PU-Palm: A PU-learning-based model for predicting plant S-acylation sites

ABSTRACT

Precisely identifying S-acylation sites is the basis of studying its biological function. While S-acylation in mammalian proteins has been extensively studied, the research on S-acylation in plants is still in its infancy. At present, only a small quantity of S-acylation sites in Arabidopsis proteins have been experimentally identified, and a larger number of S-acylation sites on proteins still remains unexplored. Under this circumstance, modeling with supervised learning lacks of a large amount of precisely labeled positive and negative data, and will lead to higher false positive rate if regarding unlabeled data as negative directly. To address this problem and make full use of unlabeled data, a PU-learning-based model, PU-Palm, was proposed in this study, which is the first PU-learning-based model used for plant S-acylation sites identification. Firstly, peptide sequences with experimentally identified S-acylation sites and unlabeled sequences were collected. Afterwards, features such as AAC, CKSAAP, PSSM and etc were extracted. Following this, PU extra trees were used to assess the significance of features for feature selection, and 213 key features were obtained. Ultimately, the PU-learning-based model was constructed with biased SVM algorithm. Our results demonstrate that AUC of our model is higher than 0.95, which indicates the advancement of our model on tasks of plant S-acylation sites identification.

0 INTRODUCTION

S-acylation, also known as palmitoylation, is a reversible post-translational modification[1]. It is a process of the addition of a saturated 16-carbon fatty acid, palmitate, to a cysteine residue of a protein through a thioester bond[2]. During the past forty years, extensive research has facilitate better understanding of how protein palmitoylation regulating protein function in a variety of biological processes. Especially since the first S-acyltransferase from yeast was discovered in 2002, the study on S-acylation has remarkably accelerated[3]. Nevertheless, compared to the great progress of the research on S-acylation in yeast and mammals, the study of plant S-acylation is still in its early stages. The knowledge of how S-acylation may affect protein function remains comparatively limited in plants, largely due to the lack of basic information regarding the specific proteins that undergo S-acylation and the location within the proteins where the modification occurs[4]. So far, the understanding of the function of S-acylation in plant protein mainly come from the limited information from targeted study of individual protein happened to be S-acylated[3].

But there is with no doubt that S-acylation impacts a large range of proteins with diverse functions, such as cellulose synthase complex subunits, heterotrimeric G protein subunits, small G proteins, involved in Ca^{2+} signaling, pathogenesis related proteins, transcription factors, among others, thereby impact a variety of events in plant life cycle[5]. In general, S-acylation alters protein interactions with membranes and has been usually described to affect subcellular and sub-membrane distribution, protein activation state, protein-protein interactions, protein turnover and protein conformation[6]. For example, within the cellulose synthase complex responsible for cellulose synthesis, all three CesA (cellulose synthase A) family proteins are S-acylated. And research indicates that mutation of the S-acylation sites in one of the three subunits will render the whole complex non-functional[6]. Heterotrimeric G proteins are another kind of S-acylated protein. They are important signal transduction components, affecting almost every aspect of plant life, such as cell division[7], pathogen defense responses[8], and hormone signaling[9]. Research demonstrates that S-acylation site plays an important role in stabilizing the newly formed G proteins and the process of targeting Arabidopsis GPA1 (the α subunits of G protein) to the plasma membrane[10]. Additionally, a lot of plant proteins involved in Ca^{2+} signaling have also been found to be S-acylated, such as CBL1[11] and CBL2[12] in Arabidopsis, LeCRK1 in tomato[13] and etc. For instance, the trafficking and anchoring of CBL1 to plasma membrane both depend on S-acylation[11].

The significant effect of S-acylation on plant growth, development and environmental adaption[14] underscores the vital importance of identifying S-acylation sites. Common methods for detecting S-acylation include mutational analysis[15], inhibition of S-acylation through PAT inhibitors[16], gas chromatography-mass spectrometry (GC-MS) analysis[11] and etc. However, these traditional biological methods of identifying S-acylation sites based on experiment is costly and time consuming. Thus, only a small part of S-acylation sites could be identified, making it not suitable for large-scale identification.

Considering that numerous potential S-acylation sites yet to be identified in the complete plant proteome, using machine learning method to predict these sites provides valuable support for researchers in conducting more targeted identification. This approach is of great importance for advancing relevant research in this field. In 2006, Zhou et al. developed CSS-Palm[17], a palmitoylation site prediction system based on a clustering and scoring strategy. This is followed by the development of NBA-Palm by Xue et al[18], a novel computational method based on Naïve Bayes algorithm for prediction of palmitoylation site. And in 2008, CSS-Palm 2.0 was developed, and in this model an updated version of CSS algorithm was used to predict palmitoylation sites[19]. In the study of Wang et al. in 2009, a new model, CKSAAP-Palm, was proposed[20]. In this model, the composition of k-spaced amino acid pairs (CKSAAP) was used to represent the sequence fragments, with support vector machine as the classifier. In 2011, Hu et al. developed a predictor with k-nearest neighbor (KNN) algorithm based on the amino acid sequence features, IFS-Palm, and the features were selected through Incremental Feature Selection (IFS) method[21]. In 2013, WAP-Palm was developed by Shi et al[22]. WAP-Palm is a new computational method that combined multiple feature extraction methods, covering weight amino acid composition (WAAC), auto-correlation functions (ACF) and position specific scoring matrix profiles (PSSM), and three algorithms, KNN, SVM and decision tree (DT), were tested to construct the online service of WAP-Palm. In 2014, Kumari et al. use structural disorder feature, secondary structure feature, and conservation feature based on PSSM to develop an SVM-based model[23]. In MDD-Palm developed by Weng et al. in 2017, amino acid composition (AAC), amino acid pair composition (AAPC), PSSM, position weight matrix (PWM), amino acid substitution matrix (BLOSUM62), and accessible surface area (ASA) were considered. This model is constructed with SVM as well. Some latest research, including PalmPred[24], GPS-Palm[25] and etc, explored the performance of other algorithms such as random forest (RF) and CNN.

However, it’s important to note that all these models are based on supervised learning methods up to now, where S-acylation sites identified through biological methods are considered as positive examples and other sites yet identified, which is actually unlabeled, as negative examples. This is unsuitable in the context of plant S-acylation sites prediction. Generally, the training data contain both positive and negative samples that are fully labeled. But present identified S-acylation sites predominantly from the result of researches on specific proteins. That is to say, in most cases, researchers often only manage to identify a small portion of S-acylation sites that is relevant to their study of specific proteins, leaving many other sites unexplored and uncertain. Consequently, there is a huge amount of unlabeled data, which is yet identified S-acylation sites, that perhaps contains a lot of positive samples. And the exist of such numerous unlabeled data raises the problem of how to make use of the huge amount of information from this data. If the unlabeled is regarded as negative data simply, the noise of the negative data might be severe. The noise will diminish the predictive efficacy of models and leading to higher false positive rate.

This kind of condition that training data only contains positive and unlabeled examples is not rare, even more prevalent, in many application scenarios. And, therefore, researchers have increasingly focused on developing classification tools for such data over the past 20 years, known as PU-learning classifier. PU learning, as a special case of semi-supervised learning, aligns with the long-term interest in developing learning algorithms that do not required fully supervised data[26]. In the field of protein functional sites identification, PU-learning has already been utilized in several prediction methods, and the results are promising. For instance, Alkuhlani et al. proposed an ensemble bagging PU learning method to predict N-Linked glycosylation sites[27], and Li et al. developed a method based on biased SVM classifier to predict substrate sites that are cleaved by HIV-1 protease[28]. Compared to supervised learning, PU learning is more fit for the task of recognizing S-acylation sites, given the limited number of identified sites and the abundance of unlabeled data in databases such as UniProt[29]. So in our work, the prediction of S-acylation sites is regarded as a task of learning from positive and unlabeled data. This approach is to effectively mining information from the large amount of unlabeled data and avoid the possible high false positive rate of supervised-learning classifiers.

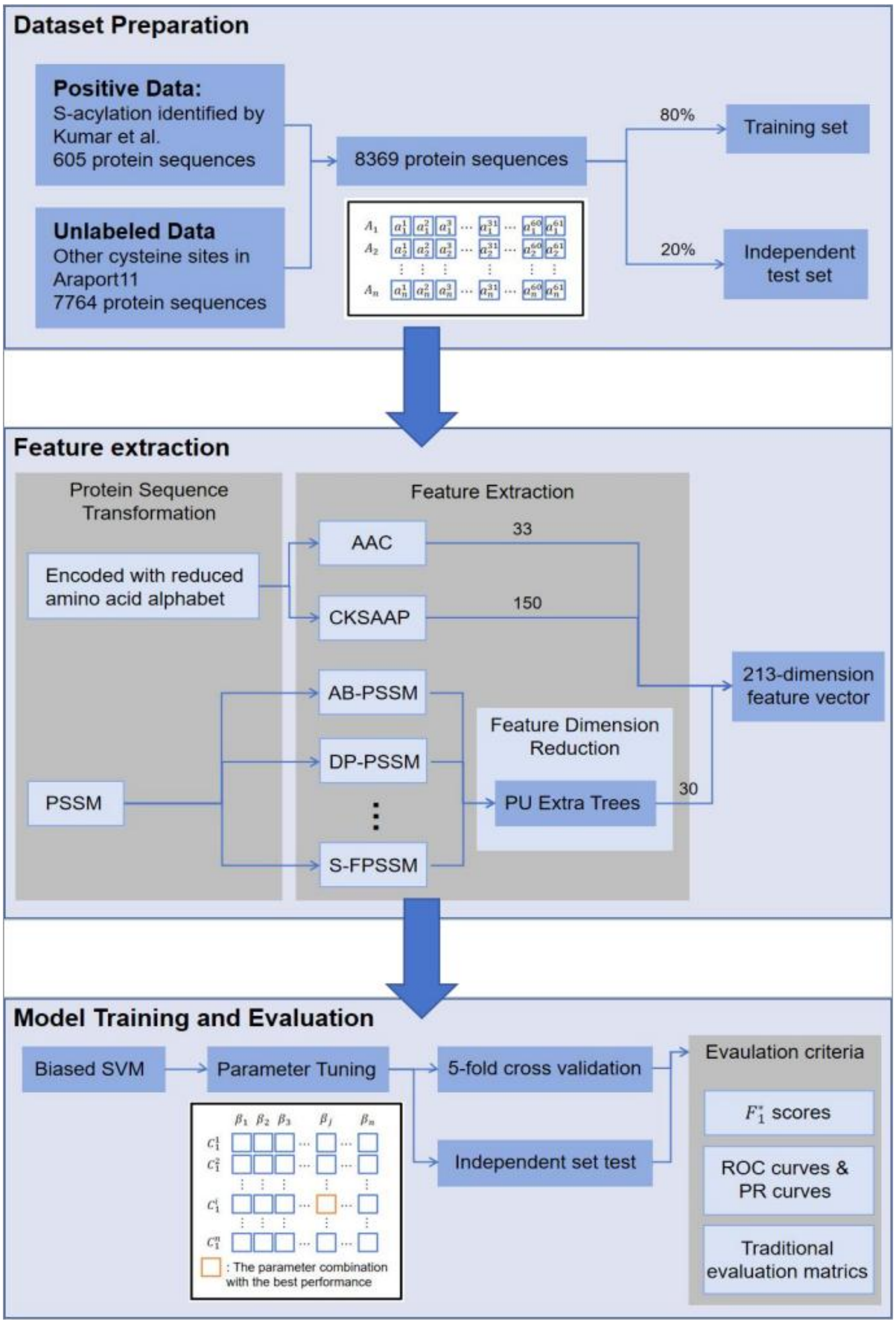


Fig. 1. The framework of the process for PU-Palm to predict new S-acylation sites through learning from positive and unlabeled data.

There are two common methods to solve the PU learning problem. The first approach is the two-step technique: firstly find the reliable negative examples within the unlabeled data, and then training the model using positive, negative and unlabeled data. But this approach requires accurate prior probability, which can be challenging to obtain. The second approach is the biased PU learning method to regard unlabeled data as examples with class label noise. And it could be implemented by placing higher penalty on misclassified positive examples[26]. This method is more fit for the task of predicting S-acylation sites whose unlabeled data is actually without any prior possibility.

Present study on plant S-acylation is still in its early stages, and most of S-acylation sites hasn't been identified. For convenience of researchers' operating more targeted and valuable sites identification, our study promote a concise and efficient PU learning model to predict more potential S-acylation sites with high accuracy to effectively and credibly extended positive data of S-acylation sites: (i) 4263 features were obtained through the feature extraction methods AAC, CKSAAP, PSSM-composition, etc, and the performance of varied sequence-based and PSSM-based features has been extensively studied. (ii) Two simplified amino acid alphabets and PU extra trees were used to reduce the dimension of features, which could sharply reduce the dimension of features with equally ideal performance at the same time. (iii) A model based on biased SVM and various feature extraction methods was developed to predict S-acylation sites. (iv) Two evaluation methods specially for PU learning were used to comprehensively exhibit the effectiveness of our model.

1 Methods

1.1 Data Collection and Preparation

The positive samples used in our study were downloaded from the research conducted by Kumar et al[30], in which they conducted a comprehensive analysis of plant protein S-acylation across six separate tissues. In their study, peptides not containing a carbamidomethyl cysteine were excluded from further analysis and the remaining 10705 peptides were then mapped to Araport11[31] proteins. Among these peptides, 9186 peptides had a single unambiguous database match (exclusive peptides), representing 4240 proteins, and 1519 peptides matched to more than one protein (ambiguous peptides) that mapped up to 1816 proteins. Subsequently, these two kinds of peptides were respectively divided into high, medium and low confidence groups based on the fold change and p value.

Its high-confidence S-acylation sites data, including both exclusive and ambiguous peptide data, were regarded as reliable positive data in our research. After these identified S-acylation sites used as positive examples in our research were removed, all the other sites of cysteine in Araport11 is collected as unlabeled data. And then, to improve the quality of the data set, Cd-hit[32] was employed to remove redundant sequences, using a clustering threshold of 0.8. The use of Cd-hit was to reduce the overall size of the data set by eliminating highly similar sequences, which was expected to keep most information within less data. The final data set used in the work was composed of 605 positive examples and 7764 unlabeled examples.

Our examples are sequences that consist of the specific site under studying along with thirty amino acids preceding and following this site. A sequence could be represented as below:

$$\mathrm{S} = a_{-30}a_{-29}\cdots a_{-1}a_0a_1\cdots a_{30} \tag{1}$$

where $\mathrm{a_i}$ represents the i th amino acid residue in the protein sequence S, and $\mathrm{a_0}$ is the potential or identified S-acylated site under study.

1.2 Feature Extraction

Varied feature extraction methods were used in our work to characterize sequence information, which could be broadly divided into two groups. The first group are based on position-specific score matrix (PSSM), while the second are based on the calculation of frequencies of residues of sequences, residue pairs of sequences and etc. These features provide us with the statistics information on the protein sequences, reflect the primary structure of proteins. Notably, PSSM and PSSM-based features was generated by using the online POSSUM package[33].

1.2.1 Amino acid composition (AAC)

Amino Acid Composition represents the frequency of twenty amino acid residue in a sequence. This feature is the most simple methods of feature extraction, which will product a 20-dimension feature vector, and has been widely used in bioinformatics[34]. Its formula is shown as follows:

$$V = [v_1 v_2 \cdots v_i \cdots v_{20}]^T \tag{2}$$

$$v_i = \frac{count_i}{L} \tag{3}$$

where i represents one of the 20 amino acid residues, $count_i$ is equal to the total of the amino acid residue i, and L is the length of the sequence.

1.2.2 Composition of k-spaced amino acid pairs (CKSAAP)

Composition of k-spaced amino acid pairs is mainly concerned with the long range correlation of two amino acid residues. There has been models for varied protein functional sites prediction that used CKSAAP to extract features, including models for the prediction of palmitoylation sites[20], phosphorylation sites[35], citrullination sites[36] and etc. CKSAAP represents all the frequencies of amino acid pairs whose gaps are no more than k, which is represented as:

$$V = V_1 V_2 \cdots V_i \cdots V_k \tag{4}$$

$$V_i = [v_1^i v_2^i \cdots v_j^i \cdots v_{400}^i]^T \tag{5}$$

$$v_j^i = \frac{count_j^i}{L - i - 1} \tag{6}$$

where j represents one kind of amino acid pair, i means the amino acid pair being studied has an interval of i peptides, L represented the length of the sequence, and $count_j^i$ is equal to the number of amino acid pair j that is i-spaced. For example, if $k = 2$, CKSAAP consists of the frequencies of adjacent amino acid pairs(400-dimension vector), the pairs that has a gap of one peptide(400-dimension vector) and the pairs with a gap of two peptides(400-dimension vector). The final feature will be a 1200-dimension vector.

1.2.3 Tripeptide composition (TPC)

Tripeptide composition represents the frequencies of three consecutive amino acids, which is a 8000-dimension feature vector($20 \times 20 \times 20$). For example, ReRF-Pred, proposed by Teng et al.[37], utilize tripeptide composition for feature extraction. Similarly, Li et al. also used TPC as one of the feature extraction method in their model, PredAmyl-MLP[38]. The formula of TPC is defined as follows:

$$V = [v_1 v_2 \cdots v_i \cdots v_{8000}]^T \tag{7}$$

$$v_i = \frac{count_i}{L - 2} \tag{8}$$

where L represented the length of the sequence, $count_i$ represented the number of the tripeptides that appear in the sequence.

1.2.4 Conjoint Triad (CT)

Conjoint triad (CT) is to calculate the frequencies of triad of the sequences simplified through a 7-letter reduced alphabet, mainly considering neighbor relationships in protein sequences. The conjoint triad method is firstly proposed by Shen et al. and was used for describing the information of protein-protein interactions[39]. The 7-letter alphabet, which is classified according to their dipoles and volumes of the side chains, is as follows:

Table. 1. The reduced 7-letter amino acid alphabet that is grouped based on the dipoles and volumes of the side chains.

7-letter alphabet	Amino acids
Group1	A, G, V
Group2	I, L, F, P
Group3	Y, M, T, S
Group4	H, N, Q, W
Group5	R, K
Group6	D, E
Group7	C

The formula of CT is similar to tripeptide composition:

$$V_i = [v_1^i v_2^i \cdots v_j^i \cdots v_{343}^i]^T \tag{9}$$

$$v_i = \frac{f_i - \min\{f_1, f_2, \cdots, f_{343}\}}{\max\{f_1, f_2, \cdots, f_{343}\}} \tag{10}$$

where $f_i = \frac{n_i}{L-2}$ represents the frequencies of triad i, and v_i is result of normalized f_i.

1.2.5 Average block (AB-PSSM)

AB-PSSM is based on the averaged PSSM profiles over blocks, with each block containing 5 percent of the sequence. Thus, a protein sequence is divided into 20 blocks regardless of its length. The feature of the jth block is a 1×20 feature vector F_j. This method is firstly used by cheol Jeong et al. to extract features from PSSMs[40], and was then extended to the task for detecting protein-protein interactions[41] and protein self-interaction[42]. The formula is represented as:

$$F_j = \frac{1}{B_j} \sum_{i=1}^{B_j} P_i^{(j)} \tag{11}$$

where $P_i^{(j)}$ is a 1×20 vector from the ith position of the jth block of the PSSM profile, and B_j represents the length of each block. And the final features of all the blocks are a 1×400 vector.

1.2.6 DP-PSSM

DP-PSSM was proposed by Juan et al. to predict protein subcellular localization for gram-negative bacteria[43]. The DP-PSSM features consist of two parts T' and G'. The formula of 40-dimensional vector T' is as follows:

$$T' = [\overline{T}_1^{\,P}, \overline{T}_1^{\,N}, \overline{T}_2^{\,P}, \overline{T}_2^{\,N}, \cdots, \overline{T}_{20}^{\,P}, \overline{T}_{20}^{\,N}] \tag{12}$$

subject to:

$$\overline{T}_j^{\,P} = \frac{1}{NP_j} \sum T_{i,j} \text{ , if } T_{i,j} \geq 0 \tag{13}$$

$$\overline{T}_j^{\,N} = \frac{1}{NN_j} \sum T_{i,j} \text{ , if } T_{i,j} < 0 \tag{14}$$

where NP_j, NN_j represent the number of positive numbers (including 0) and negative numbers. And $T_{i,j}$ could be calculate through the formulas as below:

$$mean_i = \frac{1}{20} \sum_{i=1}^{20} E_{i,k} \tag{15}$$

$$\mathrm{STD}_i = \sqrt{\frac{\sum_{u=1}^{20} [E_{i,u} -, mean]^2}{20}} \tag{16}$$

$$T_{i,j} = \frac{E_{i,j} - mean_i}{STD_i} \tag{17}$$

And the other part of DP-PSSM G' could be represented as:

$$G' = [G_1, G_2, \cdots, G_{20}] \tag{18}$$

$$G_j = [\overline{\Delta}_{1,j}^P, \overline{\Delta}_{1,j}^N, \overline{\Delta}_{2,j}^P, \overline{\Delta}_{2,j}^N, \cdots, \overline{\Delta}_{\alpha,j}^P, \overline{\Delta}_{\alpha,j}^N] \tag{19}$$

subject to:

$$\overline{\Delta}_{k,j}^P = \frac{1}{NDP_j} \sum [T_{i,j} - T_{i+k,j}]^2 \text{ , if } T_{i,j} - T_{i+k,j} \geq 0 \tag{20}$$

$$\overline{\Delta}_{k,j}^N = \frac{-1}{NDN_j} \sum [T_{i,j} - T_{i+k,j}]^2 \text{ , if } T_{i,j} - T_{i+k,j} < 0 \tag{21}$$

1.2.7 PSSM-composition

PSSM-composition is to compute PSSM composition features from original PSSM profiles. The PSSM-composition feature is a 1×400 vector generated by summing the amino acid rows in the PSSM[44]. Firstly all the corresponding rows of each kind of amino acid are summed, which means each kind of amino acid correspond to a 1×20 vector F_i. And then divide F_i by sequence length in order to scale F_i to [-1,1].

1.2.8 Residue probing method (RPM-PSSM)

RPM-PSSM borrowed the probe concept employed in microarray technologies[40], and the formula of RPM-PSSM is similar to PSSM-composition. In residue probing method, each probe is an amino acid, which corresponds to a particular column in the PSSM profiles. To calculate the features, the residue probing method firstly transform PSSMs to PPSSM, by setting all negative PSSM values as 0. For each probe, that is to say for each column, the corresponding PSSM scores of all the same kind of amino acid in the sequence are summed up respectively, which leads to a 1×20 feature vector F_i. And then after divided by sequence length to scale F_i to [-1,1], the 20 20-dimension vectors corresponding to the 20 probes are concatenated. The final feature for each protein sequence is a 1×400 vector.

1.2.9 S-FPSSM

S-FPSSM[45] is a feature extraction method based on FPSSM. FPSSM is an improved PSSM after filtering all negative scores and all positive scores greater than δ. The feature is a 1×400 vector $\mathrm{S} = (\mathrm{s}_1^1, \mathrm{s}_2^1, \cdots, \mathrm{s}_{20}^2, \cdots, \mathrm{s}_1^{20}, \mathrm{s}_2^{20}, \cdots, \mathrm{s}_{20}^{20})$, where s_j^i is the sum of the scores in the jth column of the FPSSM scores whose corresponding residue is a_i (a_1 to a_{20} represent the twenty amino acids).

1.2.10 Dipeptide composition (DPC-PSSM)

To partially reflect the local sequence-order effect, traditional dipeptide composition from the primary sequence is extended to the PSSM, denoted by DPC-PSSM[46]. DPC-PSSM is defined as a 400-dimentional vector:

$$V_{DPC} = \{d_{1,1}, \cdots, d_{1,20}, d_{2,1}, \cdots, d_{2,20}, \cdots, d_{m,n}, \cdots, d_{20,1}, \cdots, d_{20,20}\} \tag{22}$$

where

$$d_{m,n} = \frac{1}{L-1} \sum_{k=1}^{L-1} e_{k,m} \times e_{k+1,n} \quad \text{s.t.}\, 1 \le m, n \le 20 \tag{23}$$

where $e_{k,m}$ represent the PSSM elements that are scaled to the range from 0 to 1 using the following sigmoid function：

$$f(x) = \frac{1}{1+e^{-x}} \tag{24}$$

1.2.12 Evolutionary difference formula (EEDP-PSSM)

By defining evolutionary difference formula, EEDP-PSSM[47] use uniform dimensional vectors, to represent varying length proteins, which can represent evolutionary difference information between the adjacent residues. This method shows promising performance on predicting protein structural class especially for low-similarity sequences. Firstly, use the following formulas to define the average evolutionary scores of three adjacent amino acids:

$$mean_1 = \frac{e_{i-1,m} + e_{i,t}}{2}, mean_2 = \frac{e_{i,t} + e_{i+1,n}}{2} \quad \text{s.t.}\, 1 \le \mathrm{i} \le L \text{ and } 1 \le m, t, n \le 20 \tag{25}$$

where $e_{i,t}$, $e_{i+1,n}$, $e_{i-1,m}$, are elements in M_{PSSM} (the data in columns 1 to 20 of the file of PSSM), and $mean_1$, $mean_2$ represent the position score between $\mathrm{i} - 1$ and i and the position score between i and $\mathrm{i} + 1$ respectively. Position score variable AED is defined as follows:

$$\mathrm{AED}_{i-1,i+1} = (mean_1 - mean_2)^2 = (\frac{e_{i-1,m} - e_{i+1,n}}{2})^2 \tag{26}$$

So, the feature extraction method EEDP can be represented as:

$$\mathrm{x}_{m,n} = \frac{1}{L-2}\sum_{i=2}^{L-1} AED_{i-1,i+1} \quad \text{s.t.}\, 1 \le \mathrm{m}, n \le 20 \tag{27}$$

$$\mathrm{V}_{\mathrm{EEDP}} = \{x_{1,1}, \cdots, x_{1,20}, x_{2,1}, \cdots, x_{2,20}, \cdots, x_{m,n}, \cdots, x_{20,1}, \cdots, x_{20,20}\} \tag{28}$$

1.2.13 K-separated bigrams (KSB-PSSM)

K-separated bigrams (KSB-PSSM) is to extract the information of amino acids that are non-adjacent in the protein sequence (if k = 1, KSB can also be used to extract adjacent information). To accomplish this, amino acid bigram probabilities are extracted from the sequential evolution probabilities in PSSM[48]. The formula of KSB is as follows:

$$T_{m,n} = \sum_{i=1}^{L-1} e_{i,m} \times e_{i+k,n} \quad \text{s.t.}\, 0 \le m, n \le 20 \tag{29}$$

where $e_{i,j}$ is the elements of M_{KSB} (the data in columns 21 to 40 of the file of PSSM). And the feature vector is expressed as:

$$V_{KSB} = \{T_{1,1}, \cdots, T_{1,20}, T_{2,1}, \cdots, T_{2,20}, \cdots, T_{m,n}, \cdots, T_{20,1}, \cdots, T_{20,20}\} \tag{30}$$

1.3 Reduced amino acid alphabet

Encoding amino acid sequences with twenty kinds of amino acids is sometimes not the optimal choice for feature extraction, especially considering the too high dimension of features such as TPC and CKSAAP. Both experimental and theoretical studies have suggested that the full sequence complexity is not essential for the prediction of correct protein structure[49] and function[50]. And so, different works have been conducted to simplify sequence world by grouping the 20 amino acids based on their similar features. For example, in the research of Chen et al., the twenty amino acids were divided into 6 groups according to their individual hydropathies, which means a protein sequence with 20 amino acids can be represented by a sequence with 6 characters[51]. Representing sequences with reduced amino acid alphabet will not only reduces the noise and complexity in dimension of feature engineering, but also provide the model with more sufficient biological prior knowledge[52].

In our work, two different ways of grouping the amino acids was used to represent amino acid sequences. These two reduced amino acid alphabets respectively focused on the amino acids' effect on protein structure and the amino acids' chemical property. They were used in our work to reduce the dimension of sequence-based features and to obtain the biological prior knowledge of protein structure and chemical properties.

1.3.1 5-letter alphabet

Numerous works have shown that protein 3D structures are composed of a limited number of protein blocks, which defined a structural alphabet. Through analyzing equivalences between the different kinds of amino acids based on the structural alphabet, Catherine Etchebest et al. present a series of new reduced sets of amino acids while preserving the 3D fold[53]. This 5-letter alphabet used in our work is shown as follows:

Table. 2. The amino alphabet contains only 5 letters based on the study of protein blocks.

5-letter alphabet	Amino acids
Group1	G
Group2	P
Group3	I, V, F, Y, W
Group4	A, L, M, E, Q, R, K
Group5	N, D, H, S, T, C

1.3.2 Chemical-property alphabet

The other method of grouping amino acids is based on the chemical properties of amino acids. The reduced amino acids alphabet is as follows:

Table. 3. The amino alphabets that is grouped based on the chemical properties.

Chemical-property alphabet	Chemical groups	Amino acids
Group1	Sulfur-containing	C, M
Group2	Aliphatic 1	A, G, P
Group3	Aliphatic 2	I, L, V
Group4	Acidic	D, E
Group5	Basic	H, K, R
Group6	Aromatic	F, W, Y
Group7	Amide	N, Q
Group8	Small hydroxy	S, T

1.4 Feature Selection

The combination of various features might provide more valuable information for characterizing S-acylation sites. But the utilization of an excessive number of features can result in feature explosion, leading to higher sparsity, redundancy and irrelevance of features. Training models using features with such high dimension will also increase training cost and cause classifier to over-fit training data at the same time[54]. In our work, these issues was addressed by reducing the dimension of our features through a feature selection process.

In the task of eliminating irrelevant and redundant features, random forest is a promising candidate[55]. Giving the special properties of the data in our research, a random forests model specially designed for PU data was used to assess the importance of each feature. This model, called PU extra trees, is based on the method of recursive greedy risk minimization. The feature importance values generated by the model is to directly measure each feature’s contribution to risk minimization, and are instrumental in our selection of features for further analysis and modeling.

PU extra trees algorithm, proposed by Jonathan Wilton et al[56], is an efficient PU random forest algorithm designed to directly minimizes PU-data based estimators for the expected risk. The key to this algorithm is essentially a new interpretation that decision tree algorithms could be regarded as recursive greedy risk minimization algorithms. Through introducing a recursive greedy risk minimization approach to PU learning, the random forest algorithm was enabled to be used for PU learning tasks. In our study, it is the non-negative risk estimator that is chosen to estimate the risk. The formula is as follows:

$$\hat{R}_{nnPU}(g) = \sum_{x \in P} w_p l(g(x),+1) + \max\left\{0, \sum_{x \in U} w_u l(g(x),-1) - \sum_{x \in P} w_p l(g(x),-1)\right\} \tag{31}$$

where $w_p = \pi/n_p$ and $w_u = 1/n_u$. And $l : \mathbb{R} \times \{-1,+1\} \rightarrow \mathbb{R}$ is a loss function that gives the loss $\mathrm{l(v,y)}$ incurred by predicting a score v when the true label is y. This work focuses on the quadratic loss $\mathrm{l_{quad}(v,y) = (1 - vy)^2}$.

The training of this random forest model on PU data follows an iterative two-step learning process: the first step is to identify reliable negative examples, the second step is to train a model using the positive examples, the reliable negative examples and possibly unlabeled examples. To identify reliable negative examples, PU extra tree requires prior probability β (the proportion of negative samples to positive samples) of the unlabeled data. But the probability is unknown. To address this problem, AlphaMax[57] is used in our research to provide estimated prior knowledge for the application of PU extra trees. AlphaMax estimate class prior probability using two-component mixture models, and the result of the estimated class prior probability of our data β is 0.1456. The complete pseudocode for AlphaMax is as:

Algorithm 1 The AlphaMax algorithm for class prior estimation

```
Require: sample X, $X_1$
Ensure: $\alpha^*$
  // Solve level-set optimization for the following set of α; for example,
  c ← $[0.01, 0.02, \cdots, 0.98, 0.99]$
  $n_\alpha \leftarrow length(c)$
  for j = $1, 2, \cdots, n_\alpha$ do
    $l(j) \leftarrow \max_{\sum_{i=1}^{k} \beta_i \omega_i = c(j)} L(\beta | X_1, X)$
  end for
  // Smooth l using median of 2k-nearest neighbors; typically, k=3
  $l \leftarrow l_{smooth}$
  for j = $k + 1, \cdots, n_\alpha - k$ do
    $l_{smooth}(j) \leftarrow median(l(j-k), \cdots, l(j+k))$
  end for
  $l \leftarrow l_{smooth}$
  // Scale l between 0 and 1
  $l \leftarrow (l - \min(l))/(\max(l) - \min(l))$
  // Compute the difference between slopes before and after j using window win
  $\Delta slope \leftarrow 0$
  for j = $win + 1, \cdots, n_\alpha - win$ do
    $slope\_before(j) \leftarrow$ slope of the linear fit to $\{c(j), l(j)\}_{j-win}^{j}$.
    $slope\_after(j) \leftarrow$ slope of the linear fit to $\{c(j), l(j)\}_{j}^{j+win}$
    $\Delta slope(j) \leftarrow slope\_before(j) - slope\_after(j)$
  end for
  // Divide by $1 - l$ plus a small positive constant ϵ
  $\mathrm{heurist} \leftarrow \Delta\mathrm{slope}/(1 - \mathrm{l} + \epsilon)$
  $\alpha^{*} \leftarrow c(index\_of\_max(heuristic))$
```

1.5 Biased Support Vector Machine

Classical SVM models classify the positive and negative examples by calculating a hyperplane with the best performance on classification[58]. But the lack of reliable negative examples makes the classical approach infeasible. To address this challenge, biased SVM model is employed to predict S-acylation sites.

Biased SVM is a common biased learning method. Through applying different misclassification penalty C_1, C_2 to positive and unlabeled data, SVM can be adjusted to solve PU learning problems[59]. The parameter C_1, C_2 respectively represents the error tolerance on positive data and unlabeled data.The formula of biased SVM can be defined as follows:

$$Minimize: \frac{1}{2}\omega^{T}\omega + C_1\sum_{i=1}^{m-1}\delta_i + C_2\sum_{i=m}^{m-n}\delta_i \tag{32}$$

s.t. $y_i(\omega^{T}x_i + b) \geq 1 - \delta_i, \delta_i \geq 0, i = 1,2,\cdots,n$

where C_1, C_2 are two soft margin parameters, ω is the normal vector of hyperplane separating positive and unlabeled examples, δ_i refers to the corresponding slack variable used to calculate the error cost for each sequence, and b denotes the offset of hyperplane from the origin along ω. For our biased SVM model, C_1 is set a higher value and C_2 is set a lower value. A higher value for C_1 represent a greater penalty for misclassified positive examples in order to make sure the positive examples should be classified correctly as much as possible. And a lower value for C_2 is able to maximize unlabeled data as non-cleavable, but does not reject the possibility of containing positive examples[28].

In this study, the biased SVM was implemented through the Scikit-learn library. To fine tune the two penalty parameters, a new variable β was defined to control the difference between C_1 and C_2:

$$C_2 = \frac{C_1}{\beta} \tag{33}$$

where the values of C_1 was from the set [0.015625, 0.03123, 0.0625, 0.125, 0.25, 0.5, 1, 2, 4, 8, 16, 32, 64] and the values of β was from the set [2, 5, 10, 20, 30, 50, 100, 200]. After evaluating all the possible combinations of the two variables, the combination with the best performance was used as the final parameter of biased SVM classifier, and the best parameter pair was $C_1 = 64, \beta = 5$.

1.6 Performance Evaluation

In traditional modeling, evaluation metrics such as ACC, AUC, F_1, etc are commonly used to assess the performance of models trained on positive and negative examples. They usually rely on knowledge of the true class labels of each instance. However, in the context of PU learning where labeled negative examples are absent, these standard metrics cannot be calculated[60]. A common approach in such cases is to assume that all unlabeled instances are negative while evaluating performance. But research has shown that naively assuming that assuming all unlabeled examples to be negative, as is sometimes done in PU learning, will underestimate the performance[61], resulting in selecting suboptimal models according to the biased evaluation metrics.

In our study, two new methods were applied to evaluate the model performance in the PU learning context. One is by modifying the formula of F_1 score to satisfy the the condition of PU learning. This new evaluation metric is an attempt to compute the evaluation criteria based on the total number of examples and the number of positive examples. And it was widely used in various aspects of our work, such as model parameter selection, feature selection and the evaluation for the performance of our model. The other is to get the range of the evaluation metrics that is commonly used in supervised learning through calculating the upper and lower bounds. It was specifically utilized for evaluating the model performance. Additionally, this estimation method provides us an insight into the relationship between the calculated value when regarding all the unlabeled samples as negative ones and the true value of standard metrics.

1.5.1 Metrics for PU data

A commonly used performance measure for model evaluation is F_1 score, which is calculated as the harmonic mean of the precision and recall. The formula could be represent as follows:

$$F = \frac{2pr}{(p+r)} \tag{34}$$

where precision $\mathrm{p} = \Pr(\mathrm{y} = 1|\hat{\mathrm{y}} = 1)$ and recall $\mathrm{r} = \Pr(\hat{\mathrm{y}} = 1|\mathrm{y} = 1)$. It can be seen in the formula of F_1 score that a high F score indicates high values for both precision and recall. But it is important to noted that F_1 score can't be calculated directly under the circumstance that only positive and unlabeled data is available.

To address this issue, Lee et al. came up with a new evaluation criteria for PU learning, which is based on F_1 score[62]:

$$F_1^* = \frac{pr}{\Pr(\mathrm{y}=1)} = \frac{pr^2}{r\Pr(\mathrm{y}=1)} = \frac{\Pr(y=1\,|\,\hat{y}=1)r^2}{\Pr(\hat{y}=1, y=1)} = \frac{r^2}{\Pr(\hat{y}=1)} \tag{35}$$

where p stands for precision, r represents recall, $\Pr(\mathrm{y} = 1)$ is the proportion of the truly positive examples in test set, $\Pr(\hat{\mathrm{y}} = 1)$ represents the proportion of examples predicted to be positive in test set. The basis of this new measure is the equivalent relationship between $\mathrm{pr}/\Pr(\mathrm{y} = 1)$ and $\mathrm{r}^2/\Pr(\mathrm{y} = 1)$. In this formula, recall r and $\Pr(\hat{\mathrm{y}} = 1)$ can both be computed without the knowledge of negative examples, rendering it to be estimated directly from the test set without making additional assumptions.

It could also be seen from the formula that the value of the evaluation criteria F_1^* scores will grow higher, as the increasing of the values of p and r. So when using the same data set, this new performance measure is similar with F_1 score. However, it must be noted that F_1^* scores will be affected by the proportion of positive examples. As a result, this criteria could only be used for the comparison of the performance of model on the same test data set.

1.5.2 Evaluating standard evaluation metrics

As F_1^* scores could only comparing the models tested on the same data, an approach to estimate standard metrics based on contingency tables was additionally utilized in our work to assess the performance of the model[63]. This method of estimating standard metrics enables us to compare the performance of PU-Palm with other existing work.

In this method, labeled positive examples are regarded as a random set of all positives, while the unlabeled data are regarded as a set consisting of positive and negative data. This transforms the task of estimating the traditional evaluation metrics into estimating the fraction of latent positives in the unlabeled set. In stead of directly compared the predicted label values to the true label values, this approach computes contingency tables by looking at the ranking of examples produced by a model, establishing important relationships between contingency tables and rank distributions.

This method of estimating standard metrics is used in our work to calculate the the range of receiver operating characteristic curve (ROC) and precision-recall curve (PR). In order to derive the upper and lower bound of these curves, an estimated β (the fraction of latent positives in the unlabeled set) is also required. It was also generated by AlphaMax ($\beta = 0.1456$), the same as PU extra trees. These curves provide us with the insight into the proximate interval of true values of AUC and PR.

3 Result

3.1 Feature importance evaluation

Directly using high-dimensional features extracted through methods like TPC and CKSAAP is infeasible. To decrease the dimension of these high-dimension features, reduced amino acid alphabets were used to encode sequences before calculating these features. Subsequently, the importance of features was assessed for feature selection purposes. Specifically, PU extra trees was utilized to select the features with the best performance. Apart from decreasing dimension of features directly, the performance of different kinds of feature extraction methods were respectively examined as well. To evaluate these methods, their respective feature sets were employed to train PU-Palm, providing insights into the effectiveness of different feature extraction methods. The details of the feature extraction methods used in our work could be derived from supplementary materials Table 4.

When employing PU extra trees for feature selection, specific parameters were set to optimize the performance of the model. These parameters include the maximum depth of the tree set to 50, the minimum number of samples required to be at a leaf node set to 2, number of randomly chosen split points to consider for each candidate feature set to 3. And the number of features to consider when looking for the best split was the square root of the total of all features. After careful selection of parameter settings, the final F_1^* scores of PU extra trees was recorded as 2.03934.

The feature importance scores of PSSM-based features and sequence-based features are shown as below respectively. They are generated by PU extra trees.

A

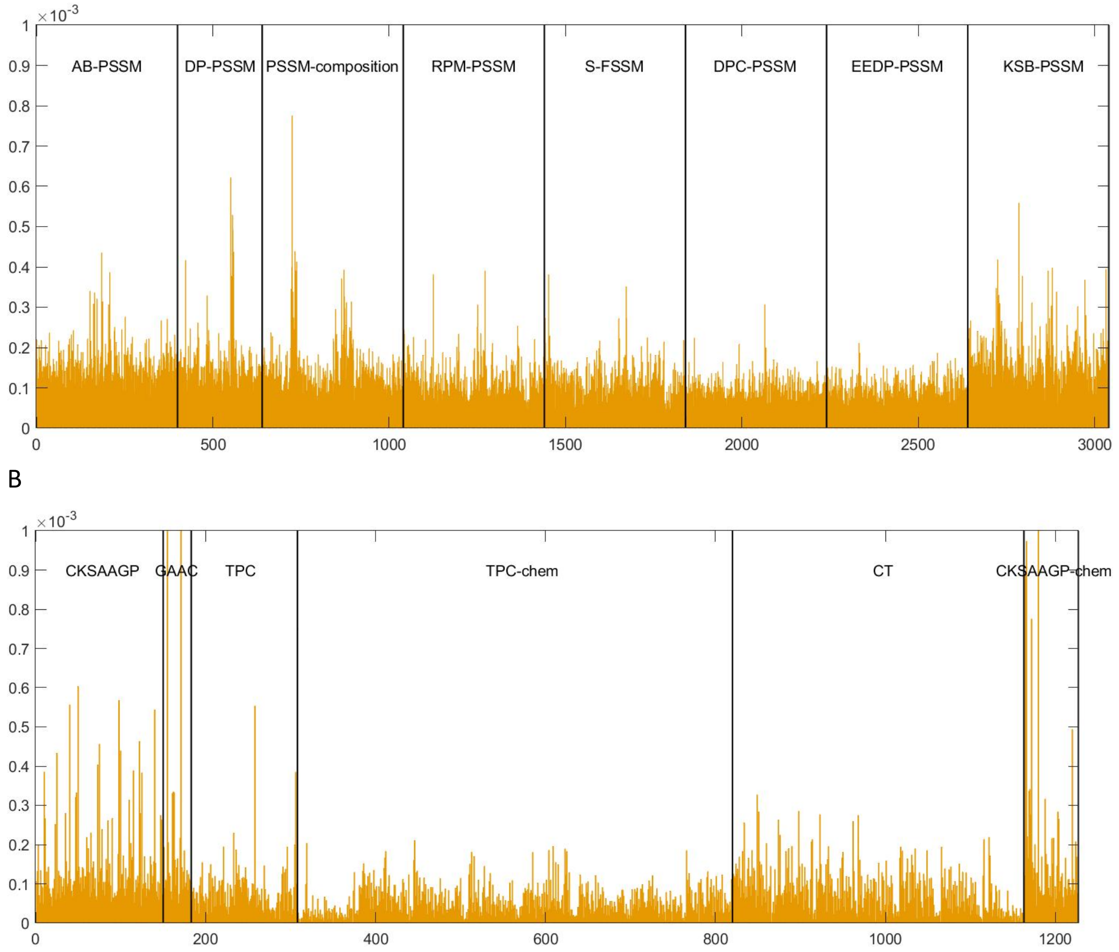


Fig. 2. The feature importance scores of different features. All the features (each sequence is represented by a 1×4267 vector), including the PSSM-based and sequence-based features, are used while training PU extra trees. But the final scores of the two groups of features are separated to present the score distribution more clearly. The X-coordinate represents the sequence number of different features, and the Y-coordinate represents the scores given by PU extra trees. (A) This figure shows the scores of features based on PSSM. The features are divided into 8 parts (AB-PSSM, DP-PSSM, PSSM-composition, RPM-PSSM, S-FPSSM, DPC-PSSM, EEDP-PSSM, KSB-PSSM) (B) This figure shows the scores of sequence-based features. The features are divided into 6 parts (CKSAAGP, GAAC, TPC, TPC-chem, CT, CKSAAGP-chem).

And the performance of biased SVM models trained on respective features sets extracted through these distinct feature extraction methods was as follows.

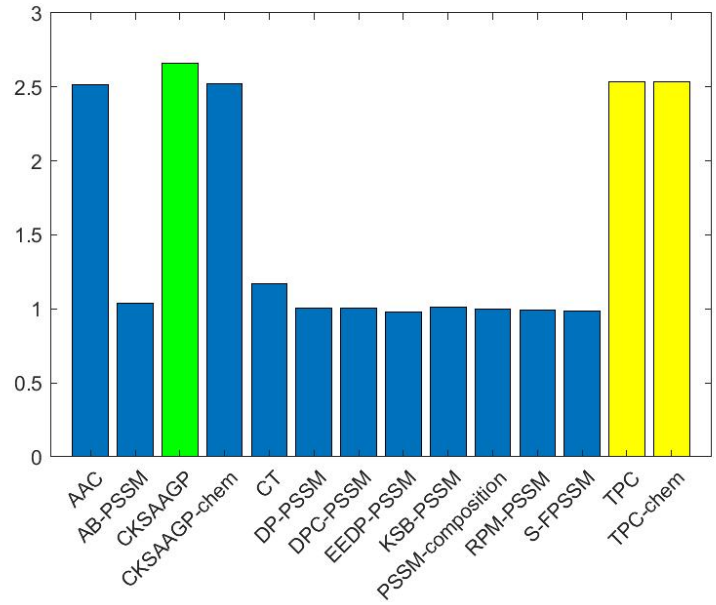


Fig. 3. The F_1^* scores of different feature extraction methods using 5-fold cross validation. It should be pointed out that the subset with the best performance is marked with green, and the subset with the second highest value ia marked with yellow. The details of the subsets has been shown in Table. 5..

Among them, CKSAAGP encompasses all the frequencies of amino acid pairs whose gaps are no more than 5 peptide (k=5), and sequences were encoded with 5-letter alphabet. CKSAAGP-chem represents the frequencies of adjacent amino acid pairs (k=0) with sequences encoded with the chemical-property alphabet. Besides, TPC-chem and TPC respectively represents Tripeptide composition features based on the chemical-property alphabet and 5-letter alphabet. And GAAC including amino acid composition features based on all the three kinds of amino acid alphabets, 20 amino acids alphabet, 5-letter alphabet, chemical-property alphabet.

The results show that CKSAAGP and AAC features have better performance on the scores of contribution to risk minimization. Although the features based on PSSM might contain more sufficient evolutionary information, they are not so ideal as expected. But the performance of PSSM-based features is still better than parts of sequence-based features, such as TPC-chem and CT. And additionally, the distinction between 5-letter alphabet and chemical-property alphabet is not notable.

3.2 Building models on different feature subsets

Because all the data including the test set used in our work was absent of labeled negative examples, new metrics F_1^* scores and estimated PR and ROC curves were applied in our research to evaluate the performance of our model. Apart from these two estimation methods that are especially for PU learning, standard metrics like Sn, AUC, F_1 scores and etc. were used in our study as well. These standard metrics are calculated based on the premise that the unlabeled data is regarded as negative examples. They are inaccurate but they provided an approach to compare with existing works and to some extent also reflect the performance of different models.

In most cases, the more the features are, the better the performance of extra random forest is. But for biased SVM, too much dimension will degrade model performance. In this work, PU-Palm is constructed based on biased SVM for the efficiency and rapidity of this algorithm, and PU extra trees is only used for feature selection. So the following choosing of feature subsets was based on the performance of biased SVM on it.

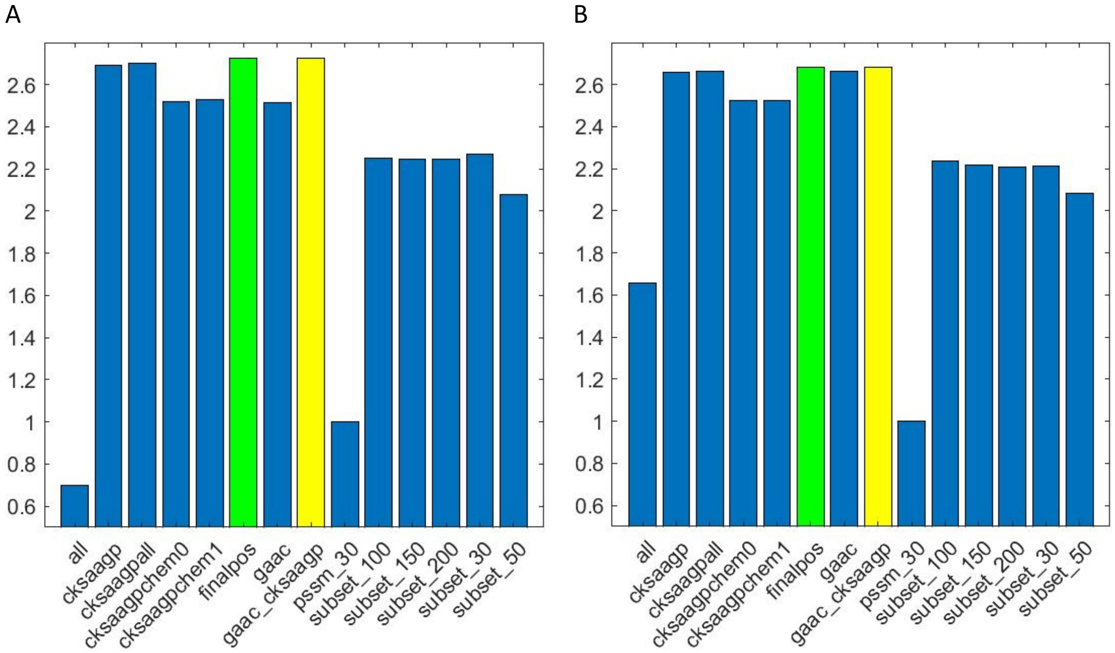
3.2.1 Feature subset partition

Based on the distribution presented in Fig. 2, we proposed different feature combinations to improve the performance of the model. One approach is to select the features with the highest scores. This is to reduce the dimension while guaranteeing the quality of the features at the same time. The other is to directly use features with the best performance like GAAC and CKSAAGP. The result of our work shows that the combination of AAC, CKSAAP and 30 optimized PSSM-based features (finalpos) has the best performance on F_1^* scores. Besides, the combination of features extracted by AAC and CKSAAGP (gaac_cksaagp) is close to the performance of the model on the subset finalpos, and could be used as a simplified substitute feature extraction method.

In our research, 13 different feature subsets are established. All the subsets could be broadly classified into two groups. One is the subsets consisting of features with highest scores of feature importance. The other is the subsets consists of the features extracted by the feature extraction methods with better performance. The details of the subsets is presented in supplementary materials Table 6.

3.2.2 Evaluation of models on different feature subsets using $\mathbf{F}_1^*$ scores

F_1^* scores, designed for PU learning, are for estimating performance of our model on different subsets.



Independent test

5-fold cross validation

Fig. 4. The F_1^* scores of the model performance on different subsets. Figure A and B represent the result of using independent test and 5-fold cross validation to verify model performance. It should be pointed out that the subset with the best performance is marked with green, and the subset with the second highest value ia marked with yellow. The details of the subsets has been shown in Table. 5.

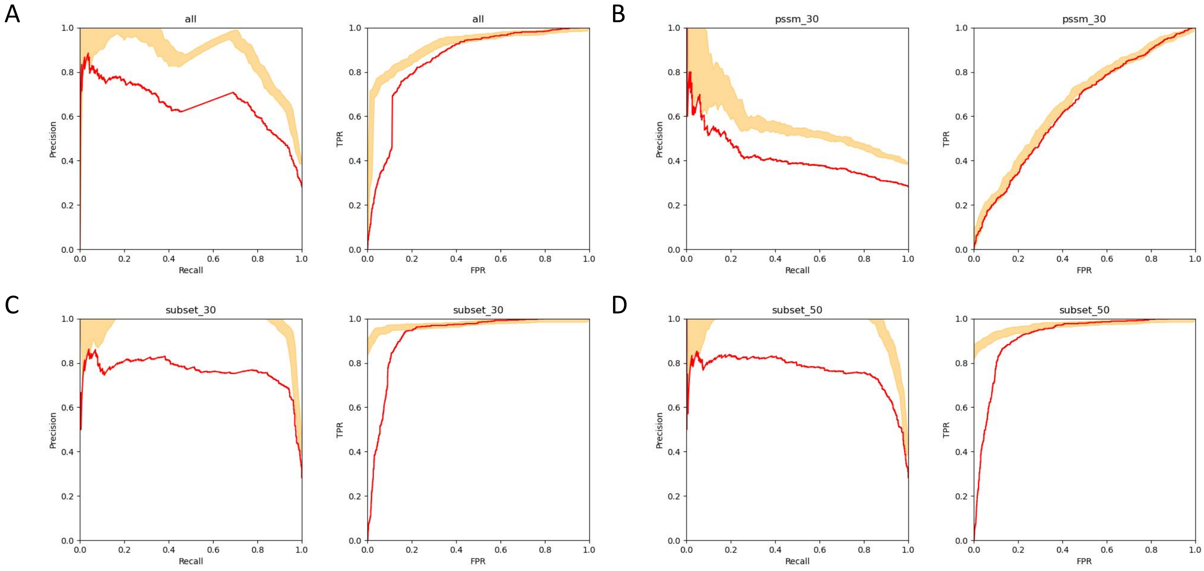
From Fig. 3., it could be clearly found that too high dimension (subset all) is not good for improving model performance. The feature subsets with close dimension (subset gaac, pssm_30 and subset_30) actually have very different performance. And the subsets consists of different number of optimized features doesn't have explicit difference. Among these subsets, finalpos have the highest F_1^* score. And F_1^* scores of the models using CKSAAP features is also promising. Among them, the performance of the combination of AAC and CKSAAP is closest to the performance of subset finalpos.

3.2.3 Evaluation of models on different feature subsets using ROC and PR curves

ROC curve is a plot of the sensitivity versus 1-specificity of a diagnostic test[64], and is an effective method to evaluate the performance of models. But the ROC curve cannot reflect the class-imbalance of the data set[65]. When dealing with highly skewed datasets, PR curves give a more information about an algorithm's performance[66]. So PR and ROC were used side by side to demonstrate the performance of models on different feature subsets. But given the absence of completely labeled data, only the upper and lower bounds of the curves could be calculated according to the confidence. In our work, the confidence was set to 95%.

Apart from the 95% confidence interval of ROC and PR curves, the curves when the fraction of latent positives in the unlabeled set set to 0 were drawn in our work as well, which represents the circumstance of unlabeled samples all regarded as negative data. This, to a certain extent, shows the relationship between the estimated values when assuming all unlabeled samples as negative data and the true values.

For the 13 feature subsets and all the features, the result of prediction based on these 14 feature combinations in total have been shown in Fig. 4 and Fig. 5. In these figures, the performance of the models trained on different feature subsets are demonstrated through PR and ROC curves. Fig. 4 is the curves of model performance trained on different number of top-scoring features selected by PU extra trees. And Fig. 5 is the curves of model performance trained on features extracted through AAC and CKSAAP.



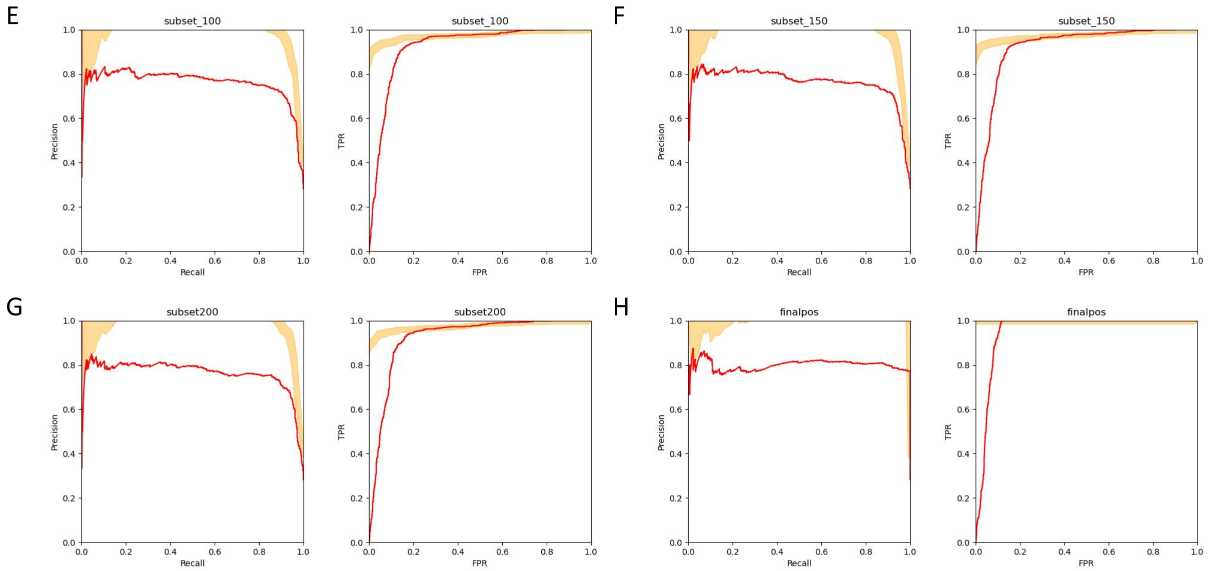
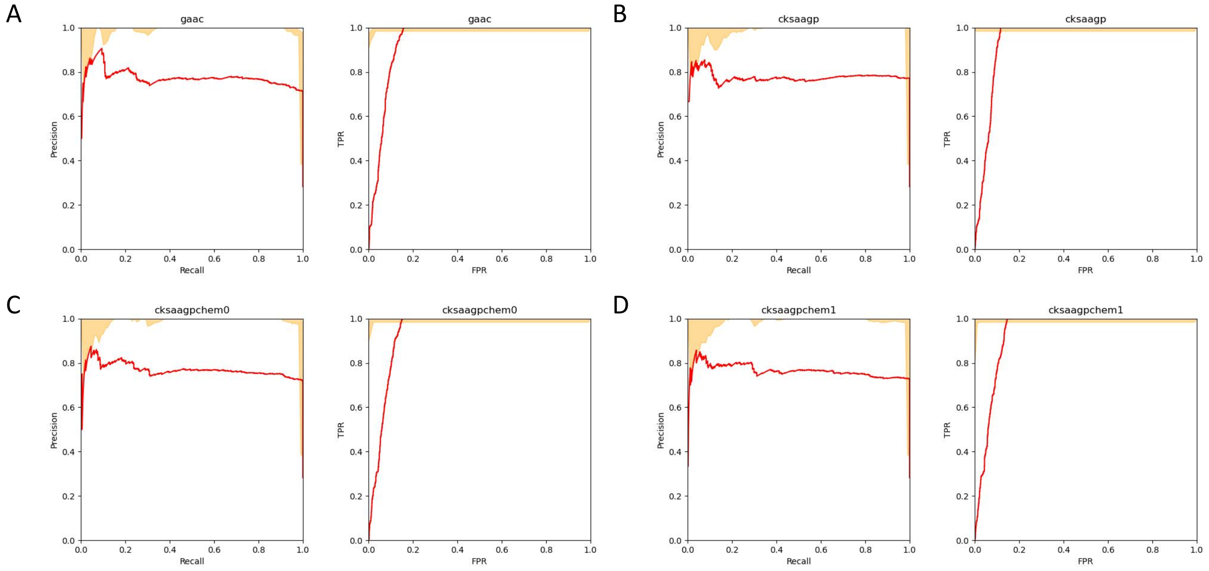


Fig. 5. ROC and PR curves ofmodels trained on top-scoring features. In the figure, the red lines are the curves of ROC and PR when the prior probability β is 0 (the unlabeled data is regarded as negative examples). And the orange areas represent the upper and lower bounds of the interval that true PR and ROC curves locate.

This part reflect the change of the performance with increasing of the number of selected top-scoring features. With the number increasing, there is a subtle improvement of the performance. But too many features, like the result shown in figure A, will lead to the significant decrease on the performance. When only using the 30 optimized PSSM-based features (figure B), the model performance is far from ideal. But the result shows that the most information of all the features could be transmitted with very few features to the model.



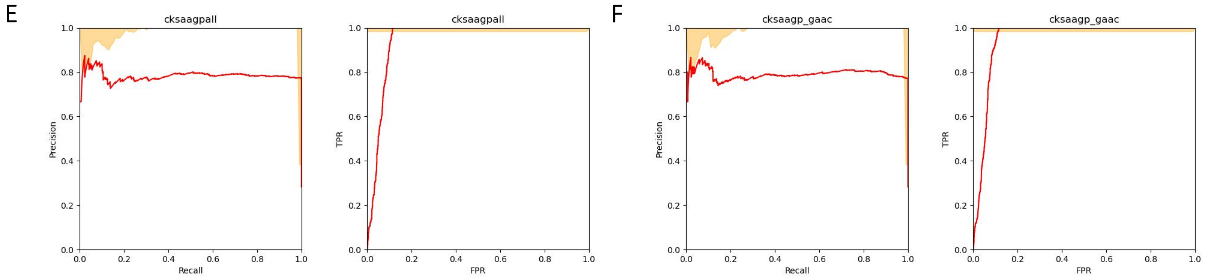


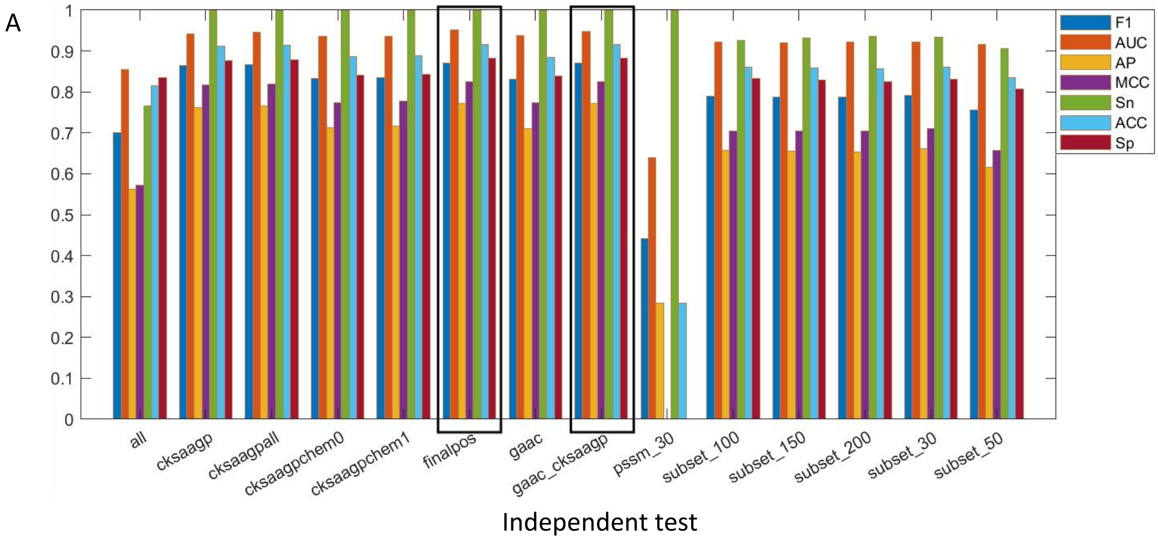
Fig. 6. ROC and PR curves of models trained on features extracted through AAC and CKSAAP. In the figure, the red lines are the curves of ROC and PR when the prior probability β is 0 (the unlabeled data is regarded as negative examples). And the orange areas represent the upper and lower bounds of the interval that true PR and ROC curves locate.

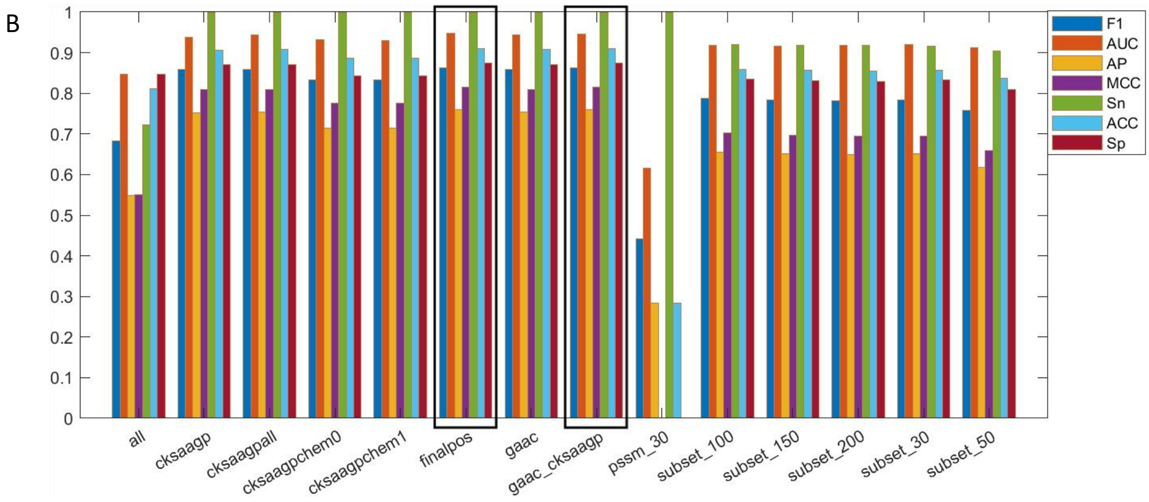
This part mainly reflect the changes of performance of CKSAAP features extracted from sequences encoded with different amino acid alphabets. The performance of CKSAAP features of 5-letter alphabet encoded sequences is better than that extracted from chemical-property alphabet. The performance of these feature subsets that use CKSAAP is outstanding. Among them, it's the combination of CKSAAP and AAC that demonstrate the best performance. And It's difficult to tell the difference between the performance of the subset final and the subset cksaagp_gaac on ROC and PR curves. This proved the feasibility of regarding the feature extraction method of the subset cksaap_gaac, which is a more simple feature extraction method, as a substitute method for the feature extraction method of finalpos.

To sum up, the result demonstrated in Fig. 4 and Fig. 5 showed the same conclusion. Using all the 4267-dimension feature vector is not good for improving the performance of biased SVM. And the model performance on subsets gaac_cksaagp and finalpos is the most ideal. And when assuming all unlabeled samples as negative ones to calculate the value of AUC and PR, the performance is actually be underestimated.

3.2.4 Evaluation of models on different feature subsets using standard metrics

The performance of the models on 14 feature combination is demonstrated through standard metrics (unlabeled examples are regarded as negative examples) in Fig. 6. This method is lack of rigor, but is sometimes used for reflecting the performance of PU-learning models. In our study, this result is exhibited in order to offer approaches to compare with existing works.





5-fold cross validation

Fig. 7. The scores of standard metrics of models trained with different subsets. A and B respectively represent the result using independent test and 5-fold cross validation to verify model performance.

Table. 4. The standard metrics using 5-fold cross validation.

Subset	F1	AUC	AP	MCC	Sn	ACC	Sp
gaac_cksaagp	0.86295	0.94561	0.75909	0.81471	1	0.90997	0.87441
finalpos	0.86310	0.94695	0.75931	0.81490	1	0.91009	0.87457

The result of model performance estimate through standard metrics support the conclusion based on the distribution of F_1^*-scores and PR and ROC curves. The subsets gaac_cksaagp and finalpos are still with the highest values on standard metrics when assuming unlabeled samples as negatives among all the subsets.

4 Discussion

Using PU learning algorithm to identify S-acylation sites accurately is challenging but meaningful for relevant researchers. In our work, the system PU-Palm, constructed based on biased SVM, is proposed to identify potential S-acylation sites.

For PU-Palm, sequences are encoded in a concise and efficient method, which not only significantly decrease the cost of training, but also lead to ideal model performance. In PU-Palm, to find the best feature conbination, 4263 features is extracted through sequence-based methods and PSSM-based methods. And two reduced amino acid alphabets and PU extra trees are used to reduce the dimension of the features. And, what's more, varied PU-learning evaluation methods are applied to evaluate our model performance on different feature subsets. The final features are consists of AAC, CKSAAP features that use reduced amino acid alphabets and optimized 30-dimension PSSM-based features. The result shows that this model could extended positive examples effectively and reliably, proving the promising future of applying PU learning methods in plant S-acylation sites prediction.

One of our novel finding is that AAC and CKSAAP features is with notable effect in the task of identifying S-acylation sites, while the performance of PSSM-based features is not as ideal as expected. Although not as ideal as features extracted through AAC and CKSAAP, the performance of the model on the subset subset_30, which consists of 30 features with the highest scores, demonstrate that the feature selection based on PU extra trees is satisfying. It can not only effectively reduce the computation cost, but the effect of identification ability is also guaranteed. And thus the combination of CKSAAP, AAC, and 30 optimized PSSM-based features have the highest values on F_1^* scores.

Present study on PU learning is still poor. So there is still no reliable metrics to accurately and comprehensively reflect the performance of PU learning models. The new evaluation metrics based on F_1 scores could only be used to compare the effect of different models on the same data set. And the estimation method could only provide an interval result instead of an accurate prediction value. In the future, more types of methods of evaluation could be used to improve evaluation ability for PU-learning models.

References

(1) Dietrich, L. E. P.; Ungermann, C. On the mechanism of protein palmitoylation. *EMBO reports* **2004**, *5* (11), 1053-1057. DOI: https://doi.org/10.1038/sj.embor.7400277.

(2) Resh, M. D. Fatty acylation of proteins: The long and the short of it. *Progress in lipid research* **2016**, *63*, 120-131.

(3) Li, Y.; Qi, B. Progress toward understanding protein S-acylation: prospective in plants. *Frontiers in Plant Science* **2017**, *8*, 346.

(4) Hemsley, P. A. Protein S-acylation in plants. *Molecular membrane biology* **2009**, *26* (1-2), 114-125.

(5) Zheng, L.; Liu, P.; Liu, Q.; Wang, T.; Dong, J. Dynamic protein S-acylation in plants. *International journal of molecular sciences* **2019**, *20* (3), 560.

(6) Hemsley, P. A. S-acylation in plants: an expanding field. *Biochemical Society Transactions* **2020**, *48* (2), 529-536.

(7) Chen, J.-G.; Gao, Y.; Jones, A. M. Differential roles of Arabidopsis heterotrimeric G-protein subunits in modulating cell division in roots. *Plant Physiology* **2006**, *141* (3), 887-897.

(8) Trusov, Y.; Rookes, J. E.; Chakravorty, D.; Armour, D.; Schenk, P. M.; Botella, J. R. Heterotrimeric G proteins facilitate Arabidopsis resistance to necrotrophic pathogens and are involved in jasmonate signaling. *Plant physiology* **2006**, *140* (1), 210-220.

(9) Pandey, S.; Assmann, S. M. The Arabidopsis putative G protein–coupled receptor GCR1 interacts with the G protein α subunit GPA1 and regulates abscisic acid signaling. *The Plant Cell* **2004**, *16* (6), 1616-1632.

(10) Adjobo-Hermans, M. J.; Goedhart, J.; Gadella Jr, T. W. Plant G protein heterotrimers require dual lipidation motifs of Gα and Gγ and do not dissociate upon activation. *Journal of cell science* **2006**, *119* (24), 5087-5097.

(11) Batistic, O.; Sorek, N.; Schultke, S.; Yalovsky, S.; Kudla, J. r. Dual fatty acyl modification determines the localization and plasma membrane targeting of CBL/CIPK Ca2+ signaling complexes in Arabidopsis. *The Plant Cell* **2008**, *20* (5), 1346-1362.

(12) Batistič, O.; Rehers, M.; Akerman, A.; Schlücking, K.; Steinhorst, L.; Yalovsky, S.; Kudla, J. S-acylation-dependent association of the calcium sensor CBL2 with the vacuolar membrane is essential for proper abscisic acid responses. *Cell research* **2012**, *22* (7), 1155-1168.

(13) Leclercq, J.; Ranty, B.; Sanchez-Ballesta, M.-T.; Li, Z.; Jones, B.; Jauneau, A.; Pech, J.-C.; Latché, A.; Ranjeva, R.; Bouzayen, M. Molecular and biochemical characterization of LeCRK1, a ripening-associated tomato CDPK-related kinase. *Journal of experimental botany* **2005**, *56* (409), 25-35.

(14) Hemsley, P. A.; Weimar, T.; Lilley, K. S.; Dupree, P.; Grierson, C. S. A proteomic approach identifies many novel palmitoylated proteins in A rabidopsis. *New Phytologist* **2013**, *197* (3), 805-814.

(15) Li, Y.; Scott, R.; Doughty, J.; Grant, M.; Qi, B. Protein S-Acyltransferase 14: a specific role for palmitoylation in leaf senescence in Arabidopsis. *Plant Physiology* **2016**, *170* (1), 415-428.

(16) Webb, Y.; Hermida-Matsumoto, L.; Resh, M. D. Inhibition of protein palmitoylation, raft localization, and T cell signaling by 2-bromopalmitate and polyunsaturated fatty acids. *Journal of Biological Chemistry* **2000**, *275* (1), 261-270.

(17) Zhou, F.; Xue, Y.; Yao, X.; Xu, Y. CSS-Palm: palmitoylation site prediction with a clustering and scoring strategy (CSS). *Bioinformatics* **2006**, *22* (7), 894-896.

(18) Xue, Y.; Chen, H.; Jin, C.; Sun, Z.; Yao, X. NBA-Palm: prediction of palmitoylation site implemented in Naive Bayes algorithm. *BMC bioinformatics* **2006**, *7*, 1-10.

(19) Ren, J.; Wen, L.; Gao, X.; Jin, C.; Xue, Y.; Yao, X. CSS-Palm 2.0: an updated software for palmitoylation sites prediction. *Protein Engineering, Design & Selection* **2008**, *21* (11), 639-644.

(20) Wang, X.-B.; Wu, L.-Y.; Wang, Y.-C.; Deng, N.-Y. Prediction of palmitoylation sites using the composition of k-spaced amino acid pairs. *Protein Engineering, Design & Selection* **2009**, *22* (11), 707-712.

(21) Hu, L.-L.; Wan, S.-B.; Niu, S.; Shi, X.-H.; Li, H.-P.; Cai, Y.-D.; Chou, K.-C. Prediction and analysis of protein palmitoylation sites. *Biochimie* **2011**, *93* (3), 489-496.

(22) Shi, S.-P.; Sun, X.-Y.; Qiu, J.-D.; Suo, S.-B.; Chen, X.; Huang, S.-Y.; Liang, R.-P. The prediction of palmitoylation site locations using a multiple feature extraction method. *Journal of Molecular Graphics and Modelling* **2013**, *40*, 125-130.

(23) Kumari, B.; Kumar, R.; Kumar, M. PalmPred: an SVM based palmitoylation prediction method using sequence profile information. *PloS one* **2014**, *9* (2), e89246.

(24) Li, S.; Li, J.; Ning, L.; Wang, S.; Niu, Y.; Jin, N.; Yao, X.; Liu, H.; Xi, L. In silico identification of protein S-palmitoylation sites and their involvement in human inherited disease. *Journal of chemical information and modeling* **2015**, *55* (9), 2015-2025.

(25) Ning, W.; Jiang, P.; Guo, Y.; Wang, C.; Tan, X.; Zhang, W.; Peng, D.; Xue, Y. GPS-Palm: a deep learning-based graphic presentation system for the prediction of S-palmitoylation sites in proteins. *Briefings in Bioinformatics* **2020**, *22* (2), 1836-1847. DOI: 10.1093/bib/bbaa038 (acccessed 11/7/2023).

(26) Bekker, J.; Davis, J. Learning from positive and unlabeled data: A survey. *Machine Learning* **2020**, *109*, 719-760.

(27) Alkuhlani, A.; Gad, W.; Roushdy, M.; Salem, A.-B. M. Pustackngly: positive-unlabeled and stacking learning for n-linked glycosylation site prediction. *IEEE Access* **2022**, *10*, 12702-12713.

(28) Li, Z.; Hu, L.; Tang, Z.; Zhao, C. Predicting HIV-1 protease cleavage sites with positive-unlabeled learning. *Frontiers in Genetics* **2021**, *12*, 658078.

(29) UniProt: the universal protein knowledgebase. *Nucleic acids research* **2017**, *45* (D1), D158-D169.

(30) Kumar, M.; Carr, P.; Turner, S. R. An atlas of Arabidopsis protein S-acylation reveals its widespread role in plant cell organization and function. *Nature Plants* **2022**, *8* (6), 670-681.

(31) Cheng, C. Y.; Krishnakumar, V.; Chan, A. P.; Thibaud-Nissen, F.; Schobel, S.; Town, C. D. Araport11: a complete reannotation of the Arabidopsis thaliana reference genome. *The Plant Journal* **2017**, *89* (4), 789-804.

(32) 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. DOI: 10.1093/bioinformatics/btl158 (acccessed 11/15/2023).

(33) Wang, J.; Yang, B.; Revote, J.; Leier, A.; Marquez-Lago, T. T.; Webb, G.; Song, J.; Chou, K.-C.; Lithgow, T. POSSUM: a bioinformatics toolkit for generating numerical sequence feature descriptors based on PSSM profiles. *Bioinformatics* **2017**, *33* (17), 2756-2758. DOI: 10.1093/bioinformatics/btx302 (acccessed 11/15/2023).

(34) Park, K.-J.; Kanehisa, M. Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs. *Bioinformatics* **2003**, *19* (13), 1656-1663. DOI: 10.1093/bioinformatics/btg222 (acccessed 11/14/2023).

(35) Zhao, X.; Zhang, W.; Xu, X.; Ma, Z.; Yin, M. Prediction of protein phosphorylation sites by using the composition of k-spaced amino acid pairs. **2012**.

(36) Ju, Z.; Wang, S.-Y. Prediction of citrullination sites by incorporating k-spaced amino acid pairs into Chou's general pseudo amino acid composition. *Gene* **2018**, *664*, 78-83.

(37) Teng, Z.; Zhang, Z.; Tian, Z.; Li, Y.; Wang, G. ReRF-Pred: predicting amyloidogenic regions of proteins based on their pseudo amino acid composition and tripeptide composition. *BMC bioinformatics* **2021**, *22*, 1-18.

(38) Li, Y.; Zhang, Z.; Teng, Z.; Liu, X. Predamyl-mlp: Prediction of amyloid proteins using multilayer perceptron. *Computational and Mathematical Methods in Medicine* **2020**, *2020*.

(39) Shen, J.; Zhang, J.; Luo, X.; Zhu, W.; Yu, K.; Chen, K.; Li, Y.; Jiang, H. Predicting protein–protein interactions based only on sequences information. *Proceedings of the National Academy of Sciences* **2007**, *104* (11), 4337-4341.

(40) cheol Jeong, J.; Lin, X.; Chen, X.-W. On position-specific scoring matrix for protein function prediction. *IEEE/ACM transactions on computational biology and bioinformatics* **2010**, *8* (2), 308-315.

(41) An, J.-Y.; You, Z.-H.; Meng, F.-R.; Xu, S.-J.; Wang, Y. RVMAB: using the relevance vector machine model combined with average blocks to predict the interactions of proteins from protein sequences. *International Journal of Molecular Sciences* **2016**, *17* (5), 757.

(42) Zhai, J.-X.; Cao, T.-J.; An, J.-Y.; Bian, Y.-T. Highly accurate prediction of protein self-interactions by incorporating the average block and PSSM information into the general PseAAC. *Journal of theoretical biology* **2017**, *432*, 80-86.

(43) Juan, E. Y.; Li, W.; Jhang, J.; Chiu, C. Predicting protein subcellular localizations for gram-negative bacteria using DP-PSSM and support vector machines. In *2009 International Conference on Complex, Intelligent and Software Intensive Systems*, 2009; IEEE: pp 836-841.

(44) Zou, L.; Nan, C.; Hu, F. Accurate prediction of bacterial type IV secreted effectors using amino acid composition and PSSM profiles. *Bioinformatics* **2013**, *29* (24), 3135-3142.

(45) Zahiri, J.; Yaghoubi, O.; Mohammad-Noori, M.; Ebrahimpour, R.; Masoudi-Nejad, A. PPIevo: Protein–protein interaction prediction from PSSM based evolutionary information. *Genomics* **2013**, *102* (4), 237-242.

(46) Liu, T.; Zheng, X.; Wang, J. Prediction of protein structural class for low-similarity sequences using support vector machine and PSI-BLAST profile. *Biochimie* **2010**, *92* (10), 1330-1334.

(47) Zhang, L.; Zhao, X.; Kong, L. Predict protein structural class for low-similarity sequences by evolutionary difference information into the general form of Chou' s pseudo amino acid composition. *Journal of theoretical biology* **2014**, *355*, 105-110.

(48) Saini, H.; Raicar, G.; Lal, S. P.; Dehzangi, A.; Imoto, S.; Sharma, A. Protein fold recognition using genetic algorithm optimized voting scheme and profile bigram. *Journal of Software* **2016**, *11* (8), 756-767.

(49) Clarke, N. D. Sequence 'minimization': exploring the sequence landscape with simplified sequences. *Current Opinion in Biotechnology* **1995**, *6* (4), 467-472.

(50) Akanuma, S.; Kigawa, T.; Yokoyama, S. Combinatorial mutagenesis to restrict amino acid usage in an enzyme to a reduced set. *Proceedings of the National Academy of Sciences* **2002**, *99* (21), 13549-13553.

(51) Chen, Y.-L.; Li, Q.-Z. Prediction of the subcellular location of apoptosis proteins. *Journal of Theoretical Biology* **2007**, *245* (4), 775-783.

(52) Liang, Y.; Yang, S.; Zheng, L.; Wang, H.; Zhou, J.; Huang, S.; Yang, L.; Zuo, Y. Research progress of reduced amino acid alphabets in protein analysis and prediction. *Computational and Structural Biotechnology Journal* **2022**.

(53) Etchebest, C.; Benros, C.; Bornot, A.; Camproux, A.-C.; De Brevern, A. A reduced amino acid alphabet for understanding and designing protein adaptation to mutation. *European Biophysics Journal* **2007**, *36*, 1059-1069.

(54) Yu, L.; Liu, H. Efficient feature selection via analysis of relevance and redundancy. *The Journal of Machine Learning Research* **2004**, *5*, 1205-1224.

(55) Kursa, M. B.; Rudnicki, W. R. The all relevant feature selection using random forest. *arXiv preprint arXiv:1106.5112* **2011**.

(56) Wilton, J.; Koay, A.; Ko, R.; Xu, M.; Ye, N. Positive-Unlabeled Learning using Random Forests via Recursive Greedy Risk Minimization. *Advances in Neural Information Processing Systems* **2022**, *35*, 24060-24071.

(57) Jain, S.; White, M.; Trosset, M. W.; Radivojac, P. Nonparametric semi-supervised learning of class proportions. *arXiv preprint arXiv:1601.01944* **2016**.

(58) Amarappa, S.; Sathyanarayana, S. Data classification using Support vector Machine (SVM), a simplified approach. *Int. J. Electron. Comput. Sci. Eng* **2014**, *3*, 435-445.

(59) Le, D.-H. Machine learning-based approaches for disease gene prediction. *Briefings in functional genomics* **2020**, *19* (5-6), 350-363.

(60) Saunders, J. D.; Freitas, A. A. Evaluating the Predictive Performance of Positive-Unlabelled Classifiers: a brief critical review and practical recommendations for improvement. *ACM SIGKDD Explorations Newsletter* **2022**, *24* (2), 5-11.

(61) Jain, S.; White, M.; Radivojac, P. Recovering true classifier performance in positive-unlabeled learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017; Vol. 31.

(62) Lee, W. S.; Liu, B. Learning with positive and unlabeled examples using weighted logistic regression. In *ICML*, 2003; Vol. 3, pp 448-455.

(63) Claesen, M.; Davis, J.; De Smet, F.; De Moor, B. Assessing binary classifiers using only positive and unlabeled data. *arXiv preprint arXiv:1504.06837* **2015**.

(64) Mandrekar, J. N. Receiver operating characteristic curve in diagnostic test assessment. *Journal of Thoracic Oncology* **2010**, *5* (9), 1315-1316.

(65) Chen, Y.; Du, X.; Guo, M. Self-paced ensemble for constructing an efficient robust high-performance classification model for detecting mineralization anomalies from geochemical exploration data. *Ore Geology Reviews* **2023**, 105418.

(66) Davis, J.; Goadrich, M. The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning*, 2006; pp 233-240.