# GOASVM: A Subcellular Location Predictor by Incorporating Term-Frequency Gene Ontology into the General Form of Chou's Pseudo Amino Acid Composition

Shibiao Wan[a], Man-Wai Mak[a,*], Sun-Yuan Kung[b]

[a]Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong SAR, China
[b]Department of Electrical Engineering, Princeton University, New Jersey, USA.

## Abstract

Prediction of protein subcellular localization is an important yet challenging problem. Recently, several computational methods based on Gene Ontology (GO) have been proposed to tackle this problem and have demonstrated superiority over methods based on other features. Existing GO-based methods, however, do not fully use the GO information. This paper proposes an efficient GO method called GOASVM that exploits the information from the GO term frequencies and distant homologs to represent a protein in the general form of Chou's pseudo amino acid composition. The method first selects a subset of relevant GO terms to form a GO vector space. Then for each protein, the method uses the accession number (AC) of the protein or the ACs of its homologs to find the number of occurrences of the selected GO terms in the Gene Ontology annotation (GOA) database as a means to construct GO vectors for support vector machines (SVMs) classification. With the advantages of GO term frequencies and a new strategy to incorporate useful homologous information, GOASVM can achieve a prediction accuracy of 72.2% on a new independent test set comprising novel proteins that were added to Swiss-Prot six years later than the creation date of the training set. GOASVM and Supplementary Materials are available online at `http://bioinfo.eie.polyu.edu.hk/mGoaSvmServer/GOASVM.html`.

*Keywords:* Protein Subcellular Localization; Gene Ontology; GO terms; Term Frequency; Support vector machines.

## 1. Introduction

As an essential and indispensable topic in proteomics research and molecular cell biology, protein subcellular localization is critically important for protein function annotation, drug target discovery, and drug design (Chou and Shen, 2007b; Lubec et al., 2005). To tackle the exponentially growing number of newly found protein sequences in the post-genomic era, many efficient and reliable computational methods have been developed to replace or assist the biological experiments such as fluorescent microscopy imaging. Recent years have witnessed an incredibly fast development in protein subcellular localization prediction.

Prediction of subcellular localization can be roughly divided into sequence-based and Gene Ontology (GO) based. The former only uses the amino-acid sequences of query proteins as input. They can be further classified into three groups: (1) composition-based methods, (2) sorting-signal based methods and (3) homology-based methods.

Composition-based methods use the relationship between subcellular locations and the composition information embedded in the amino acid sequences, such as amino-acid compositions (AA) (Nakashima and Nishikawa, 1994; Chou and Cai, 2005), amino-acid pair compositions (PairAA) (Nakashima and Nishikawa, 1994), gapped amino-acid pair compositions (GapAA) (Park and Kanehisa, 2003; Lee et al., 2006), and pseudo amino-acid composition (PseAA) (Chou, 2001; Chou and Shen, 2006b; Chou and Cai, 2003, 2004, 2005). Sorting-signal based methods – such as PSORT (Nakai and Kanehisa, 1991), WoLF PSORT (Horton et al., 2006), and TargetP (Emanuelsson et al., 2000) – predict the localization via the recognition of N-terminal sorting signals in amino acid sequences. Homology-based methods – such as Proteome Analyst (Lu et al., 2004), PairProSVM (Mak et al., 2008), and other predictors (Nair and Rost, 2002; Mott et al., 2002; Scott et al., 2004) – rely on the fact that homologous sequences are more likely to reside in the same subcellular location. Sequence-based methods are general in that they can be applied to any newly discovered proteins. However, their performance is usually poor, especially for datasets containing sequences with low-similarity. Annotation-based methods, on the contrary, are superior.

GO-based methods make use of the well-organized biological knowledge about genes and gene products in the GO databases. The GO-based methods can be viewed from the following two perspectives:

1. **GO-Terms Extraction.** There are three methods to extract GO terms from annotation databases. The first method uses a program called InterProScan (Zdobnov and Apweiler, 2001) to search against a set of protein signature databases to look for relevant GO terms (Chou and Cai, 2004; Blum et al., 2009; Wan et al., 2011; Mei et al., 2011). This method can be applied to all protein sequences; but usually they can re-

*Corresponding author

*Email addresses:* `10900600r@connect.polyu.hk` (Shibiao Wan), `enmwmak@polyu.edu.hk` (Man-Wai Mak), `kung@princeton.edu` (Sun-Yuan Kung)

trieve only a small number of GO terms, which may not be sufficient for accurate prediction of subcellular localization. The second method uses the accession numbers (ACs) of proteins to search against the Gene Ontology Annotation (GOA) database[1] to retrieve GO terms. Typical predictors using this approach include Euk-OET-PLoc (Chou and Shen, 2006c), Hum-PLoc (Chou and Shen, 2006a), Euk-mPLoc (Chou and Shen, 2007a) and Gneg-PLoc (Chou and Shen, 2006b). These predictors perform better than the ones based on InterProScan, but they are not applicable to proteins that have not been functionally annotated. The third method uses BLAST (Altschul et al., 1997) to obtain the ACs of homologs of the query proteins and then uses these ACs to search against the GOA database. Typical predictors include ProLoc-GO (Huang et al., 2008), iLoc-Virus (Xiao et al., 2011b), and Cell-PLoc 2.0 (Chou and Shen, 2010a). This method is applicable to all protein sequences and is able to retrieve more GO terms, which are essential for good prediction performance.

2. **GO-Vector Construction.** From this perspective, GO-based methods can be classified into three categories. The first category considers each GO term as a canonical basis of a Euclidean space where the coordinates can be equal to either 0 or 1. Representative methods include Euk-OET-PLoc (Chou and Shen, 2006c), Hum-PLoc (Chou and Shen, 2006a), Gneg-PLoc (Chou and Shen, 2006b) and Gpos-PLoc (Shen and Chou, 2007). Recently, a modified binary feature vector construction method is proposed to deal with many sets of GO terms for one protein (Chou and Shen, 2010a,b). This category provides a large coverage of GO terms, but it could introduce many irrelevant GO terms. The second category uses genetic algorithms to select the most informative GO terms, such as ProLoc-GO (Huang et al., 2008) and PGAC (Huang et al., 2009). One problem of this type of methods is that it may select only a small number of GO terms, increasing the chance of having a null GO vector for a test protein. The third method designs an implicit kernel function to measure the semantic similarity between two GO terms (Lei and Dai, 2006).

This paper proposes a GO-based method called GOASVM, which is based on protein homology, gene ontology, and support vector machines. GOASVM is different from other GO-based predictors in that (1) it constructs the GO vectors by using the frequency of occurrences of GO terms instead of using 1-0 values for indicating the presence or absence of some predefined GO terms; (2) it adopts a new strategy to incorporate richer and more useful information from more distant homologs instead of using only the top homologs; and (3) it constructs a GO vector subspace from the full GO vector space by selecting a set of relevant GO terms. In addition to these algorithmic perspectives, our work is different from previous works in that the training and testing sets used in our experiments are six years apart, whereas in other studies, the training and testing sets were created at the same time. This long time-separation between the training and testing sets ensures that the accuracy achieved by GOASVM is unbiased. This paper also investigates how the updated information in the GOA database affects the prediction performance of GO-based methods and in turn demonstrates the superiority of GOASVM over other GO-based methods. Experiments on a new eukaryotic dataset comprising novel proteins demonstrate the practicality and effectiveness of our proposed predictor.

GOASVM is designed for predicting single-label eukaryotic or human proteins. Actually, there are many papers (Chou and Shen, 2006c; Wang et al., 2010; Mak et al., 2008; Emanuelsson et al., 2000) focusing on single-label protein subcellular localization. It is well known that most proteins stay only at one subcellular location (Hu et al., 2012). Therefore, predicting the subcellular localization of single-label proteins is of great significance. For how to tackle proteins with both single and multiple location sites, see iLoc-Plant (Wu et al., 2011), iLoc-Hum (Chou et al., 2012), iLoc-Gpos (Wu et al., 2012), iLoc-Euk (Chou et al., 2011), iLoc-Gneg (Xiao et al., 2011a), and iLoc-Virus (Xiao et al., 2011b), as well as Cell-PLoc (Chou and Shen, 2008).

## 2. Gene Ontology for Subcellular Localization

As a result of the Gene Ontology (GO)[2] Consortium annotation effort, the GOA database has become a large and comprehensive resource for proteomics research (Camon et al., 2003). The database provides structured annotations to non-redundant proteins from many species in UniProt Knowledgebase (UniProtKB) (Apweiler et al., 2004) using standardized GO vocabularies through a combination of electronic and manual techniques. The large-scale assignment of GO terms to UniProtKB entries (or ACs) was done by converting a proportion of the existing knowledge held within the UniProKB database into GO terms (Camon et al., 2003). The GOA database also includes a series of cross-references to other databases. Thus, the systematic integration of GO annotations and UniProtKB database can be exploited for subcellular localization. Specifically, given the accession number of a protein, a set of GO terms can be retrieved from the GOA database file.[3] In UniProKB, each protein has a unique accession number (AC), and in the GOA database, each AC may be associated with zero, one or more GO terms. Conversely, one GO term may be associated with zero, one, or many different ACs. This means that the mappings between ACs and GO terms are many-to-many.

For those who are skeptical about the GO-based prediction methods, the following question is prone to be raised: If a protein has already been annotated by cellular component GO terms, is it still necessary to predict its subcellular localization? This sounds like a legitimate question because the GO terms already suggest the subcellular localization and therefore it is merely a procedure of converting the annotation into another format. In other words, all we need is to create a lookup table (hash table) using the cellular component GO terms as the

[1] http://www.ebi.ac.uk/GOA

[2] http://www.geneontology.org

[3] ftp://ftp.ebi.ac.uk/pub/databases/GO/goa/UNIPROT/

keys and the component categories as the hashed values. To answer this question, let us provide some facts here. Most of the existing 'non-GO predictors' were established based on the proteins in the Swiss-Prot database in which the subcellular locations are experimentally determined. Is it logical to consider that all of these methods have nothing to predict? Obviously, it is not. Fairly speaking, as long as the input is a query protein sequence and the output is its subcellular location(s), the predictor is deemed to be a valid protein subcellular-location predictor. In fact, most of the existing GO predictors, such as iLoc-Euk (Chou et al., 2011) and iLoc-Hum (Chou et al., 2012), use protein sequence information only to predict the subcellular locations, without adding any GO information to the input. That is to say, these GO predictors use the same input as the non-GO predictors. Therefore, GO-based predictors should also be regarded as valid predictors. According to a Nature Protocols paper (Chou and Shen, 2008), the good performance of GO-based methods is due to the fact that the features vectors in the GO space can better reflect their subcellular locations than those in the Euclidean space or any other simple geometric space.

Here, we explain why the simple table-lookup method mentioned above is undesirable. Although the cellular component ontology is directly related to the subcellular localization, we cannot simply use its GO terms to determine the subcellular locations of proteins. The reason is that some proteins do not have cellular component GO terms. Even for proteins annotated with cellular-component GO terms, it is inappropriate to use these terms only to determine their subcellular localizations. The reason is that a protein could have multiple cellular-component GO terms that map to different subcellular localizations. Another reason is that, according to Chou and Shen (2006a), proteins with annotated subcellular localization in Swiss-Prot may still be marked as 'Cellular Component Unknown' in the GO database. Because of this limitation, it is necessary to use the other two ontologies as well because they are also relevant (although not directly) to the subcellular localization of proteins. The problem of table-lookup is further exemplified in Appendix A.

## 3. Methods

According to a recent comprehensive review (Chou, 2011), the establishment of a statistical protein predictor involves the following five steps: (i) construction of a valid dataset for training and testing the predictor; (ii) formulation of effective mathematical expressions for converting proteins' characteristics to feature vectors that are relevant to the prediction task; (iii) development of classification algorithm for discriminating the feature vectors; (iv) evaluation of cross-validation tests for measuring the performance of the predictor; and (v) deployment of a user-friendly, publicly accessible web-server for other researchers to use and validate the prediction method. These steps are further elaborated below.

The GOASVM predictor uses either accession numbers (ACs) or amino acid (AA) sequences as input. The prediction process is divided into two stages: feature extraction (vectorization) and pattern classification. For the former, the query proteins are "vectorized" to high-dim GO vectors. For the latter, the GO vectors are classified by one-vs-rest linear SVMs.

### *3.1. Retrieval of GO Terms*

Given a query protein, GOASVM can handle two possible cases: (1) the AC is known and (2) the AA sequence is known. For proteins with known ACs, their respective GO terms are retrieved from the GOA database using the ACs as the searching keys. For a protein without an AC, its AA sequence is presented to BLAST (Altschul et al., 1997) to find its 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, the GO vectors constructed in Section 3.2 will contain all-zero, which are meaningless for further classification. 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 (Barrel et al., 2009), it is reasonable to assume that the homologs of the query proteins have at least one GO term (Mei, 2012). 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.

### *3.2. Construction of GO Vectors*

According to Eq. 6 of Chou (2011), the characteristics of any proteins can be represented by the general form of Chou's pseudo amino acid composition (Chou, 2001, 2005):

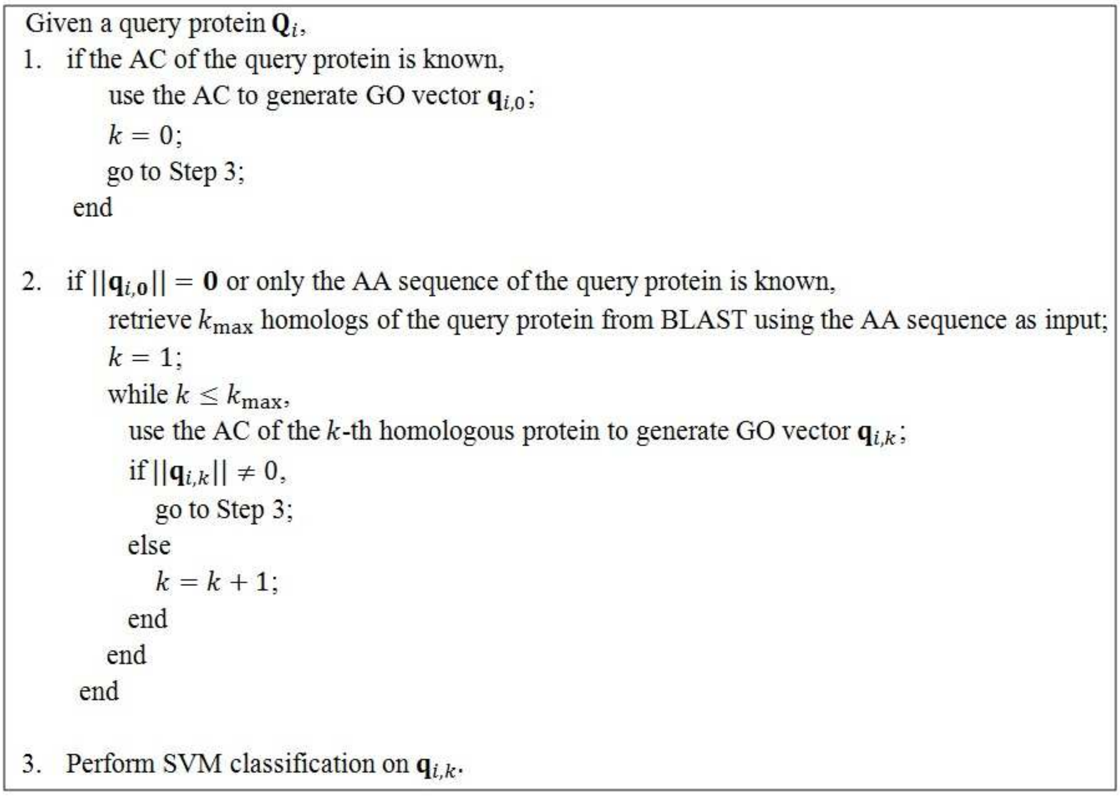
$$\mathbf{p}_i = [\phi_{i,1}, \ldots, \phi_{i,u}, \ldots, \phi_{i,\Omega}]^{\mathsf{T}}, \qquad (1)$$

where $\mathsf{T}$ is a transpose operator, $\Omega$ is a number representing the dimension of the feature vector $\mathbf{p}_i$, and the definitions of the $\Omega$ feature components $\phi_{i,u}$ ($u = 1, \ldots, \Omega$) depend on the feature extraction approaches elaborated below.

Given a dataset, we used the procedure described in Section 3.1 to retrieve the GO terms of all of its proteins. Then, we determined the number of distinct GO terms corresponding to the dataset. Suppose $\Omega$ distinct GO terms were found; these GO terms form a GO Euclidean space with $\Omega$ dimensions. For each protein in the dataset, we constructed a GO vector by matching its GO terms to all of the $\Omega$ GO terms. We have investigated four approaches to determining the elements of the GO vectors.

1. **1-0 value.** In this approach, each of the $\Omega$ GO terms represents one canonical basis of a Euclidean space, and a protein is represented by a point in this space with coordinates equal to either 0 or 1. Specifically, the GO vector of the $i$-th protein is denoted as:

$$\mathbf{p}_i = \begin{bmatrix} a_{i,1} \\ \vdots \\ a_{i,u} \\ \vdots \\ a_{i,\Omega} \end{bmatrix} \text{ where } a_{i,u} = \begin{cases} 1 & \text{, GO hit} \\ 0 & \text{, otherwise} \end{cases} \qquad (2)$$



```
Given a query protein $\mathbf{Q}_i$,
1.  if the AC of the query protein is known,
        use the AC to generate GO vector $\mathbf{q}_{i,0}$;
        $k = 0$;
        go to Step 3;
    end

2.  if $||\mathbf{q}_{i,0}|| = \mathbf{0}$ or only the AA sequence of the query protein is known,
        retrieve $k_{\max}$ homologs of the query protein from BLAST using the AA sequence as input;
        $k = 1$;
        while $k \leq k_{\max}$,
          use the AC of the $k$-th homologous protein to generate GO vector $\mathbf{q}_{i,k}$;
          if $||\mathbf{q}_{i,k}|| \neq 0$,
            go to Step 3;
          else
            $k = k + 1$;
          end
        end
    end

3.  Perform SVM classification on $\mathbf{q}_{i,k}$.
```

Figure 1: Procedures of retrieving GO terms.

where 'GO hit' means that the $u$-th GO term appears in the GOA-search result using the AC of the $i$-th protein as the searching key.

2. **Term-Frequency (TF).** This approach is similar to the 1-0 value approach in that a protein is represented by a point in the $\Omega$-dim Euclidean space. However, unlike the 1-0 approach, it uses the number of occurrences of individual GO terms as the coordinates. Specifically, the GO vector $\mathbf{p}_i$ of the $i$-th protein is defined as:

$$\mathbf{p}_i = \begin{bmatrix} b_{i,1} \\ \vdots \\ b_{i,u} \\ \vdots \\ b_{i,\Omega} \end{bmatrix} \text{ where } b_{i,u} = \begin{cases} f_{i,u} & \text{, GO hit} \\ 0 & \text{, otherwise} \end{cases} \tag{3}$$

where $f_{i,u}$ is the number of occurrences of the $u$-th GO term (term-frequency) in the $i$-th protein. The rationale is that the term-frequencies may also contain important information for classification and therefore should not be quantized to either 0 or 1. Note that $b_{i,u}$'s are analogous to the term-frequencies commonly used in document retrieval.

3. **Inverse Sequence-Frequency (ISF).** In this approach, a protein is represented by a point with coordinates determined by the existence of GO terms and the inverse sequence-frequency (ISF). Specifically, the GO vector $\mathbf{p}_i$ of the $i$-th protein is defined as:

$$\mathbf{p}_i = \begin{bmatrix} c_{i,1} \\ \vdots \\ c_{i,u} \\ \vdots \\ c_{i,\Omega} \end{bmatrix}, \; c_{i,u} = a_{i,u} \log \left( \frac{N}{|\{k : a_{k,u} \neq 0\}|} \right) \tag{4}$$

where $N$ is the number of protein sequences in the training dataset. The denominator inside the logarithm is the number of GO vectors (among all GO vectors in the dataset) having a non-zero entry in their $u$-th element, or equivalently the number of sequences with the $u$-th GO term as determined in Section 3.1. Note that the logarithmic term in Eq. 4 is analogous to the inverse document frequency commonly used in document retrieval. The idea is to emphasize (resp. suppress) the GO terms that have a low (resp. high) frequency of occurrences in the protein sequences. The reason is that if a GO term occurs in every sequence, it is not very useful for classification.

4. **Term-Frequency–Inverse Sequence-Frequency (TF-ISF).** This approach combines term-frequency (TF) and inverse sequence frequency (ISF) mentioned above. Specifically, the GO vector $\mathbf{p}_i$ of the $i$-th protein is defined

as:

$$\mathbf{p}_i = \begin{bmatrix} d_{i,1} \\ \vdots \\ d_{i,u} \\ \vdots \\ d_{i,\Omega} \end{bmatrix}, d_{i,u} = b_{i,u} \log \left( \frac{N}{|\{k : b_{k,u} \neq 0\}|} \right) \quad (5)$$

where $b_{i,u}$ is defined in Eq. 3.

By correlating Eqs. 2–5 with the general form of pseudo amino acid composition (Eq. 1), we notice that $\Omega$ is the number of distinct GO terms of the given dataset, and $\phi_{i,u}$'s in Eq. 1 correspond to $a_{i,u}$, $b_{i,u}$, $c_{i,u}$ and $d_{i,u}$ in Eqs. 2–5, respectively.

Fig. 2 and Fig. 3 illustrate the prediction process of GOASVM using protein accession numbers (ACs) and protein sequences as input, respectively.

### 3.3. Multi-class SVM Classification

GO vectors are used for training one-vs-rest SVMs. Specifically, for an $M$-class problem (here $M$ is the number of subcellular locations), $M$ independent SVMs are trained, one for each class. Denote the GO vector created by using the true AC of the $i$-th query protein as $\mathbf{q}_{i,0}$ and the GO vectors created by using the AC of the $k$-th homolog as $\mathbf{q}_{i,k}$, $k = 1, \ldots, n$, where $n$ is the number of homologs retrieved by BLAST with the default parameter setting. Then, given the $i$-th query protein $\mathbf{Q}_i$, the score of the $m$-th SVM is:

$$s_m(\mathbf{Q}_i) = \sum_{r \in \mathcal{S}_m} \alpha_{m,r} y_{m,r} K(\mathbf{p}_r, \mathbf{q}_{i,k}) + b_m, \quad (6)$$

where $\mathcal{S}_m$ is the set of support vector indexes corresponding to the $m$-th SVM, $y_{m,r} \in \{-1, +1\}$ are the class labels, $\alpha_{m,r}$ are the Lagrange multipliers, and $K(\cdot, \cdot)$ is a kernel function. In this work, linear kernels were used, i.e., $K(\mathbf{p}_r, \mathbf{q}_{i,k}) = \langle \mathbf{p}_r, \mathbf{q}_{i,k} \rangle$. The predicted class of the query protein is given by

$$m^* = \arg \max_{m=1}^{M} s_m(\mathbf{Q}_i). \quad (7)$$

Note that $\mathbf{p}_r$'s in Eq. 6 represent the GO training vectors, which may include the GO vectors created by using the true ACs of the training proteins or their homologous ACs. We have the following two cases:

1. If the true ACs are available, $\mathbf{p}_r$'s represent the GO training vectors created by using the true ACs only.
2. If only the AA sequences are known, then only the ACs of the homologous sequences can be used for training the SVM and for scoring. In that case, $\mathbf{p}_r$'s represent the GO training vectors created by using the homologous ACs only.

## 4. Results and Discussions

### 4.1. Datasets

Two benchmark datasets (EU16 (Chou and Shen, 2006c) and HUM12 (Chou and Shen, 2006a)) and a novel dataset were used to evaluate the performance of GOASVM.

Table 1: Breakdown of the novel eukaryotic-protein dataset used in this work. The dataset contains proteins that were added to Swiss-Prot created between 08-Mar-2011 and 18-Apr-2012. The sequence identity of the dataset is below 25%. $^*$: no new proteins were found in the corresponding subcellular location.

| Label | Subcellular Location | No. of sequences |
|---|---|---|
| 1 | Cell Wall | 2 |
| 2 | Centriole | $0^*$ |
| 3 | Chloroplast | 51 |
| 4 | Cyanelle | $0^*$ |
| 5 | Cytoplasm | 77 |
| 6 | Cytoskeleton | 4 |
| 7 | Endoplasmic reticulum | 28 |
| 8 | Extracellular | 103 |
| 9 | Golgi apparatus | 14 |
| 10 | Lysosome | 1 |
| 11 | Mitochondrion | 73 |
| 12 | Nucleus | 57 |
| 13 | Peroxisome | 6 |
| 14 | Plasma membrane | 169 |
| 15 | Plastid | 5 |
| 16 | Vacuole | 18 |
| Total | | 608 |

The EU16 dataset and HUM12 dataset were created from Swiss-Prot 48.2 in 2005 and Swiss-Prot 49.3 in 2006, respectively. The EU16 comprises 4150 eukaryotic proteins (2423 in the training set and 1727 in the independent test set) with 16 classes and the HUM12 has 2041 human proteins (919 in the training set and 1122 in the independent test set) with 12 classes. Both datasets were cut off at 25% sequence similarity by a culling program (Wang et al., 2003). See Supplementary Materials for more information of the two datasets. These two datasets are good benchmarks for performance comparison, because none of the proteins in either dataset has more than 25% sequence identity to any other proteins in the same subcellular location. However, the training and testing sets of these two datasets were constructed at the same period of time. Therefore, the training and testing sets are likely to share similar GO information, causing over-estimation in the prediction accuracy.

To avoid over-estimating the prediction performance and to demonstrate the effectiveness of GOASVM, a eukaryotic dataset containing novel proteins was constructed by using the criteria specified in (Chou and Shen, 2006c). To ensure that the proteins are really novel to GOASVM, the creation dates of these proteins should be significantly later than the training proteins (from EU16) and also later than the GOA database. Because EU16 was created in 2005 and the GOA database used was released on 08-Mar-2011, we selected the proteins that were added to Swiss-Prot between 08-Mar-2011 and 18-Apr-2012. Moreover, only proteins with a single subcellular location that falls within the 16 classes of the EU16 dataset were selected. After limiting the sequence similarity to 25%, 608 eukaryotic proteins distributed in 14 subcellular locations (see Table 1) were selected. This dataset can be downloaded from

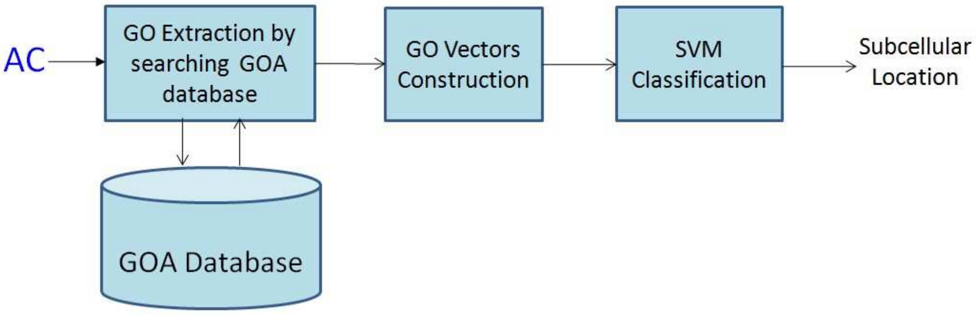


Figure 2: Flowchart of GOASVM that uses protein accession numbers (AC) only as input.

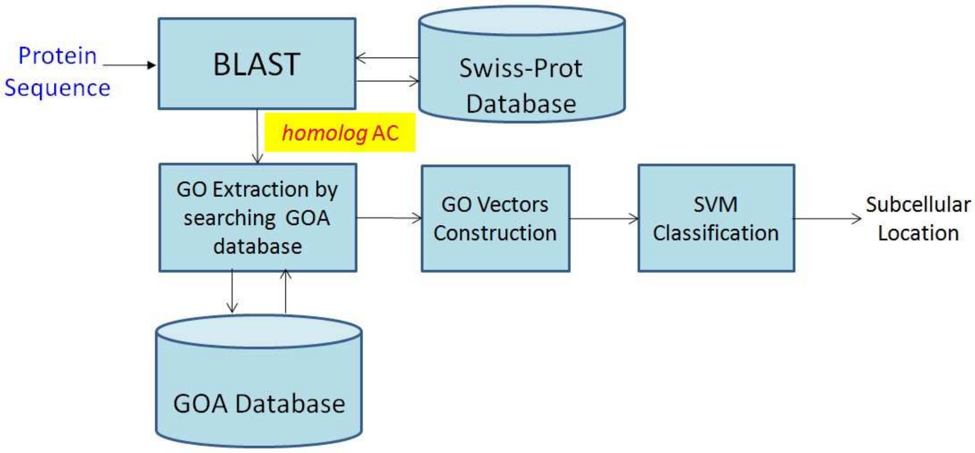


Figure 3: Flowchart of GOASVM that uses protein sequences only as input. *AC*: Accession Number.

the GOASVM web server.

*4.2. Cross Validation and Independent Tests*

We used the training set of EU16 to train the GOASVM and used the new testing set to evaluate its performance. For experiments in which EU16 and HUM12 were used, leave-one-out cross-validation (LOOCV) (Hastie et al., 2001) and independent tests were used for performance evaluation. In each fold of LOOCV, a protein of the training dataset (suppose there are $N$ proteins) was singled out as the test protein and the remaining $(N - 1)$ proteins were used as the training data. This procedure was repeated $N$ times, and in each fold a different protein was selected as the test protein. This ensures that every sequence in the dataset will be tested.

*4.3. Performance Metrics*

Several performance measures were used, including the overall accuracy (ACC), overall Mathew's correlation coefficient (OMCC) (Mak et al., 2008) and weighted average Mathew's correlation (WAMCC) (Mak et al., 2008). The latter two measures are based on Mathew's correlation coefficient (MCC) (Matthews, 1975). MCC can overcome the shortcoming of accuracy on imbalanced data and have the advantage of avoiding the performance to be dominated by the majority classes. For example, a classifier which predicts all samples as positive cannot be regarded as a good classifier unless it can also predict negative samples accurately. In this case, the accuracy and MCC of the positive class are 100% and 0%, respectively. Therefore, MCC is a better measure for imbalanced classification. Details of the performance measures are specified in Supplementary Materials.

*4.4. Prediction of Novel Proteins*

Because the novel proteins were recently added to Swiss-Prot, many of them have not been annotated in the GOA database. As a results, if we used the accession numbers of these proteins to search against the GOA database, the corresponding GO vectors will contain all zeros. This suggests that we should use the ACs of their homologs as the searching keys, i.e., the procedure shown in Fig. 3 should be adopted. However, we observed that for some novel proteins, even the top homologs do not have any GO terms annotated to them. In particular, in the new dataset, there are 169 protein sequences whose top homologs do not have any GO terms (2-nd row of Table 2), causing GOASVM unable to make any predictions. As can be seen from Table 2, by using only the first homolog, the overall prediction accuracy of GOASVM is only 57.07% (347/608). To overcome this limitation, the following strategy was adopted. For the 169 proteins (2-nd row of Table 2) whose top homologs do not have any GO terms in the GOA database, we used the second-top homolog to find the GO terms; similarly, for the 112 proteins (3-rd row of Table 2) whose top and

Table 2: Performance of GOASVM on the novel-protein dataset denoted in Table 1. The 2nd column represents the upper bound of $k$ in $\mathbf{q}_{i,k}$ shown in Fig 1. For example, when $k_{\max} = 2$, only the AC of the 1st- or 2nd homolog will be used for retrieving the GO terms. *No. of sequences without GO terms* means the number of protein sequences for which no GO terms can be retrieved. *OMCC*: Overall MCC; *WAMCC*: Weighted average MCC; *ACC*: Overall accuracy. See Supplementary Materials for the definition of these performance measures. Note for fair comparison, the *Baseline* shown here is the best performance we obtained, which also adopts the same procedure as GOASVM to obtain GO terms from homologs. *: Since the web-server of Euk-OET-PLoc is not available now, we implemented it according to Chou and Shen (2006c).

| Method | $k_{\max}$ | No. of sequences without GO terms | OMCC | WAMCC | ACC |
|---|---|---|---|---|---|
| GOASVM | 1 | 169 | 0.5421 | 0.5642 | 57.07% |
| | 2 | 112 | 0.5947 | 0.6006 | 62.01% |
| | 3 | 12 | 0.6930 | 0.6834 | 71.22% |
| | 4 | 7 | 0.6980 | 0.6881 | 71.71% |
| | 5 | 3 | 0.7018 | 0.6911 | 72.04% |
| | 6 | 3 | 0.7018 | 0.6911 | 72.04% |
| | 7 | 0 | **0.7035** | **0.6926** | **72.20%** |
| Baseline* (Euk-OET-PLoc) | 7 | 0 | 0.5246 | 0.5330 | 55.43% |

2-nd homologs do not have any GO terms, the third-top homolog was used; and so on until all the query proteins can correspond to at least one GO term. In the case where BLAST fails to find any homologs (although this case rarely happens) the default $E$-value threshold (the -e option) can be relaxed. Detailed descriptions of this strategy can be found in Section 3.1.

Table 2 shows the prediction performance of GOASVM on the 608 novel proteins. As explained earlier, to ensure that these proteins are novel to GOASVM, 2423 proteins extracted from the training set of EU16 were used for training the classifier. For fair comparison, Euk-OET-PLoc (Chou and Shen, 2006c) also uses the same version of the GOA database (08-Mar-2011) to retrieve GO terms and adopts the same procedure as GOASVM to obtain GO terms from homologs. In such case, for Euk-OET-PLoc, it is unnecessary to use the PseAA(Chou, 2001) as a backup method because a valid GO vector can be found for every protein in this novel dataset. Also, according to Euk-OET-PLoc (Chou and Shen, 2006c), several parameters are optimized and only the best performance is shown here (See the last row of Table 2). As can be seen, GOASVM performs significantly better than Euk-OET-PLoc (72.20% vs 55.43%), demonstrating that GOASVM is more capable of predicting novel proteins than Euk-OET-PLoc. Moreover, results clearly suggest that when more distant homologs are allowed to be used for searching GO terms in the GOA database, we have a higher chance of finding at least one GO terms for each of these novel proteins, thus improving the overall performance. In particular, when the most distant homolog has a rank of 7 ($k_{\max} = 7$), GOASVM is able to find GO terms for all of the novel proteins and the accuracy is also the highest, which is almost 15% (absolute) higher than that using only the top homolog. Given the novelty of these proteins and the low sequence similarity (below 25%), an accuracy of 72.2% is fairly high, suggesting that the homologs of novel proteins can provide useful GO information for protein subcellular localization.

Note that the gene association file that we downloaded from the GOA database does not provide any subcellular localization labels. This file only allows us to create a hash table storing the association between the accession numbers and their corresponding GO terms. This hash table covers all of the accession numbers in the GOA database released on 08-Mar-2011, meaning that it will cover the EU16 (dated in 2005) but not the accession numbers in the novel eukaryotic dataset. It is important to emphasize that given a query protein, having a match in this hash table does not mean that a subcellular-localization assignment can be obtained. In fact, having a match only means that a non-null GO vector can be obtained. After that, the SVMs play an important role in classifying the non-null GO vector.

### 4.5. Comparing GO Vector Construction Methods

Table 3 shows the performance of different GO-vector construction methods on the novel-protein dataset denoted in Table 1. Linear SVMs were used for all cases, and the penalty factor was set to 0.1. Also, the 2423 proteins in the training set of the EU16 dataset was used for training the classifier, which was subsequently used to classify proteins in the novel dataset. Evidently, term-frequency (TF) performs the best among these four methods, which demonstrates that the frequencies of occurrences of GO terms provide additional information for subcellular localization. The results also suggest that inverse sequence-frequency (ISF) is detrimental to classification performance, despite its proven effectiveness in document retrieval. This may be due to the differences between the frequency of occurrences of common GO terms in our datasets and the frequency of occurrences of common words in document retrieval. In document retrieval, almost all documents contain the common words; as a result, the inverse document frequency is effective in suppressing the influence of these words in the retrieval. However, the common GO terms do not appear in all of the proteins in our datasets. In fact, even the most commonly occurred GO term appears only in one-third of the proteins in EU16. We conjecture that this low-frequency of occurrences of common GO terms makes ISF not effective for subcellular localization.

Many existing GO-based methods use the 1-0 value approach to constructing GO vectors, including ProLoc-GO (Huang et al., 2008), Euk-OET-PLoc (Chou and Shen, 2006c), and Hum-PLoc (Chou and Shen, 2006a). Table 3 shows that

Table 3: Performance of different GO-vector construction methods on the novel-protein dataset. *TF*: term-frequency; *ISF*: inverse sequence-frequency; *TF-ISF*: term-frequency inverse sequence frequency. *OMCC*: Overall MCC; *WAMCC*: Weighted average MCC; *ACC*: Overall accuracy. See Supplementary Materials for the definition of these performance measures.

| GO Vector Construction Method | OMCC | WAMCC | ACC |
|---|---|---|---|
| 1-0 value | 0.6877 | 0.6791 | 70.72% |
| TF | **0.7035** | **0.6926** | **72.20%** |
| ISF | 0.6386 | 0.6256 | 66.12% |
| TF-ISF | 0.6772 | 0.6626 | 69.74% |

term-frequency (TF) performs almost 2% better than 1-0 value (72.20% vs 70.72%). Similar conclusions can be also drawn from the performance of GOASVM based on leave-one-out cross validation on the EU16 training set and the HUM12 training set (See Supplementary Materials). The results are biologically relevant because proteins of the same subcellular localization are expected to have a similar number of occurrences of the same GO term. In this regard, the 1-0 value approach is inferior because it quantizes the number of occurrences of a GO term to 0 or 1. Recently, we found that an approach similar to the TF approach had also been used in iLoc-Euk (Chou et al., 2011), iLoc-Hum (Chou et al., 2012), iLoc-Plant (Wu et al., 2011), iLoc-Gpos (Wu et al., 2012), iLoc-Gneg (Xiao et al., 2011a), and iLoc-Virus (Xiao et al., 2011b).

### *4.6. Compare with State-of-the-Art GO Methods*

To further demonstrate the superiority of GOASVM over other state-of-the-art GO methods, we also did experiments on the EU16 dataset and the HUM12 dataset, respectively. Table 4 compares the performance of GOASVM against three state-of-the-art GO-based methods on the EU16 dataset and the HUM12 dataset, respectively. As Euk-OET-PLoc and Hum-PLoc could not produce valid GO vectors for some proteins in EU16 and HUM12, both methods use PseAA as a backup. ProLoc-GO uses either the ACs of proteins as searching keys or uses the ACs of homologs returned from BLAST as searching keys. GOASVM also uses BLAST to find homologs, but unlike ProLoc-GO, GOASVM uses more than the top-ranked homologs.

Table 4 shows that for ProLoc-GO, using ACs as input performs better than using sequences (ACs of homologs) as input. However, the results for GOASVM are not conclusive in this regard because under LOOCV, using ACs as input performs better than using sequences, but the situation is opposite under independent tests. Table 4 also shows that no matter using ACs as input or sequences as input, GOASVM performs better than Euk-OET-PLoc and ProLoc-GO, for both the EU16 and HUM12 datasets.

To show that the high performance of GOASVM is not purely attribute to the homologous information obtained from BLAST, we used BLAST directly as a subcellular localization predictor. Specifically, the subcellular location of a query protein is determined by the subcellular location of its closest homolog as determined by BLAST using Swiss-Prot 2012_04 as the protein database. The subcellular location of the homologs were obtained from their CC field in Swiss-Prot. Results in Table 4 show that the performance of this approach is significantly poorer than that of other machine learning approaches, suggesting that homologous information alone is not sufficient for subcellular localization prediction. Briesemeister et al. (2009) also used BLAST to find the subcellular locations of proteins. Their results also suggest that using BLAST alone is not sufficient for reliable prediction.

Although all the datasets mentioned in this paper were cut off at 25% sequence similarity, the performance of GOASVM increased from 72.20% (Table 2) on the novel dataset to more than 90% (Table 4) on both the EU16 dataset and the HUM12 dataset. This is mainly because in Table 4, the training and testing sets were constructed at the same time, whereas there are 6 years apart between the creation of the training set and the testing set in Table 2, which causes the latter to have less similarity in GO information between the training set and test sets than the former. This in turn implies that the performance of GOASVM on our novel dataset (Table 2) can more objectively reflect the classification capabilities of the predictors.

### *4.7. GOASVM Using Old GOA Database*

The newer the version of GOA database, the more annotation information it contains. To investigate how the updated information affects the performance of GOASVM, we performed experiments using an earlier version (published in Oct. 2005) of the GOA database and compared the results with Euk-OET-PLoc on the EU16 dataset. Comparison between the last and second last rows of Table 5 reveals that using newer versions of the GOA database can achieve better performance than using older versions. This suggests that annotation information is very important to the prediction. The results also show that GOASVM significantly outperforms Euk-OET-PLoc, suggesting that the GO vector construction method and classifier (term-frequency and SVM) in GOASVM are superior to the those used in Euk-OET-PLoc (1-0 value and K-NN).

## 5. Conclusion

This paper proposes a GO-based method – GOASVM – to predict subcellular locations of proteins. The accession numbers (ACs) of query proteins are used as keys to search against the GOA database to find the GO terms. For proteins without an AC, BLAST is used to find their homologs and the ACs of these homologs are used as the searching keys. Then, GO

Table 4: Comparing GOASVM with state-of-the-art GO-based methods on (a) the EU16 dataset and (b) the HUM12 dataset. *S*: Sequences; *AC*: accession number; LOOCV: leave-one-out cross-validation. *m*(*n*): *m* means the accuracy; *n* means the WAMCC. See Supplementary Materials for the definition of WAMCC. (–) means the corresponding references do not provide the WAMCC.

| Method | Input Data | Feature | Accuracy (WAMCC) | |
|---|---|---|---|---|
| | | | LOOCV | Independent Test |
| ProLoc-GO (Huang et al., 2008) | S | GO (using BLAST) | 86.6% (0.7999) | 83.3% (0.706) |
| ProLoc-GO (Huang et al., 2008) | AC | GO (No BLAST) | 89.0% (–) | 85.7% (0.710) |
| Euk-OET-PLoc (Chou and Shen, 2006c) | S + AC | GO + PseAA | 81.6% (–) | 83.7% (–) |
| GOASVM | S | GO (usig BLAST) | **94.68% (0.9388)** | 93.86% (0.9252) |
| GOASVM | AC | GO (No BLAST) | 94.55% (0.9379) | **94.61% (0.9348)** |
| BLAST (Altschul et al., 1997) | S | – | 56.75% | 60.39% |

(a) Performance on the EU16 dataset

| Method | Input Data | Feature | Accuracy (WAMCC) | |
|---|---|---|---|---|
| | | | LOOCV | Independent Test |
| ProLoc-GO (Huang et al., 2008) | S | GO (using BLAST) | 90.0% (0.822) | 88.1% (0.661) |
| ProLoc-GO (Huang et al., 2008) | AC | GO (No BLAST) | 91.1% (–) | 90.6% (0.724) |
| Hum-PLoc (Chou and Shen, 2006a) | S + AC | GO + PseAA | 81.1% (–) | 85.0% (–) |
| GOASVM | S | GO (usig BLAST) | **91.73% (0.9033)** | 94.21% (0.9346) |
| GOASVM | AC | GO (No BLAST) | 91.51% (0.9021) | **94.39% (0.9367)** |
| BLAST (Altschul et al., 1997) | S | – | 68.55% | 65.69% |

(b) Performance on the HUM12 dataset

Table 5: Performance of GOASVM based on different versions of the GOA database on the EU16 training dataset. The 2nd column specifies the publication year of the GOA database being used for constructing the GO vectors. For proteins without a GO term in the GOA database, pseudo amino-acid composition (PseAA) was used as the backup feature. When the latest GOA database is used (last row), only one protein in the dataset does not have a GO term. Therefore, we assigned '0' to all of the elements in the GO vector of this protein instead of using PseAA. *LOOCV*: leave-one-out cross validation. Note for fair comparison, GOASVM here only uses the ACs as input and thus the backup method is needed.

| Method | Feature | | Accuracy | |
|---|---|---|---|---|
| | Main | Backup | LOOCV | Independent Test |
| Euk-OET-PLoc (Chou and Shen, 2006c) | GO (GOA2005) | PseAA | 81.6% | 83.7% |
| GOASVM | GO (GOA2005) | PseAA | 86.42% | 89.11% |
| GOASVM | GO (GOA2011) | – | **94.55%** | **94.61%** |

terms are used to construct the GO vectors, which are subsequently classified by SVMs. Comparing with the existing GO-based methods, GOASVM has the following advantages: (1) it constructs the GO vectors by using the frequency of occurrences of GO terms instead of 1-0 value; (2) it adopts a new strategy to incorporate more useful homologous GO information for classification; and (3) it selects a relevant GO-vector subspace by finding distinct GO terms instead of using the full GO-vector space. Results on a novel eukaryotic dataset and two benchmark datasets demonstrate that GOASVM outperforms the homology-based method and methods based on amino acid compositions, and GOASVM may play a complementary role to the existing state-of-the-art predictors such as iLoc-Euk (Chou et al., 2011) and iLoc-Hum (Chou et al., 2012). It was found that the frequency of occurrences of GO terms provides useful information for classification. The high performance of GOASVM on a latest eukaryotic dataset shows its practicality and effectiveness on the prediction of subcellular locations of proteins.

Because one of the future directions for subcellular localization prediction is to develop user-friendly and publicly accessible web-servers (Chou and Shen, 2009), we have provided a web-server for GOASVM at `http://bioinfo.eie.polyu.edu.hk/mGoaSvmServer/GOASVM.html`.

Our method can be extended to multi-label proteins (Chou et al., 2011, 2012). The extension from single-label protein prediction to multi-label protein prediction is a research topic in machine learning (Godbole and Sarawagi, 2004; Elisseeff and Weston, 2001) and we will address this in our future work.

## Acknowledgements

This work was in part supported by The HK RGC Grant No. PolyU5264/09E and HKPolyU Grant No. G-U877.

## Appendix A

To exemplify the discussion in Section 2, we created a lookup table (Table 6) and developed a table-lookup procedure to predict the subcellular localization of the proteins in the EU16

Table 6: Explicit GO terms for the EU16 dataset. Explicit GO terms include essential GO terms and their child terms that appear in the proteins of the dataset. The definition of essential GO terms can be found in (Huang et al., 2008). Here the relationship only includes 'is a' and 'part of', because only cellular component GO terms are analyzed here. *Relationship*: the relationship between child terms and their parent essential GO terms; *No. of Terms*: the total number of explicit GO terms in a particular class.

| Class | Cellular Component | Explicit GO Terms | | No. of Terms |
|---|---|---|---|---|
| | | Essential GO terms | Child Terms (Relationship) | |
| 1 | Cell Wall | GO:0005618 | GO:0009274 (Is a), GO:0009277 (Is a),<br>GO:0009505 (Is a), GO:0031160 (Is a) | 5 |
| 2 | Centriole | GO:0005814 | None | 1 |
| 3 | Chloroplast | GO:0009507 | None | 1 |
| 4 | Cyanelle | GO:0009842 | GO:0034060 (Part of) | 2 |
| 5 | Cytoplasm | GO:0005737 | GO:0016528 (Is a), GO:0044444 (Part of) | 3 |
| 6 | Cytoskeleton | GO:0005856 | GO:0001533 (Is a), GO:0030863 (Is a),<br>GO:0015629 (Is a), GO:0015630 (Is a),<br>GO:0045111 (Is a), GO:0044430 (Part of) | 7 |
| 7 | Endoplasmic reticulum | GO:0005783 | GO:0005791 (Is a), GO:0044432 (Part of) | 3 |
| 8 | Extracellular | GO:0030198 | None | 1 |
| 9 | Golgi apparatus | GO:0005794 | None | 1 |
| 10 | Lysosome | GO:0005764 | GO:0042629 (Is a), GO:0005765 (Part of),<br>GO:0043202 (Part of) | 4 |
| 11 | Mitochondrion | GO:0005739 | None | 1 |
| 12 | Nucleus | GO:0005634 | GO:0043073 (Is a), GO:0045120 (Is a),<br>GO:0044428 (Part of) | 4 |
| 13 | Peroxisome | GO:0005777 | GO:0020015 (Is a), GO:0009514 (Is a) | 3 |
| 14 | Plasma membrane | GO:0005886 | GO:0042383 (Is a), GO:0044459 (Part of) | 3 |
| 15 | Plastid | GO:0009536 | GO:0009501 (Is a), GO:0009507 (Is a),<br>GO:0009509 (Is a), GO:0009513 (Is a),<br>GO:0009842 (Is a) | 6 |
| 16 | Vacuole | GO:0005773 | GO:0000322 (Is a), GO:0000323 (Is a),<br>GO:0005776 (Is a) | 4 |

dataset. Table 6 has two types of GO terms: essential GO terms and child GO terms. As the name implies, the essential GO terms, as identified by Huang et al. Huang et al. (2008), are GO terms that are essential or critical for the subcellular localization prediction. In addition to the essential GO terms, their direct descendants (known as child terms) also possess direct localization information. The relationships between child terms and their parent terms include 'is a', 'part of' and 'occurs in' (Lord et al., 2003). The former two correspond to cellular component GO terms and the third one typically corresponds to biological process GO terms. As we are more interested in cellular component GO terms, the 'occurs in' relationship will not be considered. For ease of reference, we refer to both essential GO terms and their child terms as 'explicit GO terms'.

For each class in Table 6, the child terms were obtained by presenting the corresponding essential GO term to the QuickGO server (Binns et al., 2009), followed by excluding those child terms that do not appear in the proteins of the EU16 dataset.[4]

Given a query sequence, we first obtain its 'GO-term' set from the GO annotation database. Then, if only one of the terms in this set matches an essential GO term in Table 6, the subcellular location of this query protein is predicted to be the one corresponding to this matched GO term. For example, if the set of GO terms contains GO:0005618, then this query protein is predicted as 'Cell Wall'. Further, if none of the terms in this set matches any essential GO terms but one of the terms in this set matches any child terms in Table 6, then the query protein is predicted as belonging to the class associated with this child GO term. For example, if no essential GO terms can be found in the set but GO:0009274 is found, then the query protein is predicted as 'Cell Wall'.

A major problem of this table lookup procedure is that the GO terms of a query protein may contain more than one essential GO terms and/or having child terms spanning across several classes, causing inconsistent classification decisions. For example, in the EU16 dataset, 713 (out of 2423) proteins have explicit GO terms that map to more than one class, and 513

[4] Note that if we use the cellular-component names as the searching keys, QuickGO will give us more than 49 cellular-component GO terms, suggesting that the 49 explicit GO terms are only a tiny subset of all relevant GO terms (in our method, we have more than 5000 relevant GO terms). Even for such a small number of explicit GO terms, many proteins have explicit GO terms spanning several classes.

(out of 2423) proteins do not have any explicit GO terms. This means that about 51% (1226/2423) of the proteins in the dataset cannot be predicted using only explicit GO terms. Among the 2423 proteins in the dataset, only 1197 (49%) of them have explicit GO terms that map to unique (consistent) subcellular locations. This analysis suggests that direct table lookup is not a desirable approach and this motivates us to develop machine learning methods for GO-based subcellular localization prediction.

## References

Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., Lipman, D. J., 1997. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. Nucleic Acids Res. 25, 3389–3402.

Apweiler, R., Bairoch, A., Wu, C. H., Barker, W. C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., Martin, M. J., Ntale, D. A., O'Donovan, C., Redaschi, N., Yeh, L. S., 2004. UniProt: the Universal Protein knowledgebase. Nucleic Acids Res 32, D115–D119.

Barrel, D., Dimmer, E., Huntley, R. P., Binns, D., O'Donovan, C., Apweiler, R., 2009. The GOA database in 2009-an integrated Gene Ontology Annotation resource. Nucl. Acids Res. 37, D396–D403.

Binns, D., Dimmer, E., Huntley, R., Barrell, D., O'Donovan, C., Apweiler, R., 2009. QuickGO: a web-based tool for Gene Ontology searching. Bioinformatics 25 (22), 3045–3046.

Blum, T., Briesemeister, S., Kohlbacher, O., 2009. MultiLoc2: Integrating phylogeny and Gene Ontology terms improves subcellular protein localization prediction. BMC Bioinformatics 10, 274.

Briesemeister, S., Blum, T., Brady, S., Lam, Y., Kohlbacher, O., Shatkay, H., 2009. SherLoc2: A high-accuracy hybrid method for predicting subcellular localization of proteins. Journal of Proteome Research 8, 5363–5366.

Camon, E., Magrane, M., Barrel, D., Binns, D., Fleischnann, W., Kersey, P., Mulder, N., Oinn, T., Maslen, J., Cox, A., 2003. The gene ontology annotation (GOA) project: Implementation of GO in SWISS-PROT, TrEMBL and InterPro. Genome Res. 13, 662–672.

Chou, K. C., 2001. Prediction of protein cellular attributes using pseudo amino acid composition. Proteins: Structure, Function, and Genetics 43, 246–255.

Chou, K. C., 2005. Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes). Bioinformatics 21, 10–19.

Chou, K. C., 2011. Some remarks on protein attribute prediction and pseudo amino acid composition (50th Anniversary Year Review). Journal of Theoretical Biology 273, 236–247.

Chou, K. C., Cai, Y. D., 2003. Prediction and classification of protein subcellular location: sequence-order effect and pseudo amino acid composition. J. of Cell. Biochem. 90, 1250–1260.

Chou, K. C., Cai, Y. D., 2004. Prediction of protein subcellular locations by GO-FunD-PseAA predicor. Biochem. Biophys. Res. Commun. 320, 1236–1239.

Chou, K. C., Cai, Y. D., 2005. Predicting protein localization in budding yeast. Bioinformatics 21, 944–950.

Chou, K. C., Shen, H. B., 2006a. Hum-PLoc: A novel ensemble classifier for predicting human protein subcellular localization. Biochem Biophys Res Commun 347, 150–157.

Chou, K. C., Shen, H. B., 2006b. Large-scale predictions of gram-negative bacterial protein subcellular locations. Journal of Proteome Research 5, 3420–3428.

Chou, K. C., Shen, H. B., 2006c. Predicting eukaryotic protein subcellular location by fusing optimized evidence-theoretic K-nearest neighbor classifiers. J. of Proteome Research 5, 1888–1897.

Chou, K. C., Shen, H. B., 2007a. Euk-mPLoc: A fusion classifier for large-scale eukaryotic protein subcellular location prediction by incorporating multiple sites. Journal of Proteome Research 6, 1728–1734.

Chou, K. C., Shen, H. B., 2007b. Recent progress in protein subcellular location prediction. Analytical Biochemistry 1 (370), 1–16.

Chou, K. C., Shen, H. B., 2008. Cell-PLoc: A package of web-servers for predicting subcellular localization of proteins in various organisms. nature Protocols 3, 153–162.

Chou, K. C., Shen, H. B., 2009. Review: recent advances in developing web-servers for predicting protein attributes. Natural Science 2, 63–92.

Chou, K. C., Shen, H. B., 2010a. Cell-PLoc 2.0: An improved package of web-servers for predicting subcellular localization of proteins in various organisms. Nat. Sci. 2, 1090–1103.

Chou, K. C., Shen, H. B., 2010b. Plant-mPLoc: A top-down strategy to augment the power for predicting plant protein subcellular localization. PLoS ONE 5, e11335.

Chou, K. C., Wu, Z. C., Xiao, X., 2011. iLoc-Euk: A multi-label classifier for predicting the subcellular localization of singleplex and multiplex eukaryotic proteins. PLoS ONE 6 (3), e18258.

Chou, K. C., Wu, Z. C., Xiao, X., 2012. iLoc-Hum: using the accumulation-label scale to predict subcellular locations of human proteins with both single and multiple sites. Molecular BioSystems 8, 629–641.

Elisseeff, A., Weston, J., 2001. Kernel methods for multi-labelled classification and categorical regression problems. In: In Advances in Neural Information Processing Systems 14. MIT Press, pp. 681–687.

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 (4), 1005–1016.

Godbole, S., Sarawagi, S., 2004. Discriminative methods for multi-labeled classification. In: In Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, pp. 22–30.

Hastie, T., Tibshirani, R., Friedman, J., 2001. The Element of Statistical Learning. Springer-Verlag.

Horton, P., Park, K. J., Obayashi, T., Nakai, K., 2006. Protein subcellular localization prediction with WOLF PSORT. In: Proc. 4th Annual Asia Pacific Bioinformatics Conference (APBC06). pp. 39–48.

Hu, Y., Li, T., Sun, J., Tang, S., Xiong, W., Li, D., Chen, G., Cong, P., 2012. Predicting Gram-positive bacterial protein subcellular localization based on localization motifs. Journal of Theoretical Biology 308, 135–140.

Huang, W. L., Tung, C. W., Ho, S. W., Hwang, S. F., Ho, S. Y., 2008. ProLoc-GO: Utilizing informative Gene Ontology terms for sequence-based prediction of protein subcellular localization. BMC Bioinformatics 9 (80).

Huang, W. L., Tung, C. W., Ho, S. W., Hwang, S. F., Ho, S. Y., 2009. Predicting protein subnuclear localization using GO-amino-acid composition features. Biosystems 98 (2), 73–79.

Lee, K. Y., Kim, D. W., Na, D. K., Lee, K. H., Lee, D. H., 2006. PLPD: Reliable protein localization prediction from imbalanced and overlapped datasets. Nucleic Acids Research 34 (17), 4655–4666.

Lei, Z., Dai, Y., 2006. Assessing protein similarity with Gene Ontology and its use in subnuclear localization prediction. BMC Bioinformatics 7, 491.

Lord, P. W., Stevens, R. D., Brass, A., Goble, C. A., 2003. Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation . Bioinformatics 19 (10), 1275–1283.

Lu, Z., Szafron, D., Greiner, R., Lu, P., Wishart, D. S., Poulin, B., Anvik, J., Macdonell, C., Eisner, R., 2004. Predicting subcellular localization of proteins using machine-learned classifiers. Bioinformatics 20 (4), 547–556.

Lubec, G., Afjehi-Sadat, L., Yang, J. W., John, J. P., 2005. Searching for hypothetical proteins: theory and practice based upon original data and literature. Prog. Neurobiol 77, 90–127.

Mak, M. W., Guo, J., Kung, S. Y., 2008. PairProSVM: Protein subcellular localization based on local pairwise profile alignment and SVM. IEEE/ACM Trans. on Computational Biology and Bioinformatics 5 (3), 416 – 422.

Matthews, B., 1975. Comparison of predicted and observed secondary structure of t4 phage lysozyme. Biochem. Biophys. Acta 405, 442–451.

Mei, S., 2012. Multi-label multi-kernel transfer learning for human protein subcellular localization. PLoS ONE 7 (6), e37716.

Mei, S. Y., Fei, W., Zhou, S. G., 2011. Gene ontology based transfer learning for protein subcellular localization. BMC Bioinformatics 12, 44.

Mott, R., Schultz, J., Bork, P., Ponting, C., 2002. Predicting protein cellular localization using a domain projection method. Genome research 12 (8), 1168–1174.

Nair, R., Rost, B., 2002. Sequence conserved for subcellular localization. Protein Science 11, 2836–2847.

Nakai, K., Kanehisa, M., 1991. Expert system for predicting protein localization sites in gram-negative bacteria. Proteins: Structure, Function, and Genetics 11 (2), 95–110.

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.

Park, K. J., Kanehisa, M., 2003. Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs. Bioinformatics 19 (13), 1656–1663.

Scott, M., Thomas, D., Hallett, M., 2004. Predicting subcellular localization via protein motif co-occurrence. Genome research 14 (10a), 1957–1966.

Shen, H. B., Chou, K., 2007. Gpos-PLoc: An ensemble classifier for predicting subcellular localization of Gram-positive bacterial proteins. Protein Engineering, Design and Selection 20, 39–46.

Wan, S., Mak, M. W., Kung, S. Y., Sept 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.

Wang, G. L., Dunbrack, J., PISCES, R. L., 2003. A protein sequence culling server. Bioinformatics 19, 1589–1591.

Wang, W., Mak, M. W., Kung, S. Y., 2010. Speeding up subcellular localization by extracting informative regions of protein sequences for profile alignment. In: Proc. Computational Intelligence in Bioinformatics and Computational Biology (CIBCB'10). pp. 147–154.

Wu, Z. C., Xiao, X., Chou, K. C., 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.

Wu, Z. C., Xiao, X., Chou, K. C., 2012. iLoc-Gpos: A multi-layer classifier for predicting the subcellular localization of singleplex and multiplex gram-positive bacterial proteins. Protein & Peptide Letters 19, 4–14.

Xiao, X., Wu, Z. C., Chou, K. C., 2011a. A multi-label learning classifier for predicting the subcellular localization of gram-negative bacterial proteins with both single and multiple sites. PLoS ONE 6 (6), e20592.

Xiao, X., Wu, Z. C., Chou, K. C., 2011b. 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.

Zdobnov, E. M., Apweiler, R., 2001. InterProScan – an integration platform for the signature-recognition methods in InterPro. Bioinformatics 17, 847–848.