Mining Proteins with Non-Experimental Annotations Based on an Active Sample Selection Strategy for Predicting Protein Subcellular Localization

Junzhe Cao¹, Wenqi Liu¹*, Jianjun He², Hong Gu¹
¹ School of Control Science and Engineering, Dalian University of Technology, Dalian, China, ² College of Information and Communication Engineering, Dalian Nationalities University, Dalian, China

Abstract
Subcellular localization of a protein is important to understand proteins’ functions and interactions. There are many techniques based on computational methods to predict protein subcellular locations, but it has been shown that many prediction tasks have a training data shortage problem. This paper introduces a new method to mine proteins with non-experimental annotations, which are labeled by non-experimental evidences of protein databases to overcome the training data shortage problem. A novel active sample selection strategy is designed, taking advantage of active learning technology, to actively find useful samples from the entire data pool of candidate proteins with non-experimental annotations. This approach can adequately estimate the “value” of each sample, automatically select the most valuable samples and add them into the original training set, to help to retrain the classifiers. Numerical experiments with four popular multi-label classifiers on three benchmark datasets show that the proposed method can effectively select the valuable samples to supplement the original training set and significantly improve the performances of predicting classifiers.

Citation: Cao J, Liu W, He J, Gu H (2013) Mining Proteins with Non-Experimental Annotations Based on an Active Sample Selection Strategy for Predicting Protein Subcellular Localization. PLoS ONE 8(6): e67343. doi:10.1371/journal.pone.0067343

Editor: Marinus F.W. te Pas, Wageningen UR Livestock Research, The Netherlands

Received January 23, 2013; Accepted May 16, 2013; Published June 26, 2013

Copyright: © 2013 Cao et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Funding: This work was supported by a Specialized Research Fund for the Doctoral Program of Higher Education (Grant No. 20120041110008), National Science and Technology Mega-Project Program of China (Grant No. 2011ZX09101-008-09), National Natural Science Foundation of China (Grant No. 61174027), and the Program for Liaoning Excellent Talents in University (Grant No. LJQ2012005). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: wqliu@dlut.edu.cn

Introduction
A good understanding of protein subcellular location is a key for deducing protein functions, revealing disease pathogenesis, and identifying drug targets. In the last ten years, the rapid growth of protein data has made it faster and more economical to predict subcellular localization via computational methods. Since the first protein location prediction system emerged [1], many prediction approaches and predictors have been proposed. These methods are mostly based on classification algorithms, e.g. k-nearest neighbor (KNN) [2–5], support vector machine (SVM) [6–8], Bayesian methods [9,10], and neural network [11,12], etc. A comprehensive review [13] provides the process to establish a robust predictor of protein subcellular localization, with following aspects: (a) selecting or constructing an effective benchmark dataset to train and test the predictor; (b) formulating the protein samples with a valid mathematical expression; (c) proposing a powerful algorithm (classifier) for prediction tasks; and (d) performing proper tests to objectively evaluate the performance of the predictor. Among these aspects, one key factor of building a high-accuracy prediction method is to obtain a valid dataset with sufficient useful information to train a powerful classifier.

Normally, the training data of a subcellular localization predictor are acquired from the “proteins with experimental annotations (referred as PEAs hereafter)” in protein databases, which are labeled by sufficient experimental evidences. However, as we know, experimental methods require a long time to obtain conclusive evidence to assign an annotation. Therefore, these experimental protein sequences are just a small part of the overall sequences. According to the record (version 2012_05) of the central protein databank UniProtKB/Swiss-Prot, the PEAs only occupy 13.22% of the total reviewed protein contained therein. In this study, we also counted the number of the protein sequences over the past ten years in UniProtKB/Swiss-Prot and summarized the statistics in Table 1. Over the last decade, there was a tenfold increase in the amount of all protein sequences, but the growth of the experimental sequences was less than doubled. While more PEAs of all types are needed to provide useful information for increasing undetermined proteins, the gap between the amount of PEAs and the entire protein data are becoming larger and larger. In addition, for computational prediction methods, excess of homologous or similar protein data will cause the over-fitting problem and these data are redundant for training, consequently, most of these PEAs have to be abandoned in practice. Besides, some special subcellular locations are correlated with very few PEAs and it also restricts the number of data used for learning. Therefore, there are often insufficient PEAs when constructing a proper dataset for a prediction task. For instance, the virus benchmark dataset in paper [14] merely consists of 207 proteins, and there are only eight proteins located in “viral capsid”. The
problem of lacking high-quality training data nearly occurs within each species and it has been a major problem in many bioinformatics researches because the prediction with sparse data would mostly obtain disappointing results [15].

To overcome this shortage of training data, seeking extra protein training data becomes a very natural idea. Besides the PEAs, we recently find that we can take advantage of the huge number of “proteins with non-experimental annotations (referred as PNEAs hereafter)” in the central protein database UniProtKB/Swiss-Prot. Since the observations are not marked from direct experiments, non-experimental annotations are labeled based on non-experimentally proven findings such as logical or conclusive evidences, sequence analysis results, biological events and characteristics [16]. A PNEA has at least one non-experimental label in its “Subcellular location” item, and a non-experimental label corresponds to one of the following three types [17]: “Probable” - from non-direct experimental evidences; “Potential” - from computer prediction, logical or conclusive evidences; “By similarity” - from experimental evidences in a close member of the family. The details of the three non-experimental labels can be found in the UniProtKB/Swiss-Prot manual at http://www.uniprot.org/man. For protein subcellular location prediction based on computational methods, the PNEAs who are being ignored are important and valuable. Unlike unknown protein data, the PNEAs provide a lot of high reliable reference location information. Additionally, as shown in Table 1, PNEAs have a much larger number and grow much faster than PEAs. If such abundant PNEAs can be effectively exploited, they would provide a huge supplement to PEAs for training more powerful predictors. Despite the big advantage of PNEAs, not all of them can be indiscriminately used as supplementary training data. The reason is that the non-experimental evidence is still weaker than the experimental proof, so some portion of PNEAs may have inaccurate non-experimental labels. Therefore, a feasible rule is needed to select the useful members of the PNEAs with a low risk and high quality for training a classifier.

In order to develop a proper rule for the active selection process, a machine learning technique named “active learning” is adopted in our study. This active learning method is a paradigm for using unlabeled data to complement labeled data, so as to actively select and learn from the most informative unlabeled data. The idea of actively selecting new samples is suitable for our work. However, there are some issues with the active learning process that need to be resolved before it can be properly used in this study. The active learner always actively asks the user to label the unlabeled data so that it can learn a good classifier with as few manual labeled samples as possible [18]; while in our study, the candidate PNEA samples are not unlabeled but rather have special non-experimental labels, and the proposed algorithm should automatically pick out enough but not redundant samples from the whole PNEA dataset. Therefore, inspired by an active learning algorithm [19], this paper proposes such a novel active sample selection strategy for PNEAs to increase the amount of training data available. For the weak basic classifiers learned via only the original data, this strategy measures the usefulness of all candidate PNEAs, and picks out these most useful PNEAs as supplementary training data. The weak classifiers are then retrained on the new training set to obtain improved prediction performances.

The effectiveness of the proposed approach is tested on three protein benchmark datasets from virus, plant and gram-negative bacteria cells, by four popular multi-label learning classification algorithms which are based on KNN, SVM, Bayesian method and neural network. The results show that the proposed method can effectively pick out the useful PNEAs and there are obvious enhancements for the prediction performances of each basic classifier.

Materials and Methods

The Datasets

Three existing benchmark experimental datasets of different species are used for cross-validation tests, which include a virus dataset [14] consisting of 207 proteins and 6 different subcellular location classifications, a plant dataset [20] consisting of 978 proteins and 12 different subcellular location classifications, and a Gram-negative bacteria (referred as Gneg hereafter) dataset [21] consisting of 1392 proteins and 8 different subcellular location classifications. In order to obtain effective candidates for supplementary training data, we extracted numerous PNEAs of the three species by parsing the “Subcellular location” section of the “Comments” field in UniProtKB/Swiss-Prot database (release 2012_05). Protein fragments and those containing less than 50 amino acid residues were discarded. Similarly, we also collected several new PEAs which were not included in the above-mentioned benchmark datasets for an independent test. In order to reduce the redundancy and avoid homology bias, we used a public server PISCES [22] based on PSI-BLAST alignments to identify and cull protein sequences from all the sequence data extracted to ensure that none of these proteins have a ≥25% sequence similarity to one another as well as any sequence in the benchmark dataset for the same species.

After culling, we created three supplementary training sample pools as candidates for active selection, which consist of 238 virus PNEAs, 758 plant PNEAs and 248 Gneg PNEAs. We also constructed three additional independent test sets, consisting of 69 virus PEAs, 261 plant PEAs and 207 Gneg PEAs. Note that, because some proteins occur in more than one location, the concept of “locative protein” in the literature [21] is employed to compute performance indexes of the classifiers. This concept means that a protein coexisting at V(N > 1) different location sites will be counted as V locative proteins even if they have an identical sequence. The amounts of active/locative proteins in the three groups of datasets are shown in Table 2. More details about the datasets can be found in Table S1–S3 in Material S1. The new
PNEAs and new PEAs used in our research are all listed in Material S2 (Supplementary Dataset S1–S6).

Active Sample Selection Strategy

In this study, because some proteins have multiple subcellular localization sites, the final prediction task is also a multi-label learning problem. Accordingly, the active sample selection strategy should have the ability to deal with the multi-label cases. Let $D_l = \{(x_k, y_k) | 1 \leq k \leq n_l\}$ denote the original training data set consisting of $n_l$ PNEAs classified in $m$ different subcellular locations, where each protein $x_k$ can be represented by a feature vector of $d$ dimensions as $[x_{k1}, x_{k2}, \ldots, x_{kd}]^T$, and the label set $y_k = [y_{k1}, y_{k2}, \ldots, y_{km}]^T$ denotes the protein subcellular locations of $x_k$. For each protein $x_k$, if it inhabits the $t$th subcellular location, mark $y_{kt} = 1$, otherwise $y_{kt} = -1$. The basic classifier $f : \mathcal{R}^d \rightarrow \{\pm 1\}$ is trained by $D_l$ to output a set of labels for each unseen protein. Let $D_u = \{(x'_j, y'_j) | 1 \leq j \leq n_u\}$ denote the supplementary training sample pool containing $n_u$ PNEAs, where a protein $x'_j = [x'_{j1}, x'_{j2}, \ldots, x'_{jd}]^T$ has the label set $y'_j = [y'_{j1}, y'_{j2}, \ldots, y'_{jm}]^T$. For each protein $x'_j$, if it has the $t$th subcellular location labeled by experimental/non-experimental evidences, we mark that $y'_{jt} = 1$, otherwise $y'_{jt} = -1$. Note that, both $y'_{jt} = 1$, and $y'_{jt} = -1$ mean $y'_{jt} = 1$, and the subscripts are merely used for recognizing that this positive label is obtained by corresponding experimental or non-experimental annotations.

In order to actively pick out the useful samples from the supplementary training sample pool, the key is to create a feasible evaluation function to measure the usefulness of a non-experimental sample and decide which samples should be added into the original training set. In this paper, the classification risk of a sample is used for reflecting the sample’s usefulness, where a lower risk means a higher usefulness. For a sample $(x'_s, y'_s)$ in $D_u$, let $R(f, D_l, x'_s, y'_s)$ be the classification risk which is brought by adding $x'_s$ into the original training set, and the evaluation function $E(f, D_l, x'_s, y'_s)$ of $(x'_s, y'_s)$ is defined by its maximum risk. Our motivation is to evaluate the risks, and pick out the optimal $(x'_s, y'_s)^*$ by minimizing the maximum risk, that leads to the following min-max combinatorial optimization problem:

\[
(x'_s, y'_s)^* = \arg \min_{(x'_s, y'_s) \in D_u} E(f, D_l, x'_s, y'_s) = \arg \min_{(x'_s, y'_s) \in D_u} \max_{y_{se} \in \{\pm 1\}^m} R(f, D_l, x'_s, y'_s) + H
\]

where $y_{se} = [y_{s1}, y_{s2}, \ldots, y_{sm}] \in \{\pm 1\}^m$ represents the unknown actual label set for $x'_s$, where, for each label $y_{sj}$, $y_{sj} = 1$ if $y'_{sj} = 1$, $y_{sj} = -1$ if $y'_{sj} = -1$, but if $y'_{sj} = 0$ then $y_{sj}$ may be $1$ or $-1$. $1/2 \|f\|_H^2$ is the regularization item which measures the model complexity of the classifier, here $H$ is a reproducing kernel Hilbert space endowed with kernel function $K'(\cdot, \cdot) : \mathcal{R}^d \times \mathcal{R}^d \rightarrow \mathcal{R}$. $L_1(z)$ is a quadratic loss function and $L_2(z)$ is a weighted quadratic loss function, i.e.,

\[
L_1(y_k, f(x_k)) = \frac{1}{2} \sum_{k=1}^{n_l} \sum_{i=1}^{m} (y_{ki} - f_i(x_k))^2
\]

\[
L_2(y'_s, y_{se}, f(x'_s)) = \frac{1}{2} \sum_{j=1}^{m} w \left( x'_s, \tilde{y}_{sj}, y'_s \right) \left( y_{se} - f_j(x'_s) \right)^2
\]

where, $w \left( x'_s, \tilde{y}_{sj}, y'_s \right)$ is the weight function. For a PNEA, its associated label set is uncertain because its non-experimental label may not be the active label, and it is hard to directly calculate its loss. Therefore, the weight function $w \left( x'_s, \tilde{y}_{sj}, y'_s \right)$ is added to reflect the probability that a non-experimental label is the active label, which can be written as:
w(x\_s,y\_y) = \begin{cases} p(y\_y = y\_y'|x\_s,y\_y') & \text{if } y\_y = y\_y' \\ 1 - p(y\_y = y\_y'|x\_s,y\_y') & \text{if } y\_y \neq y\_y' \end{cases}

(5)

here p(y\_y = y\_y'|x\_s,y\_y') (1 \leq j \leq m) is the posterior probability of the event that y\_y just equals y\_y' when x\_s has a non-experimental label y\_y'. According to the previous description of y\_y', it can be deduced that p(\tilde{y} \_y = y\_y'|x\_s,y\_y') = 1 when y\_y' = 1\_c or y\_y' = -1. Therefore, it only needs to estimate the posterior probability for a non-experimental label y\_y' = 1\_ne. We use the Parzen-window estimation with the Gaussian kernel [23] to estimate the posterior probability of (x\_s,y\_y' = 1\_ne) as:

\[
p(y\_y|x\_s,y\_y') = \frac{p(x\_s|y\_y,y\_y')p(y\_y|y\_y')}{p(x\_s|y\_y',y\_y')p(y\_y|y\_y') + p(x\_s|y\_y',y\_y')p(y\_y|y\_y')}
\]

(6)

where, the prior probability p(y\_y|y\_y') is the confidence of the event that if y\_y = 1\_ne then y\_y = y\_y', and it is set as the parameter related to the type of the complementary non-experimental label y\_y'. p(y\_y|x\_s,y\_y') = 1 - p(\tilde{y} \_y = y\_y'|x\_s,y\_y') is the complementary set of p(\tilde{y} \_y = y\_y'|x\_s,y\_y') and p(y\_y|x\_s,y\_y') are short for p(y\_y = y\_y'|x\_s,y\_y') and p(y\_y = y\_y'|x\_s,y\_y') respectively, which are defined as:

\[
p(x\_s|y\_y,y\_y') = \frac{1}{N(y\_y)} \sum_{y\_y \in D(y\_y)} \frac{1}{\sqrt{2\pi}} \exp \left\{ -\frac{1}{2} (x\_s - x\_p)^T (x\_s - x\_p) \right\}
\]

(7)

\[
p(x\_s|y\_y',y\_y') = \frac{1}{N(y\_y')} \sum_{y\_y \in D(y\_y')} \frac{1}{\sqrt{2\pi}} \exp \left\{ -\frac{1}{2} (x\_s - x\_p)^T (x\_s - x\_p) \right\}
\]

(8)

where, D\_1 = \{x\_p|x\_p \in D\_a,y\_p = 1\} consisting of N\_1 samples is the set of all samples with certain labels; D\_2 = \{x\_q|x\_q \in D\_a,y\_q = 1\_c\} consisting of N\_2 samples defined as the set of all samples with non-experimental labels because a PNEA sample will get the maximum loss when all the actual label are opposite to the corresponding non-experimental positive label, i.e. y\_y = -1 and y\_y' = 1\_ne.

Plugging equations (3)-(5) into equation (2), this active sample selection can be written as following the min-max optimization problem:

\[
\arg \min_{x\_s,y\_y} \max_{y\_y \in [1m]} \frac{1}{2} \left( \left| f\_y\_y + \sum_{k = 1}^{m} \sum_{i = 1}^{m} (y\_k - f\_i(x\_k))^2 \right| + \sum_{j = 1}^{m} w(x\_s,\tilde{y} \_y, y\_y') \left( \tilde{y} \_y - f\_i(x\_s) \right)^2 \right)
\]

(9)

From the derivation, we have:

\[
\min_{\tilde{y} \_y \in [1m]} \frac{1}{2} \left( \left| f\_y\_y + \sum_{k = 1}^{m} \sum_{i = 1}^{m} (y\_k - f\_i(x\_k))^2 \right| + \sum_{j = 1}^{m} w(x\_s,\tilde{y} \_y, y\_y') \left( \tilde{y} \_y - f\_i(x\_s) \right)^2 \right) = \frac{1}{2} Y^T L Y
\]

(10)

where, |f\_y\_y|^2 = \sum_{i=1}^{m} f\_i(x\_y) K_{i} = K' \otimes I, K' is the kernel matrix of size (n+1) × (n+1), I is an \(m \times m\) identity matrix, \(\otimes\) is the Kronecker product, \(Y = [y\_1, y\_2, \ldots, y\_m, \tilde{x}\_y] = [Y\_c, \tilde{x}\_y]^T\), \(Y = (\lambda^{-1} K + A^{-1})^{-1}\), and

\[
A = \begin{bmatrix}
I \\
& I \\
& & A
\end{bmatrix}
\]

(11)

\[
W = \begin{bmatrix}
w(\tilde{x}\_y|x\_s, y\_y') \\
& w(\tilde{x}\_y|x\_s, y\_y') \\
& & \ldots \\
& & & & w(\tilde{x}\_y|x\_s, y\_y')
\end{bmatrix}
\]

(12)

Thus, the evaluation function \(E(f, D; x\_s, y\_y')\) is simplified as:

\[
E(f, D; x\_s, y\_y') = \max_{y\_y \in [1m]} Y^T L Y
\]

(13)

Let \(L = \begin{bmatrix} L\_l & L\_d \\ L\_d & L\_a \end{bmatrix}\), then

\[
Y^T L Y = Y\_c^T L\_l Y\_c + 2 Y\_c^T L\_d Y\_a + Y\_a^T L\_a Y\_a
\]

(14)

Except for \(\tilde{x}\_y\), all other parts in Eq.(14) can be determined and the min-max optimization problem described as Eq.(8) can be solved through using all feasible values of \(\tilde{x}\_y\) to find the optimal \((x\_s, y\_y')^\ast\) with the smallest \(E(f, D; x\_s, y\_y')\). Similarly, we can pick out other PNEA samples one by one.

Since the usefulness of all the PNEAs are being measured, the algorithm needs to decide how many samples in \(D\_a\) should be added to \(D\_l\) to help to retrain the classifier. We observe that there is a high correlation between the usefulness of PNEA samples in \(D\_l\) and the change rates of the evaluation values. If the change becomes stable, it means the latest added supplementary training samples have little or no effect. Based on this point, this paper presents a simple algorithm, which can output a proper proportion.
of all samples in the supplementary training sample pool. First, rank all the samples within $D_a$ in ascending order according to their evaluation to compose a new ordered set $D_t$. Next, denote the evaluation value of a sample $(x_i, y_i)$ in $D_t$ by $E_i(1 \leq i \leq n_u)$. Then the change rate of its evaluation value $R_i$ can be written as:

$$R_i = \left\{ \begin{array}{cl} \frac{E_{i+1} - E_i}{E_i} , & 1 \leq i < n_u - 1 \\ 0 , & i = n_u \end{array} \right. \quad (15)$$

For a given step of proportion $x$ and the corresponding number of intervals $T = 1/x$, the algorithm needs to decide which proportion is preferred for helping to retrain the basic classifier (e.g. $x = 10\%$, then $T = 10$, where the preferred proportion is one of following percentages: $10\%$, $20\%$, $30\%$, ..., and $100\%$). Let $Num_i(1 \leq i \leq T)$ be the number of the samples in the $i$-th interval, and the preferred proportion $\theta$ can be calculated as:

$$\theta = x \cdot \arg \min_{1 \leq i \leq T} \frac{1}{Num_i} \sum_{j = G(i-1) + 1}^{G(i)} R_i \quad (16)$$

Note that, it is hard to theoretically prove whether the output proportion $\theta$ is the global optimum or not, but it can be seen that $\theta$ can indeed provide excellent results in subsequent simulation experiments.

After selecting the top $\theta$ of the samples in $D_t$ and adding them into the original training set, the initial classifier is updated according to the new training set and its performance is improved. An illustration of the work process of the proposed active example selection strategy is shown in Fig. 1.

### Evaluation Metrics

In order to comprehensively evaluate the active sample selection method and compare the classifier performances with/without the proposed approach, some common evaluation metrics are used. Here, $D = \{(x_i, y_i) || \leq i \leq n \}$ denotes a test set, $g(x_i)$ returns a set of proper labels of $x_i$; $h(x_i, y)$ returns a probability indicating the confidence for $y$ to be a proper label of $x_i$; $\text{rank}_{h}(x_i, y)$ is the rank of $y$ derived from $h(x_i, y)$. Let $y_{ij}$ and $g(x_i)$ represent the

$$G(t) = \left\{ \begin{array}{cl} \sum_{j = 1}^{t} \frac{Num_{ij}}{t} > 0 \quad \text{, } t > 0 \\ 0 \quad \text{, } t = 0 \end{array} \right. \quad (17)$$
complementary sets of $y_i$ and $g(x_i)$, respectively. Therefore, we have:

- True Positives: $TP = \sum_{i=1}^{n} |y_i \cap g(x_i)|$
- False Positives: $FP = \sum_{i=1}^{n} |y_i \setminus g(x_i)|$
- True Negatives: $TN = \sum_{i=1}^{n} |y_i \cap \bar{g}(x_i)|$
- False Negatives: $FN = \sum_{i=1}^{n} |y_i \setminus \bar{g}(x_i)|$

Based on the above, three global indices: accuracy (Accu), Matthews correlation coefficient (MCC) and F1-score, and three multi-label evaluation metrics: average precision (Avgprec), ranking loss (Rloss) and coverage are computed as follows:

\[
Accu = \frac{TP + TN}{TP + FP + TN + FN}
\]  
\[
MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}}
\]  
\[
F1-score = \frac{2 \cdot Pre \cdot Rec}{Pre + Rec}, \quad \text{where } Pre = \frac{TP}{TP + FP} \quad \text{and }
Rec = \frac{TP}{TP + FN}
\]  
\[
Avgprec = \frac{1}{n} \sum_{i=1}^{n} \left| \left\{ y' | rank_{h}(x_i, y') \leq rank_{h}(x_i, y) \land y' \in y_i \right\} \right|
\]  
\[
Rloss = \frac{1}{n} \sum_{i=1}^{n} \left| \left\{ (y, y') | h(x_i, y) \leq h(x_i, y') \land (y, y') \in y_i \times y_i \right\} \right|
\]  
\[
Coverage = \frac{1}{n} \sum_{i=1}^{n} \left( \max_{y \in y_i} rank_{h}(x_i, y) - 1 \right)
\]
Results and Discussion

We performed several simulation experiments to evaluate the performance of the proposed approach through both the sub-sampling (10-fold cross validation) and independent dataset test methods using the three groups of datasets mentioned in section “Material and Methods”. In the sub-sampling tests, we performed multiple rounds of randomizations of the original training and testing data on each benchmark dataset. In the independent dataset tests, the benchmark datasets were directly used as the original training sets, and the new independent test sets were adopted for testing. The amphiphilic pseudo amino acid composition [24] was employed as the feature extraction technology to represent a protein sequence. The protein sequences were formulated with a valid mathematical expression by this method through a public online server named PseAAC at: http://www.csbio.sjtu.edu.cn/bioinf/PseAAC/. The details of PseAAC can be found in reference [25]. In this study, amino acid characters were empirically chosen to be Hydrophobicity, Hydrophilicity and Mass; the weight factor was 0.4, and the lambda parameter was 5. Four different types of multi-label classification models including IMKNN [5], SVM [26], Gaussian process [10] and ML-RBF [27], were used as basic classifiers to test our algorithm. The parameters of these classifiers were assigned the same values as the original papers and all these parameters were fixed in the whole experiments for an objective comparison.

The overall performances of the above classification algorithms following three kinds of conditions were compared. These conditions were: not using the proposed active sample selection (using no supplementary training samples), using the proposed active sample selection with a preferred proportion (top θ) of the supplementary training samples, and directly using the whole PNEA samples in the supplementary training sample pool. In the experiments, the kernel function of K was the same Gaussian kernel used for estimating the posterior probability in Eq.(6). The prior probabilities of the three levels of non-experimental labels were set according to the strength of the evidences of the three non-experimental label types: the prior probability with “Probable” label was set to be the largest, the value of “Potential” was non-experimental labels: the prior probability with “Probable” was 0.85 for “Probable”, 0.8 for “Potential” and 0.75 for “By Similarity”. The step of proportion θ was set to 10% and the number of intervals T was 10.

Through the numerical experiments, we observe the preferred proportions of active sample selection for various datasets are different. The preferred proportion of virus PNEA samples is θ = 40%, θ = 70% for plant, and θ = 50% for Gram-negative bacteria. The comparisons of the performances of these classification models by using none, preferred proportion and all of the

Table 5. Results for different basic classifiers (mean±SD) by using varied numbers of supplementary training data, trained and tested in 10-fold cross-validation on the Gram-negative bacteria dataset.

| Classifier | Evaluation metrics | Number of supplementary training data | None | Top 70% | All |
|------------|--------------------|---------------------------------------|------|--------|-----|
| IMKNN      | Accu †             |                                       | 0.8699±0.0073 | 0.8819±0.0063 | 0.8688±0.0073 |
|            | MCC †              |                                       | 0.5437±0.0233 | 0.5527±0.0290 | 0.5452±0.0225 |
|            | F1-score †         |                                       | 0.6092±0.0193 | 0.6170±0.0184 | 0.6079±0.0227 |
|            | Avgprec †          |                                       | 0.6894±0.0152 | 0.7269±0.0183 | 0.7152±0.0198 |
|            | Rloss ‡            |                                       | 0.2910±0.0243 | 0.2792±0.0299 | 0.3055±0.0254 |
|            | Coverage †         |                                       | 1.4717±0.1886 | 1.4345±0.1331 | 1.4828±0.1713 |
| SVM        | Accu †             |                                       | 0.9026±0.0067 | 0.9062±0.0013 | 0.9056±0.0034 |
|            | MCC †              |                                       | 0.5698±0.0281 | 0.5847±0.0147 | 0.5828±0.0119 |
|            | F1-score †         |                                       | 0.6258±0.0242 | 0.6390±0.0128 | 0.6366±0.0156 |
|            | Avgprec †          |                                       | 0.7193±0.0157 | 0.7210±0.0143 | 0.7199±0.0117 |
|            | Rloss ‡            |                                       | 0.3700±0.0191 | 0.3575±0.0132 | 0.3631±0.0164 |
|            | Coverage †         |                                       | 1.7672±0.0698 | 1.7003±0.0665 | 1.7115±0.0672 |
| Gaussian process | Accu †         |                                       | 0.9332±0.0035 | 0.9384±0.0042 | 0.9300±0.0057 |
|             | MCC †              |                                       | 0.6678±0.0106 | 0.6878±0.0142 | 0.6666±0.0171 |
|             | F1-score †         |                                       | 0.6990±0.0263 | 0.7096±0.0203 | 0.6984±0.0156 |
|             | Avgprec †          |                                       | 0.7307±0.0256 | 0.7417±0.0220 | 0.7264±0.0189 |
|             | Rloss ‡            |                                       | 0.3722±0.0248 | 0.3682±0.0238 | 0.3728±0.0207 |
|             | Coverage †         |                                       | 1.7689±0.0581 | 1.7191±0.0635 | 1.7989±0.0732 |
| ML-RBF     | Accu †             |                                       | 0.9159±0.0035 | 0.9319±0.0071 | 0.9127±0.0015 |
|            | MCC †              |                                       | 0.6144±0.0187 | 0.6328±0.0183 | 0.6020±0.0117 |
|            | F1-score †         |                                       | 0.6672±0.0168 | 0.6945±0.0128 | 0.6507±0.0113 |
|            | Avgprec †          |                                       | 0.8057±0.0145 | 0.8295±0.0153 | 0.7984±0.0086 |
|            | Rloss ‡            |                                       | 0.1147±0.0120 | 0.1044±0.0178 | 0.1110±0.0067 |
|            | Coverage †         |                                       | 0.8786±0.0503 | 0.8499±0.0580 | 0.8687±0.0538 |

doi:10.1371/journal.pone.0067343.t005
samples in the supplementary training sample pool are shown in Table 3, Table 4, Table 5, Table 6. Table 3, Table 4 and Table 5 show the average values of 10 randomizations, 10-fold cross-validation measures and their standard deviations, and Table 6 shows the results of the independent dataset test. For each evaluation metric, "↑" means the bigger the metric value the better the performance, and "↓" means the smaller the metric value the better the performance. It can be seen, for each case, the classifier using the supplementary training data selected by the proposed approach always performs better than the basic classifier using no supplementary training sample. Additionally, the results under the proposed approach are superior to that of indiscriminately using the whole data in the supplementary training data pool. From the simulation results, it can be concluded that, on one hand, the improvements to the original prediction indicators that the selected PNEA samples are useful and indeed provide helpful information for prediction; on the other hand, the better performance of the active sample selection over directly using all the samples in the supplementary training sample pool indicates that a part of the PNEA samples disrupts the prediction because they may have some inaccurate information. Therefore, an effective active sample selection is important to select a proper amount of valuable PNEA samples and reduce the possibility of prediction disturbance brought by the redundant supplementary training data. We also observed that the performance improvements of all the classification models are related to the size of the original training set. For the virus cases with the least original training data, each classifier’s performance improvement is superior to that of the other two datasets. On the contrary, for the Gneg cases with the most original training data, the improvement effect is the smallest. We attribute this fact to the original training set with less data having a greater data shortage, so the basic classifiers are better improved by incrementally adding useful supplementary training data. Without dependence on the original classification model, the experiment results show that the proposed active sample selection strategy provides a generic approach for the existing prediction algorithms.

It is worth noting that, the inherent problem with PNEAs is that they can only be experimentally validated. To validate that the proposed strategy is more useful than conventional analysis based simply on PEA, it is better to test it via additional biological experiments. If we can show on PNEA data that the strategy finds true positives and rejects true negatives validated against biological observation of the characteristics of these proteins, the effectiveness of this approach will be further verified. However, in our work, it is difficult to directly conduct biological experiments to validate PNEAs. In a different way, we tried to find true positive and true negative PNEA samples which can be validated against a biological observation in the Swiss-Prot database. Unfortunately, we found few true positives (e.g. the non-experimental annotation “Golgi apparatus” of the plant protein with entry number “Q9M2T1”) has been verified experimentally) and no true negatives. Although the true positives can be successfully found using this strategy, we think the amount of samples identified is too small to provide enough support for this study. Therefore, the related results of these few protein samples are not included in this paper. Moreover, the objective of this study is not to identify true positive proteins, but to make protein subcellular localization prediction tools with better performance in accuracy with the help of non-experimental proteins. According to the results in Table 3, Table 4, Table 5, Table 6, the increase in accuracy over the conventional algorithms after training with these PNEAs indicates the proposed strategy works. Therefore, the proposed method could be thought of as potentially significant, even without

### Table 6. Comparison of the prediction results of different basic classifiers by using varied numbers of supplementary training data.

| Dataset | Evaluation metrics | Number of supplementary training data | IMKNN | SVM | Gaussian process | ML-RBF |
|---------|--------------------|----------------------------------------|-------|-----|-----------------|--------|
| Non    | Top     | All | Top | All | Top | All | Top | All | Top | All | Top | All | Top | All |
| Virus   | Accu †  | 0.7476 | 0.7696 | 0.7427 | 0.7476 | 0.7672 | 0.7451 | 0.7525 | 0.7784 | 0.7672 | 0.6397 | 0.7574 | 0.7328 |
| MCC †   | 0.2518 | 0.3353 | 0.2604 | 0.2518 | 0.3257 | 0.2589 | 0.3138 | 0.2634 | 0.2095 | 0.1255 | 0.2919 | 0.1957 |
| F1-score † | 0.4114 | 0.4835 | 0.4262 | 0.4114 | 0.4751 | 0.4222 | 0.4206 | 0.3662 | 0.3166 | 0.3581 | 0.4469 | 0.3626 |
| Avgprec † | 0.5572 | 0.5982 | 0.5438 | 0.5572 | 0.5902 | 0.5431 | 0.4480 | 0.5217 | 0.5008 | 0.4994 | 0.6061 | 0.5757 |
| Rloss ↓ | 0.6380 | 0.5654 | 0.6390 | 0.6380 | 0.5713 | 0.6419 | 0.8415 | 0.7245 | 0.7485 | 0.4931 | 0.3326 | 0.3622 |
| Coverage ↓ | 2.2059 | 2.0441 | 2.2941 | 2.2059 | 2.0735 | 2.3088 | 2.5735 | 2.5294 | 2.5441 | 3.0000 | 2.0147 | 2.2941 |
| Plant   | Accu †  | 0.9061 | 0.9103 | 0.9074 | 0.9042 | 0.9087 | 0.9068 | 0.9081 | 0.9112 | 0.9090 | 0.7350 | 0.7982 | 0.7816 |
| MCC †   | 0.3906 | 0.4107 | 0.3991 | 0.4220 | 0.4410 | 0.4325 | 0.2344 | 0.2924 | 0.2634 | 0.1428 | 0.1980 | 0.1531 |
| F1-score † | 0.4346 | 0.4501 | 0.4423 | 0.4737 | 0.4893 | 0.4823 | 0.1864 | 0.2527 | 0.2276 | 0.2441 | 0.2818 | 0.2516 |
| Avgprec † | 0.5099 | 0.5142 | 0.5121 | 0.5762 | 0.5819 | 0.5784 | 0.2971 | 0.3359 | 0.3222 | 0.4367 | 0.4622 | 0.4266 |
| Rloss ↓ | 0.6040 | 0.5975 | 0.6001 | 0.5212 | 0.5136 | 0.5200 | 0.8819 | 0.8309 | 0.8478 | 0.3670 | 0.3372 | 0.3722 |
| Coverage ↓ | 4.4636 | 4.4291 | 4.4674 | 3.8867 | 3.8314 | 3.8812 | 6.2222 | 6.0153 | 6.0498 | 4.4598 | 4.0728 | 4.5402 |
| Gneg    | Accu †  | 0.8374 | 0.8441 | 0.8386 | 0.8823 | 0.8829 | 0.8811 | 0.8950 | 0.8981 | 0.8932 | 0.8811 | 0.8932 | 0.8732 |
| MCC †   | 0.4542 | 0.4683 | 0.4607 | 0.4913 | 0.4970 | 0.4918 | 0.4924 | 0.5100 | 0.4777 | 0.4399 | 0.4839 | 0.4008 |
| F1-score † | 0.5315 | 0.5429 | 0.5366 | 0.5591 | 0.5650 | 0.5605 | 0.5362 | 0.5532 | 0.5191 | 0.5000 | 0.5294 | 0.4655 |
| Avgprec † | 0.6291 | 0.6340 | 0.6218 | 0.6331 | 0.6402 | 0.6361 | 0.5722 | 0.5847 | 0.5572 | 0.6935 | 0.7267 | 0.7011 |
| Rloss ↓ | 0.3891 | 0.3743 | 0.3812 | 0.4481 | 0.4342 | 0.4391 | 0.5503 | 0.5343 | 0.5707 | 0.1715 | 0.1555 | 0.1700 |
| Coverage ↓ | 2.3010 | 2.1893 | 2.2476 | 2.6165 | 2.5388 | 2.5680 | 3.0291 | 2.9417 | 3.1214 | 1.3204 | 1.2136 | 1.3155 |

doi:10.1371/journal.pone.0067343.t006
the experimental biological validation. However, it is still worth to perform a biological validation for our algorithm, and we hope to cooperate with biochemists to improve this method in the future.

In summary, in order to overcome the shortage of experimental training data in the prediction of protein subcellular location, we mined the proteins with non-experimental annotations and designed a novel active sample selection strategy to find useful PNEA samples. As supplementary training data, these selected samples helped retrain and improve the original basic classifiers. This approach based on the min-max view provides a systematic way for measuring the usefulness of a sample with multiple labels. From the results, it can be clearly seen that the proposed algorithm is significant and valid to increase the predicting performance of all four types of classifiers. We believe that active sample selection techniques in machine learning can be used as a powerful and useful tool to alleviate the data shortage problem and it could be extended to other real-world data mining applications. We also expect that the information of a huge number of proteins with non-experimental annotations can be applied to other biological problems. Furthermore, in order to make the presented method available to compare with the predictors by other interested users, we will make efforts to provide an online prediction web-server with practical value in our future work.

**Supporting Information**

**Material S1** This includes Tables S1–S3. (PDF)

**Material S2** This includes Datasets S1-S6. (PDF)

**Acknowledgments**

The authors wish to thank the editor and anonymous reviewers for their helpful comments and suggestions.

**Author Contributions**

Conceived and designed the experiments: JC WL. Performed the experiments: JC. Analyzed the data: JC WL. Contributed reagents/materials/analysis tools: HG. Wrote the paper: JC. Designed the program used in experiments: JC JH.

**References**

1. Nakai K, Kanehisa M (1991) Expert system for predicting protein localization sites in gram-negative bacteria. Proteins: Struct Funct Genet 11: 93–110.
2. Chou KC, Shen HB (2008) Cell-PLoc: a package of web servers for predicting subcellular localization of proteins in various organisms. Nat Protocols 2: 153–166.
3. Chou KC, Shen HB (2010) Cell-PLoc 2.0: an improved package of web-servers for predicting subcellular localization of proteins in various organisms. Natural Science 2: 1090–1103.
4. Horton P, Park KJ, Ohyayashi T, Fujita N, Hazuda H, et al. (2007) WoLF PSORT: protein localization predictor. Nucleic Acids Res 35: W583–W587.
5. Cao JZ, Liu WQ, Gu H (2012) Predicting viral protein subcellular localization with Chou's pseudo amino acid composition and imbalance-weighted multi-label k-nearest neighbor algorithm. Protein Pept Lett 19: 1163–1169.
6. Gray A, Bhasin M, Raghava GPS (2005) Support vector machine-based method for subcellular localization of human proteins using amino acid compositions, their order, and similarity search. J Biol Chem 280: 14427–14432.
7. Shatkay H, Hodgus A, Brady S, Blum T, Donnes P, et al. (2007) SherLoc: high-accuracy prediction of protein subcellular localization by integrating text and protein sequence data. Bioinformatics 23: 1410–1417.
8. Briesemester S, Blum T, Brady S, Lam Y, Koldbacher O, et al. (2009) SherLoc2: a high-accuracy hybrid method for predicting subcellular localization of proteins. J Proteome Res 8: 5393–5396.
9. Bulalchiksa A, Eils R (2006) Predicting protein subcellular locations using hierarchical ensemble of Bayesian classifiers based on Markov chains. BMC Bioinformatic 7: 1–11.
10. He JJ, Gu H, Liu WQ (2012) Imbalanced multi-modal multi-label learning for subcellular localization prediction of human proteins with both single and multiple sites. PLoS ONE 7: e37155.
11. Emanuchson O, Nielsen H, Brunak S, von Heijne G (2000) Predicting subcellular localization of proteins based on their N-terminal amino acid sequence. J Mol Biol 300: 1005–1016.
12. Ma JW, Liu WQ, Gu H (2010) Using elman networks ensemble for protein subcellular location prediction. Int J Innov Comput I 6: 5093–5103.
13. Chou KC (2011) Some remarks on protein attribute prediction and pseudo amino acid composition. J Theor Biol 273: 236–247.
14. Shen HB, Chou KC (2010) Virus-mPLoc: a fusion classifier for viral protein subcellular location prediction by incorporating multiple sites. J Biomol Struct Dyn 28: 175–186.
15. Xu Q, Pan SJ, Xue HH, Yang Q (2011) Multitask learning for protein subcellular location prediction. IEEE/ACM Trans Comput Biol Bioinform 8: 740–759.
16. Junker VL, Apweiler R, Bairoch A (1999) Representation of functional information in the SWISS-PROT Data Bank. Bioinformatics 15: 1066–1073.
17. Boutet E, Lieberherr D, Tognelli M, Schneider M, Bairoch A (2007) UniProtKB/Swiss-Prot. Methods Mol Biol 496: 89–112.
18. Settles B (2009) Active Learning Literature Survey. Computer Sciences Technical Report 2009: 1648.
19. Hoi SCH, Jin R, Zhu J, Lyu MR (2007) Semi-supervised SVM batch mode active learning with applications to image retrieval. ACM T Inform Syst 27: 1–29.
20. Chou KC, Shen HB (2010) Plant-mPLoc: A top-down strategy to augment the power for predicting plant protein subcellular localization. PLoS ONE 5: e11335.
21. Shen HB, Chou KC (2010) Gneg-mPLoc: a top-down strategy to enhance the quality of predicting subcellular localization of Gram-negative bacterial proteins. J Theor Biol 264: 326–333.
22. Wang GL, Dunbrack RL (2003) PISCES: a protein sequence culling server. Bioinformatics 19: 1509–1519.
23. Li B, Chen YW, Chen YQ (2008) The nearest neighbor algorithm of local probability centers. IEEE T Syst Man Cy B 38: 141–154.
24. Chou KC (2005) Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes. Bioinformatics 21: 10–19.
25. Shen HB, Chou KC (2008) PseAAC: a flexible web server for generating various kinds of protein pseudo amino acid composition. Anal Biochem 373: 385–388.
26. Huang J, Shi F (2005) Support vector machines for predicting apoptosis proteins types. Acta Biotheor 53: 39–47.
27. Zhang ML (2009) ML-RBF: RBF neural networks for multi-label learning. Neural Process Lett 29: 61–74.