HybridGO-Loc: Mining Hybrid Features on Gene Ontology for Predicting Subcellular Localization of Multi-Location Proteins

Shibiao Wan[1], Man-Wai Mak[1]*, Sun-Yuan Kung[2]

1 Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong SAR, China, **2** Department of Electrical Engineering, Princeton University, Princeton, New Jersey, United States of America

Abstract

Protein subcellular localization prediction, as an essential step to elucidate the functions *in vivo* of proteins and identify drugs targets, has been extensively studied in previous decades. Instead of only determining subcellular localization of single-label proteins, recent studies have focused on predicting both single- and multi-location proteins. Computational methods based on Gene Ontology (GO) have been demonstrated to be superior to methods based on other features. However, existing GO-based methods focus on the occurrences of GO terms and disregard their relationships. This paper proposes a multi-label subcellular-localization predictor, namely HybridGO-Loc, that leverages not only the GO term occurrences but also the inter-term relationships. This is achieved by hybridizing the GO frequencies of occurrences and the semantic similarity between GO terms. Given a protein, a set of GO terms are retrieved by searching against the gene ontology database, using the accession numbers of homologous proteins obtained via BLAST search as the keys. The frequency of GO occurrences and semantic similarity (SS) between GO terms are used to formulate frequency vectors and semantic similarity vectors, respectively, which are subsequently hybridized to construct fusion vectors. An adaptive-decision based multi-label support vector machine (SVM) classifier is proposed to classify the fusion vectors. Experimental results based on recent benchmark datasets and a new dataset containing novel proteins show that the proposed hybrid-feature predictor significantly outperforms predictors based on individual GO features as well as other state-of-the-art predictors. For readers' convenience, the HybridGO-Loc server, which is for predicting virus or plant proteins, is available online at http://bioinfo.eie.polyu.edu.hk/HybridGoServer/.

Introduction

Proteins located in appropriate physiological contexts within a cell are of paramount importance to exert their biological functions. Subcellular localization of proteins is essential to the functions of proteins and has been suggested as a means to maximize functional diversity and economize on protein design and synthesis [1]. Aberrant protein subcellular localization is closely correlated to a broad range of human diseases, such as Alzheimer's disease [2], kidney stone [3], primary human liver tumors [4], breast cancer [5], pre-eclampsia [6] and Bartter syndrome [7]. Knowing where a protein resides within a cell can give insights on drug targets identification and drug design [8,9]. Wet-lab experiments such as fluorescent microscopy imaging, cell fractionation and electron microscopy are the gold standard for validating subcellular localization and are essential for the design of high quality localization databases such as The Human Protein Atlas (http://www.proteinatlas.org/). However, wet-lab experiments are time-consuming and laborious. With the avalanche of newly discovered protein sequences in the post-genomic era, computational methods are required to assist biologists to deal with large-scale proteomic data to determine the subcellular localization of proteins.

Conventionally, subcellular-localization predictors can be roughly divided into sequence-based and annotation-based. Sequence-based methods use (1) amino-acid compositions [10,11], (2) sequence homology [12,13], and (3) sorting signals [14,15] as features. Annotation-based menthods use information beyond the protein sequences, such as Gene Ontology (GO) terms [16–21], Swiss-Prot keywords [22], and PubMed abstracts [23,24]. A number of studies have demonstrated that methods based on GO information are superior to methods based on sequence-based features [25–28]. Note that the GO database contains not only experimental data but also predicted data (http://www.geneontology.org/GO.evidence.shtml), which may be determined by sequence-based methods. From this point of view, the GO-based prediction, which uses the GO annotation database to retrieve GO terms, is a filtering method for sequence-based predictions.

The GO comprises three orthogonal taxonomies whose terms describe the cellular components, biological processes, and molecular functions of gene products. The GO terms in each

taxonomy are organized within a directed acyclic graph. These terms are placed within structural relationships, of which the most important being the 'is-a' relationship (*parent* and *child*) and the 'part-of' relationship (*part* and *whole*) [29,30]. Recently, the GO consortium has been enriched with more structural relationships, such as 'positively-regulates', 'negatively-regulates' and 'has-part' [31,32]. These relationships reflect that the GO hierarchical tree for each taxonomy contains redundant information, for which semantic similarity over GO terms can be found.

Instead of only determining subcellular localization of single-label proteins, recent studies have been focusing on predicting both single- and multi-location proteins. Since there exist multi-location proteins that can simultaneously reside at, or move between, two or more subcellular locations, it is important to include these proteins in the predictors. Actually, multi-location proteins play important roles in some metabolic processes that take place in more than one cellular compartment, e.g., fatty acid β-oxidation in the peroxisome and mitochondria, and antioxidant defense in the cytosol, mitochondria and peroxisome [33].

Recently, several multi-label predictors based on GO have been proposed, including Plant-mPLoc [34], Virus-mPLoc [35], iLoc-Plant [36], iLoc-Virus [37], KNN-SVM [38], mGOASVM [39] and others [40,41]. These predictors have demonstrated superiority over sequence-based methods. These predictors use the occurrences of the GO terms but do not take the semantic relationships between GO terms into account.

Since the relationship between GO terms reflects the association between different gene products, protein sequences annotated with GO terms can be compared on the basis of semantic similarity measures. The semantic similarity over GO has been extensively studied and have been applied to many biological problems, including protein function prediction [42,43], subnuclear localization prediction [44], protein-protein interaction inference [45–47] and microarray clustering [48]. The performance of these predictors depends on whether the similarity measure is relevant to the biological problems. Over the years, a number of semantic similarity measures have been proposed, some of which have been used in natural language processing.

Semantic similarity measures can be applied at the GO-term level or the gene-product level. At the GO-term level, methods are roughly categorized as node-based and edge-based. The node-based measures basically rely on the concept of information content of terms, which was proposed by Resnik [49] for natural language processing. Later, Lord et al. [50] applied this idea to measure the semantic similarity among GO terms. Lin et al. [51] proposed a method based on information theory and structural information. Subsequently, more node-based measures [52–54] were proposed. Edge-based measures are based on using the length or the depth of different paths between terms and/or their common ancestors [55–58]. At the gene-product level, two most common methods are pairwise approaches [59–63] and groupwise approaches [64–67]. Pairwise approaches measure similarity between two gene products by combining the semantic similarities between their terms. Groupwise approaches, on the other hand, directly group the GO terms of a gene product as a set, a graph or a vector, and then calculate the similarity by set similarity techniques, graph matching techniques or vector similarity techniques. More recently, Pesquita et al. [68] reviewed the semantic similarity measures applied to biomedical ontologies, and Guzzi et al. [69] provides a comprehensive review on the relationship between semantic similarity measures and biological features.

This paper proposes a multi-label predictor based on hybridizing frequency of occurrences of GO terms and semantic similarity between the terms for protein subcellular localization prediction. Compared to existing multi-label subcellular-localization predictors, our proposed predictor has the following advantages: (1) it formulates the feature vectors by hybridizing GO frequency of occurrences and GO semantic similarity features which contain richer information than only GO term frequencies; (2) it adopts a new strategy to incorporate richer and more useful homologous information from more distant homologs rather than using the top homologs only; (3) it adopts an adaptive decision strategy for multi-label SVM classifiers so that it can effectively deal with datasets containing both single-label and multi-label proteins. Results on two recent benchmark datasets and a new dataset containing novel proteins demonstrate that these three properties enable the proposed predictor to accurately predict multi-location proteins and outperform several state-of-the-art predictors.

Methods

Legitimacy of Using GO Information

Despite their good performance, GO-based methods have received some criticisms from the research community. The main argument of these criticisms is that the cellular component GO terms already have the cellular component categories, i.e., if the GO terms are known, the subcellular locations will also be known. The prediction problem can therefore be easily solved by creating a lookup table using the cellular component GO terms as the keys and the cellular component categories as the hashed values. Such a naive solution, however, will lead to very poor prediction performance, as demonstrated and explained in our previous studies [28,39]. A number of studies [70–72] by other groups also strongly support the legitimacy of using GO information for subcellular localization. For example, as suggested by [72], the good performance of GO-based methods is due to the high representation power of the GO space as compared to the Euclidean feature spaces used by the conventional sequence-based methods.

Retrieval of GO Terms

The proposed predictor can use either the accession numbers (AC) or amino acid (AA) sequences of query proteins as input. Specifically, for proteins with known ACs, their respective GO terms are retrieved from the Gene Ontology annotation (GOA) database (http://www.ebi.ac.uk/GOA) using the ACs as the searching keys. For proteins without ACs, their AA sequences are presented to BLAST [73] to find their homologs, whose ACs are then used as keys to search against the GOA database.

While the GOA database allows us to associate the AC of a protein with a set of GO terms, for some novel proteins, neither their ACs nor the ACs of their top homologs have any entries in the GOA database; in other words, no GO terms can be retrieved by using their ACs or the ACs of their top homologs. In such case, the ACs of the homologous proteins, as returned from BLAST search, will be successively used to search against the GOA database until a match is found. With the rapid progress of the GOA database, it is reasonable to assume that the homologs of the query proteins have at least one GO term [17]. Thus, it is not necessary to use back-up methods to handle the situation where no GO terms can be found. The procedures are outlined in Fig 1.

GO Frequency Features

Let $\mathbb{W}$ denote a set of distinct GO terms corresponding to a data set. $\mathbb{W}$ is constructed in two steps: (1) identifying all of the GO terms in the dataset and (2) removing the repetitive GO terms.
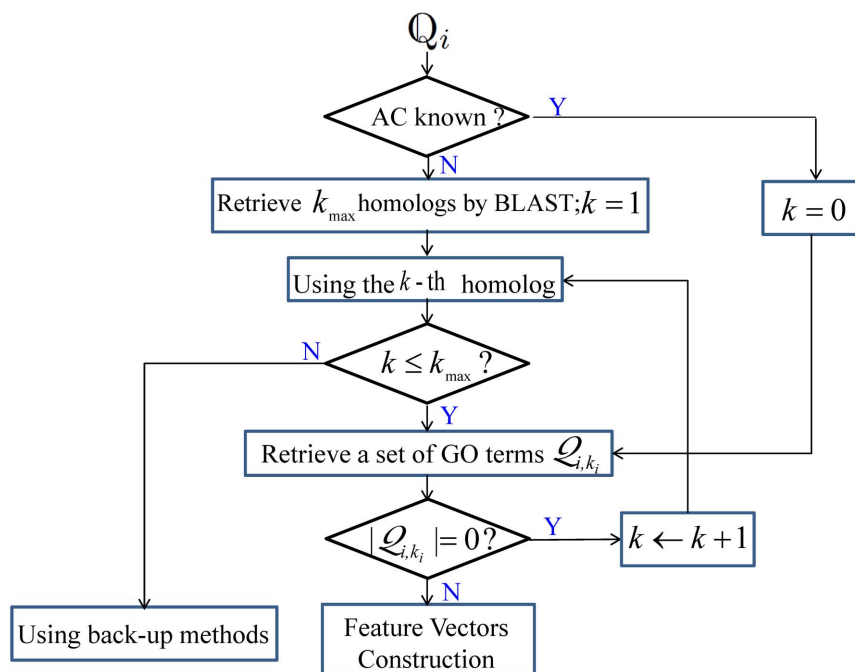
Figure 1. Procedures of retrieving GO terms. $\mathbb{Q}_i$: the i-th query protein; k_{max}: the maximum number of homologs retrieved by BLAST with the default parameter setting; $\mathcal{Q}_{i,k_i}$: the set of GO terms retrieved by BLAST using the k_i-th homolog for the i-th query protein $\mathbb{Q}_i$; k_i: the k_i-th homolog used to retrieve the GO terms.
doi:10.1371/journal.pone.0089545.g001

Suppose W distinct GO terms are found, i.e., $|\mathbb{W}| = W$; these GO terms form a GO Euclidean space with W dimensions. For each sequence in the dataset, a GO vector is constructed by matching its GO terms against $\mathbb{W}$, using the number of occurrences of individual GO terms in $\mathbb{W}$ as the coordinates. Specifically, the GO vector $\mathbf{p}_i$ of the i-th protein $\mathbb{P}_i$ is defined as:

$$\mathbf{p}_i^F = [b_{i,1}, \cdots, b_{i,j}, \cdots, b_{i,W}]^{\mathrm{T}}, b_{i,j} = \begin{cases} f_{i,j} & , \quad \text{GO hit} \\ 0 & , \quad \text{otherwise} \end{cases} \quad (1)$$

where $f_{i,j}$ is the number of occurrences of the j-th GO term (term-frequency) in the i-th protein sequence. The rationale is that the term-frequencies contain important information for classification. Note that $b_{i,j}$'s are analogous to the term-frequencies commonly used in document retrieval.

Similarly, for the t-th query protein $\mathbb{Q}_t$, the GO frequency vector is defined as:

$$\mathbf{q}_t^F = [b_{t,1}, \cdots, b_{t,j}, \cdots, b_{t,W}]^{\mathrm{T}}, b_{t,j} = \begin{cases} f_{t,j} & , \quad \text{GO hit} \\ 0 & , \quad \text{otherwise} \end{cases} \quad (2)$$

In the following sections, we use the superscript F to denote the GO frequency features in Eq. 2.

Semantic-Similarity Features

Semantic similarity (SS) is a measure for quantifying the similarity between categorical data (e.g., words in documents), where the notion of similarity is based on the likeliness of meanings in the data. It is originally developed by Resnik [49] for natural language processing. The idea is to evaluate semantic similarity in an 'is-a' taxonomy using the shared information contents of categorical data. In the context of gene ontology, the semantic similarity between two GO terms is based on their most specific common ancestor in the GO hierarchy. The relationships between GO terms in the GO hierarchy, such as 'is-a' ancestor-child, or 'part-of' ancestor-child can be obtained from the SQL database through the link: http://archive.geneontology.org/latest-termdb/go_daily-termdb-tables.tar.gz. Note here only the 'is-a' relationship is considered for semantic similarity analysis [51]. Specifically, the semantic similarity between two GO terms x and y is defined as [49]:

$$sim(x,y) = \max_{c \in A(x,y)} [-\log(p(c))], \quad (3)$$

where $A(x, y)$ is the set of ancestor GO terms of both x and y, and $p(c)$ is the probability of the number of gene products annotated to the GO term c divided by the total number of gene products annotated in the GO taxonomy.

While Resnik's measure is effective in quantifying the shared information between two GO terms, it ignores the distance between the terms and their common ancestors in the GO hierarchy. To further incorporate structural information from the GO hierarchy into the similarity measure, we have explored three extension of Resnik's measure, namely Lin's measure [51], Jiang's measure [74], and relevance similarity (RS) [52].

Given two GO terms x and y, the similarity by Lin's measure is:

$$sim_{Lin}(x,y) \equiv sim_1(x,y) = \max_{c \in A(x,y)} \left(\frac{2 \cdot [-\log(p(c))]}{-\log(p(x)) - \log(p(y))} \right) \quad (4)$$

The similarity by Jiang's measure is:

$$sim_{Jiang}(x,y) \equiv sim_2(x,y)$$
$$= \max_{c \in A(x,y)} \left(\frac{1}{1 - \log(p(x)) - \log(p(y)) + 2 \cdot [-\log(p(c))]} \right) \quad (5)$$

The similarity by RS is calculated as:

$$sim_{RS}(x,y) \equiv sim_3(x,y)$$
$$= \max_{c \in A(x,y)} \left(\frac{2 \cdot [-\log(p(c))]}{-\log(p(x)) - \log(p(y))} \cdot (1 - p(c)) \right) \quad (6)$$

Among the three measures, $sim_{Lin}(x,y)$ and $sim_{Jiang}(x,y)$ are relative measures that are proportional to the difference in information content between the terms and their common ancestors, which is independent of the absolute information content of the ancestors. On the other hand, $sim_{RS}(x,y)$ incorporates the probability of annotating the common ancestors as a weighing factor to Lin's measure. To simplify notations, we refer $sim_{Lin}(x,y)$, $sim_{Jiang}(x,y)$ and $sim_{RS}(x,y)$ as $sim_1(x,y)$, $sim_2(x,y)$ and $sim_3(x,y)$, respectively.

Based on the semantic similarity between two GO terms, we adopted a continuous measure proposed in [48] to calculate the similarity between two proteins. Specifically, given two proteins $\mathbb{P}_i$ and $\mathbb{P}_j$, we retrieved their corresponding GO terms $\mathcal{P}_i$ and $\mathcal{P}_j$ as described in the subsection "Retrieval of GO Terms". (Note that strictly speaking, $\mathcal{P}_i$ should be $\mathcal{P}_{i,k_i}$, where k_i is the k_i-th homolog used to retrieve the GO terms for the i-th protein. To simplify notations, we write it as $\mathcal{P}_i$.) Then, we computed the semantic similarity between two sets of GO terms $\{\mathcal{P}_i, \mathcal{P}_j\}$ as follows:

$$S_l(\mathcal{P}_i, \mathcal{P}_j) = \sum_{x \in \mathcal{P}_i} \max_{y \in \mathcal{P}_j} sim_l(x,y), \quad (7)$$

where $l \in \{1,2,3\}$, and $sim_l(x,y)$ is defined in Eq. 4 to Eq. 6. $S_l(\mathcal{P}_j, \mathcal{P}_i)$ is computed in the same way by swapping $\mathcal{P}_i$ and $\mathcal{P}_j$. Finally, the overall similarity between the two proteins is given by:

$$SS_l(\mathcal{P}_i, \mathcal{P}_j) = \frac{S_l(\mathcal{P}_i, \mathcal{P}_j) + S_l(\mathcal{P}_j, \mathcal{P}_i)}{S_l(\mathcal{P}_i, \mathcal{P}_i) + S_l(\mathcal{P}_j, \mathcal{P}_j)}, \quad (8)$$

where $l \in \{1,2,3\}$. In the sequel, we refer the SS measures by Lin, Jiang and RS to as SS1, SS2 and SS3, respectively.

Thus, for a testing protein $\mathbb{Q}_t$ with GO term set $\mathcal{Q}_t$, a GO semantic similarity (SS) vector $q_t^{S_l}$ can be obtained by computing the semantic similarity between $\mathcal{Q}_t$ and each of the training protein $\{\mathbb{P}_i\}_{i=1}^N$, where N is the number of training proteins. Thus, $\mathbb{Q}_t$ can be represented by an N-dimensional vector:

$$q_t^{S_l} = [SS_l(\mathcal{Q}_t, \mathcal{P}_1), \cdots, SS_l(\mathcal{Q}_t, \mathcal{P}_i), \cdots, SS_l(\mathcal{Q}_t, \mathcal{P}_N)]^T, \quad (9)$$

where $l \in \{1,2,3\}$. In other words, $q_t^{S_l}$ represents the SS vector by using the l-th SS measure.

Hybridization of Two GO Features

As can be seen from the subsections "GO Frequency Features" and "Semantic-Similarity Features", we know that the GO frequency features (Eq. 2) use the frequency of occurrences of GO terms, while GO SS features (Eq. 4 to Eq. 6) use the semantic similarity between GO terms. These two features are developed from two different perspectives. It is therefore reasonable to believe that these two kinds of information complement each other. Based on this assumption, we combine these two GO features and form a hybridized vector as:

$$q_t^{H_l} = \begin{bmatrix} q_t^F \\ q_t^{S_l} \end{bmatrix}, \quad (10)$$

where $l \in \{1,2,3\}$. In other words, $q_t^{H_l}$ represents the hybridizing-feature vector by combining the GO frequency features and the SS features derived from the l-th SS measure. We refer them to as *Hybrid1*, *Hybrid2* and *Hybrid3*, respectively.

Multi-label Multi-class SVM Classification

The hybridized-feature vectors obtained from the previous subsection are used for training multi-label one-vs-rest support vector machines (SVMs). Specifically, for an M-class problem (here M is the number of subcellular locations), M independent binary SVMs are trained, one for each class. Denote the hybrid GO vectors of the t-th query protein using the l-th SS measure as $q_t^{H_l}$. Given the t-th query protein $\mathbb{Q}_t$, the score of the m-th SVM using the l-th SS measure is

$$s_{m,l}(\mathbb{Q}_t) = \sum_{r \in \mathcal{S}_m} \alpha_{m,r} y_{m,r} K(p_r^{H_l}, q_t^{H_l}) + b_m \quad (11)$$

where $q_t^{H_l}$ is the hybrid GO vector derived from $\mathbb{Q}_t$ (See Eq. 10), $\mathcal{S}_{m,l}$ is the set of support vector indexes corresponding to the m-th SVM, $\alpha_{m,r}$ are the Lagrange multipliers, $y_{m,r} \in \{-1, +1\}$ indicates whether the r-th training protein belongs to the m-th class or not, and $K(\cdot,\cdot)$ is a kernel function. Here, the linear kernel was used.

Unlike the single-label problem where each protein has one predicted label only, a multi-label protein could have more than one predicted labels. In this work, we compared two different decision schemes for this multi-label problem. In the first scheme, the predicted subcellular location(s) of the i-th query protein are given by

$$\mathcal{M}_l^*(\mathbb{Q}_t) =$$
$$\begin{cases} \bigcup_{m=1}^M \{m : s_{m,l}(\mathbb{Q}_t) > 0\}, & \text{when } \exists \ m \in \{1,\ldots,M\} \ s.t. \ s_{m,l}(\mathbb{Q}_t) > 0 ; \\ \arg\max_{m=1}^M s_{m,l}(\mathbb{Q}_t), & \text{otherwise.} \end{cases} \quad (12)$$

The second scheme is an improved version of the first one in that the decision threshold is dependent on the test protein. Specifically, the predicted subcellular location(s) of the i-th query protein are given by:

If $\exists \ s_{m,l}(\mathbb{Q}_t) > 0$,

$$M_l(\mathbb{Q}_t) = \bigcup_{m=1}^M (m : s_{m,l}(\mathbb{Q}_t) \geq \min\{1.0, f(s_{\max,l}(\mathbb{Q}_t))\}) \quad (13)$$

otherwise,

$$M(\mathbb{Q}_t) = \arg \max_{m=1}^{M} s_{m,l}(\mathbb{Q}_t). \tag{14}$$

In Eq. 13, $f(s_{\max,l}(\mathbb{Q}_t))$ is a function of $s_{\max,l}(\mathbb{Q}_t)$, where $s_{\max,l}(\mathbb{Q}_t) = \max_{m=1}^{M} s_{m,l}(\mathbb{Q}_t)$. In this work, we used a linear function as follows:

$$f(s_{\max,l}(\mathbb{Q}_t)) = \theta s_{\max,l}(\mathbb{Q}_t), \tag{15}$$

where $\theta \in [0.0, 1.0]$ is a hyper-parameter that can be optimized through cross-validation.

In fact, besides SVMs, many other machine learning models, such as hidden Markov models (HMMs) and neural networks (NNs) [75,76], have been used in protein subcellular-localization predictors. However, HMMs and NNs are not suitable for GO-based predictors because of the high dimensionality of GO vectors. The main reason is that under such condition, HMMs and NNs can be easily overtrained and thus lead to poor performance. On the other hand, linear SVMs can well handle high-dimensional data because even if the number of training samples is smaller than the feature dimension, linear SVMs are still able to find an optimal solution.

Materials and Performance Metrics

Datasets

In this paper, a virus dataset [35,37] and a plant dataset [36] were used to evaluate the performance of the proposed predictor. The virus and the plant datasets were created from Swiss-Prot 57.9 and 55.3, respectively. The virus dataset contains 207 viral proteins distributed in 6 locations. Of the 207 viral proteins, 165 belong to one subcellular locations, 39 to two locations, 3 to three locations and none to four or more locations. This means that about 20% of the proteins in the dataset are located in more than one subcellular location. The plant dataset contains 978 plant proteins distributed in 12 locations. Of the 978 plant proteins, 904 belong to one subcellular locations, 71 to two locations, 3 to three locations and none to four or more locations. The sequence identity of both datasets was cut off at 25%.

The breakdown of these two datasets are listed in Figs. 2(a) and 2(b). Fig. 2(a) shows that the majority (68%) of viral proteins in the virus dataset are located in host cytoplasm and host nucleus while proteins located in the rest of the subcellular locations totally account only around one third. This means that this multi-label dataset is imbalanced across the six subcellular locations. Similar conclusions can be drawn from Fig. 2(b), where most of the plant proteins exist in chloroplast, cytoplasm, nucleus and mitochondrion while proteins in other 8 subcellular locations totally account for less than 30%. This imbalanced property makes the prediction of these two multi-label datasets difficult. These two benchmark datasets are downloadable from the hyperlinks in the HybridGO-Loc server.

Performance Metrics

Compared to traditional single-label classification, multi-label classification requires more complicated performance metrics to better reflect the multi-label capabilities of classifiers. Conventional single-label measures need to be modified to adapt to multi-label classification. These measures include *Accuracy, Precision, Recall, F1-score (F1)* and *Hamming Loss (HL)* [77,78]. Specifically, denote $\mathcal{L}(\mathbb{Q}_i)$ and $\mathcal{M}(\mathbb{Q}_i)$ as the true label set and the predicted label set

for the i-th protein $\mathbb{Q}_i$ ($i = 1, \ldots, N$), respectively. Here, $N = 207$ for the virus dataset and $N = 978$ for the plant dataset. Then the five measurements are defined as follows:

$$Accuracy = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{|\mathcal{M}(\mathbb{Q}_i) \cap \mathcal{L}(\mathbb{Q}_i)|}{|\mathcal{M}(\mathbb{Q}_i) \cup \mathcal{L}(\mathbb{Q}_i)|} \right) \tag{16}$$

$$Precision = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{|\mathcal{M}(\mathbb{Q}_i) \cap \mathcal{L}(\mathbb{Q}_i)|}{|\mathcal{M}(\mathbb{Q}_i)|} \right) \tag{17}$$

$$Recall = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{|\mathcal{M}(\mathbb{Q}_i) \cap \mathcal{L}(\mathbb{Q}_i)|}{|\mathcal{L}(\mathbb{Q}_i)|} \right) \tag{18}$$

$$F1 = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{2|\mathcal{M}(\mathbb{Q}_i) \cap \mathcal{L}(\mathbb{Q}_i)|}{|\mathcal{M}(\mathbb{Q}_i)| + |\mathcal{L}(\mathbb{Q}_i)|} \right) \tag{19}$$

$$HL = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{|\mathcal{M}(\mathbb{Q}_i) \cup \mathcal{L}(\mathbb{Q}_i)| - |\mathcal{M}(\mathbb{Q}_i) \cap \mathcal{L}(\mathbb{Q}_i)|}{\mathcal{M}} \right) \tag{20}$$

where $|\cdot|$ means counting the number of elements in the set therein and $\cap$ represents the intersection of sets.

Accuracy, Precision, Recall and *F1* indicate the classification performance. The higher the measures, the better the prediction performance. Among them, *Accuracy* is the most commonly used criteria. *F1-score* is the harmonic mean of *Precision* and *Recall*, which allows us to compare the performance of classification systems by taking the trade-off between *Precision* and *Recall* into account. The *Hamming Loss (HL)* [77,78] is different from other metrics. As can be seen from Eq. 20, when all of the proteins are correctly predicted, i.e., $|\mathcal{M}(\mathbb{Q}_i) \cup \mathcal{L}(\mathbb{Q}_i)| = |\mathcal{M}(\mathbb{Q}_i) \cap \mathcal{L}(\mathbb{Q}_i)|$ ($i = 1, \ldots, N$), then $HL = 0$; whereas, other metrics will be equal to 1. On the other hand, when the predictions of all proteins are completely wrong, i.e., $|\mathcal{M}(\mathbb{Q}_i) \cup \mathcal{L}(\mathbb{Q}_i)| = \mathcal{M}$ and $|\mathcal{M}(\mathbb{Q}_i) \cap \mathcal{L}(\mathbb{Q}_i)| = 0$, then $HL = 1$; whereas, other metrics will be equal to 0. Therefore, the lower the *HL*, the better the prediction performance.

Two additional measurements [37,39] are often used in multi-label subcellular localization prediction. They are overall locative accuracy (*OLA*) and overall actual accuracy (*OAA*). The former is given by:

$$OLA = \frac{1}{\sum_{i=1}^{N} |\mathcal{L}(\mathbb{Q}_i)|} \sum_{i=1}^{N} |\mathcal{M}(\mathbb{Q}_i) \cap \mathcal{L}(\mathbb{Q}_i)|, \tag{21}$$

and the overall actual accuracy (*OLA*) is:

$$OAA = \frac{1}{N} \sum_{i=1}^{N} \Delta[\mathcal{M}(\mathbb{Q}_i), \mathcal{L}(\mathbb{Q}_i)] \tag{22}$$

where

$$\Delta[\mathcal{M}(\mathbb{Q}_i), \mathcal{L}(\mathbb{Q}_i)] = \begin{cases} 1 & \text{, if } \mathcal{M}(\mathbb{Q}_i) = \mathcal{L}(\mathbb{Q}_i) \\ 0 & \text{, otherwise.} \end{cases} \tag{23}$$
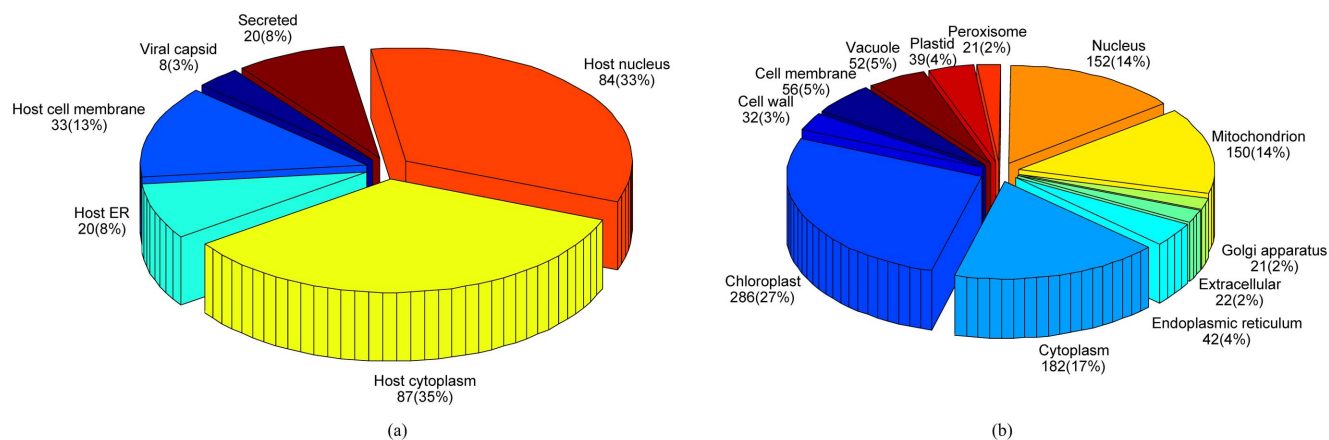
Figure 2. Breakdown of the (a) virus and (b) plant datasets. The number of proteins shown in each subcellular location represents the number of 'locative proteins' [37,39]. For (a), there are 207 actual proteins and 252 locative proteins; For (b), there are 978 actual proteins and 1055 locative proteins.

doi:10.1371/journal.pone.0089545.g002

According to Eq. 21, a locative protein is considered to be correctly predicted if any of the predicted labels matches any labels in the true label set. On the other hand, Eq. 22 suggests that an actual protein is considered to be correctly predicted only if *all* of the predicted labels match those in the true label set exactly. For example, for a protein coexist in, say 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 is considered to be incorrect. In other words, when and only when all of the subcellular locations of a query protein are exactly predicted without any overprediction or underprediction, can the prediction be considered as correct. Therefore, *OAA* is a more stringent measure as compared to *OLA*. *OAA* is also more objective than *OLA*. This is because locative accuracy is liable to give biased performance measures when the predictor tends to over-predict, i.e., giving large $|\mathcal{M}(\mathbb{Q}_i)|$ for many $\mathbb{Q}_i$. In the extreme case, if every protein is predicted to have all of the $\mathcal{M}$ subcellular locations, according to Eq. 20, the *OLA* is 100%. But obviously, the predictions are wrong and meaningless. On the contrary, *OAA* is 0% in this extreme case, which definitely reflects the real performance.

Among all the metrics mentioned above, *OAA* is the most stringent and objective. This is because if only some (but not all) of the subcellular locations of a query protein are correctly predict, the numerators of the other 4 measures (Eqs. 16 to 21) are non-zero, whereas the numerator of *OAA* in Eq. 22 is 0 (thus contribute nothing to the frequency count).

In statistical prediction, there are three methods that are often used for testing the generalization capabilities of predictors: independent tests, sub-sampling tests (or *K*-fold cross-validation) and leave-one-out cross validation (LOOCV). For independent tests, the selection of independent dataset often bears some sort of arbitrariness [79]; for the *K*-fold cross validation, different partitioning of a dataset will lead to different results, thus still being liable to statistical arbitrariness; for LOOCV, it will yield a unique outcome and is considered to be the most rigorous and bias-free method [80]. Hence, LOOCV was used to examine the performance of all predictors in this work. More detailed analysis of the statistical methods can be found in the supplementary materials. Note that the jackknife cross validation in iLoc-Plant and its variants is the same as LOOCV, as mentioned in [36,79]. Because the term jackknife also refers to the methods that estimate

the bias and variance of an estimator [81], to avoid confusion, we only use the term LOOCV in this paper.

Results

Comparing Different Features

Fig. 3(a) shows the performance of individual and hybridized GO features on the virus dataset based on leave-one-out cross validation (LOOCV). In the figure, *SS1, SS2 and SS3* represent Lin's, Jiang's and RS similarity measures, respectively. *Hybrid1, Hybrid2* and *Hybrid3* represent the hybridized features obtained from these measures. As can be seen, in terms of all the six performance metrics, the performance of the hybrid features is remarkably better than the performance of individual features, regardless of which of the GO frequency features or the three GO SS features were used. Specifically, the *OAAs* (the most stringent and objective metric) of all of the three hybrid features are at least 3% (absolute) higher than that of the individual features, which suggests that hybridizing the two features can significantly boost the prediction performance. Moreover, among the hybridized features, the performance of *Hybrid2*, namely combining GO frequency features and GO SS features by Jiang's measure, outperforms *Hybrid1* and *Hybrid3*. Another interesting thing is that although all of the individual GO SS features perform much worse than the GO frequency features, the performance of the three hybridized features is still better that of any of the individual features. This suggests that the GO frequency features and SS features are complementary to each other.

Similar conclusions can be drawn from the plant dataset shown in Fig. 3(b). However, comparison between Fig. 3(a) and Fig. 3(b) reveals that for the plant dataset, the performance of hybridized features outperforms all of the individual features in terms of all metrics except *OLA* and *Recall*, while for the virus dataset, the former is superior to the latter in terms of all metrics. However, the losses in these two metrics do not outweigh the significant improvement on other metrics, especially on *OAA*, which has around 3% (absolute) improvement in terms of hybridized features as opposed to using individual features. Among the hybridizing features, *Hybrid2* also outperforms *Hybrid1* and *Hybrid3* in terms of *OLA, Accuracy, Recall* and *F1-score*, whereas *Hybrid1* performs better than others in terms of *OAA* and *Precision*. These results demonstrate that the GO SS features obtained by Lin's measure and Jiang's measure are better candidates than the RS measure for

combining with the GO frequency features; however, there is no evidence suggesting which measure is better. It is also interesting to see that the performance of the three individual GO SS features is better than that of GO frequency features, in contrary to the results shown in Fig 3(a).

Comparing with State-of-the-Art Predictors

Table 1 and Table 2 compare the performance of the proposed predictor against several state-of-the-art multi-label predictors on the virus and plant dataset based on leave-one-out cross validation. Note that we used the best performing hybridizing features with the adaptive decision strategy. Specifically, for both the virus and plant datasets, the best performance was achieved when *Hybrid2* and the adaptive decision strategy with $\theta = 0.3$ were used. θ was determined by cross-validation as stated previously. Unless stated otherwise, we used *Hybrid2* to represent HybridGO-Loc in subsequent experiments. Our proposed predictor use the GO frequency features and GO semantic similarity features, whereas other predictors use only the GO frequency of occurrences as features. From the classification perspective, Virus-mPLoc [35] uses an ensemble OET-KNN (optimized evidence-theoretic K-nearest neighbors) classifier; iLoc-Virus [37] uses a multi-label KNN classifier; KNN-SVM [38] uses an ensemble of classifiers combining KNN and SVM; mGOASVM [39] uses a multi-label SVM classifier; and the proposed predictor use a multi-label SVM classifier incorporated with the adaptive decision scheme.

As shown in Table 1, the proposed predictor perform significantly better than the other predictors. The *OAA* and *OLA* of the proposed predictor are more than 15% (absolute) higher than that of iLoc-Virus and Virus-mPLoc. It also performs significantly better than KNN-SVM in terms of *OLA*. When comparing with mGOASVM, the proposed predictor performs remarkably better in all of the performance metrics, especially for the *OAA* (0.937 vs 0.889). These results demonstrate that hybridizing the GO frequency features and GO SS features can significantly boost prediction performance, which also suggests that these two kinds of information are proved to be complementary to each other in terms of predicting subcellular localization. Similar conclusions can be drawn for the plant dataset from

Table 2 except that the *OLA* of the proposed predictor is slightly worse than that of mGOASVM, and the *Recall* is equivalent to that of mGOASVM. Nevertheless, the small losses do not outweigh the impressive improvement in the other metrics, especially in the *OAA* (0.936 vs 0.874).

Prediction of Novel Proteins

To further demonstrate the effectiveness of HybridGO-Loc, a newer plant dataset constructed for mGOASVM [39] was used to compare with state-of-the-art multi-label predictors using independent tests. Specifically, this new plant dataset contains 175 plant proteins, of which 147 belong to one subcellular location, 27 belong to two locations, 1 belong to three locations and none to four or more locations. These plant proteins were added to Swiss-Prot between 08-Mar-2011 and 18-Apr-2012. Because the plant dataset used for training the predictors was created on 29-Apr-2008, there is an almost 3-year time gap between the training data and test data in our experiments.

Table 3 compare the performance of HybridGO-Loc against several state-of-the-art multi-label plant predictors on the new plant dataset. All the predictors use the 978 proteins of the plant dataset (See Fig. 2(b)) for training the classifier and make independent test on the new 175 proteins. As can be seen, HybridGO-Loc performs significantly better than all the other predictors in terms of all of the performance metrics. Similar conclusions can also be drawn from the performance in individual subcellular locations.

Fig. 4 shows the distribution of the E-values of the test proteins, which were obtained by using the training proteins as the repository and the test proteins as the query proteins in the BLAST search. If we use a common criteria that homologous proteins should have E-value less than 10^{-4}, then 74 out of 175 test proteins are homologs of the training proteins, which account for 42% of the test set. Note that this homologous relationship does not mean that using BLAST's homology transfers can predict all of the 74 test proteins correctly. In fact, BLAST's homology transfers (based on the CC field of the homologous proteins) can only achieve a prediction accuracy of 26.9% (47/175). As the
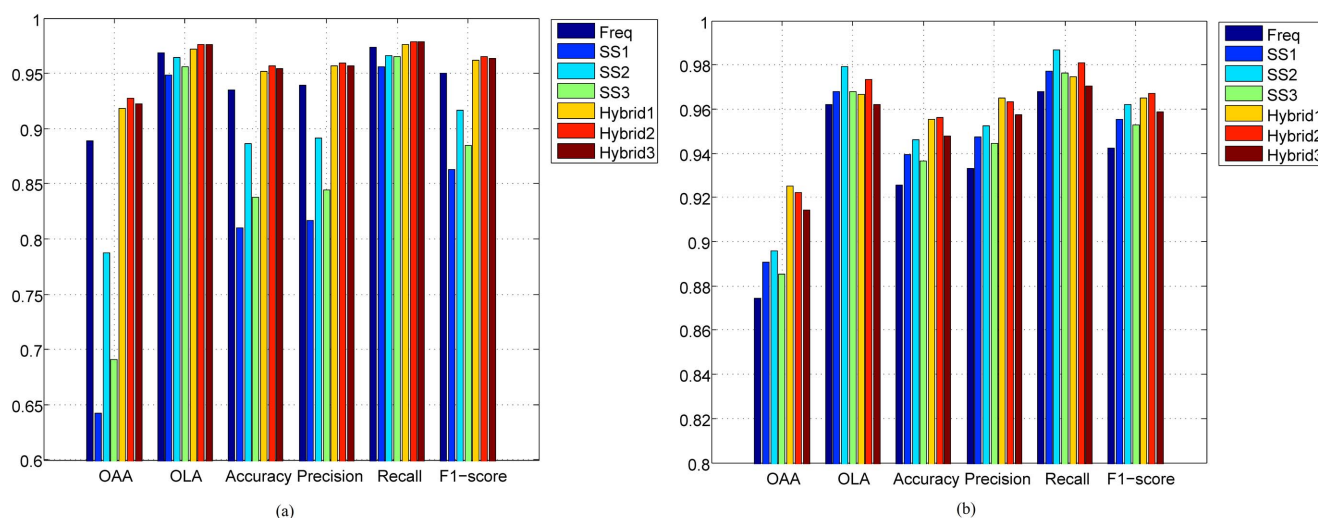
Figure 3. Performance of the hybrid features and individual features on the (a) virus and (b) plant datasets. *Freq*: GO frequency features; *SS1, SS2* and *SS3*: GO semantic similarity features by using Lin's measure [51], Jiang's measure [74] and RS measure [52], respectively; *Hybrid1, Hybrid2* and *Hybrid3*: GO hybrid features by combining GO frequency features with GO semantic similarity features based on *SS1, SS2* and *SS3*, respectively.
doi:10.1371/journal.pone.0089545.g003

Table 1. Comparing the proposed predictor with state-of-the-art multi-label predictors based on leave-one-out cross validation (LOOCV) using the virus dataset.

Label	Subcellular Location	LOOCV Locative Accuracy (LA)				
		Virus-mPLoc [35]	KNN-SVM [38]	iLoc-Virus [37]	mGOASVM [39]	HybridGO-Loc
1	Viral capsid	8/8 = 1.000	8/8 = 1.000	8/8 = 1.000	8/8 = 1.000	8/8 = 1.000
2	Host cell membrane	19/33 = 0.576	27/33 = 0.818	25/33 = 0.758	32/33 = 0.970	32/33 = 0.970
3	Host ER	13/20 = 0.650	15/20 = 0.750	15/20 = 0.750	17/20 = 0.850	18/20 = 0.900
4	Host cytoplasm	52/87 = 0.598	86/87 = 0.988	64/87 = 0.736	85/87 = 0.977	85/87 = 0.966
5	Host nucleus	51/84 = 0.607	54/84 = 0.651	70/84 = 0.833	82/84 = 0.976	82/84 = 0.988
6	Secreted	9/20 = 0.450	13/20 = 0.650	15/20 = 0.750	20/20 = 1.000	20/20 = 1.000
Overall Locative Accuracy (*OLA*)		152/252 = 0.603	203/252 = 0.807	197/252 = 0.782	244/252 = 0.968	245/252 = **0.972**
Overall Actual Accuracy (*OAA*)		–	–	155/207 = 0.748	184/207 = 0.889	194/207 = **0.937**
Accuracy		–	–	–	0.935	**0.961**
Precision		–	–	–	0.939	**0.965**
Recall		–	–	–	0.973	**0.976**
F1		–	–	–	0.950	**0.968**
HL		–	–	–	0.026	**0.016**

"–" means the corresponding references do not provide the results on the respective metrics. *Host ER*: Host endoplasmic reticulum.
doi:10.1371/journal.pone.0089545.t001

prediction accuracy of HybridGO-Loc on this test set (see Table 3) is significantly higher than this percentage, the extra information available from the GOA database plays a very important role in the prediction.

Discussion

Semantic Similarity Measures

In this paper, we have compared three of the most common semantic similarity measures for subcellular localization, including

Table 2. Comparing the proposed predictor with state-of-the-art multi-label predictors based on leave-one-out cross validation (LOOCV) using the plant dataset.

Label	Subcellular Location	LOOCV Locative Accuracy (LA)			
		Plant-mPLoc [34]	iLoc-Plant [36]	mGOASVM [39]	HybridGO-Loc
1	Cell membrane	24/56 = 0.429	39/56 = 0.696	53/56 = 0.946	51/56 = 0.911
2	Cell wall	8/32 = 0.250	19/32 = 0.594	27/32 = 0.844	28/32 = 0.875
3	Chloroplast	248/286 = 0.867	252/286 = 0.881	272/286 = 0.951	278/286 = 0.972
4	Cytoplasm	72/182 = 0.396	114/182 = 0.626	174/182 = 0.956	168/182 = 0.923
5	Endoplasmic reticulum	17/42 = 0.405	21/42 = 0.500	38/42 = 0.905	38/42 = 0.905
6	Extracellular	3/22 = 0.136	2/22 = 0.091	22/22 = 1.000	21/22 = 0.955
7	Golgi apparatus	6/21 = 0.286	16/21 = 0.762	19/21 = 0.905	19/21 = 0.905
8	Mitochondrion	114/150 = 0.760	112/150 = 0.747	150/150 = 1.000	149/150 = 0.993
9	Nucleus	136/152 = 0.895	140/152 = 0.921	151/152 = 0.993	150/152 = 0.987
10	Peroxisome	14/21 = 0.667	6/21 = 0.286	21/21 = 1.000	21/21 = 1.000
11	Plastid	4/39 = 0.103	7/39 = 0.179	39/39 = 1.000	38/39 = 0.974
12	Vacuole	26/52 = 0.500	28/52 = 0.538	49/52 = 0.942	48/52 = 0.923
Overall Locative Accuracy (*OLA*)		672/1055 = 0.637	756/1055 = 0.717	1015/1055 = **0.962**	1009/1055 = 0.956
Overall Actual Accuracy (*OAA*)		–	666/978 = 0.681	855/978 = 0.874	915/978 = **0.936**
Accuracy		–	–	0.926	**0.959**
Precision		–	–	0.933	**0.972**
Recall		–	–	**0.968**	**0.968**
F1		–	–	0.942	**0.966**
HL		–	–	0.013	**0.007**

"–" means the corresponding references do not provide the results on the respective metrics.
doi:10.1371/journal.pone.0089545.t002

Table 3. Comparing HybridGO-Loc with state-of-the-art multi-label plant predictors based on independent tests using the new plant dataset.

Label	Subcellular Location	Independent Test Locative Accuracy			
		Plant-mPLoc [34]	iLoc-Plant [36]	mGOASVM [39]	HybridGO-Loc
1	Cell membrane	8/16 = 0.500	1/16 = 0.063	7/16 = 0.438	16/16 = 1.000
2	Cell wall	0/1 = 0	0/1 = 0	0/1 = 0%	1/1 = 1.000
3	Chloroplast	27/54 = 0.500	45/54 = 0.833	39/54 = 0.722	30/54 = 0.556
4	Cytoplasm	5/38 = 0.132	15/38 = 0.395	19/38 = 0.500	31/38 = 0.816
5	Endoplasmic reticulum	1/9 = 0.111	1/9 = 0.111	3/9 = 0.333	4/9 = 0.444
6	Extracellular	0/3 = 0	0/3 = 0	1/3 = 0.333	0/3 = 0
7	Golgi apparatus	3/7 = 0.429	1/7 = 0.143	3/7 = 0.429	7/7 = 1.000
8	Mitochondrion	6/16 = 0.375	3/16 = 0.188	11/16 = 0.688	16/16 = 1.000
9	Nucleus	31/46 = 0.674	43/46 = 0.935	33/46 = 0.717	44/46 = 0.957
10	Peroxisome	4/6 = 0.667	0/6 = 0	3/6 = 0.500	4/6 = 0.667
11	Plastid	0/1 = 0	0/1 = 0	0/1 = 0	0/1 = 0
12	Vacuole	2/7 = 0.286	4/7 = 0.571	4/7 = 0.571	7/7 = 1.000
Overall Locative Accuracy (*OLA*)		87/204 = 0.427	113/204 = 0.554	123/204 = 0.603	160/204 = **0.784**
Overall Actual Accuracy (*OAA*)		60/175 = 0.343	91/175 = 0.520	97/175 = 0.554	127/175 = **0.726**
Accuracy		0.417	0.574	0.594	**0.784**
Precision		0.444	0.626	0.630	**0.826**
Recall		0.474	0.577	0.609	**0.798**
F1		0.444	0.592	0.611	**0.803**
HL		0.116	0.076	0.075	**0.037**

Lin's measure [51], Jiang's measure [74], and relevance similarity measure [52]. We excluded Resnik's measure because it ignores the distance between the terms and their common ancestors in the GO hierarchy. In addition to these measures, many online tools are also available for computing the semantic similarity at the GO-term level and gene-product level [44,82–84]. However, these measures are discrete measures whereas the measures that we used are continuous. Research has shown that continuous measures are better than discrete measures in many applications [48].

GO-Frequency Features versus SS Features

Note that we do not replace the GO frequency vectors. Instead, we augment the GO frequency feature with a more sophisticated feature, i.e. the GO SS vectors, which are to be combined with the GO frequency vectors. A GO frequency vector is found by counting the number of occurrences of every GO term in a set of distinct GO terms obtained from the training dataset, whereas an SS vector is constructed by computing the semantic similarity between a test protein with each of the training proteins at the gene-product level. That is, each element in an SS vector represents the semantic similarity of two GO-term groups. This can be easily seen from their definitions in Eq. 2 and Eq. 4–9, respectively.

The GO frequency vectors and the GO SS vectors are different in two fundamental ways.

- A). GO frequency vectors are more *primitive* in the sense that their elements are based on individual GO terms without considering the inter-term relationship, i.e., the elements in a GO frequency vectors are independent of each other.

- B). GO SS vectors are more *sophisticated* in the following two senses.

- B1) *Inter-term relationship*. SS vectors are based on inter-term relationships. They are defined on a space in which each basis corresponds to one training protein and the coordinate along that basis is defined by the semantic similarity between a testing protein and the corresponding training protein.

- B2) *Inter-group relationship*. The pairwise relationships between a test protein and the training proteins are hierarchically structured. This is because each basis of the SS space depends on a group of GO terms of the corresponding training protein, and the terms are arranged in a hierarchical structure (parent- child relationship). Because the GO terms in different groups are not mutually exclusive, the bases in the SS space are not independent of each other.

Bias Analysis

Except for the new plant dataset, we adopted LOOCV to examine the performance of all predictors in this work, which is considered to be the most rigorous and bias-free [80]. Nevertheless, determining the set of distinct GO terms $\mathbb{W}$ from a dataset is by no means without bias, which may favor the LOOCV performance. This is because the set of distinct GO terms $\mathbb{W}$ derived from a given dataset may not be representative for other datasets; in other words, the generalization capabilities of the predictors may be weakened when new GO terms outside $\mathbb{W}$ are found in the test proteins.

However, we have the following strategies to minimize the bias. First, the two benchmark datasets used in this paper were constructed based on the whole Swiss-Prot database (although in different years), which, to some extent, incorporated all the
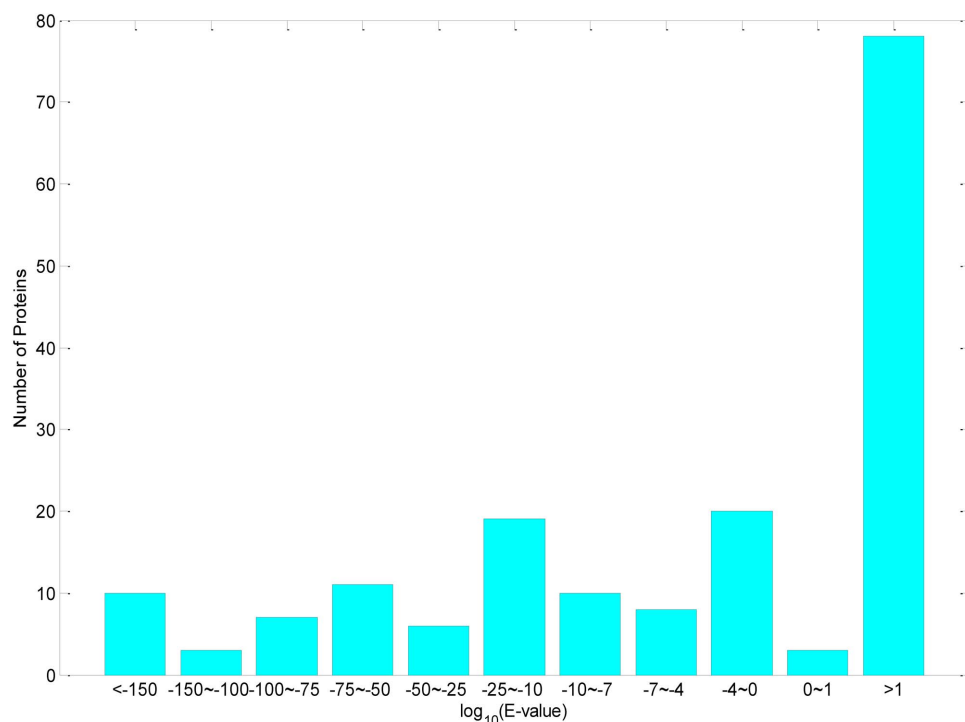
Figure 4. Distribution of the closeness between the new testing proteins and the training proteins. The *closeness* is defined as the BLAST E-values of the training proteins using the test proteins as the query proteins in the BLAST searches. *Number of Proteins*: The number of testing proteins whose E-values fall into the interval specified under the bar. Small E-values suggest that the corresponding new proteins are close homologs of the training proteins.
doi:10.1371/journal.pone.0089545.g004

possible information of plant proteins or virus proteins in the database. In other words, $\mathbb{W}$ was constructed based on all of the GO terms corresponding to the whole Swiss-Prot database, which enables $\mathbb{W}$ to be representative for all of the distinct GO terms. Second, these two benchmark datasets were collected according to strict criteria. Details of the procedures can be found in the supplementary materials. and the sequence similarity of both datasets was cut off at 25%, which enables us to use a small set of representative proteins to represent all of the proteins of the corresponding species (i.e., virus or plant) in the whole database. In other words, $\mathbb{W}$ will vary from species to species, yet still be statistically representative for all of the useful GO terms for the corresponding species. Third, using $\mathbb{W}$ for statistical performance evaluation is equivalent or at least approximate to using all of the distinct GO terms in the GOA database. This is because other GO terms that do not correspond to the training proteins will not participate in training the linear SVMs, nor will they play essential roles in contributing to the final predictions. In other words, the generalization capabilities of HybridGO-Loc will not be weakened even if some new GO terms are found in the test proteins. A mathematical proof of this statement can be found in the supplementary materials available in the HybridGO-Loc server.

One may argue that the performance bias might arise when the whole $\mathbb{W}$ was used to construct the hybrid GO vectors for both training and testing during cross validation. This is because, in each fold of the LOOCV, the training proteins and the singled-out test protein will use the same $\mathbb{W}$ to construct the GO vectors, meaning that the SVM training algorithm can *see* some information of the test protein indirectly through the GO vector space defined by $\mathbb{W}$. It is possible that for a particular fold of LOOCV, the GO terms of a test protein do not exist in any of the training proteins. However, we have mathematically proved that

this bias will not exist during LOOCV (see the accompanying supplementary materials for the proof). Furthermore, the results of the independent tests (See Table 3) for which no such bias occurs also strongly suggest that HybridGO-Loc outperforms other predictors by a large margin.

Conclusions

This paper proposes a new multi-label predictor by hybridizing GO frequency features and semantic similarity features to predict the subcellular locations of multi-label proteins. Three different semantic similarity measures have been investigated to be combined with GO frequency features to formulate GO hybrid feature vectors. The feature vectors are subsequently recognized by multi-label multi-class support vectors machine (SVM) classifiers equipped with an adaptive decision strategy that can produce multiple class labels for a query protein. Compared to existing multi-label subcellular-localization predictors, our proposed predictor has the following advantages: (1) it formulates the feature vectors by hybridizing GO frequency of occurrences and GO semantic similarity features which contains richer information than only GO term frequencies; (2) it adopts a new strategy to incorporate richer and more useful homologous information from more distant homologs rather than using the top homologs only; (3) it adopts an adaptive decision strategy for multi-label SVM classifiers so that it can effectively deal with datasets containing both single-label and multi-label proteins. Experimental results demonstrate the superiority of the proposed hybrid features over each individual features. It was also found that the proposed predictor performs remarkably better than existing state-of-the-art predictors. For readers' convenience, HybridGO-Loc is available online at http://bioinfo.eie.polyu.edu.hk/HybridGoServer/.

Author Contributions

Conceived and designed the experiments: SW MM. Performed the experiments: SW MM. Analyzed the data: SW MM SK. Contributed reagents/materials/analysis tools: SW MM. Wrote the paper: SW MM SK.

References

1. Butler GS, Overall CM (2009) Proteomic identification of multitasking proteins in unexpected locations complicates drug targeting. Nat Rev Drug Discov 8: 935–948.
2. Kaytor MD, Warren ST (1999) Aberrant Protein Deposition and Neurological Disease. J Biol Chem 274: 37507–37510.
3. Hung MC, Link W (2011) Protein localization in disease and therapy. J of Cell Sci 124: 3381–3392.
4. Krutovskikh V, Mazzoleni G, Mironov N, Omori Y, Aguelon AM, et al. (1994) Altered homologous and heterologous gap-junctional intercellular communication in primary human liver tumors associated with aberrant protein localization but not gene mutation of connexin 32. Int J Cancer 56: 87–94.
5. Chen Y, Chen CF, Riley DJ, Allred DC, Chen PL, et al. (1995) Aberrant Subcellular Localization of BRCA1 in Breast Cancer. Science 270: 789–791.
6. Lee X, Keith JCJ, Stumm N, Moutsatsos I, McCoy JM, et al. (2001) Downregulation of placental syncytin expression and abnormal protein localization in pre-eclampsia. Placenta 22: 808–812.
7. Hayama A, Rai T, Sasaki S, Uchida S (2003) Molecular mechanisms of Bartter syndrome caused by mutations in the BSND gene. Histochem & Cell Biol 119: 485–493.
8. Chou KC, Cai YD (2005) Predicting protein localization in budding yeast. Bioinformatics 21: 944–950.
9. Lubec G, Afjehi-Sadat L, Yang JW, John JP (2005) Searching for hypothetical proteins: Theory and practice based upon original data and literature. Prog Neurobiol 77: 90–127.
10. Nakashima H, Nishikawa K (1994) Discrimination of intracellular and extracellular proteins using amino acid composition and residue-pair frequencies. J Mol Biol 238: 54–61.
11. Chou KC (2001) Prediction of protein cellular attributes using pseudo amino acid composition. Proteins: Structure, Function, and Genetics 43: 246–255.
12. Lu Z, Szafron D, Greiner R, Lu P, Wishart DS, et al. (2004) Predicting subcellular localization of proteins using machine-learned classifiers. Bioinformatics 20: 547–556.
13. Mak MW, Guo J, Kung SY (2008) PairProSVM: Protein subcellular localization based on local pairwise profile alignment and SVM. IEEE/ACM Trans on Computational Biology and Bioinfor-matics 5: 416–422.
14. Emanuelsson 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.
15. Nielsen H, Engelbrecht J, Brunak S, von Heijne G (1997) A neural network method for identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. Int J Neural Sys 8: 581–599.
16. Wan S, Mak MW, Kung SY (2011) Protein subcellular localization prediction based on profile alignment and Gene Ontology. In: 2011 IEEE International Workshop on Machine Learning for Signal Processing (MLSP'11). pp. 1–6.
17. Mei S (2012) Multi-label multi-kernel transfer learning for human protein subcellular localization. PLoS ONE 7: e37716.
18. Mei SY, Fei W, Zhou SG (2011) Gene ontology based transfer learning for protein subcellular localization. BMC Bioinformatics 12: 44.
19. Yang Y, Lu B (2010) Protein subcellular multi-localization prediction using a min-max modular support vector machine. International Journal of Neural Systems 20: 13–28.
20. Liu L, Zhang Z, Mei Q, Chen M (2013) PSI: A comprehensive and integrative approach for accurate plant subcellular localization prediction. PLoS ONE 8: e75826.
21. Wan S, Mak MW, Kung SY (2013) Adaptive thresholding for multi-label SVM classification with application to protein subcellular localization prediction. In: 2013 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'13). pp. 3547–3551.
22. Lu Z, Szafron D, Greiner R, Lu P, Wishart DS, et al. (2004) Predicting subcellular localization of proteins using machine-learned classifiers. Bioinformatics 20: 547–556.
23. Brady S, Shatkay H (2008) EpiLoc: a (working) text-based system for predicting protein subcellular location. In: Pac. Symp. Biocomput. pp. 604–615.
24. Fyshe A, Liu Y, Szafron D, Greiner R, Lu P (2008) Improving subcellular localization prediction using text classification and the gene ontology. Bioinformatics 24: 2512–2517.
25. Chou KC, Shen HB (2006) Predicting eukaryotic protein subcellular location by fusing optimized evidence-theoretic K-nearest neighbor classifiers. J of Proteome Research 5: 1888–1897.
26. Huang WL, Tung CW, Ho SW, Hwang SF, Ho SY (2008) ProLoc-GO: Utilizing informative Gene Ontology terms for sequence-based prediction of protein subcellular localization. BMC Bioinfor-matics 9: 80.
27. Chi SM, Nam D (2012) Wegoloc: accurate prediction of protein subcellular localization using weighted gene ontology terms. Bioinformatics 28: 1028–1030.
28. Wan S, Mak MW, Kung SY (2013) GOASVM: A subcellular location predictor by incorporating term-frequency gene ontology into the general form of Chou's pseudo-amino acid composition. Journal of Theoretical Biology 323: 40–48.
29. Winston M, Chaffin R, Herrmann D (1987) A taxonomy of part-whole relations. Cognitive Science 11: 417–444.
30. Odell J (1998) Six different kinds of aggregation. In: Advanced object-oriented analysis and design using UML, Cambridge University Press. pp. 139–149.
31. The Gene Ontology Consortium (2012) The Gene Ontology: enhancements for 2011. Nucleic Acids Res 40: D559–D564.
32. The Gene Ontology Consortium (2010) The Gene Ontology in 2010: extensions and refinements. Nucleic Acids Res 38: D331–D335.
33. Mueller JC, Andreoli C, Prokisch H, Meitinger T (2004) Mechanisms for multiple intracellular localization of human mitochondrial proteins. Mitochondrion 3: 315–325.
34. 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.
35. Shen HB, Chou KC (2010) Virus-mPLoc: A fusion classifier for viral protein subcellular location prediction by incorporating multiple sites. J Biomol Struct Dyn 26: 175–186.
36. Wu ZC, Xiao X, Chou KC (2011) iLoc-Plant: A multi-label classifier for predicting the subcellular localization of plant proteins with both single and multiple sites. Molecular BioSystems 7: 3287–3297.
37. Xiao X, Wu ZC, Chou KC (2011) iLoc-Virus: A multi-label learning classifier for identifying the subcellular localization of virus proteins with both single and multiple sites. Journal of Theoretical Biology 284: 42–51.
38. Li LQ, Zhang Y, Zou LY, Zhou Y, Zheng XQ (2012) Prediction of protein subcellular multi-localization based on the general form of Chou's pseudo amino acid composition. Protein & Peptide Letters 19: 375–387.
39. Wan S, Mak MW, Kung SY (2012) mGOASVM: Multi-label protein subcellular localization based on gene ontology and support vector machines. BMC Bioinformatics 13: 290.
40. He J, Gu H, Liu W (2011) Imbalanced multi-modal multi-label learning for subcellular localization prediction of human proteins with both single and multiple sites. PLoS ONE 7: e37155.
41. Li LQ, Zhang Y, Zou LY, Li CQ, Yu B, et al. (2012) An ensemble classifier for eukaryotic protein subcellular location prediction using Gene Ontology categories and amino acid hydrophobicity. PLoS ONE 7: e31057.
42. Zhu M, Gao L, Guo Z, Li Y, Wang D, et al. (2007) Globally predicting protein functions based on co-expressed protein-protein interaction networks and ontology taxonomy similarities. Gene 391: 113–119.
43. Pesquita C, Faria D, Falcao AO, Lord P, Couto FM (2008) Metrics for GO based protein semantic similarity: a systematic evaluation. BMC Bioinformatics 9: S4.
44. Lei Z, Dai Y (2006) Assessing protein similarity with Gene Ontology and its use in subnuclear localization prediction. BMC Bioinformatics 7: 491.
45. Wu X, Zhu L, Guo J, Zhang DY, Lin K (2006) Prediction of yeast protein-protein interaction network: insights from the gene ontology and annotations. Nucleic Acids Res 34: 2137–2150.
46. Guo X, Liu R, Shriver CD, Hu H, Liebman MN (2006) Assessing semantic similarity measures for the characterization of human regulatory pathways. Bioinformatics 22: 967–973.
47. Xu T, Du L, Zhou Y (2008) Evaluation of GO-based functional similarity measures using S. cerevisiae protein interaction and expression profile data. BMC Bioinformatics 9: 472.
48. Yang D, Li Y, Xiao H, Liu Q, Zhang M, et al. (2008) Gaining confidence in biological interpretation of the microarray data: the functional consistence of the significant GO categories. Bioinformatics 24: 265–271.
49. Resnik P (1999) Semantic similarity in a taxonomy: an information-based measure and its application to problems of ambiguity in natural language. Journal of Artificial Intelligence Research 11: 95–130.
50. Lord PW, Stevens RD, Brass A, Goble CA (2003) Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation. Bioinformatics 19: 1275–1283.
51. Lin D (1998) An information-theoretic definition of similarity. In: Proceedings of the 15th Inter-national Conference on Machine Learning. pp. 296–304.
52. Schlicker A, Domingues FS, Rahnenfuhrer J, Lengauer T (2006) A new measure for functional similarity of gene products based on Gene Ontology. BMC Bioinformatics 7: 302.
53. Couto FM, Silva MJ, Coutinho PM (2005) Semantic similarity over the gene ontology: Family correlation and selecting disjunctive ancestors. In: Proceedings of 14-th International ACM Con-ference in Information and Knowledge Management. pp. 343–344.
54. Bodenreider O, Aubry M, Burgun A (2005) Non-lexical approaches to identifying associative rela-tions in the gene ontology. In: Pac. Symp. Biocomput. pp. 91–102.

55. Pozo AD, Pazos F, Valencia A (2008) Defining functional distances over gene ontology. BMC Bioinformatics 9: 50.

56. Wu H, Su Z, Mao F, Olman V, Xu Y (2005) Prediction of functional modules based on comparative genome analysis and gene ontology application. Nucleic Acids Res 33: 2822–2837.

57. Cheng J, Cline M, Martin J, Finkelstein D, Awad T, et al. (2004) A knowledge-based clustering algorithm driven by gene ontology. Journal of Biopharmaceutical Statistics 14: 687–700.

58. Yu H, Gao L, Tu K, Guo Z (2005) Broadly predicting specific gene function with expression similarity and taxonomy similarity. Gene 352: 75–81.

59. Sevilla JL, Segura V, Podhorski A, Guruceaga E, Mato JM, et al. (2005) Correlation between gene expression and GO semantic similarity. IEEE/ACM Transactions on Computational Biology and Bioinformatics 2: 330–338.

60. Tao Y, Sam L, Li J, Friedman C, Lussier YA (2007) Information theory applied to the sparse gene ontology annotation network to predict novel gene function. Bioinformatics 23: i529–i538.

61. Wang JZ, Du Z, Payattakool R, Yu PS, Chen CF (2007) A new method to measure the semantic similarity of GO terms. Bioinformatics 23: 1274–1281.

62. Riensche RM, Baddeley BL, Sanfilippo AP, Posse C, Gopalan B (2007) XOA: Web-enabled cross-ontological analytics. In: 2007 IEEE Congress on Services. pp. 99–105.

63. Wan S, Mak MW, Kung SY (2013) Semantic similarity over gene ontology for multi-label protein subcellular localization. Engineering 5: 68–72.

64. Huang DW, Sherman BT, Tan Q, Collins JR, Alvord WG, et al. (2007) The DAVID Gene Func-tional Classification Tool: a novel biological module-centric algorithm to functionally analyze large gene lists. Genome Biology 8.

65. Chabalier J, Mosser J, Burgun A (2007) A trasversal approach to predict gene product networks from ontology-based similarity. BMC Bioinformatics 8: 235.

66. Mistry M, Pavlidis P (2008) Gene ontology term overlap as a measure of gene functional similarity. BMC Bioinformatics 9: 327.

67. Sheehan B, Quigley A, Gaudin B, Dobson S (2008) A relation based measure of semantic similarity for Gene Ontology annotations. BMC Bioinformatics 9: 468.

68. Pesquita C, Faria D, Falcao AO, Lord P, Counto FM (2009) Semantic similarity in biomedical ontologies. PLoS Computational Biology 5: e1000443.

69. Guzzi PH, Mina M, Guerra C, Cannataro M (2012) Semantic similarity analysis of protein data: assessment with biological features and issues. Brief Bioinform 13: 569–585.

70. Wang X, Li GZ (2012) A multi-label predictor for identifying the subcellular locations of singleplex and multiplex eukaryotic proteins. PLoS ONE 7: e36317.

71. Chou KC (2013) Some remarks on predicting multi-label attributes in molecular biosystems. Molecular BioSystems 9: 1092–1100.

72. Chou KC, Shen HB (2008) Cell-PLoc: A package of web-servers for predicting subcellular localization of proteins in various organisms. Nature Protocols 3: 153–162.

73. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, et al. (1997) Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. Nucleic Acids Res 25: 3389–3402.

74. Jiang JJ, Conrath DW (1997) Semantic similarity based on corpus statistics and lexical taxonomy. In: Proceedings of International Conference Research on Computational Linguistics (ROCLING X). pp. 19–33.

75. Nair R, Rost B (2008) Protein subcellular localization prediction using artificial intelligence tech-nology. In: Functional Proteomics, Springer. pp. 435–463. Available: http://link.springer.com/protocol/10.1007/978-1-59745-398-1_27.

76. Adelfio A, Volpato V, Pollastri G (2013) SCLpredT: Ab initio and homology-based prediction of subcellular localization by N-to-1 neural networks. SpringerPlus 2: 1–11.

77. Dembczynski K, Waegeman W, Cheng W, Hullermeier E (2012) On label dependence and loss minimization in multi-label classification. Machine Learning 88: 5–45.

78. Gao W, Zhou ZH (2013) On the consistency of multi-label learning. Artificial Intelligence 199-200: 22–44.

79. Chou KC (1995) Review: Prediction of protein structural classes. Critical Reviews in Biochemistry and Molecular Biology 30: 275–349.

80. Hastie T, Tibshirani R, Friedman J (2001) The element of statistical learning. Springer-Verlag.

81. Abdi H, Williams LJ (2010) Jackknife. In: Encyclopedia of Research Design, Sage. pp. 655–660.

82. Pesquita C, Pessoa D, Faria D, Couto F (2009) CESSM: Collaborative evaluation of semantic similarity measures. JB2009: Challenges in Bioinformatics 157.

83. Faria D, Pesquita C, Couto FM, Falcão A (2007) ProteInOn: A web tool for protein semantic similarity. Available: http://docs.di.fc.ul.pt/handle/10455/3035.

84. Yu G, Li F, Qin Y, Bo X, Wu Y, et al. (2010) GOSemSim: An R package for measuring semantic similarity among GO terms and gene products. Bioinformatics 26: 976–978.