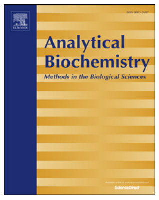
Contents lists available at ScienceDirect

Analytical Biochemistry

journal homepage: www.elsevier.com/locate/yabio



Predicting pupylation sites in prokaryotic proteins using semi-supervised self-training support vector machine algorithm

Zhe Ju, Hong Gu [*]

School of Control Science and Engineering, Dalian University of Technology, Dalian 116024, People's Republic of China

ARTICLE INFO

Article history:
Received 22 March 2016
Received in revised form
21 April 2016
Accepted 8 May 2016
Available online 16 May 2016

Keywords:
Post-translational modification
Pupylation
Semi-supervised learning
Support vector machine
k-spaced amino acid pair

ABSTRACT

As one important post-translational modification of prokaryotic proteins, pupylation plays a key role in regulating various biological processes. The accurate identification of pupylation sites is crucial for understanding the underlying mechanisms of pupylation. Although several computational methods have been developed for the identification of pupylation sites, the prediction accuracy of them is still unsatisfactory. Here, a novel bioinformatics tool named IMP–PUP is proposed to improve the prediction of pupylation sites. IMP–PUP is constructed on the composition of *k*-spaced amino acid pairs and trained with a modified semi-supervised self-training support vector machine (SVM) algorithm. The proposed algorithm iteratively trains a series of support vector machine classifiers on both annotated and non-annotated pupylated proteins. Computational results show that IMP–PUP achieves the area under receiver operating characteristic curves of 0.91, 0.73, and 0.75 on our training set, Tung's testing set, and our testing set, respectively, which are better than those of the different error costs SVM algorithm and the original self-training SVM algorithm. Independent tests also show that IMP–PUP significantly outperforms three other existing pupylation site predictors: GPS–PUP, iPUP, and pbPUP. Therefore, IMP–PUP can be a useful tool for accurate prediction of pupylation sites. A MATLAB software package for IMP–PUP is available at https://juzhe1120.github.io/.

Recently, a prokaryotic ubiquitin-like protein (Pup) has been identified in prokaryotes [1,2]. Pup is an intrinsically disordered protein with 64 amino acids and marks the target proteins that are needed to be degraded [3,4]. The process of Pup attaching substrate lysine via isopeptide bonds is named pupylation, which plays a key role in regulating various cellular processes such as protein degradation and signal transduction in prokaryotic cells [5]. Although pupylation and ubiquitylation are functional analogs, the enzymology involved in the two processes is different [6]. In contrast to eukaryotic ubiquitylation requiring three enzymes (activating enzyme, conjugating enzyme, and protein ligase), prokaryotic pupylation requires only two enzymes: deamidase of Pup (DOP) and proteasome accessory factor A (PafA). The C-terminal glutamine of Pup is first deamidated to glutamate via DOP [7]. Subsequently, the deamidated Pup is attached to specific lysine of substrate proteins by PafA [8]. However, the molecular mechanism of prokaryotic pupylation remains largely unknown.

To better understand the underlying mechanisms of pupylation, the fundamental task is the accurate identification of pupylated substrates and their pupylation sites. Recently, several large-scale proteomics methods have been applied to identify pupylated proteins and pupylation sites [9–12]. Because conventional experimental approaches are usually costly and laborious, it is urgent to develop computational methods to identify the potential pupylation sites. Up to now, several predictors have been developed for the prediction of pupylation sites. With the group-based prediction system (GPS) 2.2 algorithm, Liu and coworkers [13] developed the first predictor GPS–PUP for the prediction of the pupylation sites. Tung [14] developed a predictor, iPUP, using the composition of *k*-spaced amino acid pairs (CKSAAPs) surrounding lysine-centered peptides and support vector machine (SVM) algorithm. Chen and coworkers [15] proposed an SVM-based predictor named PupPred, in which an amino acid pair feature was employed to encode lysine-centered peptides. Recently, Hasan and coworkers [16] developed a web server named pbPUP to predict pupylation sites using profile-based CKSAAPs and SVM algorithm.

Abbreviations used: Pup, prokaryotic ubiquitin-like protein; DOP, deamidase of Pup; PafA, proteasome accessory factor A; GPS, group-based prediction system; CKSAAP, composition of *k*-spaced amino acid pair; SVM, support vector machine; DEC–SVM, different error cost SVM; Sn, sensitivity; Sp, specificity; Pre, precision; ACC, accuracy; MCC, Matthew's correlation coefficient; AUC, area under receiver operating characteristic curve; SET–SVM, self-training SVM; PTMs, post-translational modifications; RBF, radial basis function.

* Corresponding author.
E-mail address: guhong@dlut.edu.cn (H. Gu).

http://dx.doi.org/10.1016/j.ab.2016.05.005
0003-2697/© 2016 Elsevier Inc. All rights reserved.

Note that the aforementioned existing predictors were generally trained on the experimentally annotated pupylated proteins that were collected from the PupDB database [6]. However, there are only 268 annotated pupylated proteins with 311 known pupylation sites in the current version of PupDB. Because the number of annotated pupylated proteins is relatively small, they could not reflect the real distribution of pupylation sites commendably. As a result, the prediction accuracy of existing computational methods is still unsatisfactory. In fact, besides the 268 annotated pupylated proteins, there also exist 1116 pupylated proteins whose specific pupylation sites are still unknown in PupDB. Although the specific pupylation sites in these non-annotated pupylated proteins are still unknown, at least one lysine residue in each of them is pupylated. By taking advantage of these non-annotated pupylated proteins, the performance of predictor might be improved. In this study, a predictor named IMP–PUP is proposed to improve the prediction of pupylation sites by using a modified self-training SVM algorithm. Specifically, an initial SVM classifier is trained on annotated pupylated proteins (labeled samples) using the CKSAAP and *F*-score feature selection method, and the initial SVM is used to classify the non-annotated pupylated proteins (unlabeled samples). Then, the predicted unlabeled samples with the highest confidence value are added to the training set. Here, a minimum distance rule is introduced to design the confidence function, by which the proposed algorithm selects the predicted unlabeled samples nearest to the labeled set instead of selecting the samples with the highest SVM scores. The above process is repeated until the algorithm is convergent. A final SVM classifier that is used to construct IMP–PUP is obtained at the end of iteration. As illustrated by our experimental results, the performance of the predictor has been improved effectively by adding the unlabeled samples to training data. Independent tests indicated that IMP–PUP outperforms three other existing predictors significantly.

Materials and methods

Dataset

Tung's training set and independent test set [14] were used in this study. The training set contained 162 proteins, including 183 pupylation sites and 2258 non-pupylation sites. The independent test set contained 20 proteins, including 29 pupylation sites and 408 non-pupylation sites. To enlarge the training dataset, we also collected 55 *Corynebacterium glutamicum* proteins and 31 *Rhodococcus erythropolis* proteins from PupDB (version 1.3, September 2015). These 86 pupylated proteins are newly additional proteins in PupDB and have never been used to train the previous predictors. We randomly selected 20 proteins from the dataset of *C. glutamicum* and *R. erythropolis* to construct our independent testing set, and the remaining 66 proteins combined with Tung's training set were used to construct our training set. Consequently, our training set contained 228 proteins, including 257 pupylation sites and 3209 non-annotated pupylation sites, whereas our independent test set contained 20 proteins, including 21 pupylation sites and 318 non-pupylation sites.

We also collected 1116 pupylated proteins whose pupylation sites were still unknown from PupDB. These non-annotated pupylated proteins were identified by high-throughput proteomics methods [9–12]. These non-annotated pupylated proteins contained 14,955 lysine residues that were used to construct our unlabeled training set.

The sliding window method was used to encode every lysine residue K in our dataset because pupylation occurred only in lysine residues. According to previous study [14], window size was selected as 21 in this study. Thus, every sample in the aforementioned datasets was represented as a peptide segment with 10 residues upstream and 10 residues downstream of lysine residue K. To ensure the uniform length of each peptide, an added residue "X" was employed to fill the corresponding position where there was no sufficient residue. Our training set, independent testing set, and unlabeled training set are provided in Supplementary Material S1 of the online supplementary material.

Feature construction

It has been pointed out in previous studies [14,15] that the CKSAAP encoding is more suitable for representing the peptides around the pupylation sites than other encoding schemes. The CKSAAP feature has been widely applied to various post-translational modifications (PTMs) site prediction such as O-glycosylation sites [17], palmitoylation sites [18], ubiquitination sites [19], phosphorylation sites [20], and methylation sites [21]. In this study, CKSAAPs were employed to encode training peptides. The CKSAAP encoding scheme calculates the occurrence frequencies of the *k*-spaced amino acid pairs in a given sequence fragment. The *k*-spaced amino acid pair means the amino acid pair that is separated by any *k* of 21 amino acids. For example, for a sequence fragment of *m* amino acids, the CKSAAP encoding with $k = 2$ is a 441-dimensional feature vector defined as

$$\left(\frac{N_{AxxA}}{N_{\mathrm{Total}}}, \frac{N_{AxxC}}{N_{\mathrm{Total}}}, \frac{N_{AxxD}}{N_{\mathrm{Total}}}, \ldots, \frac{N_{XxxX}}{N_{\mathrm{Total}}}\right)_{441} \tag{1}$$

where *x* means any one of 21 amino acids and N_{Total} represents the number of 2-spaced amino acid pairs. After our preliminary trials, CKSAAPs with $k = 0, 1, 2, 3, 4$, and 5 were jointly used to encode training peptides in this study (see Supplementary Material S2). Thus, the feature vector of each training peptide was represented as 2646-dimensional vectors.

Feature selection

The *F*-score feature selection method [22] was used to remove the irrelevant and redundant features. The *F*-score of the *j*-th feature is defined as

$$F(j) = \frac{\left(\bar{x}_j^{(+)} - \bar{x}_j\right)^2 + \left(\bar{x}_j^{(-)} - \bar{x}_j\right)^2}{\frac{1}{m^+-1}\sum_{k=1}^{m}\left(\bar{x}_{k,j}^{(+)} - \bar{x}_j^{(+)}\right)^2 + \frac{1}{m^--1}\sum_{k=1}^{m}\left(\bar{x}_{k,j}^{(-)} - \bar{x}_j^{(-)}\right)^2}, \tag{2}$$

where $\bar{x}_j$, $\bar{x}_j^{(+)}$, $\bar{x}_j^{(-)}$ are the mean values of the *j*-th feature in whole, positive, and negative training samples, respectively. m^+ denotes the number of positive training samples, m^- denotes the number of negative training samples, $\bar{x}_{k,j}^{(+)}$ denotes the *j*-th feature of the *k*-th positive training sample, and $\bar{x}_{k,j}^{(-)}$ denotes the *j*-th feature of the *k*-th negative training sample.

Different error cost SVM

The core learning algorithm of IMP–PUP was SVM. However, our training set was unbalanced (the ratio between pupylated peptides and non-pupylated peptides is roughly 1:12). Therefore, the different error cost SVM (DEC–SVM) [23] was applied to our problem. By assigning a bigger penalty for the positive samples than for the negative samples, the effects of class imbalance can be reduced in the DEC–SVM model. Suppose a training set consists of *l* labeled samples (x_i, t_i), $i = 1, \ldots, l$, where the first *p* examples are positive examples (i.e., $t_i = 1$, $i = 1, \ldots, p$) and the rest of the $l - p$