Combinatorial Chemistry & High Throughput Screening, 2019, *22*, 455-469

RESEARCH ARTICLE

Boosting Granular Support Vector Machines for the Accurate Prediction of Protein-Nucleotide Binding Sites

Yi-Heng Zhu[1], Jun Hu[2], Yong Qi[1], Xiao-Ning Song[3] and Dong-Jun Yu[1,*]

[1]*School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, P.R. China;* [2]*College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, P.R. China;* [3]*School of Internet of Things, Jiangnan University, Wuxi 214122, P.R. China*

Abstract: ***Aim and Objective*:** The accurate identification of protein-ligand binding sites helps elucidate protein function and facilitate the design of new drugs. Machine-learning-based methods have been widely used for the prediction of protein-ligand binding sites. Nevertheless, the severe class imbalance phenomenon, where the number of nonbinding (majority) residues is far greater than that of binding (minority) residues, has a negative impact on the performance of such machine-learning-based predictors.

***Materials and Methods*:** In this study, we aim to relieve the negative impact of class imbalance by Boosting Multiple Granular Support Vector Machines (BGSVM). In BGSVM, each base SVM is trained on a granular training subset consisting of all minority samples and some reasonably selected majority samples. The efficacy of BGSVM for dealing with class imbalance was validated by benchmarking it with several typical imbalance learning algorithms. We further implemented a protein-nucleotide binding site predictor, called BGSVM-NUC, with the BGSVM algorithm.

***Results*:** Rigorous cross-validation and independent validation tests for five types of protein-nucleotide interactions demonstrated that the proposed BGSVM-NUC achieves promising prediction performance and outperforms several popular sequence-based protein-nucleotide binding site predictors. The BGSVM-NUC web server is freely available at http://csbio.njust.edu.cn/bioinf/BGSVM-NUC/ for academic use.

ARTICLE HISTORY

Received: May 04, 2019
Revised: June 21, 2019
Accepted: August 23, 2019

DOI:
10.2174/1386207322666190925125524

Keywords: Imbalance learning, granular computing, support vector machine, classifier ensemble, protein-nucleotide binding sites.

1. INTRODUCTION

Proteins often need to interact with other molecules (ligands) through binding sites to participate in various cellular and biological processes. Hence, the accurate identification of protein-ligand binding sites helps clarify protein function and facilitate the design of new drugs [1, 2]. However, traditional biochemical methods for identifying protein-ligand binding sites are time-consuming and expensive and cannot meet the urgent demands of related research. In light of this, researchers in this field have focused on developing computational methods, such as template-based methods [3-5] and machine-learning-based methods [6, 7], to quickly and accurately predict protein-ligand binding sites in recent years.

Template-based methods identify the binding sites of the query protein using the sequence and/or structure information of protein templates that were found by the appropriate alignment or comparison algorithms. For example, Roy *et al*. [3] developed COFACTOR for the identification of protein-ligand interactions by using protein structural models based on a global-to-local sequence and structural comparison algorithm; Yang *et al*. [4] designed COACH, which predicts protein-ligand binding sites based on a binding-specific substructure comparison algorithm (TM-SITE) and a sequence profile alignment (S-SITE). Other popular template-based methods include CASTp [8], FINDSITE [9], ConCavity [10], SITEHOUND [11], and 3DLigandSite [12]. However, the performances of these methods are heavily dependent on the number and quality of the protein templates found, which limits their applicability, especially in situations where an insufficient number of templates with known tertiary structures are available.

Machine-learning-based methods have emerged as a promising route for the accurate identification of protein-

*Address correspondence to this author at the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, P.R. China; E-mail: njyudj@njust.edu.cn

ligand binding sites. For example, Pupko *et al*. [13] proposed Rate4Site, which identifies the functionally important regions in proteins by using the maximum likelihood (ML) principle [14] to estimate the level of conservation of each amino acid; Shu *et al*. [15] designed a machine-learning-based method that combines a support vector machine (SVM) [16] classifier with a homology-based predictor, to identify zinc-binding sites from protein sequences; Chen *et al*. [17] developed a predictor (NsitePred) that combines SVM with the comprehensive features extracted from protein sequence, evolutionary profiles and several sequence-predicted structural descriptors, to predict protein-nucleotide binding residues with improved accuracy; Panwar *et al*. [18] proposed a SVM-based ligand-specific vitamin-binding sites predictor; Yu *et al*. [19] designed a sequence-based template-free predictor (TargetATPsite) that utilizes a novel image sparse representation technique to code input features and combines the adaptive boosting (AdaBoost) algorithm with a random under-sampling technique to eliminate the class imbalance problem in the identification of ATP-binding residues; Chen *et al*. [20] introduced a random forest (RF) [21] based predictor, called LigandRFs that ensembles multiple RFs trained on balanced datasets to solve the data imbalance phenomenon in protein-ligand prediction.

Previous studies [6, 7] have witnessed the great success of machine-learning-based methods for the prediction of protein-ligand binding sites. Nevertheless, an inevitable critical issue for all machine-learning-based methods is the class imbalance phenomenon where the number of binding sites (minority class) is significantly fewer than that of non-binding sites (majority class) [19, 22, 23]. Traditional statistical machine learning algorithms will fail to achieve good performance under the class imbalance scenario because the prediction results tend to be biased toward the majority class [24-26]. Taking SVM, which is one of the most used machine learning algorithms for the prediction of protein-ligand binding sites, as an example, related studies have demonstrated that SVM often performs effectively on balanced datasets but could generate suboptimal results with imbalanced datasets [27-29]. The underlying reason can be explained as follows: the basic idea of SVM is mapping the samples from original feature space to a new high-dimension feature space and finding a separating hyperplane to classify the samples in this new space; therefore, the performance of SVM is depended on the separating hyperplane; if SVM is trained on an imbalanced dataset, the corresponding separating hyperplane will be pushed towards the minority class, which leads to an unexpected result that SVM more likely predicts minority samples to majority ones [27-29]. As another example, K-Nearest Neighbor (KNN) [30, 31], one of the classical machine-learning algorithms, also obtains the unsatisfied performances on imbalanced datasets [32] due to the following reason: for a query sample, KNN first finds k samples (neighbors), which are nearest to it in feature space, from training dataset, and then predicts it as one class which has the highest frequency among these k neighbors; thus, the predicted result of the query sample by KNN is completely determined by its k nearest neighbors; if training dataset is imbalanced, the k nearest neighbors of a query sample will be mainly composed of majority class; as a result, KNN tends to predict minority samples as majority ones.

To address the negative impact of class imbalance, many solutions, such as sample rescaling [33-35], active learning [36, 37], and kernel learning [38, 39], have been developed. Among these solutions, sampling rescaling, which balances the sizes of samples of different classes by changing the numbers of samples and distribution between classes, is the most straight-forward one and has been widely used as a basic strategy for obtaining a balanced dataset for training machine learning models [20, 40, 41].

Among a number of sample-rescaling-based methods, random under-sampling, which can get a parsimonious sampled training dataset, is the simplest and most straight forward one. Considering the simplicity of random under-sampling as well as the efficiency of SVM mentioned above, researchers attach more importance to deal with the class imbalance problem by combining SVM with random under-sampling. For example, Kang *et al*. [33] proposed an ensemble of under-sampled SVMs (EUS SVMs), which involves three ensemble methods, namely majority voting, weighted voting, and function value aggregation, to incorporate multiple SVMs trained on subsets of the original imbalanced training dataset *via* random under-sampling; Yu *et al*. [42] developed an algorithm to ensemble multiple SVMs trained on several sub-datasets sampled from the original imbalanced dataset by using random under-sampling.

However, random under-sampling does not always provide optimal performance because it can result in information loss. In the unique scenario where random under-sampling is combined with SVM, the information loss of samples may cause the loss of cues about the ideal hyperplane of SVM, which can lead to an unexpected result. Substantial effort has been devoted to finding more effective sampling-rescaling-based methods for solving class imbalance problems [43, 44]. Recently, Tang *et al*. [45, 46] developed a granular SVMs-repetitive under-sampling model, called GSVM-RU, which trains the SVM model based on granular computing [47] and under-sampling. Specifically, GSVM-RU first generates multiple subsets from the original imbalanced training datasets based on the concept of granular computing and then trains the SVM model on the final dataset, which is formed using the proposed aggregation methods, including "Discard" and "Combine", to aggregate the multiple subsets obtained above.

We noticed that GSVM-RU is an effective model that elegantly combines SVM with under-sampling and has been demonstrated to be superior to traditional SVM on a series of imbalanced datasets. However, the performance of GSVM-RU can be further improved by avoiding potential defects (loss and redundancy of sample information) in its aggregation methods (we will carefully investigate this point in Section 2.3).

Motivated by the merits of granular computing and the potential disadvantages of aggregation strategies in GSVM-RU, in this study, we proposed an improved version of GSVM-RU by boosting multiple granular SVMs, called BGSVM, to address the class imbalance. More specifically, in BGSVM, we first obtain multiple granular SVMs, each of which is trained on a granular subset sampled from the original training dataset; then, the obtained multiple granular

Table1. Statistical compositions of the benchmark datasets.

Dataset	Ligand Type	No. of Sequences	(Num_Pos, Num_Neg) [a]	Ratio [b]
Train-NUC	ATP	227	(3393, 80409)	24
	ADP	321	(4688, 121158)	26
	AMP	140	(1756, 44009)	25
	GDP	105	(1577, 36561)	23
	GTP	56	(875, 21401)	24
Test-NUC	ATP	17	(248, 6974)	28
	ADP	26	(405, 10553)	26
	AMP	20	(263, 6057)	23
	GDP	7	(94, 2420)	26
	GTP	7	(134, 2678)	20

[a] Num_Pos and Num_Neg represent the numbers of positive and negative samples, respectively;
[b] Ratio = Num_Neg/Num_Pos, which measures the imbalance degree of a dataset.

SVMs are ensembled by using an enhanced AdaBoost (EAdaBoost) algorithm [48]. On one hand, BGSVM retains the merits of granular computing of GSVM-RU; on the other hand, the potential disadvantages of the aggregation methods in GSVM-RU are relieved by introducing the EAdaBoost algorithm to ensemble the multiple granular SVMs.

We performed rigorous comparison experiments regarding the prediction of binding sites for five types of nucleotide ligands that show severe class imbalance phenomena. The experimental results demonstrate that the proposed BGSVM is superior to GSVM-RU under the class imbalance scenario and that the predictor implemented with BGSVM, called BGSVM-NUC, outperforms the state-of-the-art sequence-based protein-nucleotide binding site predictors. The BGSVM-NUC web server is available at http://csbio.njust.edu.cn/bioinf/BGSVM-NUC/ for academic use.

2. MATERIALS AND METHODS

2.1. Benchmark Datasets

In this study, we used the dataset constructed by Chen *et al*. [17] as a benchmark dataset to evaluate the efficacy of the proposed BGSVM and compare the proposed predictor with existing protein-nucleotide binding site predictors. This benchmark dataset consists of a training set, called Train-NUC, and an independent validation set, called Test-NUC.

Train-NUC is composed of 227, 321, 140, 105, and 56 protein sequences (released into PDB before 10 March 2010) that bind to ATP, ADP, AMP, GDP, and GTP, respectively. For each type of the five nucleotides, the maximal pairwise sequence identity of the corresponding protein sequences is reduced less than 40% with CD-HIT software [49]. Test-NUC consists of 17, 26, 20, 7, and 7 protein sequences (released into PDB after 10 March 2010) interacting with ATP, ADP, AMP, GDP, and GTP, respectively. Also, the maximal pairwise identity of the sequences for each type of the five nucleotides in Test-NUC is reduced less than 40% by CD-HIT. In addition, for each type of the five nucleotides, no sequence in Test-NUC shares more than 40% pairwise identity to sequences in Train-NUC. Train-NUC and Test-NUC can be easily downloaded at http://csbio.njust.edu.cn/bioinf/BGSVM-NUC/Data.html, and their detailed statistical compositions are summarized in Table **1**.

2.2. Feature Representation

In this study, two typical features, *i.e*., position-specific scoring matrix (PSSM) and predicted protein secondary structure (PPSS), are serially combined to form the feature representation of each residue in a protein sequence.

2.2.1. Position-Specific Scoring Matrix Feature

Position-specific scoring matrix, which is one of the most important feature sources used in protein-ligand binding sites prediction, encodes evolutionary conservation information of a protein. For a given protein sequence with L residues, we obtain its PSSM, which is L rows and 20 columns numeric matrix, by using PSI-BLAST [50] to search against the Swiss-Prot database [51] through three iterations with *E*-value = 0.001 as the cutoff. Then, the obtained PSSM is further normalized with the following logistic function (eq. 1):

$$f(x) = \frac{1}{1+\exp(-x)} \tag{1}$$

where x is the original value in PSSM. Considering the fact that whether a residue will interact with ligands depends on not only the residue itself but also its neighboring residues, a sliding window of size W centered on the residue is used to extract its PSSM feature vector. Previous studies [17, 42, 52] have demonstrated that $W = 17$ is a better choice. Thus, the dimensionality of PSSM-derived feature vector of a residue is $17 \times 20 = 340$.

2.2.2. Predicted Protein Secondary Structure Feature

We extract predicted protein secondary structure (PPSS) feature of a residue by using PSIPRED [53] as follows: for a given protein sequence with L residues, the output of PSIPRED is a $L \times 3$ matrix. The three values in the *i*-th row of the matrix measure the probabilities of the *i*-th residue for belonging to three secondary structure classes, *i.e*., coil (C), helix (H), and strand (E), respectively. Again, a sliding

window of size 17 is used to extract the PPSS feature of each residue and the dimensionality of the extracted feature vector is $17\times3=51$.

2.3. Granular SVMs-Repetitive Under-Sampling Model (GSVM-RU)

GSVM-RU [45, 46] is based on granular computing [47]. The basic idea of granular computing is representing information in the form of granules and solving the target problem in each information granule [46]. The granule can be a subset, subspace, class, or cluster. In GSVM-RU, a granule refers to a subset of the original training dataset. More specifically, GSVM-RU extracts all positive samples to form a positive information granule, called *PS*, and generates multiple negative information granules by the following under-sampling steps: initially, GSVM-RU constructs an SVM on the original training dataset and then extracts all negative samples, which are represented by the negative support vectors of the trained SVM, to form a negative information granule; these negative samples are called negative local support vectors (*NSLV*); in the next step, a new training dataset is formed by removing the *NSLV* from the original training dataset; then, GSVM-RU constructs an SVM on the new training dataset and extracts all the negative support vectors of the newly trained SVM to form a new negative granule; the above procedure is repeated several times to generate multiple negative granules.

After obtaining multiple negative granules (*i.e*., *NSLV*), the goal of GSVM-RU is to aggregate the positive information granule (*PS*) with multiple negative information granules (*NSLV*) to form a final training dataset, denoted as *FD*; then, a final SVM classifier is trained on *FD*. Considering that it is difficult to determine the specific number of *NSLVs* before performing aggregation operation, GSVM-RU executes under-sampling and aggregation operation in turns: initially, the *FD* only contains *PS*; when a new *NSLV* is generated, it is aggregated with *FD* by using the reasonable strategies and an SVM is then trained on the new aggregated dataset *FD*. The procedure is continued until the newly generated *NSLV* cannot further improve the classification performance of the SVM trained on *FD*.

There are two aggregation strategies, *i.e*., "Discard" and "Combine", in GSVM-RU [45, 46]. In "Discard" strategy, when a new $NLSV$ is generated, only the negative samples in this granule are added into *FD* and all negative samples in old negative information granules are removed from *FD*. By continuously removing *NSLVs*, "Discard" strategy pushes the hyperplane of an SVM towards the negative class to seek the ideal hyperplane. However, removing a large number of negative samples may cause serious information loss. To reduce information loss, "Combine" strategy has been developed. In "Combine" strategy, when a new *NSLV* is extracted, it is directly added into *FD* and all old negative granules are reserved in *FD*. Unfortunately, blindly combining the current granule with old granules easily leads to information redundancy. In light of this, we thus try to circumvent this issue by boosting multiple granular support vector machines.

2.4. Boosting Granular Support Vector Machines

As an improved version of GSVM-RU, the proposed BGSVM aims to enhance the performance of GSVM-RU for dealing with class imbalance by improving the aggregation strategy while preserving the merit of granular computing.

GSVM-RU aggregates multiple negative granules with a positive granule and then trains a global SVM on the aggregated dataset [45, 46]. Unlike GSVM-RU, the proposed BGSVM aggregates multiple SVMs, which are trained on different granules sampled from the original training dataset, by using the EAdaBoost algorithm [48]. The information loss and redundancy incurred by data level aggregation in GSVM-RU could be partially relieved by decision level aggregation in BGSVM, hence the importance of this work. Fig. (**1**) presents a schematic diagram of the proposed BGSVM. As described in Fig. (**1**), the basic idea of BGSVM can be roughly described as follows.

In the training stage, two procedures are performed: granular SVMs generation (Procedure I) and granular SVMs ensemble (Procedure II). In the first procedure, BGSVM generates N subsets (granules) of the training dataset, denoted as $\{N\_Tr_i\}_{i=1}^{N}$, by under-sampling, and constructs SVM on each granule to form a team of granular SVMs, denoted as SVM_Team. Second, the EAdaBoost algorithm is performed on SVM_Team to select M ($M \le N$) SVMs, which form a new set of SVMs denoted as $SVM\_Selected$, and the corresponding weight of each selected SVM is calculated. In the test stage, first, a given query input x is predicted by each SVM in $SVM\_Selected$, and a result set *P*, which contains the predicted result of each selected SVM, is generated; then, a post-processing procedure, based on the values in *P* and the weights of the SVMs in $SVM\_Selected$, is performed to obtain the final predicted result, denoted as $H(x)$.

It should be noted that we use EAdaBoost rather than the original AdaBoost [54] in Procedure II (this decision is explained in detail in Section 2.4.2). In EAdaBoost, an independent evaluation dataset (*IED*) that has no samples in common with the training dataset of Procedure I was used. Therefore, given a training dataset, denoted as *TD*, for BGSVM, we randomly select 20% of the samples from the *TD* to form the *IED* in Procedure II and use the remaining samples as the training dataset of Procedure I, denoted as *RTD*. As shown in Fig. (**1**), each procedure of BGSVM can be described in detail as follows.

2.4.1. Granular SVMs Generation

The procedure for generating the granular SVMs can be further divided into the following two steps:

Step I: Extract a positive information granule and multiple negative information granules

We extract all the positive samples in the training dataset (*RTD*) as a positive information granule, *PS*, which is the same as what was done in GSVM-RU; then, an SVM is trained on *RTD* and all the negative support vectors of the trained SVM are extracted as a negative information granule,

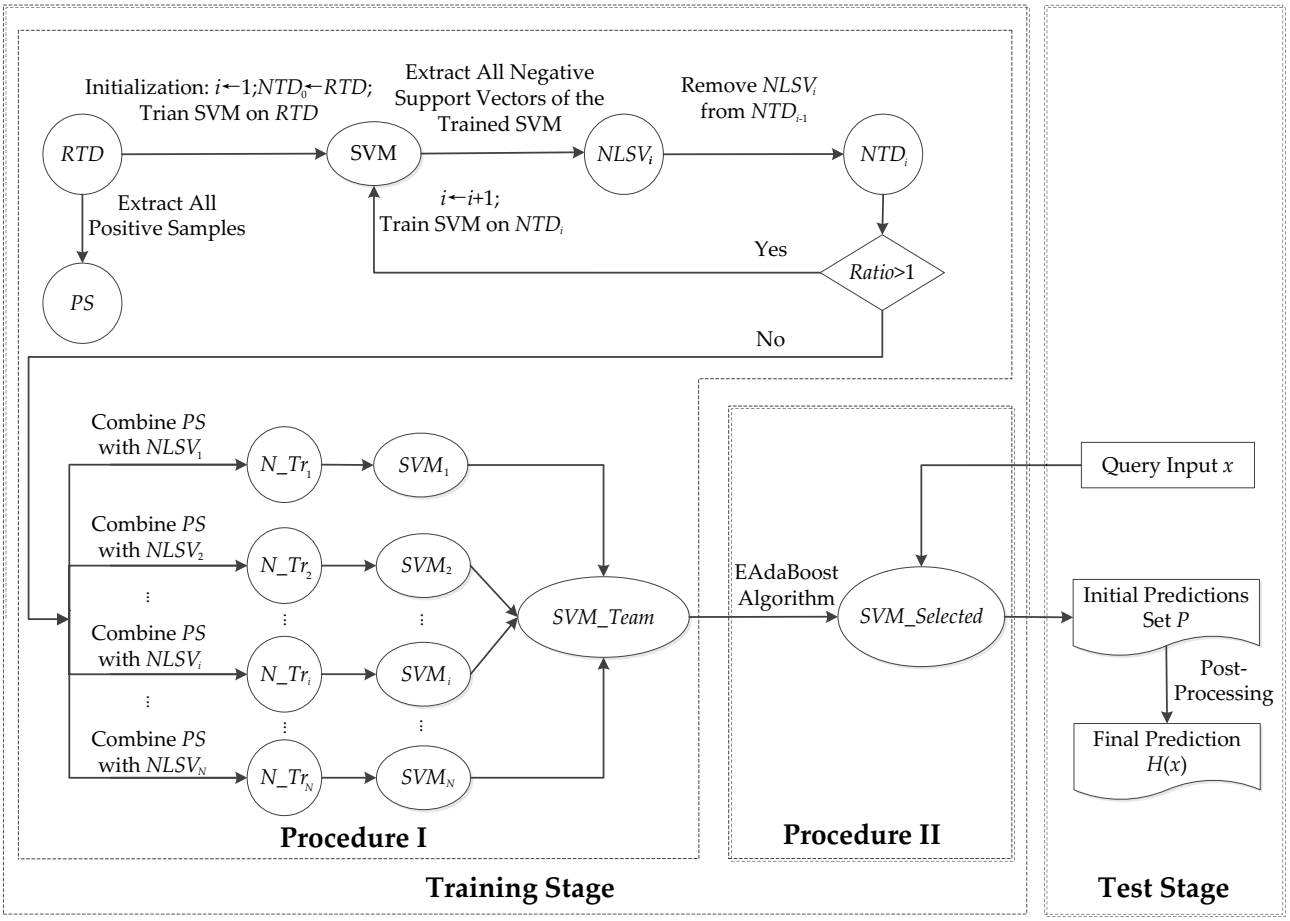

Fig. (1). The schematic diagram of the proposed BGSVM.

denoted as $NLSV_1$; after that, the negative samples in $NLSV_1$ are removed from the *RTD* and the remaining set is taken as the new training dataset, represented as NTD_1; then, a new SVM is trained on NTD_1, and all negative support vectors of the newly trained SVM are extracted to form a new negative granule, called $NLSV_2$; this practice continues until the ratio between the number of negative samples and the number of positive samples in the newly generated training dataset is equal to or less than 1. At this point, we extract a set of negative information granules, denoted as $NLSV\_Set = \{NLSV_i\}_{i=1}^{N}$, where N is the number of extracted negative information granules.

Step II: Train a team of base classifiers based on PS and $NLSV\_Set$

Each $NLSV_i \in NLSV\_Set$ is combined with PS to form a granule-specific training subset, denoted as N_Tr_i. Then, we train a granular base SVM, called SVM_i, on each N_Tr_i. Accordingly, we obtain a team of granular base SVMs, denoted as $SVM\_Team = \{SVM_i\}_{i=1}^{N}$.

In this study, we implement the SVM classifier by using LIBSVM software [55], which is freely available at http://www.csie.ntu.edu.tw/~cjlin/libsvm/. Here, the radial basis function (RBF) is selected as the kernel function, and two parameters, *i.e.*, penalty parameter C and RBF kernel width parameter γ, are optimized by the grid search strategy of the LIBSVM tool over five-fold cross-validation.

2.4.2. Granular SVMs Ensemble

In this procedure, BGSVM provides the final ensembled classifier by boosting multiple granular SVMs in SVM_Team. Among various boosting algorithms, AdaBoost algorithm is the most frequently used one. As described in [54], AdaBoost is an iterative algorithm; in each iteration, it first evaluates the error rate of the base classifier using evaluation samples and calculates the weight of the base classifier. After multiple iterations, the final ensemble classifier is generated by combining several base classifiers with their weights. However, the original AdaBoost often leads to over-fitting. The underlying reason is that samples in the training dataset are used as evaluation samples; in other words, the evaluation samples and training samples originate from the same dataset; as a result, the ensemble classifier shows outstanding performance on the training dataset but poor generalization performance on the test dataset.

To overcome over-fitting, we adopted an EAdaBoost algorithm [48] in this work. Compared with the original AdaBoost, EAdaBoost uses an independent evaluation dataset (*IED*), which has no samples in common with the training dataset used in Procedure I, to evaluate the error rates of the base classifiers. The procedure for ensembling multiple granular SVMs with EAdaBoost is summarized in Algorithm **1**.

Algorithm 1: Ensembling multiple granular SVMs with EAdaBoost

Input: $SVM\_Team = \{SVM_i\}_{i=1}^{N}$: a team of granular base SVMs;

$IED = \{X_j^e\}_{j=1}^{m}$: independent evaluation dataset, where X_j^e is the *j*-th evaluation sample and m is the number of evaluation samples

Output: $SVM\_Selected$: a set of selected granular SVMs; SVM_Weight : the weights set of selected granular SVMs

Initialization: $i \leftarrow 1$; $k \leftarrow 1$; $SVM\_Selected \leftarrow \varnothing$; $SVM\_Weight \leftarrow \varnothing$; $w_j^i = 1/m$, $j = 1, 2, ..., m$, where w_j^i is the weight of the *j*-th evaluation sample in the *i*-th iteration

Calculate the error rate of SVM_i , denoted as ε_i , by Eq. (2)

1

$$\varepsilon_i \leftarrow \sum_{j=1}^{m} w_j^i \cdot I_j^i \qquad (2)$$

where $I_j^i = 1$ if SVM_i misclassifies X_j^e ; otherwise, $I_j^i = 0$.

2 If $\varepsilon_i = 0$ or $\varepsilon_i > 0.5$, $i \leftarrow i+1$, $w_j^i = 1/m$, $j = 1, 2, ..., m$, go to **Step 1**;

Calculate the weight of SVM_i , denoted as β_i , by Eq. (3)

3

$$\beta_i = \frac{1}{2}\log\frac{1-\varepsilon_i}{\varepsilon_i} \qquad (3)$$

$E_k \leftarrow i$, where E_k is the index of the *k*-th base SVM selected from SVM_Team ;

4 Add SVM_{E_k} and β_{E_k} to $SVM\_Selected$ and SVM_Weight , respectively: $SVM\_Selected \leftarrow SVM\_Selected \cup \{SVM_{E_k}\}$, $SVM\_Weight \leftarrow SVM\_Weight \cup \{\beta_{E_k}\}$;

5 Update the weight of each evaluation sample by Eq. (4)

$$w_j^i \leftarrow \frac{w_j^i \cdot \exp((2 \cdot I_j^i - 1) \cdot \beta_i)}{\sum_{j=1}^{m} w_j^i \cdot \exp((2 \cdot I_j^i - 1) \cdot \beta_i)} \qquad (4)$$

6 $i \leftarrow i+1$; $k \leftarrow k+1$; if $i < N$, go to **Step 1**; otherwise, normalize the values of SVM_Weight : $\beta_{E_k} = \beta_{E_k} / \sum_{k=1}^{M} \beta_{E_k}$, $k = 1, 2, ..., M$, where M is the number of selected base SVMs.

Return $SVM\_Selected$, SVM_Weight

As described in Algorithm **1**, taking the *i*-th iteration as an example, first, the *IED* is used to calculate the error rate of SVM_i, denoted as ε_i, by Eq. (2); then, if $0 < \varepsilon_i < 0.5$, the SVM_i is selected from *SVM_Team* and the corresponding weight, denoted as β_i , can be calculated by Eq. (3); otherwise, the SVM_i is discarded; finally, the weight of each evaluation sample is updated by Eq. (4). After *N* iterations, we will obtain a set of selected base SVMs, denoted as $SVM\_Selected = \{SVM_{E_k}\}_{k=1}^{M}$, with a corresponding set of base classifier weights, denoted as $SVM\_Weight = \{\beta_{E_k}\}_{k=1}^{M}$, where SVM_{E_k} denotes the E_k -th SVM in SVM_Team , $E_k \in [1, N]$, $E_1 < E_2 < \cdots < E_k < \cdots < E_M$, $\beta_{E_1} + \beta_{E_2} + ... + \beta_{E_M} = 1$, and M is the number of selected base classifiers. Then, the decision function of the ensembled classifier can be formulated as follows:

$$H(x) = \sum_{k=1}^{M} \beta_{E_k} \cdot SVM_{E_k}(x) \qquad (5)$$

where x is the query input and $SVM_{E_k}(x)$ is the output of base classifier SVM_{E_k} under input x . Without loss of generality, in this study, we suppose that each base classifier predicts the probability of a query sample x for belonging to the positive class.

2.4.3. Post-Processing Procedure

For a query input x , we can obtain the initial prediction of the ensembled classifier, *i.e.*, $H(x)$, by using Eq. (5). Let $P = \{p_{E_k}\}_{k=1}^{M}$ be the set of predictions of the M base

classifiers for a query input x, where $p_{E_k} = SVM_{E_k}(x)$ is the probability of query sample x for belonging to the positive class predicted by base classifier SVM_{E_k}.

To further improve the prediction performance, we propose a post-processing technique based on the following observations.

Observation: in most cases, the probability of a query sample for belonging to positive class predicted by SVM_{E_k} is lower than that predicted by $SVM_{E_{k+1}}$, *i.e.*, $p_{E_k} \le p_{E_{k+1}}$ ($1 \le k < M$).

Table 2. The percent of testing samples that conform to the observation for the five types of nucleotides over five-fold cross-validation.

	ATP	ADP	AMP	GDP	GTP
Percent (%)	83.3	80.8	81.8	75.3	83.1

We calculated the percent of testing samples that conform to the above observation for each type of nucleotide over five-fold cross-validation, as shown in Table **2**. It can be found over 80% of testing samples conform to this observation for four out of the five types of nucleotides, *i.e.*, ATP, ADP, AMP, and GTP. The underlying reason for this observation can be qualitatively explained as follows:

The SVM_{E_k} is trained on the dataset that combines PS with $NLSV_{E_k}$, while the $SVM_{E_{k+1}}$ is trained on the dataset that combines PS with $NLSV_{E_{k+1}}$. By revisiting the training procedures of the proposed BGSVM described above, we know that $E_k < E_{k+1}$ and $NLSV_{E_k}$ is more close to the positive samples than $NLSV_{E_{k+1}}$. The relative positions of $NLSV_{E_k}$ and $NLSV_{E_{k+1}}$ are intuitively illustrated in Fig. (**2**). In other words, the separating hyperplane of SVM_{E_k} is more close to the positive samples than that of $SVM_{E_{k+1}}$, which explains why $SVM_{E_{k+1}}$ is more likely to predict a query sample as positive if compared with SVM_{E_k}.

Considering the rationality of the observation mentioned-above, we thus developed a simple post-processing procedure for those query samples that do not conform to the observation by re-arranging the predictions of base classifiers as follows:

For a query input x, the predictions of the M base classifiers are formulated as $P = \{p_{E_k}\}_{k=1}^{M}$. If there exist $p_{E_i} \ge p_{E_j} (1 \le i < j \le M)$, *i.e.*, the predictions of base classifiers do not conform to the observation, we re-arrange the M predictions in P in ascending order and get $P' = \{p_{E_k'}\}_{k=1}^{M}$, where $p_{E_i'} \le p_{E_j'}$ for any $i < j$ ($1 \le i, j \le M$). Then, the re-arranged $P' = \{p_{E_k'}\}_{k=1}^{M}$ is considered as the predictions of the M base classifiers, *i.e.*, $p_{E_k'} \triangleq SVM_{E_k}(x)$. After this post-processing procedure, the final prediction of the ensembled classifier can be formulated as eq (6):

$$H(x) = \sum_{k=1}^{M} \beta_{E_k} \cdot p_{E_k'} \tag{6}$$

where β_{E_k} is the weight of the base classifier SVM_{E_k}.

2.5. Evaluation Indices

In this work, four evaluation indices [56-60], *i.e.*, *Sensitivity* (*Sen*), *Specificity* (*Spe*), *Accuracy* (*Acc*), and *Matthew's Correlation Coefficient* (*MCC*) were used to evaluate the prediction performances of predictors as follows eq (7-10):

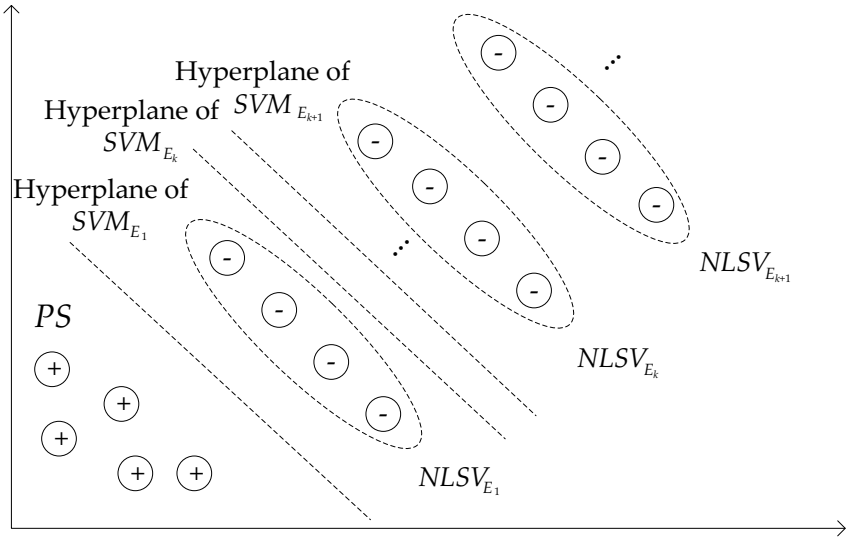

Fig. (2). The relative positions of $NLSV_{E_k}$ and $NLSV_{E_{k+1}}$. The circled points with '+' inside denote the positive samples, the circled points with '-' inside denote the negative support vectors, and the dashed lines are separating hyperplanes.

Table 3. The *AUC* performances of base granular SVMs on Train-NUC over five-fold cross-validation for five types of nucleotides.

Base Granular SVM	ATP	ADP	AMP	GDP	GTP
SVM_1	0.895	0.915	0.859	0.922	0.862
SVM_2	0.892	0.918	0.864	0.928	0.865
SVM_3	0.867	0.897	0.834	0.905	0.846
SVM_4	0.823	0.863	0.733	0.895	0.748
SVM_5	0.739	0.804	-	0.814	-
SVM_6	0.657	0.710	-	-	-
SVM_Num [a]	6	6	4	5	4
SVM_Best [b]	SVM_1	SVM_2	SVM_2	SVM_2	SVM_2

[a] *SVM_Num* indicates the number of SVMs in *SVM_Team*;
[b] *SVM_Best* is the SVM which has the highest *AUC* in *SVM_Team*;
'-' indicates that the corresponding value does not exist.

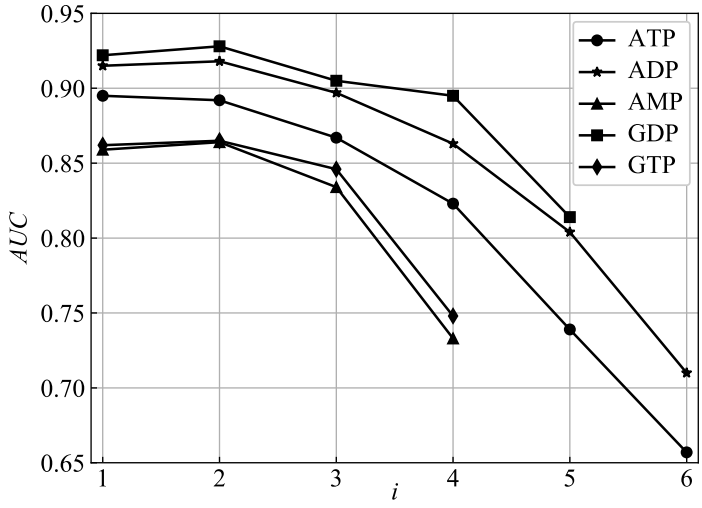

Fig. (3). The variation curve of *AUC* versus base granular SVM (SVM_i) for each type of the five nucleotides.

$$Sen = \frac{TP}{TP + FN} \quad (7)$$

$$Spe = \frac{TN}{TN + FP} \quad (8)$$

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \quad (9)$$

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}} \quad (10)$$

where *TP* is the number of correctly classified positive samples, *FN* is the number of the positive samples misclassified as negatives, *TN* is the number of the correctly classified negative samples, and *FP* is the number of negative samples misclassified as positives.

The above four indices, including the *MCC* which provides the overall measurement of the quality of the binary predictions, are threshold-dependent. In all experiments of this study, we selected the threshold which maximizes the value of *MCC* on the training set over five-fold cross-validation test. To further evaluate the overall performance of a predictor on imbalanced datasets, we used the area under the receiver operating characteristic (*ROC*) curve (*AUC*), which is threshold-independent, as another critical evaluation index.

3. RESULT AND DISCUSSION

3.1. Classification Performances of Base Granular SVMs in BGSVM

We evaluated the classification performance of each base granular SVM_i in *SVM_Team*. Since each SVM_i is trained on a dataset that combines *PS* with $NLSV_i$, we thus can investigate the relative importance of each $NLSV_i$ by evaluating the performance of the corresponding SVM_i.

For each type of the five nucleotides, we evaluated the overall performance, measured by *AUC*, of each granular SVM_i in *SVM_Team* in BGSVM on the corresponding training dataset over five-fold cross-validation. Table **3** summarizes the *AUC* performances of base granular SVMs on Train-NUC over five-fold cross-validation for five types of nucleotides, while Fig. (**3**) plots the variation curve of *AUC* versus base granular SVM (SVM_i) for each type of the five nucleotides.

Interesting phenomena can be observed from Table **3** and Fig. (**3**) as follows:

Table 4. Performances of the proposed BGSVM on the training datasets for five types of nucleotides over five-fold cross-validation

Ligand Type	*Sen* (%)	*Spe* (%)	*Acc* (%)	*MCC*	*AUC*
ATP	48.4	99.1	97.0	0.561	0.901
ADP	62.1	99.1	97.7	0.657	0.924
AMP	37.3	98.9	96.6	0.449	0.873
GDP	67.5	99.6	98.3	0.765	0.933
GTP	47.3	99.6	97.5	0.609	0.872

For four out of the five types of nucleotides, *i.e*., ADP, AMP, GDP, and GTP, it is found that the *AUC* performance of base granular SVM_i first enhances and then decreases with the increase of the value of *i*. The *AUC* performance reaches maximum when $i = 2$ for the 4 ligands, denoting that SVM_2 trained on a dataset obtained by combining *PS* with $NLSV_2$ can achieve the best classification performance. When $i > 2$, the value of *AUC* gradually decreases with the increase of the value of *i*. This phenomenon can be explained as follows: initially, the separating hyperplane of SVM_1 trained on a dataset that combines $NLSV_1$ with PS may be too close to positive samples, thus SVM_1 more likely predicts positive samples to negative ones; after gradually removing $NLSV_i$ ($i > 1$) from the original training dataset, the separating hyperplane of SVM_i is moved towards negative samples; hence, the classification performance of SVM_i could be improved; when the separating hyperplane of SVM_i is moved at or close to the ideal hyperplane, SVM_i achieves the best performance (*e.g*., *i*=2 for ADP, AMP, GDP, and GTP in this study); after arriving at or close to the ideal hyperplane, the separating hyperplane of SVM_i (*e.g*., $i > 2$ for ADP, AMP, GDP, and GTP in this study) may be moved more and more towards negatives samples if we further remove *NLSVs*, which makes the prediction of SVM_i is skewed to positive class, leading to a deteriorate *AUC* performance.

We also observe that the *AUC* performance of base granular SVM_i for ATP continuously decreases with the increase of the value of *i*. The best *AUC* performance is achieved when $i = 1$, indicating that the separating hyperplane of SVM_1 trained on a dataset obtained by combining PS with $NLSV_1$ is considerably ideal compared with that of other base granular SVM_i ($i \geq 2$).

Two conclusions can be drawn from the above two phenomena: first, it is feasible to seek a better separating hyperplane by continuously removing *NLSVs* from an imbalanced training dataset; second, the number of base granular SVMs and the optimal base granular SVM in *SVM_Team* are dataset-dependent.

3.2. Classification Performance of BGSVM

In this selection, we evaluated the prediction performance of BGSVM. For five types of nucleotides, the performances of the proposed BGSVM on the corresponding training datasets over a rigorous five-fold cross-validation procedure are summarized in Table **4**.

From Table **4**, it is found that the values of the evaluation indices of BGSVM for five types of ligands vary in different ranges. For examples, the *Sensitivity* (*Sen*) varies from 37.3% to 67.5%, the *Specificity* (*Spe*) from 98.9% to 99.6%, and the *Accuracy* (*Acc*) from 96.6% to 98.3%. Moreover, it can be found that BGSVM yields *AUC*>0.87 and *MCC*>0.44 for all five types of nucleotides. With respect to *AUC* and *MCC*, the GDP reaches the highest values, which are 0.933 and 0.765, respectively, among the five types of nucleotides. On the contrary, AMP has the lower values of *AUC* and *MCC*, which are 0.873 and 0.449, respectively. There exist about 6% gap of *AUC* and 22% gap of *MCC* between GDP and AMP. We speculate that these gaps are caused by the imbalanced distribution of training datasets.

To investigate the mechanism of BGSVM, we further compared it with the *SVM_Best*, which has the highest *AUC* in *SVM_Team*. Fig. (**4**) illustrates the detailed performance comparisons between BGSVM and *SVM_Best* on the corresponding training datasets over five-fold cross-validation.

From Fig. (**4**), we can observe that the performance of the proposed BGSVM always performs better than *SVM_Best* in terms of *AUC*, *Spe*, *Acc*, and *MCC* for all the five nucleotides. For example, the values of *MCC* of BGSVM are 0.561, 0.657, 0.449, 0.765, and 0.609, which are approximately 0.1%, 4.9%, 8.4%, 3.7%, and 11.9% higher than the *MCC* values produced by *SVM_Best*, for ATP, ADP, AMP, GDP, and GTP, respectively. Results in Fig. (**4**) demonstrate that the ensembled predictor obtained by boosting multiple granular SVMs does help to improve prediction performance even on imbalanced dataset.

3.3. Performance Comparisons between BGSVM, GSVM-RU, and SVM-RU

We compared the performance of BGSVM with that of two under-sampling-based prediction models. Here, SVM with random under-sampling, denoted as SVM-RU, is used as the baseline model. In SVM-RU, randomly under-sampling is applied to the majority class and a balanced dataset is obtained; then, a global SVM model is trained on the balanced dataset. Considering that BGSVM is based on GSVM-RU [45, 46], we also compared BGSVM with GSVM-RU. It is noted that GSVM-RU has two aggregation

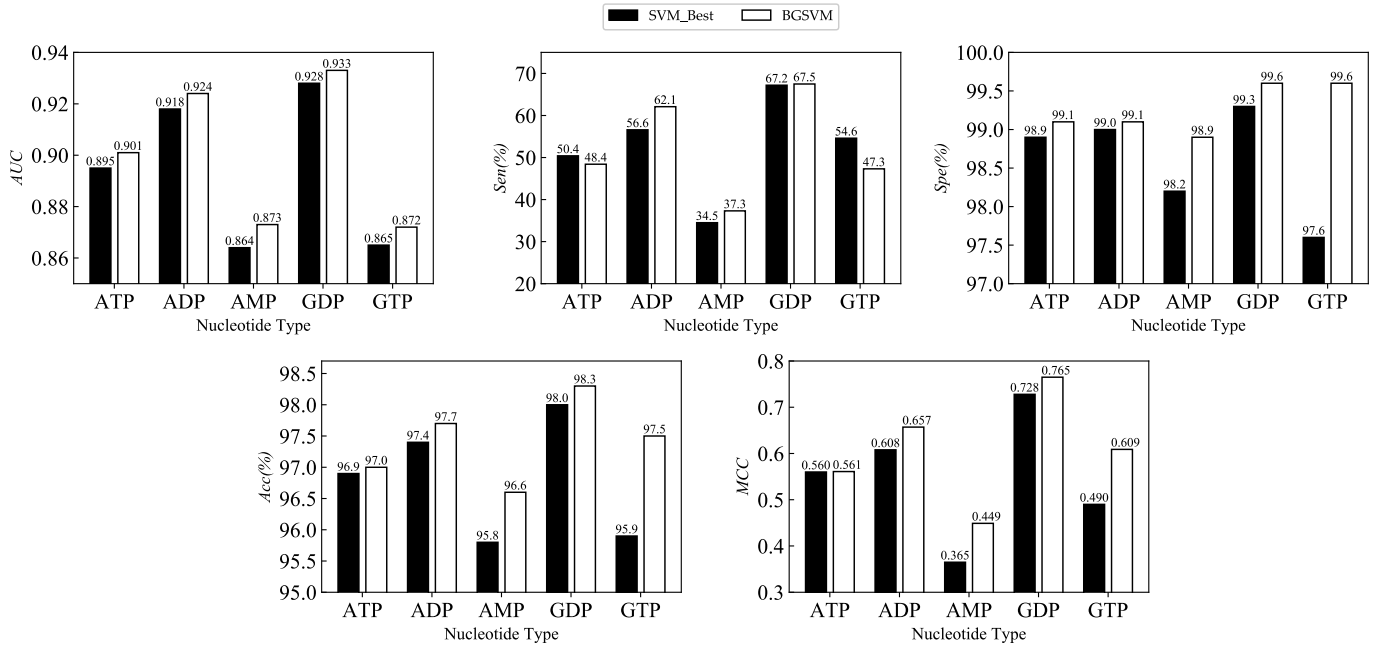

Fig. (4). Performance comparisons between the proposed BGSVM and *SVM_Best* over five-fold cross-validation.

Table 5. Performance comparisons between BGSVM, GSVM-RU, and SVM-RU for five types of ligands over five-fold cross-validation.

Ligand Type	Method	*Sen* (%)	*Spe* (%)	*Acc* (%)	*MCC*	*AUC*
ATP	BGSVM	**48.4**	99.1	**97.0**	**0.561**	**0.901**
	GSVM-RU	37.6	**99.5**	**97.0**	0.523	0.891
	SVM-RU	41.9	98.7	96.4	0.474	0.885
ADP	BGSVM	**62.1**	99.1	**97.7**	**0.657**	**0.924**
	GSVM-RU	53.2	**99.5**	**97.7**	0.638	0.917
	SVM-RU	51.9	99.0	97.3	0.576	0.909
AMP	BGSVM	**37.3**	98.9	96.6	**0.449**	**0.873**
	GSVM-RU	31.8	**99.3**	**96.7**	0.437	0.862
	SVM-RU	30.4	98.9	96.2	0.379	0.850
GDP	BGSVM	**67.5**	99.6	**98.3**	**0.765**	**0.933**
	GSVM-RU	59.7	**99.7**	98.1	0.727	0.928
	SVM-RU	64.6	99.4	98.0	0.717	0.923
GTP	BGSVM	**47.3**	99.6	**97.5**	**0.609**	**0.872**
	GSVM-RU	42.2	**99.7**	97.4	0.589	0.837
	SVM-RU	45.6	99.4	97.2	0.569	0.862

strategies ("Discard" and "Combine") and choosing the appropriate aggregation strategy is a critical step of GSVM-RU. In light of this, here we adopted the hybrid aggregation strategy recently developed by Tang *et al*. [46] to perform aggregation in GSVM-RU. This strategy can be described as follows: both the "Discard" and the "Combine" aggregations are executed when the second negative granule is extracted; then, the winner ("Discard" or "Combine") which can achieve better performance will be used for next aggregation. Table **5** summarizes the detailed performance comparisons between BGSVM, GSVM-RU, and SVM-RU for five types of ligands over five-fold cross-validation.

It is easy to find from Table **5** that the proposed BGSVM outperforms both GSVM-RU and SVM-RU for all the five ligands with highest values of *AUC* and *MCC*. We notice that the proposed BGSVM performs much better than SVM-RU with highest improvements of 2.3% and 8.7% regarding *AUC* and *MCC*, respectively, for AMP and ATP. Compared with GSVM-RU, which is the second-best performer, BGSVM also achieves approximately averaged improvements of 1.4% and 2.5% regarding *AUC* and *MCC*, respectively. In terms of *AUC*, the maximal improvement (3.5%) over GSVM-RU is achieved by BGSVM on GTP. As to *MCC*, BGSVM achieves the maximal improvement (3.8%) over GSVM-RU for both ATP and GDP.

Table 6. Performance comparisons between BGSVM, GSVM-RU, and SVM-RU on the independent validation datasets for the five types of nucleotides.

Ligand Type	Method	*Sen* (%)	*Spe* (%)	*Acc* (%)	*MCC*	*AUC*
ATP	BGSVM	**55.6**	99.0	**97.5**	**0.595**	**0.920**
	GSVM-RU	37.1	**99.4**	97.2	0.490	0.919
	SVM-RU	47.2	98.7	96.9	0.498	0.904
ADP	BGSVM	58.0	98.6	**97.1**	**0.578**	**0.929**
	GSVM-RU	**70.9**	94.3	93.5	0.452	0.920
	SVM-RU	44.7	**98.9**	96.9	0.510	0.909
AMP	BGSVM	**43.0**	99.0	**96.7**	**0.512**	**0.895**
	GSVM-RU	38.4	**99.3**	**96.7**	0.503	0.892
	SVM-RU	39.9	98.9	96.5	0.481	0.878
GDP	BGSVM	**35.1**	99.6	**97.2**	**0.514**	**0.881**
	GSVM-RU	26.6	**99.8**	97.0	0.453	0.871
	SVM-RU	**35.1**	98.4	96.1	0.384	0.870
GTP	BGSVM	56.0	**99.6**	97.5	0.687	**0.913**
	GSVM-RU	**56.7**	**99.6**	**97.6**	**0.697**	0.912
	SVM-RU	53.7	99.2	97.0	0.631	0.907

To demonstrate the generalization capability of the proposed BGSVM, we further compared it with GSVM-RU and SVM-RU on independent validation datasets. For each of the three methods, we trained it on the training dataset for a given ligand type and then tested the trained model with the corresponding independent validation set as described in Table **1**. Table **6** summarizes the performance comparisons between BGSVM, GSVM-RU, and SVM-RU on the independent validation datasets for the five types of nucleotides.

From Table **6**, we can conclude that the generalization performance of the proposed BGSVM outperforms that of GSVM-RU and SVM-RU with respect to *AUC* and *MCC*, which are two global metrics for evaluating prediction quality. In terms of *AUC*, the corresponding values of BGSVM on ATP, ADP, AMP, GDP, and GTP under independent validation tests are 0.920, 0.929, 0.895, 0.881, and 0.913 respectively, which are 1.6%, 2.0%, 1.7%, 1.1%, and 0.6% higher than that of SVM-RU. As to *MCC*, BGSVM is superior to SVM-RU with improvements of 9.7%, 6.8%, 3.1%, 13.0%, and 5.6% on ATP, ADP, AMP, GDP, and GTP, respectively. Compared with GSVM-RU, the *MCC* of BGSVM on GTP is 1.0% lower. However, it still achieves improvements of 10.5%, 12.6%, 0.9%, and 6.1% on ATP, ADP, AMP, and GDP regarding *MCC*. Moreover, the values of *AUC* of BGSVM on ADP and GDP are both almost 1.0% higher than the *AUC* values yielded by GSVM-RU.

3.4. Comparison with Existing Predictors

To further demonstrate the efficacy of BGSVM, we compared the predictor implemented with BGSVM, called BGSVM-NUC, to other popular sequence-based protein-ligand binding site predictors including Rate4Site [13], SVMPred [17], NsitePred [17], and TargetS [42]. Table **7** illustrates the performances of BGSVM-NUC and the above-mentioned existing predictors on Train-NUC dataset over five-fold cross-validation for comparison.

First, we compared the overall performances, measured by *AUC* and *MCC*, of the five protein-nucleotide predictors considered in this study. As shown in Table **7**, the proposed BGSVM-NUC obviously shows the best performance, as it offers the highest values of *AUC* and *MCC* and consistently outperforms the other four predictors for all five types of nucleotide ligands. More specifically, we observed that the proposed BGSVM-NUC overwhelms Rate4site, SVMPred, and NsitePred. For example, BGSVM-NUC achieves averaged *AUC* improvements of approximately 15.4%, 4.1%, and 3.3% relative to those of Rate4site, SVMPred, and NsitePred, respectively, on the five nucleotide training datasets. Regarding TargetS, which is the second-best performer among the listed predictors, BGSVM-NUC still achieves an averaged improvement of approximately 1.0% in *AUC*. In terms of *MCC*, BGSVM-NUC significantly outperforms Rate4site, SVMPred, and NsitePred. For example, BGSVM-NUC achieves average improvements of approximately 9.7% and 7.9% in *MCC* over SVMPred and NsitePred, respectively. Compared with TargetS, the *MCC* of BGSVM-NUC showed an improvement of approximately 2.5% on average.

We further compared the protein-nucleotide predictors based on three other metrics, *i.e.*, *Sen*, *Spe*, and *Acc*. For ATP and GDP, BGSVM-NUC consistently performs better than or equal to the other four predictors in terms of *Sen*, *Spe*, and *Acc*. Regarding ADP, AMP, and GTP, BGSVM-NUC also provides the best performance on *Acc* while showing comparable or even better performance on *Spe* compared with SVMPred, NsitePred, and TargetS. For 3 out of 5 ligands, *i.e.*, ATP, ADP, and GDP, BGSVM-NUC performs best in terms of *Sen*, with the highest values of 48.4%, 62.1%, and 67.5%, respectively. It has not escaped from our notice that Rate4Site performs much better than the other 4 predictors with the highest *Sen* values of 56.2% and 56.9% for AMP and GTP, respectively. However, the corresponding *Spe* values of Rate4Site for AMP and GTP are significantly lower than those of the other 4 predictors. In other words, Rate4Site tends to predict too many false positives. Together with the fact that the number of negative

Table 7. Performance comparisons between BGSVM-NUC, TargetS, NsitePred, SVMPred, and Rate4Site on Train-NUC dataset over five-fold cross-validation.

Ligand Type	Predictor	*Sen* (%)	*Spe* (%)	*Acc* (%)	*MCC*	*AUC*
ATP	BGSVM-NUC	**48.4**	**99.1**	**97.0**	**0.561**	**0.901**
	TargetS [a]	44.6	99.0	96.7	0.531	0.896
	NsitePred [a]	44.4	98.2	96.0	0.460	0.861
	SVMPred [a]	36.1	98.8	96.2	0.433	0.854
	Rate4Site [a]	44.6	87.0	85.2	0.182	0.749
ADP	BGSVM-NUC	**62.1**	99.1	**97.7**	**0.657**	**0.924**
	TargetS [a]	58.7	99.0	97.5	0.631	0.918
	NsitePred [a]	54.4	98.8	97.1	0.572	0.893
	SVMPred [a]	45.8	**99.3**	97.3	0.555	0.885
	Rate4Site [a]	47.2	84.4	83.0	0.161	0.749
AMP	BGSVM-NUC	37.3	98.9	**96.6**	**0.449**	**0.873**
	TargetS [a]	36.8	98.6	96.1	0.418	0.857
	NsitePred [a]	30.4	98.8	96.2	0.377	0.829
	SVMPred [a]	20.8	**99.6**	**96.6**	0.360	0.820
	Rate4Site [a]	**56.2**	79.9	79.0	0.174	0.755
GDP	BGSVM-NUC	**67.5**	**99.6**	**98.3**	**0.765**	**0.933**
	TargetS [a]	65.0	**99.6**	98.1	0.741	0.920
	NsitePred [a]	64.6	99.1	97.6	0.675	0.910
	SVMPred [a]	62.3	98.9	97.7	0.655	0.905
	Rate4Site [a]	51.6	82.3	81.1	0.170	0.733
GTP	BGSVM-NUC	47.3	99.6	**97.5**	**0.609**	**0.872**
	TargetS [a]	44.3	99.6	97.4	0.595	0.863
	NsitePred [a]	47.3	99.1	96.8	0.562	0.844
	SVMPred [a]	37.3	**99.7**	97.0	0.551	0.836
	Rate4Site [a]	**56.9**	80.6	79.6	0.180	0.748

[a] Data are excerpted from [42].

samples is far larger than that of positive samples, Rate4Site produces the lowest performances in terms of *MCC* for all 5 types of ligands.

Table **8** summarizes the comparison between the performance of the five considered protein-ligand predictors on the independent validation datasets Test-NUC. For four out of five ligands, *i.e.*, ATP, ADP, AMP, and GTP, the proposed BGSVM-NUC provides the best overall performance in terms of *MCC* and *AUC*. The *MCC* values of BGSVM-NUC on ATP, ADP, AMP, and GTP reach 0.595, 0.578, 0.512, and 0.687, which are 6.1%, 6.2%, 0.9%, and 3.4%, respectively, higher than the corresponding values of the second-best predictor, TargetS. Regarding *AUC*, the proposed BGSVM-NUC still shows improvements of 0.4%, 0.6%, 1.1%, and 0.3% for ATP, ADP, AMP, and GTP, respectively, when compared with TargetS. For GDP, we found that BGSVM-NUC also achieves a good *AUC* performance that is comparable to those of TargetS, NsitePred, and SVMPred. Nevertheless, BGSVM-NUC is inferior to TargetS, NsitePred, and SVMPred in terms of *MCC* for GDP. We speculate that the insufficient distribution information in the training dataset may account for the inferior performance of BGSVM-NUC for GDP.

CONCLUSIONS

In this work, we proposed a new machine-learning algorithm, called BGSVM, for addressing the class imbalance by boosting multiple granular support vector machines. Based on the proposed BGSVM, we implemented an effective protein-nucleotide binding site predictor, called BGSVM-NUC, which can currently perform binding site predictions for five types of nucleotide ligands. The experimental results with a training dataset and an independent test dataset demonstrated that the proposed BGSVM-NUC outperforms other existing sequence-based protein-nucleotide binding site predictors. The superior performance of our predictor mainly stems from the impressive ability of the BGSVM algorithm to deal with class imbalance, which is a common phenomenon in protein-ligand prediction problems.

Although our method has provided some improvement compared with other sequence-based protein-ligand predictors,

Table 8. Performance comparisons of the proposed BGSVM-NUC with other protein-ligand binding sites predictors on independent validation datasets of Test-NUC.

Ligand Type	Predictor	*Sen* (%)	*Spe* (%)	*Acc* (%)	*MCC*	*AUC*
ATP	BGSVM-NUC	**55.6**	99.0	**97.5**	**0.595**	**0.920**
	TargetS [a]	50.4	98.9	97.2	0.534	0.916
	NsitePred [b]	46.0	98.5	96.7	0.476	0.875
	SVMPred [b]	36.7	**99.1**	96.9	0.451	0.868
	Rate4Site [b]	46.4	86.2	84.9	0.167	0.741
ADP	BGSVM-NUC	**58.0**	98.6	**97.1**	**0.578**	**0.929**
	TargetS [a]	50.9	98.5	96.8	0.516	0.923
	NsitePred [b]	47.4	98.7	96.8	0.512	0.893
	SVMPred [b]	38.8	**99.3**	**97.1**	0.500	0.886
	Rate4Site [b]	52.1	82.3	81.2	0.166	0.735
AMP	BGSVM-NUC	43.0	99.0	96.7	**0.512**	**0.895**
	TargetS [a]	44.5	98.7	96.5	0.503	0.884
	NsitePred [b]	42.3	98.7	**96.9**	0.501	0.876
	SVMPred [b]	33.5	**99.4**	96.7	0.478	0.870
	Rate4Site [b]	**52.0**	82.4	81.1	0.175	0.752
GDP	BGSVM-NUC	35.1	**99.6**	97.2	0.514	0.881
	TargetS [a]	45.9	99.4	**97.4**	0.571	**0.884**
	NsitePred [b]	**58.5**	98.5	97.0	**0.576**	0.867
	SVMPred [b]	51.1	98.8	97.1	0.553	0.855
	Rate4Site [b]	54.5	79.3	78.1	0.173	0.748
GTP	BGSVM-NUC	56.0	**99.6**	**97.5**	**0.687**	**0.913**
	TargetS [a]	**62.6**	98.7	97.0	0.653	0.910
	NsitePred [b]	60.4	98.8	96.9	0.640	0.909
	SVMPred [b]	48.5	99.3	96.9	0.602	0.887
	Rate4Site [b]	53.1	81.7	80.6	0.168	0.745

[a] Results are calculated by TargetS [42] models trained on the corresponding datasets of Train-NUC;
[b] Data are excerpted from [17].

there is room for further improvement due to two potential disadvantages. First, the dimension of feature in this study is fairly high, which may cause information redundancy. Therefore, reducing the feature dimension may be a promising way to further improve the accuracy of prediction. Another disadvantage is the relatively long computation time of BGSVM-NUC because BGSVM-NUC performs PSI-BLAST [50], PSIPRED [53], and LIBSVM [55] software in a linear manner to extract features and predict protein-nucleotide binding sites. In future work, we will try to speed up the computation by using multiple servers to concurrently perform these computations.

In addition, the BGSVM model, proposed in this work, is specially used to learn from an imbalanced dataset in the prediction of protein-binding residues. In the future, we will further investigate the ability of our model to other prediction problems that involve imbalanced datasets, such as protein-protein binding site prediction [61], and sumoylation site prediction in proteins [62].

ETHICS APPROVAL AND CONSENT TO PARTICIPATE

Not applicable.

HUMAN AND ANIMAL RIGHTS

No Animals/Humans were used for studies that are the basis of this research.

CONSENT FOR PUBLICATION

Not applicable.

AVAILABILITY OF DATA AND MATERIALS

The data and materials can be freely downloaded from http://csbio.njust.edu.cn/bioinf/BGSVM-NUC/.

FUNDING

This work was supported by the National Natural Science Foundation of China (No. 61772273, 61373062, 61876072, and 61902352), the Fundamental Research Funds for the Central Universities (No. 30918011104), and the National Key Research and Development Program: Key Projects of International Scientific and Technological Innovation Cooperation between Governments (No. 2016YFE0108000).

CONFLICT OF INTEREST

The authors declare no conflict of interest, financial or otherwise.

ACKNOWLEDGEMENTS

Declared none.

REFERENCES

[1] Gao, M.; Skolnick, J. The distribution of ligand-binding pockets around protein-protein interfaces suggests a general mechanism for pocket formation. *Proc. Natl. Acad. Sci. USA,* **2012**, *109*(10), 3784-3789. [http://dx.doi.org/10.1073/pnas.1117768109] [PMID: 22355140]

[2] Kokubo, H.; Tanaka, T.; Okamoto, Y. Ab initio prediction of protein-ligand binding structures by replica-exchange umbrella sampling simulations. *J. Comput. Chem.,* **2011**, *32*(13), 2810-2821. [http://dx.doi.org/10.1002/jcc.21860] [PMID: 21710634]

[3] Roy, A.; Zhang, Y. Recognizing protein-ligand binding sites by global structural alignment and local geometry refinement. *Structure,* **2012**, *20*(6), 987-997. [http://dx.doi.org/10.1016/j.str.2012.03.009] [PMID: 22560732]

[4] Yang, J.; Roy, A.; Zhang, Y. Protein-ligand binding site recognition using complementary binding-specific substructure comparison and sequence profile alignment. *Bioinformatics,* **2013**, *29*(20), 2588-2595. [http://dx.doi.org/10.1093/bioinformatics/btt447] [PMID: 23975762]

[5] Wang, C.; Liu, J.; Luo, F.; Deng, Z.; Hu, Q.N. Predicting target-ligand interactions using protein ligand-binding site and ligand substructures. *BMC Syst. Biol.,* **2015**, *9*(Suppl. 1), S2-S11. [http://dx.doi.org/10.1186/1752-0509-9-S1-S2] [PMID: 25707321]

[6] Chen, P.; Hu, S.; Zhang, J.; Gao, X.; Li, J.; Xia, J.; Wang, B. A sequence-based dynamic ensemble learning system for protein ligand-binding site prediction. *IEEE/ACM Trans. Comput. Biol. Bioinformatics,* **2016**, *13*(5), 901-912. [http://dx.doi.org/10.1109/TCBB.2015.2505286] [PMID: 26661785]

[7] Yu, D.J.; Hu, J.; Tang, Z.M.; Shen, H.B.; Yang, J.; Yang, J.Y. Improving protein-ATP binding residues prediction by boosting SVMs with random under-sampling. *Neurocomputing,* **2013**, *104*, 180-190. [http://dx.doi.org/10.1016/j.neucom.2012.10.012]

[8] Dundas, J.; Ouyang, Z.; Tseng, J.; Binkowski, A.; Turpaz, Y.; Liang, J. CASTp: computed atlas of surface topography of proteins with structural and topographical mapping of functionally annotated residues. *Nucleic Acids Res.,* **2006**, *34*(Web Server issue), W116-8. [http://dx.doi.org/10.1093/nar/gkl282] [PMID: 16844972]

[9] Brylinski, M.; Skolnick, J. A threading-based method (FINDSITE) for ligand-binding site prediction and functional annotation. *Proc. Natl. Acad. Sci. USA,* **2008**, *105*(1), 129-134. [http://dx.doi.org/10.1073/pnas.0707684105] [PMID: 18165317]

[10] Capra, J.A.; Laskowski, R.A.; Thornton, J.M.; Singh, M.; Funkhouser, T.A. Predicting protein ligand binding sites by combining evolutionary sequence conservation and 3D structure. *PLOS Comput. Biol.,* **2009**, *5*(12), e1000585. [http://dx.doi.org/10.1371/journal.pcbi.1000585] [PMID: 19997483]

[11] Hernandez, M.; Ghersi, D.; Sanchez, R. SITEHOUND-web: A server for ligand binding site identification in protein structures. *Nucleic Acids Res.,* **2009**, *37*(Web Server issue), W413-6. [http://dx.doi.org/10.1093/nar/gkp281] [PMID: 19398430]

[12] Wass, M.N.; Kelley, L.A.; Sternberg, M.J. 3DLigandSite: predicting ligand-binding sites using similar structures. *Nucleic Acids Res.,* **2010**, *38*(Web Server issue), W469-73. [http://dx.doi.org/10.1093/nar/gkq406] [PMID: 20513649]

[13] Pupko, T.; Bell, R.E.; Mayrose, I.; Glaser, F.; Ben-Tal, N. Rate4Site: An algorithmic tool for the identification of functional regions in proteins by surface mapping of evolutionary determinants within their homologues. *Bioinformatics,* **2002**, *18*(Suppl. 1), S71-S77. [http://dx.doi.org/10.1093/bioinformatics/18.suppl_1.S71] [PMID: 12169533]

[14] Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum Likelihood from Incomplete Data *via* the EM Algorithm. *J. R. Stat. Soc.,* **1977**, *39*, 1-38.

[15] Shu, N.; Zhou, T.; Hovmöller, S. Prediction of zinc-binding sites in proteins from sequence. *Bioinformatics,* **2008**, *24*(6), 775-782. [http://dx.doi.org/10.1093/bioinformatics/btm618] [PMID: 18245129]

[16] Suykens, J.A.; Vandewalle, J. Least squares support vector machine classifiers. *Neural Process. Lett.,* **1999**, *9*, 293-300. [http://dx.doi.org/10.1023/A:1018628609742]

[17] Chen, K.; Mizianty, M.J.; Kurgan, L. Prediction and analysis of nucleotide-binding residues using sequence and sequence-derived structural descriptors. *Bioinformatics,* **2012**, *28*(3), 331-341. [http://dx.doi.org/10.1093/bioinformatics/btr657] [PMID: 22130595]

[18] Panwar, B.; Gupta, S.; Raghava, G.P. Prediction of vitamin interacting residues in a vitamin binding protein using evolutionary information. *BMC Bioinformatics,* **2013**, *14*, 44-57. [http://dx.doi.org/10.1186/1471-2105-14-44] [PMID: 23387468]

[19] Yu, D.J.; Hu, J.; Huang, Y.; Shen, H.B.; Qi, Y.; Tang, Z.M.; Yang, J.Y. TargetATPsite: A template-free method for ATP-binding sites prediction with residue evolution image sparse representation and classifier ensemble. *J. Comput. Chem.,* **2013**, *34*(11), 974-985. [http://dx.doi.org/10.1002/jcc.23219] [PMID: 23288787]

[20] Chen, P.; Huang, J.Z.; Gao, X. LigandRFs: random forest ensemble to identify ligand-binding residues from sequence information alone. *BMC Bioinformatics,* **2014**, *15*(Suppl. 15), S4-S15. [http://dx.doi.org/10.1186/1471-2105-15-S15-S4] [PMID: 25474163]

[21] Liaw, A.; Wiener, M. Classification and regression by randomforest. *R News,* **2002**, *2/3*, 1-22.

[22] Chen, K.; Mizianty, M.J.; Kurgan, L. ATPsite: Sequence-based prediction of ATP-binding residues. *Proteome Sci.,* **2011**, *9*(Suppl. 1), S4. [http://dx.doi.org/10.1186/1477-5956-9-S1-S4] [PMID: 22165846]

[23] Yu, D.J.; Hu, J.; Yan, H.; Yang, X.B.; Yang, J.Y.; Shen, H.B. Enhancing protein-vitamin binding residues prediction by multiple heterogeneous subspace SVMs ensemble. *BMC Bioinformatics,* **2014**, *15*, 297-310. [http://dx.doi.org/10.1186/1471-2105-15-297] [PMID: 25189131]

[24] He, H.; Garcia, E.A. Learning from Imbalanced Data. *IEEE Trans. Knowl. Data Eng.,* **2009**, *21*, 1263-1284. [http://dx.doi.org/10.1109/TKDE.2008.239]

[25] Chawla, N.V.; Japkowicz, N.; Kotcz, A. Editorial: Special issue on learning from imbalanced data sets. *Sigkdd Explor.,* **2004**, *6*, 1-6. [http://dx.doi.org/10.1145/1007730.1007733]

[26] Ganganwar, V. An overview of classification algorithms for imbalanced datasets. *Int. J. Emerg. Technol. Adv. Eng.,* **2012**, *2*, 42-47.

[27] Guyon, I.J. Weston; S. Barnhill; V. Vapnik, gene selection for cancer classification using support vector machines. *Mach. Learn.,* **2002**, *46*, 389-422. [http://dx.doi.org/10.1023/A:1012487302797]

[28] Akbani, R.; Kwek, S.; Japkowicz, N. Applying support vector machines to imbalanced datasets. *Proceedings of European Conference on Machine Learning,* **2004**, pp. 39-50. [http://dx.doi.org/10.1007/978-3-540-30115-8_7]

[29] Wang, B.X.; Japkowicz, N. Boosting support vector machines for imbalanced data sets. *Knowl. Inf. Syst.,* **2010**, *25*, 1-20. [http://dx.doi.org/10.1007/s10115-009-0198-y]

[30] Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory,* **1953**, *13*, 21-27. [http://dx.doi.org/10.1109/TIT.1967.1053964]

[31] Keller, J.M.; Gray, M.R.; Givens, J.A. Fuzzy K-Nearest neighbor algorithm. *IEEE Trans. Syst. Man Cybern.,* **2012**, *SMC-15*, 580-585. [http://dx.doi.org/10.1109/TSMC.1985.6313426]

[32] Tan, S. Neighbor-weighted K-nearest neighbor for unbalanced text corpus. *Expert Syst. Appl.,* **2005**, *28*, 667-671. [http://dx.doi.org/10.1016/j.eswa.2004.12.023]

[33] Kang, P.; Cho, S. EUS SVMs: Ensemble of under-sampled svms for data imbalance problems. *Proceedings of International Conference on Neural Information Processing,* **2006**, pp. 837-846. [http://dx.doi.org/10.1007/11893028_93]

[34] He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning. *Proceedings of IEEE International Joint Conference on Neural Networks,* **2008**, pp. 1322-1328.

[35] Liu, Y.; Yu, X.; Huang, J.X.; An, A. Combining integrated sampling with svm ensembles for learning from imbalanced datasets. *Inf. Process. Manage.,* **2011**, *47*, 617-631. [http://dx.doi.org/10.1016/j.ipm.2010.11.007]

[36] Tong, S.; Koller, D. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.,* **2001**, *2*, 45-66.

[37] Ertekin, S.; Huang, J.; Giles, C.L. Active Learning for Class Imbalance Problem. *Proceedings of 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* **2007**, pp. 823-824.

[38] Wu, G.; Chang, E.Y. KBA: Kernel boundary alignment considering imbalanced data distribution. *IEEE Trans. Knowl. Data Eng.,* **2005**, *17*, 786-795. [http://dx.doi.org/10.1109/TKDE.2005.95]

[39] Hong, X.; Chen, S.; Harris, C.J. A kernel-based two-class classifier for imbalanced data sets. *IEEE Trans. Neural Netw.,* **2007**, *18*(1), 28-41. [http://dx.doi.org/10.1109/TNN.2006.882812] [PMID: 17278459]

[40] Jia, J.; Liu, Z.; Xiao, X.; Liu, B.; Chou, K.C. iCar-PseCp: Identify carbonylation sites in proteins by Monte Carlo sampling and incorporating sequence coupled effects into general PseAAC. *Oncotarget,* **2016**, *7*(23), 34558-34570. [http://dx.doi.org/10.18632/oncotarget.9148] [PMID: 27153555]

[41] Jia, J.; Liu, Z.; Xiao, X.; Liu, B.; Chou, K.C. iSuc-PseOpt: Identifying lysine succinylation sites in proteins by incorporating sequence-coupling effects into pseudo components and optimizing imbalanced training dataset. *Anal. Biochem.,* **2016**, *497*, 48-56. [http://dx.doi.org/10.1016/j.ab.2015.12.009] [PMID: 26723495]

[42] Yu, D.J.; Hu, J.; Yang, J.; Shen, H.B.; Tang, J.; Yang, J.Y. Designing template-free predictor for targeting protein-ligand binding sites with classifier ensemble and spatial clustering. *IEEE/ACM Trans. Comput. Biol. Bioinformatics,* **2013**, *10*(4), 994-1008. [http://dx.doi.org/10.1109/TCBB.2013.104] [PMID: 24334392]

[43] García, S.; Herrera, F. Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy. *Evol. Comput.,* **2009**, *17*(3), 275-306. [http://dx.doi.org/10.1162/evco.2009.17.3.275] [PMID: 19708770]

[44] Galar, M.; Fernández, A.; Barrenechea, E.; Herrera, F. EUSBoost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling. *Pattern Recognit.,* **2013**, *46*, 3460-3471. [http://dx.doi.org/10.1016/j.patcog.2013.05.006]

[45] Tang, Y.; Zhang, Y-Q. Granular SVM with repetitive undersampling for highly imbalanced protein homology prediction. *Proceedings of IEEE International Conference on Granular Computing,* **2006**, pp. 457-460.

[46] Tang, Y.; Zhang, Y-Q.; Chawla, N.V.; Krasser, S. SVMs modeling for highly imbalanced classification. *IEEE Trans. Syst. Man Cybern. B Cybern.,* **2009**, *39*(1), 281-288. [http://dx.doi.org/10.1109/TSMCB.2008.2002909] [PMID: 19068445]

[47] Yao, J.; Vasilakos, A.V.; Pedrycz, W. Granular computing: Perspectives and challenges. *IEEE Trans. Cybern.,* **2013**, *43*(6), 1977-1989. [http://dx.doi.org/10.1109/TSMCC.2012.2236648] [PMID: 23757594]

[48] Zhu, Y.H.; Hu, J.; Song, X.N.; Yu, D.J.; DNAPred: Accurate identification of DNA-binding sites from protein sequence by ensembled hyperplane-distance-based support vector machines. *J. Chem. Inf. Mod.*, **2019**, *59*(6), 3057-3071. [https://pubs.acs.org/doi/pdf/10.1021/acs.jcim.8b00749] [PMID: 30668479]

[49] Li, W.; Godzik, A. Cd-hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics,* **2006**, *22*(13), 1658-1659. [http://dx.doi.org/10.1093/bioinformatics/btl158] [PMID: 16731699]

[50] Schäffer, A.A.; Aravind, L.; Madden, T.L.; Shavirin, S.; Spouge, J.L.; Wolf, Y.I.; Koonin, E.V.; Altschul, S.F. Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. *Nucleic Acids Res.,* **2001**, *29*(14), 2994-3005. [http://dx.doi.org/10.1093/nar/29.14.2994] [PMID: 11452024]

[51] Bairoch, A.; Apweiler, R. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Res.,* **2000**, *28*(1), 45-48. [http://dx.doi.org/10.1093/nar/28.1.45] [PMID: 10592178]

[52] Zhang, Y.N.; Yu, D.J.; Li, S.S.; Fan, Y.X.; Huang, Y.; Shen, H.B. Predicting protein-ATP binding sites from primary sequence through fusing bi-profile sampling of multi-view features. *BMC Bioinformatics,* **2012**, *13*, 118-128. [http://dx.doi.org/10.1186/1471-2105-13-118] [PMID: 22651691]

[53] Jones, D.T. Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol. Biol.,* **1999**, *292*(2), 195-202. [http://dx.doi.org/10.1006/jmbi.1999.3091] [PMID: 10493868]

[54] Freund, Y.; Schapire, R.E. Experiments with a new bosting algorithm. *Proceedings of International Conference on Machine Learning,* **1996**, pp. 148-156.

[55] Chang, C.C.; Lin, C.J. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.,* **2011**, *2*, 1-27. [http://dx.doi.org/10.1145/1961189.1961199]

[56] Liu, G.H.; Shen, H.B.; Yu, D.J. Prediction of protein-protein interaction sites with machine-learning-based data-cleaning and post-filtering procedures. *J. Membr. Biol.,* **2016**, *249*(1-2), 141-153. [http://dx.doi.org/10.1007/s00232-015-9856-z] [PMID: 26563228]

[57] He, X.; Han, K.; Hu, J.; Yan, H.; Yang, J.Y.; Shen, H.B.; Yu, D.J. TargetFreeze: Identifying antifreeze proteins *via* a combination of weights using sequence evolutionary information and pseudo amino acid composition. *J. Membr. Biol.,* **2015**, *248*(6), 1005-1014. [http://dx.doi.org/10.1007/s00232-015-9811-z] [PMID: 26058944]

[58] Xiao, X.; Hui, M.; Liu, Z. iAFP-Ense: An ensemble classifier for identifying antifreeze protein by incorporating grey model and PSSM into PseAAC. *J. Membr. Biol.,* **2016**, *249*(6), 845-854. [http://dx.doi.org/10.1007/s00232-016-9935-9] [PMID: 27812737]

[59] Hu, J.; Zhou, X.; Zhu, Y.H.; Yu, D.J.; Zhang, G.; Target, D.B.P. TargetDBP: Accurate DNA-binding protein prediction *via* sequence-based multi-view feature learning. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, **2019**, 1-1. [http://dx.doi.org/ 10.1109/TCBB.2019.2893634] [PMID: 30668479]

[60] Ahmad, K.; Waris, M.; Hayat, M. Prediction of protein submitochondrial locations by incorporating dipeptide composition into Chou's general pseudo amino acid composition. *J. Membr. Biol.,* **2016**, *249*(3), 293-304. [http://dx.doi.org/10.1007/s00232-015-9868-8] [PMID: 26746980]

[61] Jia, J.; Liu, Z.; Xiao, X.; Liu, B.; Chou, K.C. iPPBS-Opt: A sequence-based ensemble classifier for identifying protein-protein binding sites by optimizing imbalanced training datasets. *Molecules,* **2016**, *21*(1), E95. [http://dx.doi.org/10.3390/molecules21010095] [PMID: 26797600]

[62] Jia, J.; Zhang, L.; Liu, Z.; Xiao, X.; Chou, K.C. pSumo-CD: Predicting sumoylation sites in proteins with covariance discriminant algorithm by incorporating sequence-coupled effects into general PseAAC. *Bioinformatics,* **2016**, *32*(20), 3133-3141. [http://dx.doi.org/10.1093/bioinformatics/btw387] [PMID: 27354696]