molecules

MDPI

Article

EPuL: An Enhanced Positive-Unlabeled Learning Algorithm for the Prediction of Pupylation Sites

Xuanguo Nan [1], Lingling Bao [1], Xiaosa Zhao [1], Xiaowei Zhao [1], Arun Kumar Sangaiah [2], Gai-Ge Wang [3,*] and Zhiqiang Ma [1,*]

[1] School of Information Science and Technology, Northeast Normal University, Changchun 130117, China; biocs_nenu@126.com (X.N.); baoll601@nenu.edu.cn (L.B.); zhaoxs686@nenu.edu.cn (X.Z.); zhaoxw303@nenu.edu.cn (X.Z.)

[2] School of Computing Science and Engineering, VIT University, Vellore 632014, Tamil Nadu, India; arunkumarsangaiah@gmail.com

[3] School of Computer Science and Technology, Jiangsu Normal University, Xuzhou 221116, China

[*] Correspondence: gaigewang@163.com (G.-G.W.); zhiqiang.ma967@gmail.com (Z.M.); Tel.: +86-0431-8453-6338 (G.-G.W.); Fax: +86-0431-8453-6338 (G.-G.W.)

Received: 23 July 2017; Accepted: 30 August 2017; Published: 5 September 2017

Abstract: Protein pupylation is a type of post-translation modification, which plays a crucial role in cellular function of bacterial organisms in prokaryotes. To have a better insight of the mechanisms underlying pupylation an initial, but important, step is to identify pupylation sites. To date, several computational methods have been established for the prediction of pupylation sites which usually artificially design the negative samples using the verified pupylation proteins to train the classifiers. However, if this process is not properly done it can affect the performance of the final predictor dramatically. In this work, different from previous computational methods, we proposed an enhanced positive-unlabeled learning algorithm (EPuL) to the pupylation site prediction problem, which uses only positive and unlabeled samples. Firstly, we separate the training dataset into the positive dataset and the unlabeled dataset which contains the remaining non-annotated lysine residues. Then, the EPuL algorithm is utilized to select the reliably negative initial dataset and then iteratively pick out the non-pupylation sites. The performance of the proposed method was measured with an accuracy of 90.24%, an Area Under Curve (AUC) of 0.93 and an MCC of 0.81 by 10-fold cross-validation. A user-friendly web server for predicting pupylation sites was developed and was freely available at http://59.73.198.144:8080/EPuL.

Keywords: positive-unlabeled learning algorithm; pupylation sites; prediction; web server; support vector machine

1. Introduction

Prokaryotic ubiquitin-like proteins (Pup) are the first identified post-translational small modifier in prokaryotes [1,2]. They are disordered proteins, including 64 amino acids and an important signal for the protein's selective degradation [3]. Pup usually attaches to substrate lysine via isopeptide bonds, and this process is called pupylation. Although the function of pupylation and ubiquitylation is similar, the enzymology participating in these processes is not the same [4]. Ubiquitylation requires three types of enzymes, including activating enzymes, ligases, and conjugating enzymes [5–7]. Pupylation only requires two types of enzymes (proteasome accessory factor A and deamidase of Pup).

Accurate identification of pupylation sites is an essential first step to better understand the underlying mechanism of protein pupylation. Though some large-scale proteomics technologies have been adopted to find the pupylation sites, they are usually time-consuming and laborious, especially for large-scale protein samples. Thus, computational methods were needed to effectively and accurately identify the potential pupylation sites in protein sequences. Lin et al. [8] developed the first pupylation site predictor, named GPS-PUP, the GPS means Group-based Prediction System. Tung et al. [9] constructed a predictor, iPup, in which the composition of k-spaced amino acid pairs feature (CKSAAP) was used. Zhao et al. [10] created a predictive model with five features and adopted feature selection methods to find the optimal feature set. Chen et al. [11] proposed a predictor, PupPred, which is based on the SVM and some sequence-derived features. Hasem et al. [12] introduced a profile-based CKSAAP to encode the pupylation sites and built a predictor called pbPUP. Wand et al. [13] employed the non-annotated lysine sites as unlabeled training samples and then used a two-class SVM to expand reliable negative set at each iteration. More recently, Jiang et al. [14] applied the positive-unlabeled learning technique to the prediction of pupylation sites, which combined the SVM and CKSAAP to construct the predictor PUL-PUP.

However, most of these computational methods artificially constructed the negative samples which included all the remaining non-annotated lysine residues. This negative samples dataset may contain some pupylation sites which were not validated. Then the classifiers trained on the experimentally-verified positive samples and such negative samples may be problematic and biased, and the final prediction performance was unsatisfactory. In this paper, we proposed an enhanced positive unlabeled learning algorithm to identify pupylation sites, EPuL, which enhanced the reliability of initial negative samples and then iteratively identified the non-pupylation sites from the unlabeled samples. Experimental results showed that our method achieved better performance when compared with other existing methods. Meanwhile, a user-friendly webserver of our proposed predictor was freely accessible at reference [15].

2. Results and Discussion

2.1. The Development of EPuL

The training dataset consisted of two kinds of subsets: (1) the positive dataset P and (2) the unlabeled dataset U. Positive-unlabeled learning has been used in bioinformatics and obtains satisfactory performance [16–18]. In this study, we proposed an enhanced positive-unlabeled learning algorithm called EPuL to predict pupylation sites. The detailed process of the algorithm is described as follows (Stage 1 is our proposed part. Stage 2 and Stage 3 are the same as PUL-PUP [14]):

Stage 1: Select the reliably negative initial set

The reliably negative dataset RN is initialized to an empty set and we use a vector V_{s_i} to represent each sample in P and U by using the CKSAAP encoding scheme. By summing up all the vectors in P, we built the 'positive representative vector (pr)' and normalized it by using the formula below:

$$pr = \sum_i^{|P|} V_{s_i} / |P| \tag{1}$$

Then, maximum distance rule is adopted, and the Euclidean distance was utilized to compute the average distance of each sample s_i in U to pr:

$$Avg\_dist+ = \sum_{\mathrm{i}}^{|U|} dist(pr, V_{s_i}) / |U| \tag{2}$$

For each sample s_i in U, the likely initial negative set LN was selected from U by Avg_dist; that is, if dist(pr,V_{s_i}) is more than $Avg\_dist \times \partial$ ($\partial = 1.05$), we regard s_i as a likely negative sample and put it into LN: $LN = LN \cup \{s_i\}$.

To select the reliably negative initial set RN^0 and enhance the reliability of RN^0, we randomly divide LN into five likely negative subsets and each of them builds a model with P, which is based on the SVM. Subsequently, the remaining dataset $U - RN$ is classified by the five models, respectively.

The common sequences cs which are predicted by five models and the negative support vectors N_{sv} of the five models are all used to represent the reliably negative initial set RN^0, in which, $RN^0 = cs + N_{sv}$.

Stage 2: Expand the reliably negative set

After the selection of reliably negative initial set, the reliable negative set was expanded by iteratively adding the negative examples from U using a series of two-class SVM classifiers. Specifically, at the ith iteration, the SVM classifier f^i is firstly trained using dataset $P + RN^i$; then, f^i was used to classify the U^i and each sample x^i in U^i, and each sample was obtained a decision value $f(x^i)$. To insure the reliability, samples belonging to the negative set need to satisfy:

$$f\left(x^i\right) \leq T$$

Here, we set $T = -0.50$.

To overcome the problem of imbalance at each iteration, the negative support vectors N_{sv}^i and the newly-predicted negative samples N_{pred}^i are used to represent the existing negative set RN^i, and we control the size of N_{pred}^i less than $2 \times |P|$. Then, at the $i + 1$th iteration, $U^{i+1} = U^i - N_{pred}^i$; $RN^{i+1} = N_{pred}^i \cup N_{sv}^i$. Classifier f^{i+1} is trained on P and current reliable negative training set RN^{i+1}.

With the expansion of negative set, the size of the remaining unlabeled set becomes less and less. Thus, iteration should be terminated at some point. When the number of the remaining unlabeled sets goes below the threshold $5 \times |P|$, the unlabeled data with the positive data would correspond to the maximum MCC.

Stage 3: Return the final classifier

After the extraction of the reliably negative set, a final SVM classifier is trained on P and the reliable negative set RN.

Algorithm 1 summarizes the detailed procedures of the proposed method EPuL.

2.2. The Performance of EPuL on the Training Dataset

To evaluate the effectiveness of the proposed method for pupylation site prediction, we compare EPuL with other methods, including PUL-PUP [14], PSoL [13], and SVM balance on the training dataset. In PSoL [13] algorithms, a two-class SVM is applied to filter the negative set from the non-annotated lysine sites and expand the negative set at each iteration. Additionally, in PUL-PUP [14] algorithms, the non-annotated lysine sites are treated as unlabeled samples and positive-unlabeled learning technique is used to predict of pupylation sites. The difference for us is on the selection of the initial negative set. As for SVM_balance, the negative training dataset is randomly selected from the non-annotated lysine sites. The ratio of the positive and negative training datasets is 1:1, which can avoid the imbalanced problem. The 10-fold cross-validation is performed on the positive set P and the reliably negative set RN, the results are shown in Table 1. We can see from Table 1 that EPuL yielded the best performance, a Sn of 84.21%, Sp of 95.45%, ACC of 90.24, and MCC of 0.81. EPuL achieves an improvement on the training dataset. Among this, the results of PSoL and SVM_balance are taken from PUL-PUP. To further demonstrate the superiority of EPuL, we also draw the ROC curve, as shown in Figure 1.

Table 1. Ten-fold cross-validation performance of EPuL, PUL-PUP, PSoL and SVM_balance.

Method	Sn (%)	Sp (%)	ACC (%)	MCC	AUC
EPuL	84.21	95.45	90.24	0.81	0.93
PUL-PUP	82.24	91.57	88.92	0.74	0.92
PSoL	67.50	73.60	70.55	0.42	0.80
SVM_balance	76.71	63.65	69.88	0.40	0.77

```
Algorithm 1. An enhanced positive-unlabeled learning algorithm.

Input       $P$—Positive training set; $U$—Unlabeled training set;
            $\partial$—The distance coefficient; $V_{s_i}$—Sequence $s_i$ in $P$ and $U$;
            $Model_{1,2,3,4,5}$ —Five models trained by five subsets with $P$ respectively;
            $N_{1,2,3,4,5}$ —Five negative sets predicted by $Model_{1,2,3,4,5}$ on the remaining unlabeled
            training set respectively;
            cs—Common sequences of five negative sets $N_{1,2,3,4,5}$
            $N_{sv}$—Negative support vectors of five $Model_{1,2,3,4,5}$
Output      $F$—Final classifier.
Stage 0:    Initialization
            $l \leftarrow 0; Avg\_dist = 0; LN = \varnothing; RN = \varnothing; i$
Stage 1:    Select the reliably negative initial set
            $pr = \sum_{\mathrm{i}}^{|P|} V_{s_i} / |P|;$
            $Avg\_dist += \sum_{i}^{|U|} dist(pr, V_{s_i}) / |U|;$
            FOR $i$ from 1 to $|U|$
                IF $dist(pr, V_{s_i}) > Avg\_dist * \partial$
                    $LN = LN \cup \{S_i\};$
                END IF
            END FOR
            Randomly divide the $LN$ into five subsets $D_1, D_2, D_3, D_4, D_5$.
            FOR $i$ from 1 to 5
                $Model_i = \mathrm{SVM}(P, D_i);$
                $N_i = Model_i(U - LN);$
            END FOR
            The common sequence are represented to reliably negative initial set
            $cs = N_1 \cap N_2 \cap N_3 \cap N_4 \cap N_5; RN^0 = RN^0 \cup cs;$
            then the negative support vectors $N_{sv}$ of five models are included in $RN^0 = RN^0 \cup N_{sv}$.
Stage 2     Expand the reliably negative set
            WHILE TRUE
            IF $U^l > 5 * |P|$
            $U^{l+1} = U^l - N_{pred}^l;$
            $RN^{l+1} = N_{pred}^l \cup N_{sv}^l;$
            ELSE IF $U^l < 5 * |P|$
            Go to Stage 3
            END IF
            Train a SVM classifier $f^{l+1}$ on the $P \cup RN^{l+1}$ with optimal parameter $C$ and $\gamma$.
            Each sequence $x_i$ in $U^{l+1}$ would have a decision value $f(x_i)$ through the obtained $f^{l+1}$,
            use the threshold T to get the reliably negative set.
            $l \leftarrow l + 1$
Stage 3     Return the final classifier
            Return $F = (P, RN)$
```

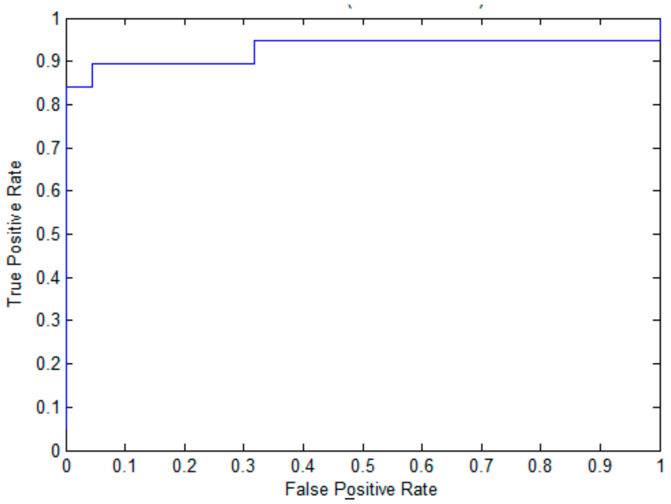

Figure 1. The ROC curve of EPuL on the training dataset.

2.3. The Performance Evaluation on the Independent Testing Dataset

In order to further evaluate the performance of the proposed predictor, the independent testing dataset was utilized, which was completely blind to the training dataset. Table 2 presents the comparison of the results among EPuL, PUL-PUP, PSoL, and SVM-balance. Although SVM_balance can avoid the imbalanced problem, its prediction performance was the lowest, because the negative set of SVM_balance is randomly selected and are not the reliably negative samples. The PUL-PUP, which also uses the positive-unlabeled learning technique, mainly improves the performances through containing more information in *RN* at each iteration. However, the performance of PUL-PUP was not better than EPuL because the contained points are only based on the distance and not very precise. Especially, the stage 2 of EPuL is similar to PSoL, but we select the reliably negative initial set at stage 1, enhancing the positive-unlabeled learning at the beginning which would contribute to the selection of a more accurate negative set and make our algorithm more effective than PSoL.

Table 2. Independent test performance of EPuL, PUL-PUP, and PSoL.

Method	Sn (%)	Sp (%)	ACC (%)	MCC	AUC
EPuL	72.41	71.57	71.63	0.24	0.78
PUL-PUP	68.97	70.83	70.71	0.22	0.77
PSoL	51.72	73.14	71.62	0.13	0.74
SVM-balance	62.07	67.4	67.05	0.15	0.7

We also compare our method with four existing predictors: iPUP, GPS-PUP, pbPUP, and PUL-PUP. We predefined three thresholds according to the SVM scores; that is, high (0.9672), medium (0.4032), and low (0.1088). Table 3 presents the detailed prediction performances on the independent testing dataset. The performance of our algorithm outperforms the existing predictors. For example, at the threshold low, the MCC of EPuL is 0.24, which is higher than that of GPS-PUP with an MCC of 0.1, iPUP with MCC of 0.15, pbPUP with MCC of 0.07, and PUL-PUP with MCC of 0.23. Moreover, our method obtains the best AUC value (0.78). Our classifier is iteratively trained on *P* and *RN*. Only with the reliable initial negative set can was obtain a more reliable negative set in the subsequent iterations. Thus, our method is more accurate and suitable for predicting pupylation sites than other methods.

Table 3. The performance of EPuL and four exiting pupylation sites predictors on the independent testing dataset.

Predictors	Thresholds	Sn (%)	Sp (%)	ACC (%)	MCC	AUC
GPS-PUP	High	31.03	89.46	85.62	0.16	0.6
	Medium	34.48	85.54	82.19	0.14	
	Low	41.38	76.72	74.43	0.1	
iPUP	High	48.28	82.84	80.55	0.2	0.66
	Medium	51.72	76.47	74.83	0.16	
	Low	55.17	72.06	70.94	0.15	
pbPUP	High	17.24	88.48	83.75	0.04	0.6
	Medium	31.03	80.15	76.89	0.07	
	Low	41.38	69.85	67.96	0.07	
PUL-PUP	High	51.72	83.33	81.24	0.22	0.77
	Medium	65.52	76.72	75.97	0.24	
	Low	68.97	72.79	72.54	0.23	
EPuL	High	37.93	89.46	86.04	0.21	0.78
	Medium	58.62	79.90	78.49	0.23	
	Low	68.97	74.02	73.68	0.24	

2.4. Feature Analysis

Through the feature selection method, we can find the ranked features generated by the CKSAAP encoding scheme in Figure 2. The importance of these features was also clearly and intuitively shown in Figure 3. For example, the feature ExE which represents the EE residue pair spaced by any amino acid, is enriched in the positive pair and not in the negative pair. From Figure 2, we can see that the features frequently appeared in the top 25 amino acid pairs, which also frequently occurred in Figure 3.

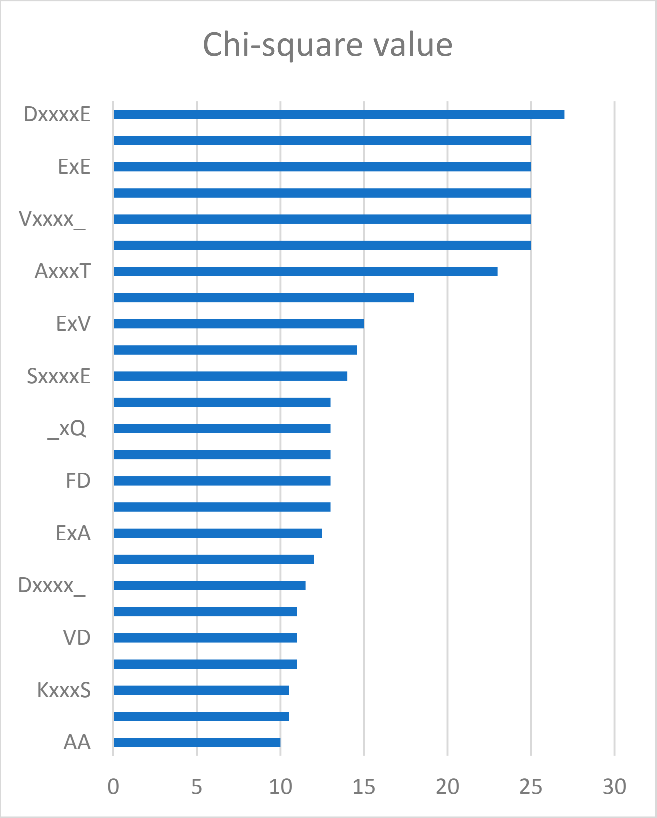

Figure 2. Top 25 *k*-spaced amino acid pairs.

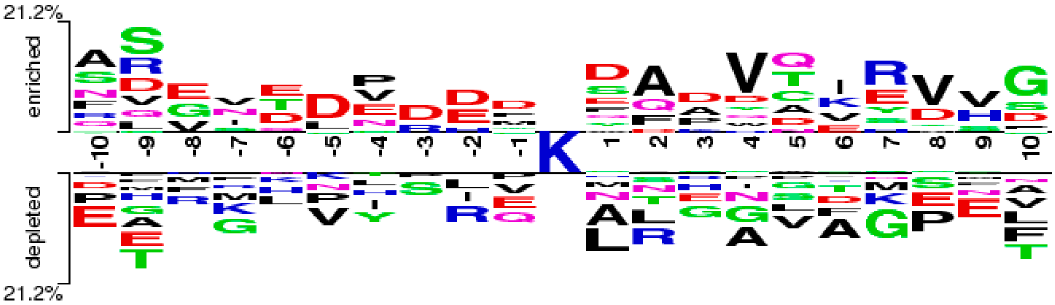

Figure 3. The two-sample-logos of the composition of *k*-spaced amino acid pairs surrounding the pupylation site and non-pupylation site.

2.5. Case Study

To further verify the generalization of our model, we adopted EPuL for a total of 1116 pupylated proteins, which are identified by high-throughput proteomics methods [19] and have unknown pupylation sites. Among the total proteins, EPuL successfully identified 2102, 3265, and 3899 pupylation sites at the threshold of 'high', 'medium' and 'low', respectively. The result of the predicted pupylation sites is available in Supplementary File 1.

3. Materials and Methods

3.1. Datasets

In this paper, the training dataset and the independent testing dataset of iPup [9] were used. The training dataset included 162 proteins, which consisted of 183 experimentally-validated pupylation sites and 2258 artificial generated non-annotated pupylation sites. The former were regarded as positive samples, and the latter were regarded as unlabeled samples. The independent testing dataset included 20 proteins, including 29 experimentally-verified pupylation sites and 408 non-annotated pupylation sites. Though the independent testing dataset was highly imbalanced, it can reflect the real effectiveness of different methods. Similar to the current pupylation site prediction methods [8–14], the sliding window method was adopted to encode each sample in the dataset. The window size was set to 21 here, in accordance with [9].

3.2. Construction of Feature Vectors

In this study, the composition of the *k*-spaced amino acid pairs (CKSAAP)-based encoding scheme was applied to encode each sample. CKSAAP could show the association of the residues surrounding pupylation sites and it has been successfully applied to other kinds of PTM site prediction problems [20–22]. Taking $k = 0$ as an example, for a sequence fragment including 2n + 1 amino acids, there are 441 0-spaced residue pairs (i.e., AA, AC, . . .). Then a 441-dimensional feature vector can be defined as:

$$\left(\frac{N_{AA}}{N_{total}}, \frac{N_{AC}}{N_{total}}, \ldots\ldots\ldots, \frac{N_{\_\_}}{N_{total}}\right)_{441}$$

The value of each component is the probability of each amino acid pair. When there are *n* AA pairs in the sequence fragment, the value of N_{total} is 441 for any window size, and the value of $\frac{n}{N_{total}}$ is the probability of the corresponding AA pair.

With the increase of *k*, the accuracy and the sensitivity increase, while the computational complexity and the required time also increase. In this paper, the value of parameter *k* in CKSAAP was set to 0, 1, 2, 3, and 4. Thus, each sample is represented by 2205 dimension features. For example, for the pair A and D, the *k*-spaced amino acid pairs for $k = 0, 1, 2, 3$, and 4 are represented as AD, AxD, AxxD, AxxxD, AxxxxD.

3.3. Feature Selection

In order to remove the irrelevant and redundant features, we utilized the chi-square test and sequential backward feature elimination algorithm, which was the same as iPUP [9]. Each feature would have a value by chi-square test and sequential backward feature elimination algorithm was used to select optimal feature subset. Firstly, we ranked the 2205 dimension features according to the value of chi-square. Then, we iteratively removed 10 features with the lowest value in a sequential backward feature elimination algorithm. Finally, the feature subset with the highest performance was used as the optimal feature subset to train the model. Figure 2 shows the top 25 CKSAAP features ranked by using the chi-square test and we used the top 150 features as the optimal feature subset, which was the same as iPUP [9]. The complete list of the optimal feature subset is shown in Supplementary File 2.

3.4. Support Vector Machine

A support vector machine with the kernel radial basis function (RBF) was the core learning algorithm of EPuL. The LibSVM [23] package widely used in the area of bioinformatics [24–26] was used to train the final prediction model. A grid search strategy based on 10-fold cross-validation was utilized to find the optimal parameters.

3.5. Performance Evaluation of EPuL

Five measurements were employed to evaluate the performance of our proposed predictor [21]. These measurements included sensitivity (SN), specificity (SP), accuracy (ACC), and Matthews' correlation coefficient (MCC). These measurements are defined as the following formulas:

$$SN = \frac{TP}{TP + FN}$$

$$SP = \frac{TN}{TN + FP}$$

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

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

where *TP*, *FP*, *TN*, and *FN* denote the number of true positives, false positives, true negatives, and false negatives, respectively. Matthews' correlation coefficient (*MCC*) provides an overall performance of binary classification.

4. Conclusions

In this paper, we proposed a new predictor, EPuL, to identify the protein pupylation sites. We aim to make the initial selected negative set reliable, and then a more and more reliable negative set will be selected in later iterations. As this process continues, the final negative set will be as reliable as possible. The proposed enhanced positive-unlabeled learning algorithm outperforms the existing predictors. Moreover, the most likely pupylation and non-pupylation sites can be predicted in non-annotated lysine sites by using EPuL. We are confident that the proposed method could also be applied in the identification of other types of PTMs sites. A user-friendly web server is freely available at reference [15]. In our future research, except for the predictor EPuL proposed in this paper, we will use some state-of-the-art metaheuristic algorithms to identify the protein pupylation sites, such as monarch butterfly optimization (MBO) [27], earthworm optimization algorithm (EWA) [28], elephant herding optimization (EHO) [29], moth search (MS) algorithm [30], and krill herd (KH) [31–35].

Supplementary Materials: The following are available online. Supplementary File 1: The result of the predicted pupylation sites, Supplementary File 2: The complete list of the optimal feature subset.

Acknowledgments: This research is partially supported by the China Postdoctoral Science Founda funded project (2014M550166, 2015T80285), National Natural Science Foundation of China (61403077, 61503165), Natural Science Foundation of Jiangsu Province (No. BK20150239), and the Natural Science Foundation of the Education Department of JiLin Province (2016-505).

Author Contributions: X. Nan and L. Bao conceived and designed the experiments; X. Zhao performed the experiments; X. Zhao and Z. Ma analyzed the data; G.-G. Wang contributed analysis tools; A.K. Sangaiah wrote the paper.

References

1. Pearce, M.J.; Mintseris, J.; Ferreyra, J.; Gygi, S.P.; Darwin, K.H. Ubiquitin-like protein involved in the proteasome pathway of Mycobacterium tuberculosis. *Science* **2008**, *322*, 1104–1107. [CrossRef] [PubMed]
2. Burns, K.E.; Liu, W.T.; Boshoff, H.I.; Dorrestein, P.C.; Barry, C.E. Proteasomal protein degradation in mycobacteria is dependent upon a prokaryotic ubiquitin-like protein. *J. Biol. Chem.* **2009**, *284*, 3069–3075. [CrossRef] [PubMed]
3. Chen, X.; Solomon, W.C.; Kang, Y.; Cerda-Maira, F.; Darwin, K.H.; Walters, K.J. Prokaryotic ubiquitin-like protein pup is intrinsically disordered. *J. Mol. Biol.* **2009**, *392*, 208–217. [CrossRef] [PubMed]
4. Tung, C.W. PupDB: A database of pupylated proteins. *BMC Bioinf.* **2012**, *13*, 40–45. [CrossRef] [PubMed]
5. Striebel, F.; Imkamp, F.; Sutter, M.; Steiner, M.; Mamedov, A.; Weber-Ban, E. Bacterial ubiquitin-like modifier Pup is deamidated and conjugated to substrates by distinct but homologous enzymes. *Nat. Struct. Mol. Biol.* **2009**, *16*, 647–651. [CrossRef] [PubMed]
6. Poulsen, C.; Akhter, Y.; Jeon, A.H.-W.; Schmitt-Ulms, G.; Meyer, H.E.; Stefanski, A.; Stühler, K.; Wilmanns, M.; Song, Y.H. Proteome-wide identification of mycobacterial pupylation targets. *Mol. Syst. Biol.* **2010**, *6*, 386–392. [CrossRef] [PubMed]
7. Georgiou, D.N.; Karakasidis, T.E.; Megaritis, A.C. A short survey on genetic sequences, chou's pseudo amino acid composition and its combination with fuzzy set theory. *Open Bioinf. J.* **2013**, *7*, 41–48. [CrossRef]
8. Liu, Z.; Ma, Q.; Cao, J.; Gao, X.; Ren, J.; Xue, Y. GPS-PUP: Computational prediction of pupylation sites in prokaryotic proteins. *Mol. Biosyst.* **2011**, *7*, 2737–2740. [CrossRef] [PubMed]
9. Tung, C.W. Prediction of pupylation sites using the composition of k-spaced amino acid pairs. *J. Theor. Biol.* **2013**, *336*, 11–17. [CrossRef] [PubMed]
10. Zhao, X.; Dai, J.; Ning, Q.; Ma, Z.; Yin, M.; Sun, P. Position-specific analysis and prediction of protein pupylation sites based on multiple features. *BioMed. Res. Int.* **2013**, *12*, 109549. [CrossRef] [PubMed]
11. Chen, X.; Qiu, J.D.; Shi, S.P.; Suo, S.B.; Liang, R.P. Systematic analysis and prediction of pupylation sites in prokaryotic proteins. *PLoS ONE* **2013**, *8*, 130. [CrossRef] [PubMed]
12. Hasan, M.M.; Zhou, Y.; Lu, X.; Li, J.; Song, J.; Zhang, Z. Computational identification of protein pupylation sites by using profile-based composition of k-spaced amino acid pairs. *PLoS ONE* **2015**, *10*, e0129635. [CrossRef] [PubMed]
13. Wang, C.; Ding, C.; Meraz, R.F.; Holbrook, S.R. PSoL: A positive sample only learning algorithm for finding non-coding RNA genes. *Bioinformatics* **2006**, *22*, 2590. [CrossRef] [PubMed]
14. Jiang, M.; Cao, J.Z. Positive-Unlabeled learning for pupylation sites prediction. *Biomed. Res. Int.* **2016**, *16*, 1–5. [CrossRef] [PubMed]
15. EPuL: An Enhanced Positive-Unlabeled Learning Algorithm for the Prediction of Pupylation Sites. Available online: http://59.73.198.144:8080/EPuL (accessed on 30 August 2017).
16. Zeng, X.; Liao, Y.; Liu, Y.; Zou, Q. Prediction and validation of disease genes using HeteSim Scores. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **2016**, 1–10. [CrossRef] [PubMed]
17. Wei, L.; Liao, M.; Gao, Y.; Ji, R.; He, Z.; Zou, Q. Improved and promising identification of human microRNAs by incorporating a high-quality negative set. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **2013**, *11*, 192–201. [CrossRef] [PubMed]
18. Zou, Q.; Zeng, J.; Cao, L.; Ji, R. A novel features ranking metric with application to scalable visual and bioinformatic sdata classification. *Neurocomputing* **2016**, *173*, 346–354. [CrossRef]
19. Cerda-Maira, F.A.; McAllsiter, F.; Bode, N.J.; Burns, K.E.; Gygi, S.P.; Darwin, K.H. Reconstitution of the Mycobackterium tuberculosis pupylation pathway in Escherichia coli. *EMBO Rep.* **2011**, *12*, 863–870. [CrossRef] [PubMed]

20. Zhe, J.; Cao, J.Z.; Gu, H. ILM-2L: A two-level predictor for identifying protein lysine methylation sites and their methylation degrees by incorporating K-gap amino acid pairs into Chou's general PseAAC. *J. Theor. Biol.* **2015**, *385*, 50–57.
21. Zhe, J.; Cao, J.Z.; Gu, H. Predicting lysine phosphoglycerylation with fuzzy SVM by incorporating k-spaced amino acid pairs into Chou's general PseAAC. *J. Theor. Biol.* **2016**, *397*, 145–150.
22. Ju, Z.; Gu, H. Predicting pupylation sites in prokaryotic proteins using semi-supervised self-training support vector machine algorithm. *Anal. Biochem.* **2016**, *507*, 1–6. [CrossRef] [PubMed]
23. Chang, C.C.; Lin, C.J. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 389–396. [CrossRef]
24. Yan, R.X.; Si, J.N.; Wang, C.; Zhang, Z. DescFold: A web server for protein fold recognition. *BMC Bioinf.* **2008**, *10*, 1949. [CrossRef] [PubMed]
25. Song, J.; Tan, H.; Shen, H.; Mahmood, K.; Boyd, S.E.; Webb, G.I.; Akutsu, T.; Whisstock, J.C. Cascleave: towards more accurate prediction of caspase substrate cleavage sites. *Bioinformatics* **2010**, *26*, 752–760. [CrossRef] [PubMed]
26. Si, J.N.; Yan, R.X.; Wang, C.; Zhang, Z.; Su, X.D. TIM-Finder: A new method for identifying TIM-barrel proteins. *BMC Struct. Biol.* **2009**, *9*, 73. [CrossRef] [PubMed]
27. Wang, G.G.; Deb, S.; Cui, Z. Monarch Butterfly Optimization. *Neural Comput. Appl.* **2015**, 1–20. [CrossRef]
28. Wang, G.G.; Deb, S.; Coelho, L.D.S. Earthworm optimization algorithm: A bio-inspired metaheuristic algorithm for global optimization problems. *Int. J. Bio-Inspired Comput.* **2015**. [CrossRef]
29. Wang, G.G.; Coelho, L.D.S.; Gao, X.Z.; Coelho, L.D.S. A new metaheuristic optimization algorithm motivated by elephant herding behaviour. *Int. J. Bio-Inspired Comput.* **2016**, *8*, 394. [CrossRef]
30. Wang, G.G. Moth search algorithm: A bio-inspired metaheuristic algorithm for global optimization problems. *Memet. Comput.* **2016**, 1–14. [CrossRef]
31. Wang, G.G.; Guo, L.; Wang, H.; Duan, H.; Liu, L.; Li, J. Incorporating mutation scheme into krill herd algorithm for global numerical optimization. *Neural Comput. Appl.* **2014**, *24*, 1231. [CrossRef]
32. Wang, G.G.; Gandomi, A.H.; Alavi, A.H. Stud krill herd algorithm. *Neorucomputing* **2014**, *128*, 363–370. [CrossRef]
33. Wang, G.G.; Guo, L.; Gandomi, A.H.; Hao, G.S.; Wang, H. Chaotic Krill Herd algorithm. *Inf. Sci.* **2014**, *274*, 17–34. [CrossRef]
34. Wang, G.G.; Gandomi, A.H.; Alavi, A.H. An effective krill herd algorithm with migration operator in biogeography-based optimization. *Appl. Math. Model.* **2014**, *38*, 2454–2462. [CrossRef]
35. Wang, G.G.; Gandomi, A.H.; Alavi, A.H.; Gong, D. A comprehensive review of krill herd algorithm: variants, hybrids and applications. *Artif. Intell. Rev.* **2017**, 1–30. [CrossRef]

Sample Availability: Samples of the compounds are not available from the authors.