Mathematical Biosciences and Engineering

MBE, 19(1): 775–791.
DOI: 10.3934/mbe.2022035
Received: 23 August 2021
Accepted: 22 September 2021
Published: 22 November 2021

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

Research article

UPFPSR: a ubiquitylation predictor for plant through combining sequence information and random forest

Shuwan Yin[1], Jia Zheng[1], Cangzhi Jia[1,*], Quan Zou[2,3], Zhengkui Lin[4,*] and Hua Shi[5,*]

[1] School of Science, Dalian Maritime University, Dalian 116026, China
[2] Yangtze Delta Region Institute (Quzhou), University of Electronic Science and Technology of China, Quzhou, China
[3] Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu, China
[4] School of Maritime Economics and Management, Dalian Maritime University Dalian 116026, China
[5] School of Opto-electronic and Communication Engineering, Xiamen University of Technology, Xiamen, China

* **Correspondence:** Email: cangzhijia@dlmu.edu.cn, dalianjx@163.com, shihua@xmut.edu.cn.

Abstract: As one of the most significant protein post-translational modifications (PTMs) in eukaryotes, ubiquitylation plays an essential role in regulating diverse cellular functions, such as apoptosis, cell division, DNA repair and replication, intracellular transport and immune reactions. Traditional experimental methods have the defect of being time-consuming, costly and labor-intensive. Therefore, it is highly desired to develop automated computational methods that can recognize potential ubiquitylation sites rapidly and accurately. In this study, we propose a novel predictor, named UPFPSR, for predicting lysine ubiquitylation sites in plant. UPFPSR is developed using multiple physicochemical properties of amino acids and sequence-based statistical information. In order to find a suitable classification algorithm, four traditional algorithms and two deep learning networks are compared, and the random forest with superior performance is selected ultimately. An extensive benchmarking shows that UPFPSR outperforms the most advanced ubiquitylation prediction tool on each measurement indicator, with the accuracy of 77.3%, precision of 75%, recall of 81.7%, F1-score of 0.7824, and AUC of 0.84 on the independent test dataset. The results indicate that UPFPSR can provide new guidance for further experimental study on ubiquitylation. The data sets and source code used in this study are freely available at https://github.com/ysw-sunshine/UPFPSR.

Keywords: protein post-translational modifications; lysine ubiquitylation; traditional machine learning; deep learning; test evaluation

1. Introduction

To date, more than 400 different types of protein post-translational modifications (PTMs) have been discovered in the whole process of life activities [1,2]. Ubiquitylation is a modification process in which ubiquitin molecules may attach themselves to substrate proteins on lysine residues under the action of E1 activation enzyme, E2 conjugation enzyme and E3 ligation enzyme [3–5]. Ubiquitination is widely involved in various physiological processes due to its diversity and multivalence, including cell proliferation, apoptosis, autophagy, DNA damage repair and immune response [6,7]. Similar to the phosphorylation pathway, the ubiquitin modification pathway is reversible, that is, ubiquitin protein modifications can be removed by deubiquitinase. Therefore, it is difficult to study protein ubiquitination. However, there are many experimental approaches which have been developed, mainly including high-throughput mass spectrometry techniques, ubiquitin antibodies, ubiquitin binding proteins, and so on [8,9].

To study the ubiquitination of proteins, we need to make clear that: 1) Which proteins can be ubiquitinated; 2) For ubiquitinated proteins, which lysine residues can be ubiquitinated; 3) Quantitative analysis should be performed to find the motifs of ubiquitinated protein sequences. After clarifying the above points, we need to further understand how the ubiquitination happens, and what the key molecules that affect this ubiquitination process are. In other words, what is the role of E3 enzyme in this process? To complete these studies, a large amount of statistical data is needed. With the development of molecular biology and computer aided calculation, motif identification has become a helpful method for digging valuable information from biological sequences. Recently, a variety of machine learning approaches have been developed for automatic recognition of protein ubiquitylation sites.

The first online predictor for identifying protein ubiquitylation sites, called Ubipred [10], took advantage of 31 informative attributes out of 531 physicochemical properties as features, and support vector machine (SVM) as classifier. After that, many researches have proposed new models to predict ubiquitylation sites based on traditional classification algorithm, such as SVM [8,11–14], random forest (RF) [15,16], gray system model. In these studies, the composition of k-spaced amino acid pairs, binary amino acid encoding, physicochemical properties of amino acids, pseudo-amino acid composition, and so on, are adopted to characterize sequence information. Chen et al. [17] proposed a predictor called hCKSAAP_UbSite, which used SVM classifier incorporated with the composition of k-spaced amino acid pairs (CKSAAP), the binary amino acid encoding, the AAindex physicochemical property, and the protein aggregation propensity, to recognize protein ubiquitylation sites of human. Qiu et al. [18] developed a predictor called iUbiq-Lys, which adopted the evolutionary information, pseudo-amino acid composition (PseAAC), as well as the gray system model to predict protein ubiquitylation sites. Cai and Jiang [19] employed various traditional machine learning methods for the ubiquitylation site identification based on physicochemical properties of amino acids concerning protein sequences. Wang et al. [20] designed a tool, ESA-UbiSite, using physicochemical properties together with support vector machine to identify human ubiquitylation sites. And they also proposed the evolutionary screening algorithm (ESA) to select negative samples from non-validated sites effectively.

In recent years, deep learning has been extensively used in the field of bioinformatics [21–24]. In 2017, He et al. [21] proposed the first deep learning architecture by utilizing raw protein sequence fragments, selected physicochemical properties of amino acids, and corresponding position-specific scoring matrix as input. More recently, Wang et al. [24] proposed an improved training scheme with word-embedding model, incorporated with the multilayer convolutional neural network to predict plant ubiquitylation sites. Although many models have been proposed to recognize ubiquitination sites of different species, there are only several models are designed for plant [13,15,16,24]. The latest model is built by Wang et al. [24], which achieved the accuracies of 0.782 on 10-fold cross-validation and 0.756 on independent test, respectively. Comparing the performance with other species, there is still a lot of room to improve the prediction performance. One solution is to use more sequence order information and position information of whole sequences. Another solution is to consider both traditional classification algorithm and deep learning. For the convenience of research, we show the related works and major information in Table 1.

In this work, we present a novel prediction model called UPFPSR to further improve the predictive performance for plant potential ubiquitylation sites. We build and optimize our model from three aspects. First, in order to extract more effective and representative information from protein fragments, four sequence feature extraction methods, namely DBPB (di-amino acid bi-profile Bayes), EGAAC (enhanced grouped amino acid composition), Pse-AAC (pseudo-amino acid composition) and PWAA (position-weight amino acid composition) are used to transform sequence fragments of length 31 into numerical feature vectors efficiently. Second, deep learning algorithms and several extensively used traditional machine learning algorithms were compared during model construction, and random forest (RF) is the chosen classification algorithm to establish our lysine ubiquitination site prediction model UPFPSR. Last, we perform a 10-fold cross-validation test, as well as an independent test to compare and evaluate the performance of the constructed model objectively using five common measures, i.e., accuracy (Acc), precision, recall, F1-score and the area under the ROC curve (AUC) values. When compared with one of the most advanced prediction tool CNN+word2vec on an independent dataset, UPFPSR shows its advantage over CNN+word2vec with the accuracy of 77.3%, precision of 75.0%, recall of 81.7%, F1-score of 0.782, and AUC of 0.84.

2. Materials and methods

Figure 1 shows the overall framework of UPFPSR, which contains four major steps: 1) Data collection and preprocessing; 2) Feature encoding schemes; 3) Model construction; 4) Model evaluation. In the first step, the training dataset and the independent testing set originating from the PLMD database [25], are collected and pre-processed. Then in the second step, we adopt four different sequence-based feature-encoding techniques to extract effective feature vectors. We perform a 10-fold cross-validation on multiple classifiers in the third step to select the optimal model for plant ubiquitylated site recognition. Finally, the trained RF model is further evaluated by an independent test set, and the predictive performance is compared with the existing predictor of Wang et al. [24]. Details are described in following sections.

2.1. Data collection and pre-processing

In this study, we use datasets constructed by Wang et al. [24] to train and validate our model. The

experimentally verified lysine ubiquitylation proteins are collected from the PLMD database [25]. We select ubiquitylation sites from *Oryza sativa subsp indica*, *O. sativa subsp japonica*, and *Arabidopsis thaliana* for the plant subset. The protein peptide sequences of length 31 with experimentally verified ubiquitylation lysine in the center are collected as positive dataset. If the number of upstream or downstream amino acids is less than 15, the lacking amino acids are complemented with the same number of pseudo amino acid "X"s. The negative samples (non-ubiquitination sites) were generated by satisfying the requirement that 31 long sequences with lysine in the center. Meanwhile, the negative sample should not be annotated experimentally. After a series of treatment, a total of 7000 protein peptide sequences are obtained for species of plant with sequence similarity less than 30%, which contains 3500 positive peptides and 3500 negative peptides. We randomly select 2750 peptides from the 3500 positive peptides and 2750 peptides from the 3500 negative peptides, separately, as the positive training dataset and the negative training dataset. The training dataset, consisting of 5500 protein fragments, is used to train and optimize the prediction model. The remaining 1500 protein peptides including 750 positive samples and 750 negative samples are used to evaluate the generalization ability of the established predictor. For the specific procedure of constructing data set, please refer to the work of Wang et al. [24].

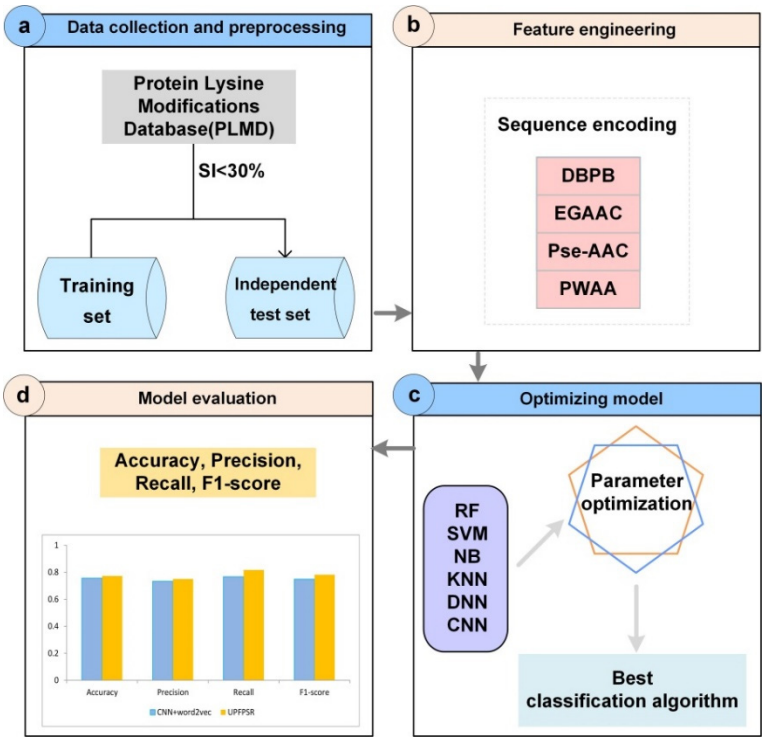

Figure 1. Overall framework of UPFPSR.

2.2. Sequence encoding schemes

Four different protein fragment encoding methods are used in this study, namely DBPB, EGAAC, Pse-AAC and PWAA. These encoding schemes consider 20 natural amino acids and a pse-amino acid (A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y, X) presented in protein sequence fragments, and transform them into numerical feature vectors.

Table 1. A comprehensive list of the existing methods for prediction of ubiquitylation sites.

Tools	Yaer	Reference	Features	Algorithm	Species
Ubipred	2008	Tung et al.	PCPs	SVM	Generic
hCKSAAP_UbSite	2013	Chen et al.	CKSAAP + BE + Aaindex + aggregation propensity	SVM	Human
UbiProber	2013	Chen et al.	AAC + PCPs + KNN	SVM	General and species-specific
iUbiq-Lys	2015	Qiu et al.	evolutionary information + PseAAC	Gray system model	Generic
Cai and Jiang, 2016	2016	Cai et al.	PCPs	NB + FSNB + MANB + EBMC + SVM + LR + LASSO	Generic
Nguyen et al., 2016	2016	Nguyen et al.	AAC + AAPC + PSSM	SVM	Generic
UbiSite	2016	Huang et al.	MDDLogo-identified substrate motifs + PSSM	a two-layered SVM model	Generic
ESA-UbiSite	2017	Wang et al.	PCPs	SVM	Human
He et al., 2018	2018	He et al.	One hot + PCPs + PSSM	CNN	Generic
UbiNets	2018	Yadav et al.	PCPs	DenseNet	Generic
DeepUbi	2019	Fu et al.	One-Hot + CKSAAP	CNN	Generic
AraUbiSite	2019	Chen et al.	AAC + CKSAAP	SVM	A. thaliana
UbiSitePred	2019	Cui et al.	BE + PseAAC + CKSAAP + PSPM	SVM	Generic
in silico	2019	Mosharaf et al.	BE	RF	A. thaliana
Mosharaf et al., 2020	2020	Mosharaf et al.	CKSAAP	RF	A. thaliana
Wang et al., 2020a	2020	Wang et al.	word embedding	CNN	Plant

Notes: A. thaliana: Arabidopsis thaliana; RF: Random forest; SVM: Support vector machine; CNN: Convolutional neural network; DenseNet: Densely connected convolutional neural networks; NB: Naïve bayes; FSNB: Feature selection NB; MANB: Model averaged NB; EBMC: Efficient bayesian multivariate classifier; LR: Logistic regression; LASSO: Least absolute shrinkage and selection operator; CKSAAP: Composition of k-spaced amino acid pairs; PCPs: Physicochemical properties; PseAAC: Pseudo amino acid composition; AAC: Amino acid composition; BE: Binary encoding; PSPM: Position-specific propensity matrices; AAPC: amino acid pair composition; PSSM: Position-specific scoring matrix; KNN: K nearest neighbor; AAindex : Amino acid index database.

2.2.1. Di-amino acid bi-profile Bayes (DBPB)

DBPB [26] considers the frequency of every two adjacent amino acids at each position in all the positive and negative samples. It has been widely used in the field of post translational modification of proteins. For example, Zhu et al. [27] proposed Inspector, a novel succinylation prediction tool that used random forest algorithm to identify key determinants of succinylation among six sequence-based features: di-amino acid bi-profile Bayes, position-specific di-amino acid propensity, pseudo-amino acid composition, position-weight amino acid composition, enhanced grouped amino acid composition and composition of *k*-spaced amino acid group pairs. Jia et al. [28] proposed an ensemble model O-GlcNAcPRED-II to predict O-GlcNAcylation sites by fusing multiple features incorporated di-amino acid bi-profile Bayes. Given a protein peptide sequence *S*, the DBPB feature vector is defined as:

$$P = \left(p_1, p_2, \cdots, p_{n-1}, p_n, \cdots, p_{2\times(n-1)} \right) \tag{1}$$

where n is the length of the sequence fragment after omitting amino acid K in the central position. (i.e., n=30), $p_j\,(j = 1, 2, \cdots, n-1)$ denotes the posterior probability of two adjacent amino acids at the j-th position in all positive samples, while $p_j\left(j = n, n+1, \cdots, 2(n-1)\right)$ represents the posterior probability of two adjacent amino acids at the $(j-n+1)$-th position in all negative samples.

2.2.2. Enhanced grouped amino acid composition (EGAAC)

The EGAAC feature encoding [29] firstly classifies the 20 amino acids into five categories, according to their physicochemical properties, e.g., charge, hydrophobicity and molecular size. These five categories are aliphatic group (g1): (G, A, V, L, M, I), aromatic group (g2): (F, Y, W), positive charge group (g3): (K, R, H), negative charge group (g4): (D, E) and uncharged group (g5): (S, T, C, P, N, Q), respectively. EGAAC descriptor calculates the frequency of each amino acid group in windows of fixed length, which is defined as:

$$f_{EGAAC}(g, w) = \frac{N(g, w)}{N(w)},\ g \in \{g1, g2, g3, g4, g5\},\ w \in \{w1, w2, \ldots, w31\} \tag{2}$$

where $N(w)$ is the size of the sliding window w, and $N(g, w)$ is the number of amino acids in group g within the sliding window w. In this study, we used the default setting w = 5, the size of the sliding window is 5.

2.2.3. Pseudo-amino acid composition (Pse-AAC)

Considering both local and global sequence-order information of protein sequences, Pse-AAC [30] has been developed and widely used to represent protein sequences [31–33]. Pse-AAC expresses the protein sequence as a $(20+\lambda)$-dimensional feature vector, where the first 20 dimensions contain information about the composition of amino acids, while the last λ-dimensional vector represents a range of physicochemical properties. This method can effectively avoid the loss of information in amino acid order and the loss of physicochemical information in protein sequence. The Pse-AAC feature vector for a protein sequence can be formulated as:

$$P = \left(p_1, p_2, \cdots, p_{n-1}, p_n, \cdots, p_{2\times(n-1)} \right) \tag{3}$$

where

$$d_t = \begin{cases} \dfrac{f_t}{\sum_{r=1}^{20} f_r + w\sum_{k=1}^{\lambda} \theta_k} & (1 \le t \le 20) \\ \dfrac{w\theta_{t\text{-}20}}{\sum_{r=1}^{20} f_r + w\sum_{k=1}^{\lambda} \theta_k} & (20+1 \le t \le 20+\lambda) \end{cases} \tag{4}$$

$f_r\ (r = 1,\ 2,\ \ldots,\ 20)$ represents the normalized occurrence frequency of 20 natural amino acids in the protein sequence [27,31]. Parameter λ is the integer representing the top counted grade (or rank) of the correlation along a protein sequence, and λ = 30 is adopted in this section. In addition, w (ranging from 0 to 1) is a weight factor used to improve accuracy and is set to w = 0.05 in this work. $\theta_k\ (k = 1,\ 2,\ \ldots,\ \lambda)$ is referred to as the *j*-tier correlation factor imaging the sequence-order correlation among all the *j*-th most contiguous residues along the protein chain. θ_k can be calculated as follows:

$$\theta_k = \frac{\sum_{i=1}^{L-k} \theta(R_i,\ R_{i+k})}{L-k} \quad (k \le \lambda) \tag{5}$$

The correlation function $\theta(R_i,\ R_{i+k})$ is given by:

$$\theta(R_i,\ R_{i+k}) = \frac{1}{\mu}\sum_{j=1}^{\mu}\left(I_j(R_i) - I_j(R_{i+k}) \right) \tag{6}$$

where μ is the number of physical and chemical indices considered, and $I_j(R_i)$ is the *j*-th physicochemical index value of the amino acid R_i. In this section, we employed thirty physicochemical properties, and so, μ is equal to 30 in Eq (6).

It should be noted that before replacing the physicochemical index values through Eq (6), the standard conversion described by the following formula is performed:

$$I_j(A_i) = \frac{I'(A_i) - \sum_{m=1}^{20} \frac{I'(A_i)}{20}}{\sqrt{\frac{\sum_{k=1}^{20} I'_j(x_k) - \sum_{m=1}^{20} \frac{I'(x_m)}{20}}{20}}} \tag{7}$$

where $I'_j(A_i)$ is the *j*-th original physical and chemical value of the *i*-th amino acid A_i. x_k and x_m represent 20 natural amino acids $(m, k = 1, 2,\ \ldots, 20)$. The amino acid "X" is omitted here. We apply the iLearn package [34] to calculate the Pse-AAC features.

2.2.4. Position-weight amino acid (PWAA) composition

PWAA [35] is the improvement of the traditional amino acid composition vector and can reflect the sequence position information of amino acid around the intermediate site. Therefore, this feature

can avoid the loss of sequence-order information effectively. Given a natural amino acid $A_i\,(i=1,2,\ldots,20)$, the position information of A_i in a protein peptide fragment can be calculated as follows:

$$w_i = \frac{1}{D(D+1)} \sum_{j=-D}^{D} x_{i,j} \left(j + \frac{|j|}{D} \right) \tag{8}$$

where D represents the number of upstream residues or downstream residues from the center site in a protein or peptide fragment. $x_{i,j}=1$ if A_i is the j-th residue in a protein peptide fragment, otherwise $x_{i,j}=0$. According to Eq (8), w_i demonstrates the distance information between A_i and intermediate site.

2.3. Model training

Random Forest (RF) [36] is an algorithm that integrates a lot of decision trees through the idea of ensemble learning, and has been widely used in computational biology, since it is non-parametric, efficient and interpretable. For example, it has been used in identifying protein succinylation sites [37–39], phosphorylation sites [40], glutarylation sites [41], et al..

The basic unit of random forest is a decision tree, which casts a unit vote for each subset of samples. Then the forest is constructed based on the majority voting strategy. In general, the number of trees has a great impact on the performance of the RF classification algorithm. Therefore, we search for optimal RF parameters in the training process, by setting the tree number from set $\{50,100,150,200,250,300\}$, respectively. The performance results are shown in Supplementary Table S1.

Support vector machine (SVM) is a classical machine learning method originally proposed in 1963 by Vapnik et al. [42], and has been widely used to solve data classification problems [11,43–46]. Based on the statistical learning theory, the main idea of SVM is to design a kernel function and look for the optimal separating maximum margin hyperplane which can differentiate between ubiquitylation sites and non-ubiquitylation sites. The Gaussian radial basis function (RBF) is adopted as the kernel function in this study. The RBF-SVM needs optimizing two key parameters: penalty parameter C and kernel parameter γ. For SVM algorithm, we apply grid search by setting $C \in \{2^2, 2^3, 2^4, 2^5\}$ and $\gamma \in \{0.01, 0.1, 1\}$, and finally select the pair of parameters which show the best prediction performance on the 10-fold cross validation.

The *k*-nearest neighbor (KNN) algorithm is the most common and simplest among all machine learning classifiers, and is also called lazy learning algorithm because it requires less training time [47–49]. KNN algorithm selects k training samples which are nearest to the input sample in the feature space according to certain decision rules. In this section, Euclidean distance is used as the measure of the difference between two data points and we assign the sample to be determined to the class label with the maximum voted class among these neighboring classes [50]. The Euclidean distance d between two samples x and y can be calculated through the following formula:

$$d(x,y) = \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2} \tag{9}$$

where N is the total number of feature dimensions, and x_i, y_i are the i-th dimension feature of x and y respectively.

It is important to select the optimal k value, which affects the performance of the KNN classifier significantly. We choose the optimal k value by 10-fold cross validation, and set the parameter k from set $\{3,5,7,9,11,13,15\}$.

The Naïve Bayes (NB) is an efficient statistical classification algorithm, constructing model through the joint probability $P(x,y) = P(x \mid y)P(y)$. It achieves posterior probability $P(y \mid x)$ based on Bayes theorem, and then the prediction is given according to the category label with the maximum posterior probability [50–52].

Despite its simplicity, the NB algorithm tends to outperform some more complex classification approaches, and is a widely used algorithm in bioinformatics researches. Because of its simple implementation, no iteration and high learning efficiency, Naïve Bayes algorithm has been extensively used in classification and other decision support applications [50,53,54].

2.4. Performance evaluation

To evaluate the performance of the constructed model and compare it with existing methods objectively, statistical analyses, including k-fold cross-validation tests and independent tests, are performed in this study. We also adopt four common performance measures including accuracy, precision, recall, and F1-score [24,55,56], which are defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{13}$$

where TP, FP, TN, and FN indicate the number of true positives, false positives, true negatives and false negatives, respectively. In addition, we plot the receiver operating characteristic (ROC) curves and calculate the area under curve (AUC) values to further assess the performance of our model. The higher the AUC value is, the better the model performs.

3. Results and discussion

3.1. Selection of the classification algorithm

To find the most suitable algorithm for distinguishing ubiquitination sites in plant, we firstly try four traditional algorithms, namely support vector machine (SVM), *k*-nearest neighbor (KNN), naive Bayes (NB), and random forest (RF). For each machine-learning algorithm, we perform 10-fold cross validation to search for optimal model parameters. The detailed performance results of each parameter combination are provided in supplementary Tables S1, S2 and S3.

When training the RF model, we set the number of decision trees as 50, 100, 150, 200, 250 and 300, separately, to screen the optimal number of decision trees. Table S1 demonstrates that the optimal accuracy achieves the best at 0.811 with tree number of 200. Moreover, other measurement precision, such as recall and F1-score, also reach the best values simultaneously.

During the process of SVM algorithm optimization, the penalty parameter C is selected from the set $\{2^2, 2^3, 2^4, 2^5\}$ and the kernel parameter γ is selected from the set $\{0.01, 0.1, 1\}$. Table S2 shows that the maximum accuracy achieves 0.747 by the parameter combinations C = 2^2, γ = 0.1 or C = 2^3, γ = 0.1.

For k-nearest neighbor algorithm, the parameter k is set as 3, 5, 7, 9, 11, 13 and 15 orderly. For each neighbor number k, the 10-fold cross-validation results are shown in Supplementary Table S3, where the k value of 11 achieves the best accuracy of 0.676.

We report the best performance for each classifier with the measures of accuracy, precision, recall, and F1-score in Table 2 and Figure 2. Moreover, we also plot their ROC curves and provide the corresponding AUC values, respectively, as shown in Figure 3. It is noted that the RF classification algorithm demonstrates its superiority to other classifiers on all of these measures. Specifically, the optimal model RF on 10-fold cross validation achieves an accuracy of 81.1%, precision of 81.0%, recall of 81.2%, F1-score of 0.811, and AUC of 0.888.

Table 2. Performance comparison of six different classifiers on 10-fold cross-validation test.

Algorithm	Accuracy	Precision	Recall	F1-score
NB	0.700	0.707	0.684	0.695
KNN	0.676	0.645	0.781	0.707
SVM	0.747	0.748	0.745	0.746
DNN	0.743	0.747	0.735	0.741
CNN	0.698	0.701	0.691	0.696
RF	**0.811**	**0.810**	**0.812**	**0.811**

Due to their outperforming learning ability, deep learning algorithms have also been extensively applied in prediction realms [23,56–59]. We test two deep neural network architectures, deep neural network (DNN) and convolutional neural network (CNN) on our training data as well. The 10-fold cross-validation prediction results show that RF is superior to CNN and DNN once again, with accuracy of 0.811 versus 0.698 of CNN and 0.743 of DNN. As far as we know, deep learning frameworks tend to show good performance on large-scale data. The plant data size we used is not large enough for deep neural networks, but suitable for traditional machine learning algorithms. If enough ubiquitination sites are discovered, deep learning framework will still be the first choice in the future.

3.2. Comparison of UPFPSR with existing method CNN+word2vec on the independent test set

As far as we know, the most up-to-date ubiquitylation prediction tool CNN+word2vec [24]

demonstrated the best performance compared to other state-of-the art methods [8,12,18,21,23]. Therefore, we only compare the predictive performance of UPFPSR with CNN+word2vec on the independent test dataset including 750 positive samples and 750 negative samples. The specific performance comparison results are shown in Figure 4 and Table S4. The results illustrate that between these two predictors, UPFPSR achieves a better predictive performance than CNN+word2vec on each measurement index. In particular, UPFPSR achieves the recall value of 81.7%, while the recall value of CNN+word2vec is 76.7%. In addition, UPFPSR improves the accuracy by 1.7% over CNN+word2vec. What is more, UPFPSR gives the AUC of 0.84 versus 0.81 of CNN+word2vec. In conclusion, all results demonstrate that our proposed model has high confidence on plant ubiquitylation site prediction and is more appropriate for recognizing the plant ubiquitylation site.

3.3. Amino acid preferences of ubiquitylation sites

We analyze the amino acid preferences around ubiquitylation sites as compared with non-ubiquitylation sites using the Two Sample Logo [60] web server and show the statistical results in Figure 5. The larger font indicates that this kind of amino acid is enriched in this position, which is statistically significant. It is clearly shown that arginine (R) and glutamic acid (E) are more likely to appear around ubiquitylated lysine than non-ubiquitylated lysine in plant, especially on the -11th to -1th and 1th to 10th positions. By contrast, serine (S) and lysine (K) are less likely to appear in ubiquitylated peptides than in non-ubiquitylated peptides. This analysis indicates that there is no obvious motif for ubiquitylated sites. Therefore, it is significant to build a model for the prediction of plant ubiquitylation sites.

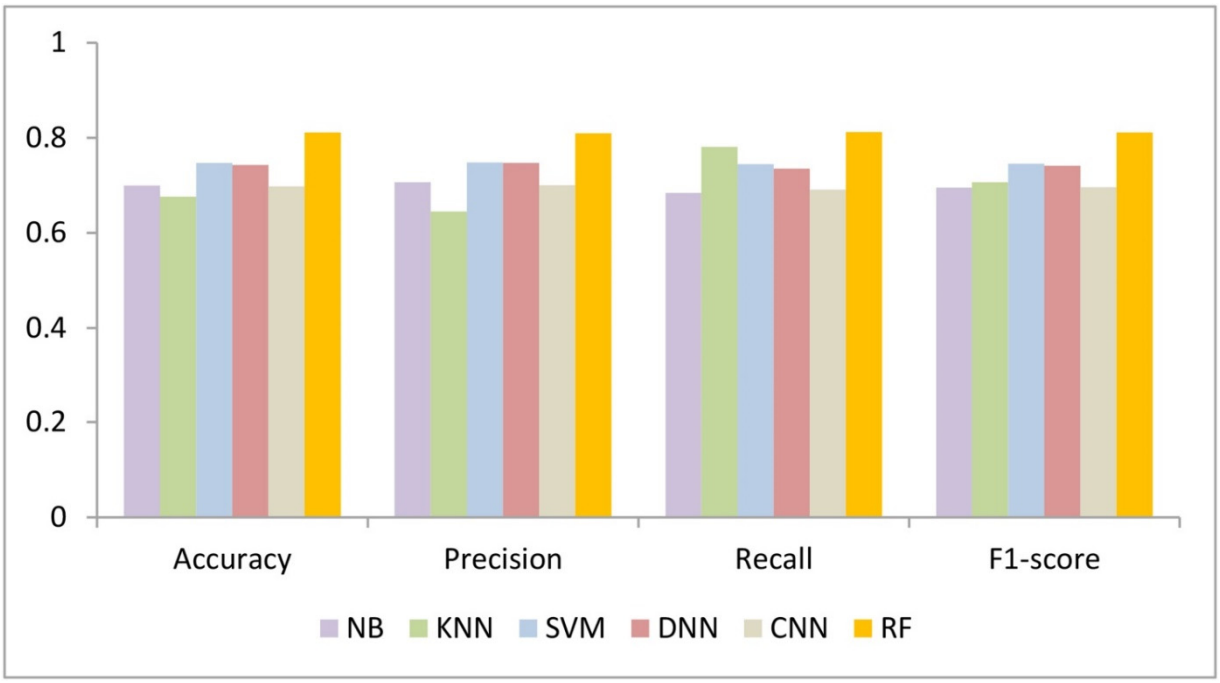

Figure 2. Performance comparison of six classification algorithms on 10-fold cross-validation test.

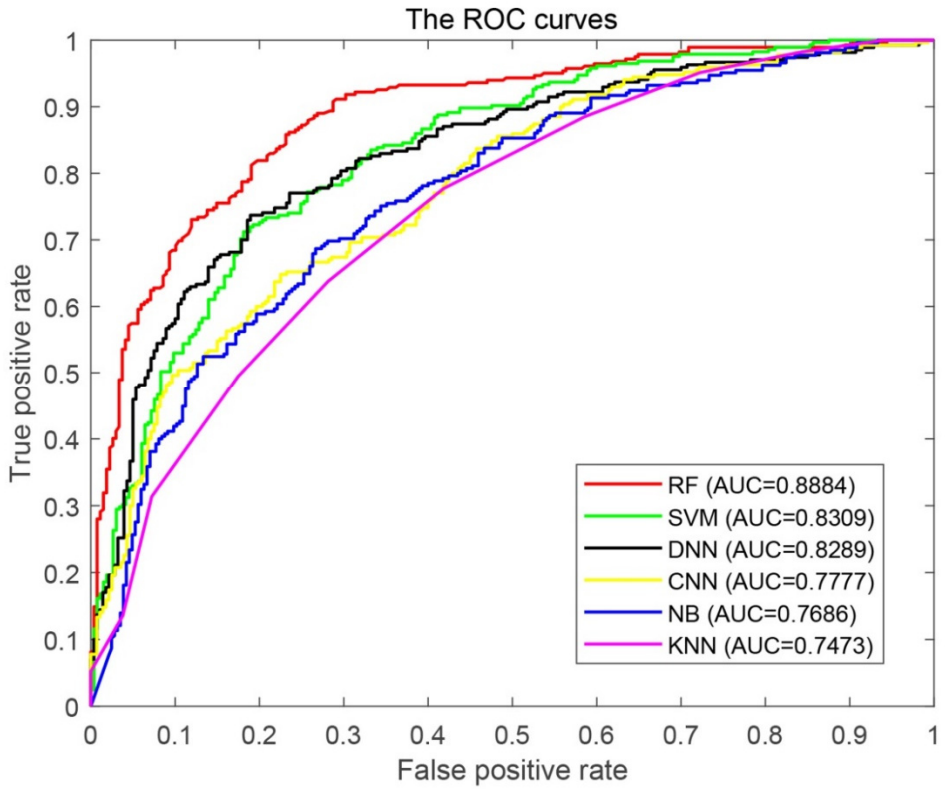

Figure 3. ROC curves of six different classifiers on 10-fold cross-validation test.

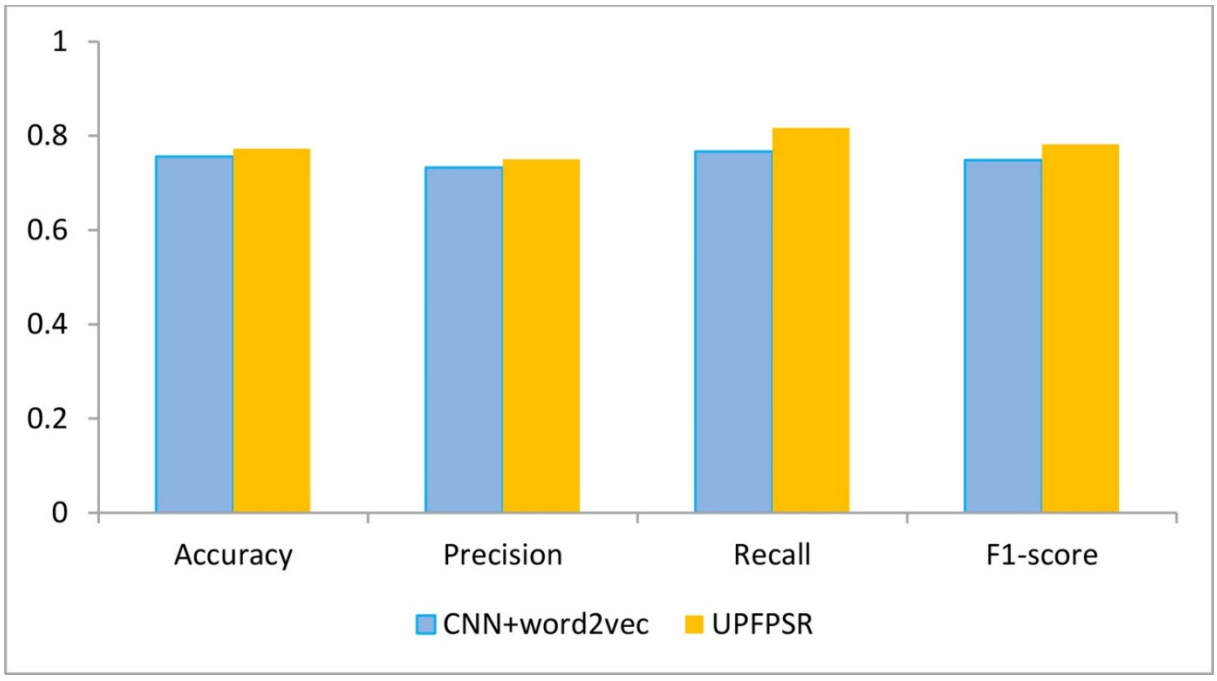

Figure 4. Performance comparison results between UPFPSR and CNN+word2vec on the independent test dataset for *plant.*

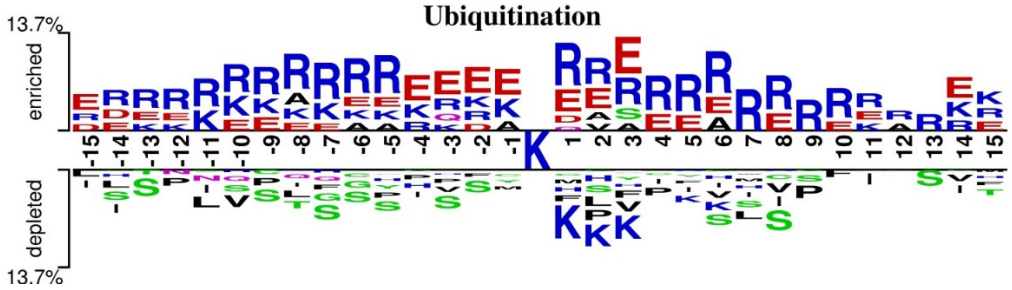

Figure 5. Two-sample Logo (http://www.twosamplelogo.org/) of the amino acid preferences around ubiquitylation sites as compared with non-ubiquitylation sites ($P < 0.05$; t-test).

4. Results

In this study, a novel model named UPFPSR is developed to identify lysine ubiquitylation sites for plant. UPFPSR incorporates various sequence-based information including DBPB, EGAAC, Pse-AAC and PWAA. Rigorous benchmarking tests based on 10-fold cross validation and an independent test set have illustrated that this novel method is efficient and promising for improving the prediction of lysine ubiquitylation sites. But the overall prediction performance is less than 90%, the model still needs to be further improved. Besides the types of features adopted, more secondary and tertiary structure information of proteins should be considered in the future. In addition, more up-to-date deep learning neural networks ensemble with traditional algorithms can be employed for further researches.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No: 62071079 and 61803065).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. A. R. Farley, A. J. Link, Identification and quantification of protein posttranslational modifications, *Methods Enzymol.*, **463** (2009), 725–763. doi: 10.1016/S0076-6879(09)63040-8.
2. J. Jia, Y. Shen, W. Qiu, Identifying Lysine Succinylation Sites in Proteins by Broad Learning System and Optimizing Imbalanced Training Dataset via Randomly Labeling Samples, *Wuhan Univ. J. Nat. Sci.*, (2021), 81–88. doi: 10.19823/j.cnki.1007-1202.2021.0005.
3. C. Ou, H. Pi, C. Chien, Control of protein degradation by E3 ubiquitin ligases in Drosophila eye development, *Trends Genet.*, **19** (2003), 382–389. doi: 10.1016/S0168-9525(03)00146-X.
4. J. Herrmann, L. Lerman, A. Lerman, Ubiquitin and Ubiquitin-Like Proteins in Protein Regulation, *Circul. Res.*, **100** (2007), 1276–1291. doi: 10.1161/01.res.0000264500.11888.f0.
5. R. Welchman, C. Gordon, R. Mayer, Ubiquitin and ubiquitin-like proteins as multifunctional signals, *Nat. Rev. Mol. Cell Biol.*, **6** (2005), 599–609. doi: 10.1038/nrm1700.

6. Y. Tu, C. Chen, J. Pan, J. Xu, Z. Zhou, C. Wang, The Ubiquitin Proteasome Pathway (UPP) in the regulation of cell cycle control and DNA damage repair and its implication in tumorigenesis, *Int. J. Clin. Exp. Pathol.*, **5** (2012), 726–738. doi: 10.3109/15513815.2012.659410.
7. A. Schwartz, A. Ciechanover, The ubiquitin-proteasome pathway and pathogenesis of human diseases, *Annu. Rev. Med.*, **50** (1999), 57–74. doi: 10.1146/annurev.med.50.1.57.
8. X. Chen, J. Qiu, S. Shi, S. Suo, S. Huang, R. Liang, Incorporating key position and amino acid residue features to identify general and species-specific Ubiquitin conjugation sites, *Bioinformatics*, **29** (2013), 1614–1622. doi: 10.1093/bioinformatics/btt196.
9. W. Qiu, C. Xu, X. Xiao, D. Xu, Computational Prediction of Ubiquitination Proteins Using Evolutionary Profiles and Functional Domain Annotation, *Curr. Genomics*, **20** (2019), 389–399. doi: 10.2174/1389202919666191014091250.
10. C. Tung, S. Ho, Computational identification of ubiquitylation sites from protein sequences, *BMC Bioinf.*, **9** (2008), 310. doi: 10.1186/1471-2105-9-310.
11. V. Nguyen, K. Huang, C. Huang, K. Lai, T. Lee, A new scheme to characterize and identify protein ubiquitination sites, *IEEE/ACM Trans. Comput. Biol. Bioinf.*, **14** (2016), 393–403. doi: 10.1109/TCBB.2016.2520939.
12. C. Huang, M. Su, H. Kao, J. Jhong, S. Weng, T. Lee, UbiSite: incorporating two-layered machine learning method with substrate motifs to predict ubiquitin-conjugation site on lysines, *BMC Syst. Biol.*, **10** (2016), 6. doi: 10.1186/s12918-015-0246-z.
13. J. Chen, J. Zhao, S. Yang, Z. Chen, Z. Zhang, Prediction of protein ubiquitination sites in Arabidopsis thaliana, *Curr. Bioinf.*, **14** (2019), 614–620. doi: 10.2174/1574893614666190311141647.
14. X. Cui, Z. Yu, B. Yu, M. Wang, B. Tian, Q. Ma, UbiSitePred: A novel method for improving the accuracy of ubiquitination sites prediction by using LASSO to select the optimal Chous pseudo components, *Chemometr. Intellig. Lab. Syst.*, **184** (2019), 28–43. doi: 10.1016/j.chemolab.2018.11.012.
15. M. Mosharaf, F. Ahmed, M. Hassan, S. Tasmia, M. Mollah, In Silico Prediction of Protein Ubiquitination Sites by Using Binary Encoding on Arabidopsis thaliana, *Int. J. Statist. Sci.*, **18** (2019), 65–76.
16. M. Mosharaf, M. Hassan, F. Ahmed, M. Khatun, M. Moni, M. Mollah, Computational prediction of protein ubiquitination sites mapping on Arabidopsis thaliana, *Comput. Biol. Chem.*, **85** (2020), 107238. doi: 10.1016/j.compbiolchem.2020.107238.
17. Z. Chen, Y. Zhou, J. Song, Z. Zhang, hCKSAAP_UbSite: improved prediction of human ubiquitination sites by exploiting amino acid pattern and properties, *BBA-Proteins Proteomics*, **1834** (2013), 1461–1467. doi: 10.1016/j.bbapap.2013.04.006.
18. W. Qiu, X. Xiao, W. Lin, K. Chou, iUbiq-Lys: prediction of lysine ubiquitination sites in proteins by extracting sequence evolution information via a gray system model, *J. Biomol. Struct. Dyn.*, **33** (2015), 1731–1742. doi: 10.1080/07391102.2014.968875.
19. B. Cai, X. Jiang, Computational methods for ubiquitination site prediction using physicochemical properties of protein sequences, *BMC Bioinf.*, **17** (2016), 1–12. doi: 10.1186/s12859-016-0959-z.
20. J. Wang, W. Huang, M. Tsai, K. Hsu, H. Huang, S. Ho, ESA-UbiSite: accurate prediction of human ubiquitination sites by identifying a set of effective negatives, *Bioinformatics*, **33** (2017), 661–668. doi: 10.1093/bioinformatics/btw701.

21. F. He, R. Wang, J. Li, L. Bao, D. Xu, X. Zhao, Large-scale prediction of protein ubiquitination sites using a multimodal deep architecture, *BMC Syst. Biol.*, **12** (2018), 81–90. doi: 10.1186/s12918-018-0628-0.
22. S. Yadav, M. Gupta, A. Bist, Prediction of ubiquitination sites using UbiNets, *Adv. Fuzzy Syst.*, **2018** (2018). doi: 10.1155/2018/5125103.
23. H. Fu, Y. Yang, X. Wang, H. Wang, Y. Xu, DeepUbi: a deep learning framework for prediction of ubiquitination sites in proteins, *BMC Bioinf,*, **20** (2019), 86. doi: 10.1186/s12859-019-2677-9.
24. H. Wang, Z. Wang, Z. Li, T. Lee, Incorporating Deep Learning With Word Embedding to Identify Plant Ubiquitylation Sites, *Front. Cell. Dev. Biol.*, **8** (2020), 572195. doi: 10.3389/fcell.2020.572195.
25. H. Xu, J. Zhou, S. Lin, W. Deng, Y. Xue, PLMD: An updated data resource of protein lysine modifications, *J. Genet. Genomics*, **44** (2017), 243–250. doi: 10.1016/j.jgg.2017.03.007.
26. B. Li, L. Hu, S. Niu, Y. Cai, K. Chou, Predict and analyze S-nitrosylation modification sites with the mRMR and IFS approaches, *J. Proteomics*, **75** (2012), 1654–1665. doi: 10.1016/j.jprot.2011.12.003.
27. Y. Zhu, C. Jia, F. Li, J. Song, Inspector: a lysine succinylation predictor based on edited nearest-neighbor undersampling and adaptive synthetic oversampling, *Anal. Biochem.*, **593** (2020), 113592. doi: 10.1016/j.ab.2020.113592.
28. C. Jia, Y. Zuo, Q. Zou, O-GlcNAcPRED-II: an integrated classification algorithm for identifying O-GlcNAcylation sites based on fuzzy undersampling and a K-means PCA oversampling technique, *Bioinformatics*, **34** (2018), 2029–2036. doi: 10.1093/bioinformatics/bty039.
29. Z. Chen, P. Zhao, F. Li, L. André, T. Marquez-Lago, Y. Wang, et al., iFeature: a python package and web server for features extraction and selection from protein and peptide sequences, *Bioinformatics*, **34** (2018), 2499–2502. doi: 10.1093/bioinformatics/bty140.
30. H. Shen, K. Chou, PseAAC: A flexible web server for generating various kinds of protein pseudo amino acid composition, *Anal. Biochem.*, **373** (2008), 386–388. doi: 10.1016/j.ab.2007.10.012.
31. T. Li, R. Song, Q. Yin, M. Gao, Y. Chen, Identification of S-nitrosylation sites based on multiple features combination, *Sci. Rep.*, **9** (2019), 3098. doi: 10.1038/s41598-019-39743-9.
32. Q. Wuyun, W. Zheng, Y. Zhang, J. Ruan, G. Hu, Improved Species-Specific Lysine Acetylation Site Prediction Based on a Large Variety of Features Set, *PLoS One*, **11** (2016), e0155370. doi: 10.1371/journal.pone.0155370.
33. W. Qiu, A. Xu, Z. Xu, C. Zhang, X. Xiao, Identifying Acetylation Protein by Fusing Its PseAAC and Functional Domain Annotation, *Front. Bioeng. Biotechnol.*, **7** (2019), 311. doi: 10.3389/fbioe.2019.00311.
34. Z. Chen, P. Zhao, F. Li, T. Marquez-Lago, A. Leier, J. Revote, et al., iLearn: an integrated platform and meta-learner for feature engineering, machine learning analysis and modeling of DNA, RNA and protein sequence data, *Brief. Bioinf.*, **21** (2019), 1047–1057. doi: 10.1093/bib/bbz041.
35. S. Shi, J. Qiu, X. Sun, S. Suo, S. Huang, R. Liang, PMeS: Prediction of Methylation Sites Based on Enhanced Feature Encoding Scheme, *PLoS One*, **7** (2012), e38772. doi: 10.1371/journal.pone.0038772.
36. L. Breiman, Random forests, *Mach. Learn.*, **45** (2001), 5–32. doi: 10.1023/A:1010933404324.
37. J. Jia, Z. Liu, X. Xiao, B. Liu, K. Chou, pSuc-Lys: Predict lysine succinylation sites in proteins with PseAAC and ensemble random forest approach, *J. Theor. Biol.*, **394** (2016), 223–230. doi: 10.1016/j.jtbi.2016.01.020.

38. M. Hasan, S. Yang, Y. Zhou, M. Mollah, SuccinSite: a computational tool for the prediction of protein succinylation sites by exploiting the amino acid patterns and properties, *Mol. Biosyst.*, **12** (2016), 786–795. doi: 10.1039/C5MB00853K.
39. M. Hasan, M. Khatun, M. Mollah, C. Yong, D. Guo, A systematic identification of species-specific protein succinylation sites using joint element features information, *Int. J. Nanomed.*, **12** (2017), 1–13. doi: 10.2147/IJN.S140875.
40. H. Ismail, A. Jones, J. Kim, R. Newman, D. Kc, RF-Phos: A Novel General Phosphorylation Site Prediction Tool Based on Random Forest, *BioMed. Res. Int.*, **2016** (2016), 3281590. doi: 10.1155/2016/3281590.
41. H. AL-barakati, H. Saigo, R. Newman, B. Dukka, RF-GlutarySite: a random forest based predictor for glutarylation sites, *Mol. Omics*, **15** (2019), 189–204. doi: 10.1039/C9MO00028C.
42. V. Vapnik, A. Lerner, Recognition of patterns with help of generalized portraits, *Avtomat. Telemekh.*, **24** (1963), 774–780.
43. C. Jia, M. Zhang, C. Fan, F. Li, J. Song, Formator: predicting lysine formylation sites based on the most distant undersampling and safe-level synthetic minority oversampling, *IEEE/ACM Trans. Comput. Biol. Bioinf.*, **18** (2019), 1937–1945. doi: 10.1109/TCBB.2019.2957758.
44. R. Wang, Z. Wang, H. Wang, Y. Pang, T. Lee, Characterization and identification of lysine crotonylation sites based on machine learning method on both plant and mammalian, *Sci. Rep.*, **10** (2020), 20447. doi: 10.1038/s41598-020-77173-0.
45. S. Suo, J. Qiu, S. Shi, X. Sun, S. Huang, X. Chen, et al., Position-specific analysis and prediction for protein lysine acetylation based on multiple features, *PLoS One*, **7** (2012), e49108. doi: 10.1371/journal.pone.0049108.
46. H. Al-Barakati, E. McConnell, L. Hicks, L. Poole, R. Newman, SVM-SulfoSite: a support vector machine based predictor for sulfenylation sites, *Sci. Rep.*, **8** (2018), 1–9. doi: 10.1038/s41598-018-29126-x.
47. J. Raikwal, K. Saxena, Performance Evaluation of SVM and K-Nearest Neighbor Algorithm over Medical Data set, *Int. J. Comput. Appl.*, **50** (2012), 35–39. doi: 10.5120/7842-1055.
48. Q. Ning, Z. Ma, X. Zhao, dForml(KNN)-PseAAC: Detecting Formylation sites from protein sequences using K-nearest neighbor algorithm via Chous 5-step rule and Pseudo components, *J. Theor. Biol.*, **470** (2019), 43–49. doi: 10.1016/j.jtbi.2019.03.011.
49. K. Mittal, G. Aggarwal, P. Mahajan, Performance study of K-nearest neighbor classifier and K-means clustering for predicting the diagnostic accuracy, *Int. J. Inf. Technol.*, **11** (2019), 535–540. doi: 10.1007/s41870-018-0233-x.
50. A. Singh, N. Malka, R. Lakshmiganthan, Impact of Different Data Types on Classifier Performance of Random Forest, Naïve Bayes, and K-Nearest Neighbors Algorithms, *Int. J. Adv. Comput. Sci. Appl.*, **8** (2017), 1–10. doi: 10.14569/ijacsa.2017.081201.
51. M. Khatun, M. Hasan, Prediction of protein Post-Translational Modification sites: An overview, *Ann. Proteom. Bioinf.*, **2** (2018), 49–57. doi: 10.29328/journal.apb.1001005.
52. A. Zamir, H. Khan, T. Iqbal, N. Yousaf, F. Aslam, A.Anjum, et al., Phishing web site detection using diverse machine learning algorithms, *Electron. Libr.*, **38** (2020), 65–80. doi: 10.1108/EL-05-2019-0118.
53. Y. Pan, H. Gao, H. Lin, Z. Liu, L. Tang, S. Li, Identification of Bacteriophage Virion Proteins Using Multinomial Nave Bayes with g-Gap Feature Tree, *Int. J. Mol. Sci.*, **19** (2018), 1779. doi: 10.3390/ijms19061779.

54. G. Webb, N. Bayes, Encyclopedia of Machine Learning, in *Springer US* (eds. C. Sammut and G. I. Webb), Academic Press, (2010), 613–624. doi: 10.1007/978-0-387-30164-8.
55. F. Li, J. Chen, Z. Ge, Y. Wen, Y. Yue, M. Hayashida, et al., Computational prediction and interpretation of both general and specific types of promoters in Escherichia coli by exploiting a stacked ensemble-learning framework, *Brief. Bioinf.*, **22** (2020), 2126–2140. doi: 10.1093/bib/bbaa049.
56. R. Xie, J. Li, J. Wang, W. Dai, A. Leier, T. Marquez-Lago, et al., DeepVF: a deep learning-based hybrid framework for identifying virulence factors using the stacking strategy, *Brief. Bioinf.*, **22** (2020). doi: 10.1093/bib/bbaa125.
57. M. Wu, S. Pan, L. Du, X. Zhu, Learning Graph Neural Networks with Positive and Unlabeled Nodes, preprint, arXiv:2103.04683.
58. A. Strokach, T. Lu, P. Kim, ELASPIC2 (EL2): Combining Contextualized Language Models and Graph Neural Networks to Predict Effects of Mutations, *J. Mol. Biol.*, **433** (2021), 166810. doi: 10.1016/j.jmb.2021.166810.
59. L. Zhang, P. Yang, H. Feng, Q. Zhao, H. Liu, Using network distance analysis to predict lncRNA–miRNA interactions, *Interdiscip. Sci.*, **13** (2021), 535–545. doi: 10.1007/s12539-021-00458-z.
60. V. Vacic, L. Iakoucheva, P. Radivojac, Two Sample Logo: a graphical representation of the differences between two sets of sequence alignments, *Bioinformatics*, **22** (2006), 1536–1537. doi: 10.1093/bioinformatics/btl151.

AIMS Press