron Vulnerability to Age-Related Neurodegeneration. *Aging cell* 20 (5), e13365. doi:10.1111/acel.13365

Chen, Y., Ma, T., Yang, X., Wang, J., Song, B., and Zeng, X. (2021). MUFFIN: Multi-Scale Feature Fusion for Drug-Drug Interaction Prediction. *Bioinformatics* 37 (17), 2651–2658. doi:10.1093/bioinformatics/btab169

Cheret, C., Ganzella, M., Preobraschenski, J., Jahn, R., and Ahnert-Hilger, G. (2021). Vesicular Glutamate Transporters (SLCA17 A6, 7, 8) Control Synaptic Phosphate Levels. *Cell Rep.* 34 (2), 108623. doi:10.1016/j.celrep.2020.108623

Chou, K.-C., and Shen, H.-B. (2007). MemType-2L: a Web Server for Predicting Membrane Proteins and Their Types by Incorporating Evolution Information through Pse-PSSM. *Biochem. biophysical Res. Commun.* 360 (2), 339–345. doi:10.1016/j.bbrc.2007.06.027

Consortium, G. O. (2004). The Gene Ontology (GO) Database and Informatics Resource. *Nucleic acids Res.* 32 (Suppl. l_1), D258–D261. doi:10.1093/nar/gkh036

Consortium, U. (2019). UniProt: a Worldwide Hub of Protein Knowledge. *Nucleic Acids Res.* 47 (D1), D506–D515. doi:10.1093/nar/gky1049

Ding, Y., Tang, J., and Guo, F. (2019). Protein Crystallization Identification via Fuzzy Model on Linear Neighborhood Representation. *IEEE/ACM Trans. Comput. Biol. Bioinform* 18 (5), 1986–1995. doi:10.1109/TCBB.2019.2954826

Ding, Y., He, W., Tang, J., Zou, Q., and Guo, F. (2021). Laplacian Regularized Sparse Representation Based Classifier for Identifying DNA N4-Methylcytosine Sites via L2, 1/2-matrix Norm. *IEEE/ACM Trans. Comput. Biol. Bioinforma.* 99, 1. doi:10.1109/tcbb.2021.3133309

Ding, Y., Jiang, L., Tang, J., and Guo, F. (2020a). Identification of Human microRNA-Disease Association via Hypergraph Embedded Bipartite Local Model. *Comput. Biol. Chem.* 89, 107369. doi:10.1016/j.compbiolchem.2020.107369

Ding, Y., Tang, J., and Guo, F. (2020b). Human Protein Subcellular Localization Identification via Fuzzy Model on Kernelized Neighborhood Representation. *Appl. Soft Comput.* 96, 106596. doi:10.1016/j.asoc.2020.106596

Ding, Y., Tang, J., and Guo, F. (2020c). Identification of Drug-Target Interactions via Dual Laplacian Regularized Least Squares with Multiple Kernel Fusion. *Knowledge-Based Syst.* 204, 106254. doi:10.1016/j.knosys.2020.106254

Ding, Y., Tang, J., Guo, F., and Zou, Q. (2022a). Identification of Drug–Target Interactions via Multiple Kernel-Based Triple Collaborative Matrix Factorization. *Briefings Bioinforma.* 23 (2), bbab582. doi:10.1093/bib/bbab582

Ding, Y., Tiwari, P., Zou, Q., Guo, F., and Pandey, H. M. (2022b). C-Loss Based Higher-Order Fuzzy Inference Systems for Identifying DNA N4-Methylcytosine Sites. *IEEE Trans. Fuzzy Syst.* 2022, 12. doi:10.1109/tfuzz.2022.3159103

Ding, Y., Yang, C., Tang, J., and Guo, F. (2022c). Identification of Protein-Nucleotide Binding Residues via Graph Regularized K-Local Hyperplane Distance Nearest Neighbor Model. *Appl. Intell.* 52 (6), 6598–6612. doi:10.1007/s10489-021-02737-0

Feehan, R., Franklin, M. W., and Slusky, J. S. G. (2021). Machine Learning Differentiates Enzymatic and Non-enzymatic Metals in Proteins. *Nat. Commun.* 12 (1), 1–11. doi:10.1038/s41467-021-24070-3

Fu, J., Zhang, Y., Wang, Y., Zhang, H., Liu, J., Tang, J., et al. (2022a). Optimization of Metabolomic Data Processing Using NOREVA. *Nat. Protoc.* 17 (1), 129–151. doi:10.1038/s41596-021-00636-9

Fu, T., Li, F., Zhang, Y., Yin, J., Qiu, W., Li, X., et al. (2022b). VARIDT 2.0: Structural Variability of Drug Transporter. *Nucleic Acids Res.* 50 (D1), D1417–D1431. doi:10.1093/nar/gkab1013

He, S., Guo, F., and Zou, Q. (2020). MRMD2. 0: a python Tool for Machine Learning with Feature Ranking and Reduction. *Curr. Bioinforma.* 15 (10), 1213–1221. doi:10.2174/1574893615999200503030350

Hofmann, M. (2006). Support Vector Machines-Kernels and the Kernel Trick. *Notes* 26 (3), 1–16.

Hong, J., Luo, Y., Mou, M., Fu, J., Zhang, Y., Xue, W., et al. (2020a). Convolutional Neural Network-Based Annotation of Bacterial Type IV Secretion System Effectors with Enhanced Accuracy and Reduced False Discovery. *Brief. Bioinform* 21 (5), 1825–1836. doi:10.1093/bib/bbz120

Hong, J., Luo, Y., Zhang, Y., Ying, J., Xue, W., Xie, T., et al. (2020b). Protein Functional Annotation of Simultaneously Improved Stability, Accuracy and False Discovery Rate Achieved by a Sequence-Based Deep Learning. *Brief. Bioinform* 21 (4), 1437–1447. doi:10.1093/bib/bbz081

Jin, S., Zeng, X., Xia, F., Huang, W., and Liu, X. (2021). Application of Deep Learning Methods in Biological Networks. *Briefings Bioinforma.* 22 (2), 1902–1917. doi:10.1093/bib/bbaa043

Johnson, M., Zaretskaya, I., Raytselis, Y., Merezhuk, Y., McGinnis, S., and Madden, T. L. (2008). NCBI BLAST: a Better Web Interface. *Nucleic Acids Res.* 36 (Suppl. l_2), W5–W9. doi:10.1093/nar/gkn201

Le, N. Q. K., Yapp, E. K. Y., Nagasundaram, N., Chua, M. C. H., and Yeh, H.-Y. (2019). Computational Identification of Vesicular Transport Proteins from Sequences Using Deep Gated Recurrent Units Architecture. *Comput. Struct. Biotechnol. J.* 17, 1245–1254. doi:10.1016/j.csbj.2019.09.005

Li, F., Eriksen, J., Finer-Moore, J., Edwards, R. H., and Stroud, R. M. (2021). Structure of a Vesicular Glutamate Transporter Determined by Cryo-Em. *Biophysical J.* 120 (3), 104a. doi:10.1016/j.bpj.2020.11.844

Li, F., Zhou, Y., Zhang, Y., Yin, J., Qiu, Y., Gao, J., et al. (2022a). POSREG: Proteomic Signature Discovered by Simultaneously Optimizing its Reproducibility and Generalizability. *Brief. Bioinform* 23 (2), bbac040. doi:10.1093/bib/bbac040

Li, J., Pu, Y., Tang, J., Zou, Q., and Guo, F. (2020). DeepAVP: a Dual-Channel Deep Neural Network for Identifying Variable-Length Antiviral Peptides. *IEEE J. Biomed. Health Inf.* 24 (10), 3012–3019. doi:10.1109/jbhi.2020.2977091

Li, X., Ma, S., Liu, J., Tang, J., and Guo, F. (2022b). Inferring Gene Regulatory Network via Fusing Gene Expression Image and RNA-Seq Data. *Bioinformatics* 38 (6), 1716–1723. doi:10.1093/bioinformatics/btac008

Liu, X., Zhang, X., Zhang, Y., Ding, Y., Shan, W., Huang, Y., et al. (2021). Kernelized K-Local Hyperplane Distance Nearest-Neighbor Model for Predicting Cerebrovascular Disease in Patients with End-Stage Renal Disease. *Front. Neurosci.* 15, 773208. doi:10.3389/fnins.2021.773208

Lv, H., Dao, F.-Y., Guan, Z.-X., Yang, H., Li, Y.-W., and Lin, H. (2021). Deep-Kcr: Accurate Detection of Lysine Crotonylation Sites Using Deep Learning Method. *Brief. Bioinform* 22 (4), bbaa255. doi:10.1093/bib/bbaa255

Mazere, J., Dilharreguy, B., Catheline, G., Vidailhet, M., Deffains, M., Vimont, D., et al. (2021). Striatal and Cerebellar Vesicular Acetylcholine Transporter Expression Is Disrupted in Human DYT1 Dystonia. *Brain* 144 (3), 909–923. doi:10.1093/brain/awaa465

Pan, X., Lin, X., Cao, D., Zeng, X., Yu, P. S., He, L., et al. (2022). Deep Learning for Drug Repurposing: Methods, Databases, and Applications. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* 2022, e1597. doi:10.1002/wcms.1597

Qiu, Y., Ching, W. K., and Zou, Q. (2021). Matrix Factorization-Based Data Fusion for the Prediction of RNA-Binding Proteins and Alternative Splicing Event Associations during Epithelial-Mesenchymal Transition. *Brief. Bioinform* 22 (6), bbab332. doi:10.1093/bib/bbab332

Shen, Y., Tang, J., and Guo, F. (2019). Identification of Protein Subcellular Localization via Integrating Evolutionary and Physicochemical Information into Chou's General PseAAC. *J. Theor. Biol.* 462, 230–239. doi:10.1016/j.jtbi.2018.11.012

Song, B., Li, F., Liu, Y., and Zeng, X. (2021). Deep Learning Methods for Biomedical Named Entity Recognition: a Survey and Qualitative Comparison. *Brief. Bioinform* 22 (6), bbab282. doi:10.1093/bib/bbab282

Song, B., Luo, X., Luo, X., Liu, Y., Niu, Z., and Zeng, X. (2022). Learning Spatial Structures of Proteins Improves Protein-Protein Interaction Prediction. *Briefings Bioinforma.* 23, bbab558. doi:10.1093/bib/bbab558

Su, R., He, L., Liu, T., Liu, X., and Wei, L. (2021). Protein Subcellular Localization Based on Deep Image Features and Criterion Learning Strategy. *Brief. Bioinform* 22 (4), bbaa313. doi:10.1093/bib/bbaa313

Tang, J., Fu, J., Wang, Y., Li, B., Li, Y., Yang, Q., et al. (2020). ANPELA: Analysis and Performance Assessment of the Label-free Quantification Workflow for Metaproteomic Studies. *Brief. Bioinform* 21 (2), 621–636. doi:10.1093/bib/bby127

Tao, Z., Li, Y., Teng, Z., and Zhao, Y. (2020). A Method for Identifying Vesicle Transport Proteins Based on LibSVM and MRMD. *Comput. Math. Methods Med.* 2020, 8926750. doi:10.1155/2020/8926750

Tzeng, S., Chen, C.-S., Li, Y.-F., and Chen, J.-H. (2022). On Summary ROC Curve for Dichotomous Diagnostic Studies: an Application to Meta-Analysis of COVID-19. *J. Appl. Statistics*, 1–17. doi:10.1080/02664763.2022.2041565

Vincent, P., and Bengio, Y. (2001). K-Local Hyperplane and Convex Distance Nearest Neighbor Algorithms. *Adv. neural Inf. Process. Syst.* 14, 985–992.

Wang, H., Ding, Y., Tang, J., Zou, Q., and Guo, F. (2021). Identify RNA-Associated Subcellular Localizations Based on Multi-Label Learning Using Chou's 5-steps Rule. *Bmc Genomics* 22 (1), 56. doi:10.1186/s12864-020-07347-7

Xiong, G., Wu, Z., Yi, J., Fu, L., Yang, Z., Hsieh, C., et al. (2021). ADMETlab 2.0: an Integrated Online Platform for Accurate and Comprehensive Predictions of ADMET Properties. *Nucleic Acids Res.* 49 (W1), W5–W14. doi:10.1093/nar/gkab255

Yang, H., Ding, Y., Tang, J., and Guo, F. (2021). Drug-disease Associations Prediction via Multiple Kernel-Based Dual Graph Regularized Least Squares. *Appl. Soft Comput.* 112, 107811. doi:10.1016/j.asoc.2021.107811

Yang, Q., Li, B., Tang, J., Cui, X., Wang, Y., Li, X., et al. (2020). Consistent Gene Signature of Schizophrenia Identified by a Novel Feature Selection Strategy from Comprehensive Sets of Transcriptomic Data. *Brief. Bioinform* 21 (3), 1058–1068. doi:10.1093/bib/bbz049

Zeng, X., Tu, X., Liu, Y., Fu, X., and Su, Y. (2022). Toward Better Drug Discovery with Knowledge Graph. *Curr. Opin. Struct. Biol.* 72, 114–126. doi:10.1016/j.sbi.2021.09.003

Zhang, D., Chen, H.-D., Zulfiqar, H., Yuan, S.-S., Huang, Q.-L., Zhang, Z.-Y., et al. (2021a). iBLP: An XGBoost-Based Predictor for Identifying Bioluminescent Proteins. *Comput. Math. Methods Med.* 2021, 6664362. doi:10.1155/2021/6664362

Zhang, J., Zhang, Z., Pu, L., Tang, J., and Guo, F. (2021b). AIEpred: An Ensemble Predictive Model of Classifier Chain to Identify Anti-inflammatory Peptides. *IEEE/ACM Trans. Comput. Biol. Bioinf.* 18 (5), 1831–1840. doi:10.1109/tcbb.2020.2968419

Zhang, W., Xue, X., Xie, C., Li, Y., Liu, J., Chen, H., et al. (2021c). CEGSO: Boosting Essential Proteins Prediction by Integrating Protein Complex, Gene Expression, Gene Ontology, Subcellular Localization and Orthology Information. *Interdiscip. Sci. Comput. Life Sci.* 13 (3), 349–361. doi:10.1007/s12539-021-00426-7

Zhang, Z.-Y., Sun, Z.-J., Yang, Y.-H., and Lin, H. (2022). Towards a Better Prediction of Subcellular Location of Long Non-coding RNA. *Front. Comput. Sci.* 16 (5), 1–7. doi:10.1007/s11704-021-1015-3

Zheng, Y., Wang, H., Ding, Y., and Guo, F. (2021). CEPZ: A Novel Predictor for Identification of DNase I Hypersensitive Sites. *IEEE/ACM Trans. Comput. Biol. Bioinf.* 18 (6), 2768–2774. doi:10.1109/tcbb.2021.3053661

Zhou, D., Huang, J., and Schölkopf, B. (2006). Learning with Hypergraphs: Clustering, Classification, and Embedding. *Adv. neural Inf. Process. Syst.* 19, 1601–1608.

Zhou, Y., Zhang, Y., Lian, X., Li, F., Wang, C., Zhu, F., et al. (2022). Therapeutic Target Database Update 2022: Facilitating Drug Discovery with Enriched Comparative Data of Targeted Agents. *Nucleic Acids Res.* 50 (D1), D1398–D1407. doi:10.1093/nar/gkab953

Zhu, X.-J., Feng, C.-Q., Lai, H.-Y., Chen, W., and Hao, L. (2019). Predicting Protein Structural Classes for Low-Similarity Sequences by Evaluating Different Features. *Knowledge-Based Syst.* 163, 787–793. doi:10.1016/j.knosys.2018.10.007

Zou, Y., Wu, H., Guo, X., Peng, L., Ding, Y., Tang, J., et al. (2021). MK-FSVM-SVDD: a Multiple Kernel-Based Fuzzy SVM Model for Predicting DNA-Binding Proteins via Support Vector Data Description. *Cbio* 16 (2), 274–283. doi:10.2174/1574893615999200607173829

Zulfiqar, H., Yuan, S.-S., Huang, Q.-L., Sun, Z.-J., Dao, F.-Y., Yu, X.-L., et al. (2021). Identification of Cyclin Protein Using Gradient Boost Decision Tree Algorithm. *Comput. Struct. Biotechnol. J.* 19, 4123–4131. doi:10.1016/j.csbj.2021.07.013

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.