A Multi-Label Predictor for Identifying the Subcellular Locations of Singleplex and Multiplex Eukaryotic Proteins

Xiao Wang, Guo-Zheng Li*

The MOE Key Laboratory of Embedded System and Service Computing, Department of Control Science and Engineering, Tongji University, Shanghai, China

Abstract

Subcellular locations of proteins are important functional attributes. An effective and efficient subcellular localization predictor is necessary for rapidly and reliably annotating subcellular locations of proteins. Most of existing subcellular localization methods are only used to deal with single-location proteins. Actually, proteins may simultaneously exist at, or move between, two or more different subcellular locations. To better reflect characteristics of multiplex proteins, it is highly desired to develop new methods for dealing with them. In this paper, a new predictor, called Euk-ECC-mPLoc, by introducing a powerful multi-label learning approach which exploits correlations between subcellular locations and hybridizing gene ontology with dipeptide composition information, has been developed that can be used to deal with systems containing both singleplex and multiplex eukaryotic proteins. It can be utilized to identify eukaryotic proteins among the following 22 locations: (1) acrosome, (2) cell membrane, (3) cell wall, (4) centrosome, (5) chloroplast, (6) cyanelle, (7) cytoplasm, (8) cytoskeleton, (9) endoplasmic reticulum, (10) endosome, (11) extracellular, (12) Golgi apparatus, (13) hydrogenosome, (14) lysosome, (15) melanosome, (16) microsome, (17) mitochondrion, (18) nucleus, (19) peroxisome, (20) spindle pole body, (21) synapse, and (22) vacuole. Experimental results on a stringent benchmark dataset of eukaryotic proteins by jackknife cross validation test show that the average success rate and overall success rate obtained by Euk-ECC-mPLoc were 69.70% and 81.54%, respectively, indicating that our approach is quite promising. Particularly, the success rates achieved by Euk-ECC-mPLoc for small subsets were remarkably improved, indicating that it holds a high potential for simulating the development of the area. As a user-friendly web-server, Euk-ECC-mPLoc is freely accessible to the public at the website http://levis.tongji.edu.cn:8080/bioinfo/Euk-ECC-mPLoc/. We believe that Euk-ECC-mPLoc may become a useful high-throughput tool, or at least play a complementary role to the existing predictors in identifying subcellular locations of eukaryotic proteins.

Introduction

Proteins perform their appropriate functions only when they are located in the correct subcellular locations. Therefore, one of the fundamental goals in cell biology and proteomics is to identify the subcellular locations of these proteins. Although the subcellular localization of a protein may be determined by carrying out various biochemical experiments, the approach by purely doing experiments is both time consuming and high cost. In the post-genomic age, the gap between newly found protein sequences and biological functions worthy of our special notice [19,20]. In particular, recent evidences have indicated that an increasing number of proteins have multiple locations in the cell, as indicated by Millar et al. [21].

In this paper, we focus on predicting the subcellular locations of eukaryotic proteins with both singleplex and multiplex sites. So far, only three existing predictors, i.e., Euk-mPLoc [22], Euk-mPLoc 2.0 [23] and iLoc-Euk [24], were developed that can be used to predict the subcellular locations of both singleplex and multiplex eukaryotic proteins. To the best of our knowledge, iLoc-Euk is at present the best predictor with capacity to deal with multiple-location or multiplex proteins when predicting eukaryotic proteins.

Unfortunately, the aforementioned methods don’t take multiple-location or multiplex proteins into account when predicting protein subcellular localization. In general, they were established under the assumption that a protein resides at one, and only one, subcellular location. However, proteins may simultaneously reside at, or move between, two or more different subcellular locations. Proteins with multiple location sites or dynamic feature of this kind are particularly interesting, because they may have some unique biological functions worthy of our special notice [19,20].
protein subcellular localization. However, ML-KNN prediction engine used by \texttt{iLoc-Euk} is not optimal because it doesn’t take correlations among subcellular locations into account.

In this paper, to better reflect the characteristics of multiplex proteins, a new predictor, called \texttt{Euk-ECC-mPLoc}, has been developed that can be used to deal with the systems containing both singleplex and multiplex eukaryotic proteins by introducing a powerful multi-label learning algorithm which exploits correlations between subcellular locations and by hybridizing the gene ontology information with the dipeptide composition information. Our experimental results on a benchmark dataset consisting of 7,766 eukaryotic protein sequences by jackknife cross validation test show that the overall success rates thus obtained by our proposed predictor \texttt{Euk-ECC-mPLoc} outperforms that by \texttt{iLoc-Euk} predictor. Moreover, for some subcellular locations with training proteins of very small size, the success rates achieved by \texttt{Euk-ECC-mPLoc} are $35\% - 90\%$ higher than those by \texttt{iLoc-Euk}. Therefore, \texttt{Euk-ECC-mPLoc} significantly improve the predictive performance on those “difficult” subcellular locations.

According to a recent comprehensive review [25], to establish a practically useful statistical predictor for a protein system, we need to consider the following procedures: (i) construct or select a valid benchmark dataset to train and test the predictor; (ii) formulate the protein samples with an effective mathematical expression that can truly reflect their intrinsic correlation with the target concerned; (iii) introduce or develop a powerful algorithm (or engine) to operate the prediction; (iv) properly perform cross-validation tests to objectively evaluate the anticipated accuracy of the predictor; (v) establish a user-friendly web-server for the predictor that is accessible to the public. Below, let us describe in detail how to deal with these steps one-by-one.

**Materials and Methods**

**Dataset**

In this paper, the dataset $X$ from \texttt{iLoc-Euk} [24] is used as the benchmark dataset for the current study. The dataset can be obtained from the Online Supporting Information S1 of [24]. The dataset is constructed specialized for eukaryotic proteins, where none of proteins included in $X$ has greater than or equal to 25% pairwise sequence identity to any other one in a same subcellular location compared with most of the other benchmark datasets in this area. Using the dataset $X$ will make it more reliable and easier to compare our new predictor with the existing ones.

The dataset $X$ contains 7,766 different eukaryotic protein sequences, of which 6,687 belong to one subcellular location, 1,029 to two locations, 48 to three locations, and 2 to four locations. The dataset covers 22 different subcellular locations as shown in \textbf{Fig. 1}, and hence can be represented as

$$X = X_1 \cup X_2 \cup X_3 \cup X_4 \cup X_3 \cup \ldots \cup X_{22},$$  \hspace{1cm} (1)

where $X_1$ represents the subset for the subcellular location of “acrosome”, $X_2$ for “cell membrane”, $X_3$ for “cell wall”, and so forth. A breakdown of the 7,766 eukaryotic proteins in the benchmark dataset $X$ according to their 22 location sites is given in Table 1. To avoid redundancy and homology bias, none of the proteins in $X$ has greater than or equal to 25% pairwise sequence identity to any other in a same subset. For convenience, hereafter let us just use the subscripts of Eq.(1) as the codes of the 22 location sites; i.e., “1” for “acrosome”, “2” for “cell membrane”, “3” for “cell wall”, and so forth (Table 1).

Note that because some proteins may occur in two different locations, the 7,766 different proteins actually correspond to 8,897

**Feature Extraction**

To develop a powerful method for statistically predicting protein subcellular localization, one of the most important steps is to extract core and essential features of protein samples that are closely correlated with their subcellular locations. To avoid losing important information hidden in protein sequences, the pseudo amino acid composition (PseAAC) was proposed [28,29] to replace the simple amino acid composition (AAC) for representing the sample of a protein. For a brief introduction about Chou’s PseAAC, please visit the Wikipedia web-page at http://en.wikipedia.org/wiki/Pseudo_amino_acid_composition. For a summary about its recent developments and applications, see a comprehensive review [30]. Ever since the concept of PseAAC was proposed by Chou [28] in 2001, it has rapidly penetrated into almost all the fields of protein attribute prediction, such as identifying bacterial virulent proteins [31], predicting homologemic proteins [32], predicting protein secondary structure content [33], predicting supersecondary structure [34], predicting protein structural classes [35,36], predicting protein quaternary structure [37], predicting enzyme family and sub-family classes [38–40], predicting protein subcellular location [41–44], predicting subcellular localization of apoptosis proteins [45–48], predicting protein subnuclear location [49], predicting protein submitochondria locations [50–52], identifying cell wall lytic enzymes [53], identifying risk type of human papillomaviruses [54], identifying DNA-binding proteins [55], predicting G-Protein-Coupled Receptor Classes [56,57], predicting protein folding rates [58], predicting outer membrane proteins [59], predicting cyclin proteins [60], predicting GABA/A receptor proteins [61],
identifying bacterial secreted proteins [62], identifying the cofactors of oxidoreductases [63], identifying lipase types [64], identifying protease family [65], predicting Golgi protein types [66], classifying amino acids [67], among many others. Actually, according to a recent comprehensive review [25], the Chou’s PseAAC is generally formulated as
\[ P = [f_1, f_2, f_3, \ldots, f_{n}, \ldots, f_{\Omega}]^T, \]  
where the subscript \( \Omega \) is an integer, and its value as well as the components depends on how to extract the desired features from the amino acid sequence of \( P \).

In the present study, we adopt Gene Ontology and Diptope Composition feature extraction methods to generate features of protein examples, which are widely used in many existing protein subcellular localization systems [22–24,26,27,68–74]. For reader’s convenience, a brief introduction on Gene Ontology and Diptope Composition is given below.

**Gene Ontology.** GO database [75] was established according to the molecular function, biological process, and cellular component. The following questions might be raised by those who do not really understand GO (Gene Ontology): One of the three aspects of GO is ‘Cellular Compartment’ [75], which is just another name for subcellular location. If a protein already has GO annotation, why does one need to predict its subcellular location? Is it merely a procedure of converting the annotation into another format? Is it true that the high success rate obtained via the GO approach was due to a trivial utilization of the subcellular component annotations in the GO database? To really understand these questions, the readers should carefully read the paper [14], particularly the profound and penetrating analysis on the left column of page 155 of that paper [14]. For readers’ convenience, it can be briefly summarized as follows: (i) Although GO database is constructed based on protein function and cellular component, for those proteins with ‘subcellular location unknown’ annotation in Swiss-Prot database, most (more than 99%) of their corresponding GO numbers in GO database are also annotated with ‘cellular component unknown’. (ii) Even for those proteins whose subcellular locations are clearly annotated in Swiss-Prot database, their corresponding GO numbers in GO database do not always directly indicate their corresponding subcellular locations. In some cases they are actually annotated with ‘cellular component unknown’. (iii) More important, it should be emphasized that during the course of prediction, only the GO numbers of a query protein but not its GO annotations were used, just like the case of using all the other predictors in identifying the protein subcellular location that only the sequence of a query protein but not its Swiss-Prot annotation was used. (iv) Finally, as shown by the compelling statistical analysis given in Table 6 of the paper [14], the percentage (45.02%) of proteins with GO annotations to indicate their subcellular components is even less than the percentage (31.76%) of proteins with known subcellular location annotation in the Swiss-Prot database. Accordingly, the high success rate obtained by the method via the GO approach was by no means due to a trivial procedure of converting the annotation from one into another format, as often misinterpreted by some people. Furthermore, it can be seen from Table 6 of the paper [14] that there is a huge number of proteins with given accession numbers and the corresponding GO numbers, but their subcellular locations are still unknown. Actually, the essence of why using GO approach to represent protein samples can significantly improve the prediction quality is due to the fact that proteins mapped into the GO database space would be clustered in a better reflecting their subcellular locations, thus to significantly enhance the success rate of prediction for those proteins that do not have significant sequence homology to proteins with known locations, as elaborated in [18,76]. So far, there are two main approaches to extract features from GO database space. However, in order to incorporate more information, instead of only using 0 and 1 elements as done in [23], here let us use another better approach [24] as described below.

**Step 1.** Compression and reorganization of the existing GO numbers. The GO database (version 94 released on 08 April 2011) contains many GO numbers. However, these numbers do not increase successively and orderly. For easier handling, some reorganization and compression procedures are taken to remember them. The GO database obtained through such a treatment is called GO_compress database, which contains 18,844 numbers increasing successively from 1 to the last one.

**Step 2.** Using Eq.(2) with \( \Omega = 18,844 \), the protein P is represented as
\[ P_{GO} = [f^G_1, f^G_2, f^G_3, \ldots, f^G_{18844}]^T, \]  
where \( f^G_u (u = 1,2,\ldots,18,844) \) are defined via the following steps.

**Step 3.** Use BLAST [77] to search the homologous proteins of the protein P from the Swiss-Prot database (version 55.3), with the expect value \( E \leq 0.001 \) as the BLAST parameter.
Step 4. Those proteins which have \( \geq 60\% \) pairwise sequence identity with the protein \( P \) are collected into a set, \( \mathcal{X}^{P-homo} \), called the “homology set” of \( P \). All the elements in \( \mathcal{X}^{P-homo} \) are deemed as the “representative proteins” of \( P \), sharing some similar attributes such as structural conformations and biological functions [76–80]. Because they were retrieved from the Swiss-Prot database, these representative proteins must have their own accession numbers.

Step 5. Search the GO database at http://www.ebi.ac.uk/GOA/ to find the corresponding GO number(s) [81] for each of the accession numbers collected in Step 4, and then convert the GO numbers thus obtained to their GO_compress numbers as described in Step 1. (Note that the relationships between the UniProtKB/Swiss-Port protein entries and the GOA/ to find the corresponding accession numbers.

Step 6. The elements in Eq.(3) is given by

\[
f_u^G = \frac{\sum_{k=1}^{N(rep)} g(u,k)}{N(rep)} (u=1,2,\ldots,18844),
\]

where \( N(rep) \) is the number of representative proteins in \( \mathcal{X}^{P-homo} \), and

\[
g(u,k) = \begin{cases} 
1, & \text{if the } k \text{-th representative protein hits the } u \text{-th GO compress number} \\
0, & \text{otherwise}
\end{cases}
\]

where \( \Delta_i \) is the number of representative proteins in \( \mathcal{X}^{P-homo} \), and

\[
g(u,k) = \begin{cases} 
1, & \text{if the } k \text{-th representative protein hits the } u \text{-th GO compress number} \\
0, & \text{otherwise}
\end{cases}
\]

Note that the GO feature extraction method may become a naught vector or meaningless under any of the following situations: (1) the protein \( P \) does not have significant homology to any protein in the Swiss-Prot database, i.e., \( \Delta_i = 0 \) meaning the homology set \( \mathcal{X}^{P-homo} \) is an empty one; (2) its representative proteins do not contain any useful GO information for statistical prediction based on a given training dataset.

Under such a situation, let us consider using the dipptide composition method as backup to extract features for the protein \( P \), as described below.

Dipeptide Composition. Dipeptide composition (abbreviated as DC) represents the co-occurrence frequency of each two adjacent amino acid residues. It is used to describe the global information about each protein sequence in the form of 420-dimensional (420-D) feature vector. An advantage of DC over amino acid composition is that it uses some sequence-order information. Dipeptide composition generates 420 components for each protein sequence, the first 20 components are the conventional amino acid composition(AAC); the following 400 components are the fractions of 400 dipeptides, i.e. AA, AC, AD, …, YV, YW, YY; the 400 components are calculated using the following equation

\[
\text{fraction of dip}(i) = \frac{\text{total number of dip}(i)}{\text{total number of all possible dipeptides}}
\]

where \( \text{dip}(i) \) is the \( i \)-th dipeptide of the 400 dipeptides, \( i = 1,2,\ldots,400 \).

Prediction Algorithm: Ensemble of Classifier Chains

To enhance the success rate, the powerful ECC (Ensemble of Classifier Chains) classifier [82] is adopted to perform prediction. Below, let us introduce the Ensemble of Classifier Chains classifier.

Without lose of generality, let us consider a system or dataset \( \mathcal{X} \) that contains \( N \) eukaryotic proteins classified into \( M = 22 \) subcellular location sites. The dataset \( \mathcal{X} \) can be represented by the following matrix:

\[
\begin{pmatrix}
\Delta_1^1 & \Delta_1^2 & \cdots & \Delta_1^M \\
\Delta_2^1 & \Delta_2^2 & \cdots & \Delta_2^M \\
\vdots & \vdots & \ddots & \vdots \\
\Delta_N^1 & \Delta_N^2 & \cdots & \Delta_N^M \\
\end{pmatrix}
\]

where \( \Delta_i^j = 1 \) (\( i = 1,2,\ldots,N; j = 1,2,\ldots,M \)) if the \( i \)-th eukaryotic protein belongs to the \( j \)-th subcellular location site, 0 otherwise.

According to Eq.(7), we know that if \( \sum_{i=1}^{M} \Delta_i^j > 1 \), the \( i \)-th eukaryotic protein is a multiplex protein, while if \( \sum_{i=1}^{M} \Delta_i^j = 1 \), the \( i \)-th eukaryotic protein is a single-location protein. In this study, we deal with the case that there is at least one eukaryotic protein of \( \sum_{i=1}^{M} \Delta_i^j > 1 \), that is to say, the systems that contain both single-location and multiple-location eukaryotic proteins.

Before introducing Ensemble of Classifier Chains, we firstly present a simple method, called Binary relevance (BR) [83], which converts a multi-label learning problem into a number of independent binary classification ones. Taking the above system or dataset \( \mathcal{X} \) for example, \( M \) independent binary classifiers are separately constructed for the \( M \) eukaryotic subcellular location sites, i.e.,

\[
\{C_1,C_2,\cdots,C_M\},
\]

where \( C_i \) is the prediction model for the \( i \)-th subcellular location site, \( i = 1,2,\ldots,M \). We collected according to the following formula:

\[
L_1: X \rightarrow C_1 \rightarrow P \rightarrow y_1
\]

\[
L_2: X \rightarrow C_2 \rightarrow P \rightarrow y_2
\]

\[
L_M: X \rightarrow C_M \rightarrow P \rightarrow y_M
\]

Figure 2. Figure to illustrate the complete process of BR method.

doi:10.1371/journal.pone.0036317.g002
where $D_j^i$ represents the label information as shown in Eq. (7), $P_i(D_j^i=1)$ represents the protein that belongs to the $j$-th subcellular location site, $\cup$ is the union symbol in the set theory.

For the prediction of a query protein, BR outputs the union of the class labels that are predicted by the $M$ classifiers:

$$\{y_1, y_2, \ldots, y_M\},$$

where $y_j \in \{-1, +1\} (j=1,2,\ldots,M)$ is the result predicted by the $j$-th classifier, $y_j = +1$ representing the query protein belonging to the $j$-th subcellular location site, otherwise not. To provide an intuitive picture, it is shown in Fig. 2 to illustrate the complete process of BR method.

BR is conceptually simple and easy to implement, whereas may be less effective since it don’t take label correlations into account. In the experiment below, we will compare our proposed ECC method with the BR method in order for proving the effectiveness of considering label correlations.

Now we begin to introduce ECC algorithm. ECC algorithm is proposed by J. Read in [82], which aggregates multiple CC (Classifier Chain). CG is the core of the ECC algorithm, which is based on the framework of BR and consists of $M$ classifiers as in BR. However, in contrast to BR, classifiers are linked along a chain where each classifier is responsible for prediction of presence or absence of one class label. The feature space of each classifier in the chain is extended with the 0/1 class label associations of all previous classifiers. In other words, assuming that the classifier chain $\{C_{k_1}, C_{k_2}, \ldots, C_{k_M}\}$ is a random permutation of $\{1,2,\ldots,M\}$ is constructed, each classifier $C_{k_i}$ in the chain is responsible for predicting the binary association of class label $k_j$ given the feature space, augmented by all prior binary relevance associations in the chain $k_1, \ldots, k_{j-1}$. An intuitive illustration is provided in Fig. 3.

The chaining method passes label information between classifiers, allowing CC to take into account label correlations and thus overcoming the label independence problem of BR method. However, the order of the chain itself clearly has an effect on accuracy. In [82], the issue is solved by using an ensemble framework with a random chain ordering for each iteration.

In contrast to the traditional single-label ensemble learning, ECC is an ensemble of multiple multi-label methods, i.e. the CC...
method. Following the typical strategy of ensemble learning, ECC also has two steps, in which the first is to train \( q \) CC classifiers \( C_1, C_2, \ldots, C_q \) and the second is to combine their predictions. In the first step, each \( C_k \) is trained with both a random chain ordering and a random subset of original training data set. In the second step, multi-label predictions of each \( C_k \) classifier are summed by label so that each label gets some votes, and then, we use a threshold to select the most possible labels which form the final multi-label prediction. Specifically, each \( C_k \) classifier predicts a vector \( \hat{y}_k = (\hat{y}_{k1}, \ldots, \hat{y}_{km}) \in \{0,1\}^m \). The sums are stored in a vector \( W = (l_1, \ldots, l_m) \in \mathbb{R}^m \) such that \( l_j = \sum_{k=1}^{q} \hat{y}_{kj} \). Hence each \( l_j \in W \) represents the sum of the votes for the \( j \)th label. We then normalize \( W \) to \( W\text{norm} \), which represents a distribution of scores for each label in \([0, 1] \). A threshold is used to choose the final multi-label set \( Y \) such that class label \( \neq Y \) if \( l_j \geq \theta \) for threshold \( \theta \). Here we simply set the threshold to be 0.5. Hence the relevant labels in \( Y \) represent the final multi-label prediction.

Support vector machine (SVM) [84] is a powerful binary classifier in the field of machine learning and pattern recognition. The basic ideas behind SVM is to map the input vectors into a high dimensional feature space and then find an Optimal Separating Hyperplane (OSH) which maximizes the margin, i.e., the distances between the hyperplane and the nearest data points of each class in the mapped feature space. SVM classifier has been largely and successfully used in the field of prediction of protein subcellular localization [3–5,8–11]. In this study, we also use Support vector machine (SVM) as base classifier in both BR and ECC. The software package used to train SVM with default parameters is the very efficient LIBLINEAR library [85] which is specially designed for large scale and high dimensional datasets as the benchmark eukaryotic protein dataset for the current study.

The entire predictor thus established is called \( \text{Euk-ECC-mPLoc} \), which can predict the subcellular localization of both singleplex and multiplex eukaryotic proteins. To provide an intuitive picture, a flowchart is provided in Fig. 4 to illustrate the prediction process of \( \text{Euk-ECC-mPLoc} \).

Results and Discussion

In statistical prediction, it is needed to evaluate the quality of different prediction methods. The following three commonly used methods, that is, the independent data set test, K-fold cross validation test, and jackknife test, are often used for evaluating the power of a statistical prediction method. Of the three methods, the jackknife test is deemed as the most objective because it always generates a unique result for a given benchmark dataset, as elucidated in a comprehensive review [18]. Therefore, the jackknife test has been increasingly and widely employed by researchers to examine the accuracy of various prediction methods (see, e.g., [23,24,26,86–88]). Accordingly, in the present study, we use jackknife test to evaluate the power of \( \text{Euk-ECC-mPLoc} \).

Actually, for such a stringent and complicated dataset containing both single-location and multiple-location eukaryotic proteins distributed among 22 subcellular location sites, so far only three existing predictors, i.e., \( \text{Euk-mPLoc} \) [22], \( \text{Euk-mPLoc 2.0} \) [23] and \( \text{iLoc-Euk} \) [24], were able to deal with it. It has been reported from [23] that, \( \text{Euk-mPLoc 2.0} \), which is an updated version of \( \text{Euk-mPLoc} \), can significantly outperform \( \text{Euk-mPLoc} \). Moreover, as can be seen from [24], the overall jackknife success rate achieved by \( \text{iLoc-Euk} \) was about 15% higher than that by \( \text{Euk-mPLoc 2.0} \) when tested on the dataset \( \mathcal{X} \). As a result, \( \text{iLoc-Euk} \) is currently the best one. Therefore, to demonstrate the power of the proposed predictor, it would suffice to just compare \( \text{Euk-ECC-mPLoc} \) with \( \text{iLoc-Euk} \).

Table 2 reports the detailed results on the 22 eukaryotic subcellular locations obtained with \( \text{iLoc-Euk} \) and \( \text{Euk-ECC-mPLoc} \) on the aforementioned benchmark dataset \( \mathcal{X} \) by the jackknife test. For a fair algorithmic comparison between \( \text{Euk-ECC-mPLoc} \) and \( \text{iLoc-Euk} \), we use the same GOA database version that is described in this study to extract GO features for \( \text{Euk-ECC-mPLoc} \) and \( \text{iLoc-Euk} \). As can be seen from Table 2, for such a stringent and complicated dataset, the average jackknife success rate achieved by \( \text{Euk-ECC-mPLoc} \) is 69.70%, which is about 19% higher than that achieved by \( \text{iLoc-Euk} \) [24]. \( \text{Euk-ECC-mPLoc} \) achieves very satisfactory performance on most subcellular locations, whereas \( \text{iLoc-Euk} \) achieves very poor performance on some subcellular locations, e.g., “acrosome”, “endosome”, “hydrogenosome”, “melanosome” and “microsome”. It is indicated that \( \text{Euk-ECC-mPLoc} \) is more balanced than \( \text{iLoc-Euk} \). Meanwhile, \( \text{Euk-ECC-mPLoc} \) obtains 81.54% overall jackknife success rate, with about 3% performance improvement against \( \text{iLoc-Euk} \). For the benchmark dataset containing both singleplex and multiplex eukaryotic proteins, the prediction accuracy is mainly influenced by the multiplex characteristics of proteins in that location. Roughly speaking, the bigger multiplex protein ratio in a location, the lower success rate will be obtained. For example, there are about 32% and 60%
proteins respectively in the "melanosome" and "synapse" location belonging to two or more locations, iLoc-Euk obtains only 2.13% and 38.30% success rates respectively. Euk-ECC-mPLoc, however, achieves 53.19% and 46.81% success rates in the two locations respectively, with largely 51% improvement in the "melanosome" location and over 8% improvement in the "synapse" location. The main reason is that correlations between different subcellular location sites have been taken into account in our proposed Euk-ECC-mPLoc, while iLoc-Euk only transforms the problem of predicting multiplex eukaryotic protein subcellular locations to a number of problems of prediction of singleplex eukaryotic protein subcellular localization, and thus iLoc-Euk lose much important information related to multi-label learning problems, e.g., correlations between different subcellular locations as utilized in Euk-ECC-mPLoc. Therefore, Euk-ECC-mPLoc reaches better performance than iLoc-Euk in predicting multiplex proteins. Moreover, for some subcellular locations with smaller number of training proteins, the success rates achieved by Euk-ECC-mPLoc are 35% ~ 90% higher than those by iLoc-Euk. For example, the success rate by Euk-ECC-mPLoc in "hydrogenosome" is 90% higher than that by iLoc-Euk, and the success rate by Euk-ECC-mPLoc in "acrosome" is about 64% higher than that by iLoc-Euk. This may be caused by the inherent advantage of SVM base classifier used in Euk-ECC-mPLoc.

Table 3 illustrates the "exact match" success rate between predicted outputs and real annotations on the same benchmark dataset \( \% \) by the jackknife test. The "exact match" means that both the predicted number and annotations of the subcellular locations for a query protein are the same as real observations. For a protein belonging to three subcellular locations, if only two of the three are correctly predicted, or the predicted result contains a location not belonging to the three, the prediction score will be counted as 0. In other words, when and only when all the subcellular locations of a query protein are exactly predicted without any underprediction or overprediction, can the prediction be scored with 1. Meanwhile, the success rates by the random predictor are also shown. Because iLoc-Euk didn’t provide the accuracy value specific to each subset in terms of the number of subcellular locations, the corresponding values are set to be "-". As can be seen from Table 3, the overall "exact match" success rate forms the problem of predicting multiplex eukaryotic protein subcellular locations to a number of problems of prediction of singleplex eukaryotic protein subcellular localization, and thus iLoc-Euk lose much important information related to multi-label learning problems, e.g., correlations between different subcellular locations as utilized in Euk-ECC-mPLoc. Therefore, Euk-ECC-mPLoc reaches better performance than iLoc-Euk in predicting multiplex proteins. Moreover, for some subcellular locations with smaller number of training proteins, the success rates achieved by Euk-ECC-mPLoc are 35% ~ 90% higher than those by iLoc-Euk. For example, the success rate by Euk-ECC-mPLoc in "hydrogenosome" is 90% higher than that by iLoc-Euk, and the success rate by Euk-ECC-mPLoc in "acrosome" is about 64% higher than that by iLoc-Euk. This may be caused by the inherent advantage of SVM base classifier used in Euk-ECC-mPLoc.

Table 3. A comparison of the jackknife “exact match” success rates by iLoc-Euk [24] and the proposed Euk-ECC-mPLoc on the benchmark dataset \( \% \) that covers 22 location sites of eukaryotic proteins in which none of the proteins included has \( \geq 25\% \) pairwise sequence identity to any other in a same location.

| Number of Locations | Euk-ECC-mPLoc | iLoc-Euk | Random |
|---------------------|---------------|----------|--------|
| 1                   | 75%           | -        | \( \frac{1}{C_{22}} = 4.55\% \) |
| 2                   | 59.09%        | -        | \( \frac{1}{C_{22}} = 0.43\% \) |
| 3                   | 10.42%        | -        | \( \frac{1}{C_{22}} = 0.06\% \) |
| 4                   | 0%            | -        | \( \frac{1}{C_{22}} = 0.01\% \) |
| Overall             | 72.59%        | 71.27%   | -      |

doi:10.1371/journal.pone.0036317.t003

Table 4. The predicted outputs by iLoc-Euk and Euk-ECC-mPLoc as well as the corresponding experimental annotations from DBMLoc [89].

| UniProt entry | UniProt entry name | Locations predicted by iLoc-Euk | Locations predicted by Euk-ECC-mPLoc | Annotations in DBMLoc |
|---------------|--------------------|---------------------------------|-------------------------------------|-----------------------|
| P38143        | GPX2_YEAST         | Cytoplasm                       | Cytoplasm                           | Cytoplasm             |
|               |                    |                                 |                                     |                       |
| P28823        | TUD_DROME          | Mitochondrion                   | Cytoplasm                           | Cytoplasm             |
|               |                    |                                 | Mitochondrion                       | Mitochondrion         |
| P28829        | BYR2_SCHPO         | Cytoplasm                       | Cell membrane                       | Cell membrane         |
|               |                    |                                 | Cytoplasm                           | Cytoplasm             |
| P32614        | FRDS_YEAST         | Cytoplasm                       | Cytoplasm                           | Cytoplasm             |
|               |                    |                                 | Mitochondrion                       | Mitochondrion         |
| Q9H190        | SDCB2_HUMAN        | Cytoplasm                       | Cell membrane                       | Cell membrane         |
|               |                    |                                 | Cytoplasm                           | Cytoplasm             |
| Q9Y7Q2        | GST1_SCHPO         | Cytoplasm                       | Cytoplasm                           | Cytoplasm             |
|               |                    |                                 | Nucleus                             | Nucleus               |
| O59827        | GST2_SCHPO         | Cytoplasm                       | Cytoplasm                           | Cytoplasm             |
|               |                    |                                 | Nucleus                             | Nucleus               |
| P27476        | NSR1_YEAST         | Nucleus                         | Mitochondrion                       | Mitochondrion         |
|               |                    |                                 | Nucleus                             | Nucleus               |
| P47119        | ITPA_YEAST         | Nucleus                         | Cytoplasm                           | Cytoplasm             |
|               |                    |                                 | Nucleus                             | Nucleus               |

doi:10.1371/journal.pone.0036317.t004
achieved by Euk-ECC-mPLoc is 72.59%, which is slightly higher than 71.27%, the corresponding “exact match” success rate achieved by iLoc-Euk [24]. The “exact match” accuracy of Euk-ECC-mPLoc is significantly superior to the random predictor. Therefore, our approach is quite promising for handling multiplex proteins, or at least play a complementary role to the existing predictors in identifying the subcellular locations of eukaryotic proteins.

In order to make the readers understand the superiority of our approach over other existing predictors more easily and intuitively, several typical proteins that are localized in multiple subcellular locations are selected from DBMLoc [89] which is a database of proteins with multiple subcellular localizations, and thus make prediction by inputting them into our Euk-ECC-mPLoc and iLoc-Euk online web servers respectively. Results are listed in Table 4 with the predicted outputs by the two predictors and the corresponding experimental annotations. As can be seen from Table 4, predicted subcellular locations achieved by our approach are all identical to the corresponding true annotations, whereas iLoc-Euk fails to get fully accurate results.

**Conclusion**

Prediction of protein subcellular localization is a challenging problem, particularly when the system concerned contains both singleplex and multiplex proteins. In this paper, we have proposed a novel multi-label predictor, called Euk-ECC-mPLoc, for predicting eukaryotic protein subcellular locations based on the powerful EOC algorithm and a hybrid of GO and DC feature extraction methods, which has been demonstrated powerful for dealing with both singleplex and multiplex proteins. Since user-friendly and publicly accessible web-servers represent the future direction for developing practically more useful predictors [90], here we have provided a web-server for the method presented in this paper at http://levis.tongji.edu.cn:8080/bioinfo/Euk-ECC-mPLoc/. The current approach represents a new strategy to deal with the multi-label biological problems, and hence may become a useful tool in the areas of bioinformatics and proteomics.

**Acknowledgments**

The authors wish to thank anonymous reviewers for their constructive comments, which are very helpful for strengthening the presentation of this paper.

**Author Contributions**

Conceived and designed the experiments: G-ZL XW. Performed the experiments: XW. Analyzed the data: XW G-ZL. Contributed reagents/materials/analysis tools: XW. Wrote the paper: XW.

**References**

1. Reinhardt A, Hubbard T (1998) Using neural networks for prediction of the subcellular location of proteins. Nucleic Acids Research 26: 2230–2236.
2. Chou KC, Elrod DW (1999) Protein subcellular location prediction. Protein Engineering 12: 107–118.
3. Hua S, Sun Z (2001) Support vector machine approach for protein subcellular location prediction. Bioinformatics 17: 721–729.
4. Chou KC, Cai YD (2002) Using functional domain composition and support vector machines for prediction of protein subcellular location. Journal of Biological Chemistry 277: 45763–45769.
5. Park KJ, Kanehisa M (2003) Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs. Bioinformatics 19: 1656–1663.
6. Huang Y, Li Y (2004) Prediction of protein subcellular locations using fuzzy k-NN method. Bioinformatics 20: 21–28.
7. Lu Z, Szafron D, Greiner R, Lu P, Wishart D, et al. (2004) Predicting subcellular localization of proteins using machine-learned classifiers. Bioinformatics 20: 547–556.
8. Yu CS, Lin CJ, Hwang JK (2004) Predicting subcellular localization of proteins for gram-negative bacteria by support vector machines based on n-peptide compositions. Protein Science 13: 1402–1406.
9. Bhasin M, Raghava GPS (2004) Euk-mPLoc: SVM-based method for subcellular localization of eukaryotic proteins using dipeptide composition and PSI-BLAST. Nucleic Acids Research 32: W414–W419.
10. Wang J, Sung WK, Krishnan A, Li KB (2005) Protein subcellular localization prediction for gram-negative bacteria using amino acid subalphanets and a combination of multiple support vector machines. BMC Bioinformatics 6: 174.
11. Garg 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. Journal of Biological Chemistry 280: 14427–14432.
12. Chou KC, Shen HB (2006) Predicting eukaryotic protein subcellular location by fusing optimized Evidence-Theoretic K-Nearest neighbor classifiers. Journal of Proteome Research 5: 1858–1897.
13. Pierleoni A, Martelli PL, Fasilli R, Casadio R (2006) BaCelLo: a balanced subcellular localization predictor. Bioinformatics 22: e408–e416.
14. Chou KC, Shen HB (2006) Hum-PLoc: a novel ensemble classifier for predicting human protein subcellular localization. Biochemical and Biophysical Research Communications 347: 150–157.
15. Shen HB, Yang J, Chou KC (2007) Euk-PLoc: an ensemble classifier for large-scale eukaryotic protein subcellular location prediction. Amino Acids 33: 57–67.
16. Niu B, Jin YH, Feng KY, Lu WC, Cai YD, et al. (2008) Using AdaBoost for the prediction of subcellular location of prokaryotic and eukaryotic proteins. Molecular Diversity 12: 41–45.
17. Nakai K (2000) Protein sorting signals and prediction of subcellular localization. Advances in Protein Chemistry 54: 277–344.
18. Chou KC, Shen HB (2007) Recent progress in protein subcellular location prediction. Analytical Biochemistry 367: 1–16.
19. Glory E, Murphy RF (2007) Automated subcellular location determination and High-Throughput microscopy. Developmental Cell 12: 7–16.
36. Saha SS, Panda G (2010) A novel feature representation method based on chou's pseudo amino acid composition for protein structural class prediction. Computational Biology and Chemistry 34: 320–327.

37. Zhang SW, Chen W, Yang F, Pan Q (2008) Using chou's pseudo amino acid composition to predict protein quaternary structure: a sequence-segmented PseAAC approach. Amino Acids 35: 591–598.

38. Qiu JD, Huang JH, Shi SP, Liang RP (2010) Using the concept of chou's pseudo amino acid composition to predict enzyme family classes: An approach with support vector machine based on discrete wavelet transform. Protein and Peptide Letters 17: 715–722.

39. Zhou XB, Chen C, Li ZC, Zou XY (2007) Using chou's amphiphilic pseudo-amino acid composition and support vector machine for prediction of enzyme subfamily classes. Journal of Theoretical Biology 248: 546–553.

40. Wang YC, Wang XB, Yang ZX, Deng NY (2010) Prediction of enzyme subfamily class via pseudo amino acid composition by incorporating the conjoint triple feature. Protein and Peptide Letters 17: 1441-1449.

41. Li FM, Li QZ (2008) Predicting protein subcellular localization using chou's pseudo amino acid composition and improved hybrid approach. Protein and Peptide Letters 15: 612–616.

42. Zhang W, Zhang YL, Yang HF, Zhao CH, Pan Q (2008) Using the concept of chou’s pseudo amino acid composition to predict protein subcellular localization: an approach by incorporating evolutionary information and von neumann entropies. Amino Acids 34: 563–572.

43. Lin J, Wang Y (2011) Using a novel AdaBoost algorithm and chou’s pseudo amino acid composition for predicting protein subcellular localization. Protein and Peptide Letters 18: 1219–1225.

44. Lin J, Wang Y, Xu X (2011) A novel ensemble and composite approach for classifying proteins based on chou’s pseudo amino acid composition. African Journal of Biotechnology 10: 16063–16068.

45. Ding YS, Zhang TL (2008) Using chou’s pseudo amino acid composition to predict subcellular localization of apoptosis proteins: An approach with immune genetic algorithm-based ensemble classifier. Pattern Recognition Letters 29: 1837–1892.

46. Lin H, Wang H, Ding H, Chen YL, Li QZ (2009) Prediction of subcellular localization of apoptosis protein using chou’s pseudo amino acid composition. Acta Biotheoretica 57: 321–330.

47. JuiN X, Wei R, Zhan T, Gu Q (2008) Using the concept of chou’s pseudo amino acid composition to predict apoptosis proteins subcellular location: An approach by approximate entropy. Protein and peptide letters 15: 392–396.

48. Kandhaswamy KK, Papadimitriu G, Moller S, Hartmann E, Kalies KL, et al. (2010) Prediction of apoptosis protein locations with genetic algorithms and support vector machines through a new mode of pseudo amino acid composition. Protein and Peptide Letters 17: 1473–1479.

49. Jiang X, Wei R, Zhao Y, Zhang T (2008) Using chou’s pseudo amino acid composition based on approximate entropy and an ensemble of AdaBoost classifiers to predict protein subnuclear location. Amino Acids 34: 669–675.

50. Lin H, Ding H, Guo FB, Zhang YH, Huang J (2008) Predicting subcellular localization of mycobacterial proteins by using chou’s pseudo amino acid composition. Protein and Peptide Letters 15: 739–744.

51. Zeng Y, Guo YZ, Xiao RQ, Yang Y, Yu LZ, et al. (2009) Using the augmented chou’s pseudo amino acid composition for predicting protein submitochondria locations based on auto covariance approach. Journal of Theoretical Biology 259: 366–372.

52. Nanini L, Laminini A (2008) Genetic programming for creating chou’s pseudo amino acid based features for submitochondria localization. Amino Acids 34: 653–660.

53. Ding H, Luo L, Lin H (2009) Prediction of cell wall lytic enzymes using chou’s amphiphilic pseudo amino acid composition. Protein and Peptide Letters 16: 351–355.

54. Esmaili M, Mohabatkar M, Mohsenzadeh S (2010) Using the concept of chou’s pseudo amino acid composition for risk type prediction of human papilloma viruses. Journal of Theoretical Biology 263: 203–209.

55. Fang Y, Guo Y, Feng Y, Li M (2008) Predicting DNA-binding proteins: an approach from discrete wavelet transform. Analytical Biochemistry 380: 68–75.

56. Gu D, Qiu JD, Zhang YL, Wang XB, Pan Q (2009) Prediction of G-Protein-Coupled receptor classes in low homology using chou’s pseudo amino acid composition with approximate entropy and hydrophobicity patterns. Protein and Peptide Letters 17: 559–567.

57. Qu JD, Huang JH, Liang RP, Lu NQ (2009) Prediction of G-protein-coupled receptor classes based on the concept of chou’s pseudo amino acid composition: An approach with predict-seq-bio-seq tool. Analytical Biochemistry 386: 101–109.

58. Guo J, Rao N, Lin G, Yang Y, Wang G (2011) Predicting protein folding states using the concept of chou’s pseudo amino acid composition. Journal of Computational Chemistry 32: 1612-1617.

59. Hao I (2008) The modified mahalanobis discriminant for predicting outer membrane proteins by using chou’s pseudo amino acid composition. Journal of Theoretical Biology 252: 350–356.

60. Mohabatkar H (2010) Prediction of cyclin proteins using chou’s pseudo amino acid composition. Protein and Peptide Letters 17: 1207–1214.

61. Mohabatkar H, Beigi MM, Esmaili A (2011) Prediction of GABAA receptor proteins using the concept of chou’s pseudo-amino acid composition and support vector machine. Journal of Theoretical Biology 281: 18–23.