

# PROTEIN SUBCELLULAR LOCALIZATION PREDICTION BASED ON PROFILE ALIGNMENT AND GENE ONTOLOGY

*Shibiao Wan , Man-Wai Mak*

*Sun-Yuan Kung*

Dept. of Electronic and Information Engineering  
The Hong Kong Polytechnic University  
Hung Hom, Hong Kong SAR, China  
email: {10900600r,enmwak}@polyu.edu.hk

Dept. of Electrical Engineering  
Princeton University  
New Jersey, USA  
email: kung@princeton.edu

## ABSTRACT

The functions of proteins are closely related to their subcellular locations. Computational methods are required to replace the laborious and time-consuming experimental processes for proteomics research. This paper proposes combining homology-based profile alignment methods and functional-domain based Gene Ontology (GO) methods to predict the subcellular locations of proteins. The feature vectors constructed by these two methods are recognized by support vector machine (SVM) classifiers, and their scores are fused to enhance classification performance. The paper also investigates different approaches to constructing the GO vectors based on the GO terms returned from InterProScan. The results demonstrate that the GO methods are comparable to profile-alignment methods and overshadow those based on amino-acid compositions. Also, the fusion of these two methods can outperform the individual methods.

**Index Terms**— Protein subcellular localization; Gene Ontology; Profile Alignment; InterProScan; PairProSVM; Support vector machines.

## 1. INTRODUCTION

Identifying the functions of proteins is one of the fundamental targets in proteomics research. The subcellular locations of proteins can have significant influence on their functional characteristics. However, determination of subcellular localization entirely by laboratory tests is both time-consuming and laborious; while at the same time, the number of newly found protein sequences has been growing rapidly in the post-genomic era. Therefore, more reliable, efficient and automatic methods are highly required for the prediction of where a protein resides in a cell. Over the years, a number of in-silico methods have been proposed to deal with this problem. Conventional methods can be generally divided into four categories described below.

**Composition-based methods** are one of the earliest methods for subcellular localization prediction. This category focuses on the relationship between subcellular locations and the information embedded in the amino acid sequences such as amino-acid compositions (AA) [1], [2], amino-acid pair compositions (PairAA) [1], and gapped amino-acid pair compositions (GapAA) [3]. Based on these early approaches, Chou [4] proposed a method called pseudo amino-acid composition (PseAA) using a sequence-order correlation factor to discover more biochemical properties from protein sequences.

**Sorting-signals based methods** predict the localization via the recognition of N-terminal sorting signals in amino acid sequences [5]. Nakai in 1991 [6] proposed the earliest predictor using sorting signals—PSORT. Recently, more advanced approaches based on the composition of sorting-signal have been proposed [7].

**Homology-based methods** use the fact that homologous sequences are more likely to reside in the same subcellular location. This kind of methods can achieve a very high accuracy as long as the homologs of the query sequences can be found in protein databases [8]. Over the years, a number of homology-based predictors have been proposed. For example, Proteome Analyst [9] computes the feature vectors for classification by using the presence or absence of some tokens from certain fields of the homologous sequences in the Swiss-Prot database. Recently, a predictor called PairProSVM was proposed by Mak et al. [10], which applies profile alignment to detect weak similarity between protein sequences.

**Functional-domain based methods** make use of the correlation between the function of a protein and its subcellular location. In [11], a sequence is mapped into the GO database so that a feature vector can be formed by determining which GO terms the sequence holds. Moreover, based on deeper biological knowledge, [12] proposes a searching algorithm called GOMining to discover the informative GO terms and classify them into instructive GO terms and essential GO terms to leverage the information in the GO database.

Among all the methods mentioned above, sorting-signal

This work was in part supported by The Hong Kong Research Grant Council, Grant No. PolyU5251/08E and HKPolyU Grant No. G-U877.

based methods could only deal with datasets containing a few subcellular locations. For example, the popular TargetP [13], [14] could only detect three locations: chloroplast, mitochondria and secretory pathway. Homology based methods, on the other hand, can detect as many locations as appeared in the dataset and can achieve comparatively high accuracy [15]. But when the dataset contains sequences with low sequence similarity or the numbers of samples in different classes are imbalanced, the performance is still very poor. Although the functional-domain based methods can often outperform sequence-based methods (as they can leverage the annotation in functional domain databases), they can only be applied to datasets where the sequences possess the required information as so far not all sequences are functionally annotated. Thus, they must be complemented by other types of methods.

This paper proposes a method based on the fusion of functional-domain based Gene Ontology (GO) methods and homology-based Pairwise Profile Alignment SVM (Pair-ProSVM). The GO-based and homology-based methods are detailed in Sections 2 and 3, respectively, and the fusion of these two methods is explained in Section 4. Section 5 details the experiments and results, which show that the proposed predictor can leverage the advantages of both methods, leading to better classification performance.

## 2. GENE ONTOLOGY METHOD

Gene Ontology (GO)<sup>1</sup> is a set of standardized vocabularies that annotate the function of genes and gene products across different species. The term ‘ontology’ originally refers to a systematic account of existence. In the GO database, the annotations of gene products are organized in three related ontologies: cellular components, biological processes, and molecular functions. A cellular component is a component of a cell. It is a part of some larger objects such as an anatomical structure or a gene product group. A biological process is a sequence of events achieved by one or more ordered assemblies of molecular functions. A molecular function is achieved by activities that can be performed by individual or by assembled complexes of gene products at the molecular level.

Although the cellular component ontology is directly related to the subcellular localization, we cannot simply use its GO terms to annotate the subcellular locations of proteins. The reason is that the percentage of proteins that have annotation of cellular components in the GO database is less than the percentage of proteins that have subcellular locations annotations in the Swiss-Prot database [16]. In fact, for those proteins that are annotated as ‘Subcellular Location Unknown’ in Swiss-Prot, many of them have GO terms also labelled as ‘Cellular Component Unknown’ in the GO database. On the other hand, proteins with subcellular locations clearly anno-

tated in Swiss-Prot may still be marked as ‘Cellular Component Unknown’ in the GO database [16]. Because of this limitation, it is necessary to make use of the other two ontologies as they are also relevant (although not directly) to the subcellular localization of proteins.

This paper investigates several approaches to extracting subcellular localization information from the GO database. This is realized through a GO Processor, which consists of two parts: GO vector construction and GO vector post-processing.

### 2.1. Construction of GO Vectors

The construction of GO vectors is divided into two steps. First, a collection of distinct GO terms is obtained by presenting all of the sequences in the dataset to InterProScan.<sup>2</sup> For each query sequence, InterProScan returns a file containing the GO terms found by various protein-signature recognition algorithms (we used all available algorithms in this work). Using the dataset described in Section 5, we found 1203 distinct GO terms, from GO:0019904 to GO:0016719. These GO terms form a GO Euclidean space with 1203 dimensions.

In the second step, for each sequence in the dataset, we constructed a GO vector by matching its GO terms to all of the 1203 GO terms determined in the first step. We have investigated four approaches to determine the elements of the GO vectors.

1. **1-0 value.** In this approach, each of the 1203 GO terms represents one canonical basis of a Euclidean space, and a protein sequence is represented by a point with coordinates equal to either 0 or 1. Specifically, the GO vector of the  $i$ -th protein is denoted as:

$$\mathbf{p}_i = \begin{bmatrix} a_{i,1} \\ \vdots \\ a_{i,j} \\ \vdots \\ a_{i,1203} \end{bmatrix} \quad \text{where } a_{i,j} = \begin{cases} 1 & , \text{GO hit} \\ 0 & , \text{otherwise} \end{cases} \quad (1)$$

where ‘GO hit’ means that the GO term appears in the protein.

2. **Term-Frequency.** This approach is similar to the 1-0 value approach in that a protein is represented by a point in a Euclidean space. However, unlike the 1-0 approach, it uses the number of occurrences of individual GO terms as the coordinates. Specifically, the GO vec-

<sup>1</sup><http://www.geneontology.org>

<sup>2</sup><http://www.ebi.ac.uk/Tools/pfa/iprscan/#>

tor  $\mathbf{p}_i$  of the  $i$ -th protein is defined as:

$$\mathbf{p}_i = \begin{bmatrix} b_{i,1} \\ \vdots \\ b_{i,j} \\ \vdots \\ b_{i,1203} \end{bmatrix} \quad \text{where } b_{i,j} = \begin{cases} f_{i,j} & , \text{GO hit} \\ 0 & , \text{otherwise} \end{cases} \quad (2)$$

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 may also contain important information for classification and therefore should not be quantized to either 0 or 1. Note that  $b_{i,j}$ 's are analogous to the term-frequencies commonly used in document retrieval.

- 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:

$$\mathbf{p}_i = \begin{bmatrix} c_{i,1} \\ \vdots \\ c_{i,j} \\ \vdots \\ c_{i,1203} \end{bmatrix}, \quad c_{i,j} = a_{i,j} \log \left( \frac{N}{|\{k : a_{k,j} \neq 0\}|} \right) \quad (3)$$

where  $N$  is the number of protein sequences in the 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  $j$ -th element, or equivalently the number of sequences with the  $j$ -th GO term as determined by InterProScan. Note that the logarithmic term in Eq. 3 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.

- 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:

$$\mathbf{p}_i = \begin{bmatrix} d_{i,1} \\ \vdots \\ d_{i,j} \\ \vdots \\ d_{i,1203} \end{bmatrix}, \quad d_{i,j} = b_{i,j} \log \left( \frac{N}{|\{k : b_{k,j} \neq 0\}|} \right) \quad (4)$$

where  $b_{i,j}$  is defined in Eq. 2.

## 2.2. Post-processing of GO Vectors

While the raw GO vectors can be directly applied to support vector machines (SVMs) for classification, better performance may be obtained by post-processing the raw vectors before SVM classification. Here we introduce two methods of post-processing: (1) vector norm and (2) geometric mean.

- Vector Norm.** Given the  $i$ -th GO training vector  $\mathbf{p}_i$ , the vector is normalized as:

$$\mathbf{x}_i^{(v)} = [x_{i,1}^{(v)}, \dots, x_{i,1203}^{(v)}]^\top \quad \text{where } x_{i,j}^{(v)} = \frac{p_{i,j}}{\|\mathbf{p}_i\|} \quad (5)$$

where the superscript  $(v)$  stands for vector norm, and  $p_{i,j}$  is the  $j$ -th element of  $\mathbf{p}_i$ . Similarly, given the  $i$ -th test vector  $\mathbf{p}'_i$ , the GO test vector is normalized as:

$$\mathbf{x}_i^{(v)'} = [x_{i,1}^{(v)'}, \dots, x_{i,1203}^{(v)'}]^\top \quad \text{where } x_{i,j}^{(v)'} = \frac{p'_{i,j}}{\|\mathbf{p}'_i\|}$$

- Geometric Mean.** This method involves pairwise comparison of GO vectors, followed by normalization.

**-Pairwise Comparison:** Denote  $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_T]^\top$  as a  $T \times 1203$  matrix whose rows are the raw GO vectors of  $T$  training sequences. Given the  $i$ -th GO training vector  $\mathbf{p}_i$ , we compute the dot products between  $\mathbf{p}_i$  and each of the training GO vectors to obtain a  $T$ -dim vector:

$$\mathbf{x}_i = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_T]^\top \mathbf{p}_i = \mathbf{P} \mathbf{p}_i, \quad i = 1, \dots, T.$$

During testing, given the  $i$ -th test vector  $\mathbf{p}'_i$ , we compute

$$\mathbf{x}_i' = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_T]^\top \mathbf{p}'_i = \mathbf{P} \mathbf{p}'_i, \quad i = 1, \dots, T'$$

where  $T'$  is the number of test vectors (sequences).

**-Normalization:** The  $j$ -th elements of  $\mathbf{x}_i$  is divided by the geometric mean of the  $i$ -th element of  $\mathbf{x}_i$  and the  $j$ -th element of  $\mathbf{x}_j$ , leading to the normalized vectors:

$$\mathbf{x}_i^{(g)} = [x_{i,1}^{(g)}, \dots, x_{i,T}^{(g)}]^\top \quad \text{where } x_{i,j}^{(g)} = \frac{x_{i,j}}{\sqrt{x_{i,i} x_{j,j}}} \quad (6)$$

where the superscript  $(g)$  stands for geometric mean. Note that pairwise comparison guarantees that the elements  $x_{i,i}$  and  $x_{j,j}$  exist for  $i, j = 1, \dots, T$ .

## 2.3. Multiclass SVM Classification

After GO vector construction and post-processing, the vectors  $\mathbf{p}_i$ ,  $\mathbf{x}_i^{(v)}$ , or  $\mathbf{x}_i^{(g)}$  can be used for training one-vs-rest SVMs. Specifically, for an  $M$ -class problem (here  $M$  is the number of subcellular locations),  $M$  independent SVM classifiers are

trained. During testing, given an unknown protein with GO vector  $\mathbf{p}'$ , the output of the  $m$ -th SVM is

$$s_m^{\text{GO}}(\mathbf{p}') = \sum_{r \in \text{SV}_m^{\text{GO}}} \alpha_{m,r}^{\text{GO}} y_{m,r}^{\text{GO}} K^{\text{GO}}(\mathbf{p}_r, \mathbf{p}') + b_m^{\text{GO}}$$

where  $\text{SV}_m^{\text{GO}}$  is the set of support vector indexes corresponding to the  $m$ -th SVM,  $y_{m,r}^{\text{GO}} \in \{-1, +1\}$  are the class labels,  $\alpha_{m,r}^{\text{GO}}$  are the Lagrange multipliers, and  $K^{\text{GO}}(\mathbf{p}_r, \mathbf{p}')$  is a kernel function. The form of  $K^{\text{GO}}(\mathbf{p}_r, \mathbf{p}')$  depends on the post-processing method being used. For example, if vector norm is used for normalization, the kernel becomes

$$K^{\text{GO}}(\mathbf{p}_r, \mathbf{p}') = \langle \mathbf{x}_r^{(v)}, \mathbf{x}^{(v)'} \rangle$$

The SVM score  $s_m^{\text{GO}}(\mathbf{p}')$  will be fused with the score of the profile alignment SVM described next.

### 3. PROFILE ALIGNMENT METHOD

This method extracts the features from protein sequences by aligning the profiles of the sequences with each of the training profiles [10]. A profile is a matrix in which elements in a column (sequence position) specify the frequency of individual amino acids appeared in the corresponding position of some homologous sequences. The profile of a sequence can be obtained by presenting the sequence to PSI-BLAST [17] that searches against a protein database for homologous sequences. The information pertaining to the aligned sequences is represented by two matrices: position-specific scoring matrix (PSSM) and position-specific frequency matrix (PSFM). Each entry of a PSSM represents the log-likelihood of the residue substitutions at the corresponding position in the query sequence. The PSFM contains the weighted observation frequencies of each position of the aligned sequences.

Given the  $i$ -th test protein sequence, we align its profile with each of the training profiles to obtain a profile-alignment test vector  $\mathbf{q}'_i$ , whose elements are then normalized by the geometric mean as follows:

$$\mathbf{q}'_i = [q'_{i,1}, \dots, q'_{i,T}]^T \text{ where } q'_{i,j} = \frac{q_{i,j}}{\sqrt{q'_{i,i} q_{j,j}}}$$

where  $q_{j,j}$  is the  $j$ -th element of the  $j$ -th training alignment vector.

Similar to the GO method, a one-versus-rest SVM classifier was used to classify the profile-alignment vectors. Specifically, the score of the  $m$ -th profile-alignment SVM is

$$s_m^{\text{PA}}(\mathbf{q}') = \sum_{r \in \text{SV}_m^{\text{PA}}} \alpha_{m,r}^{\text{PA}} y_{m,r}^{\text{PA}} K^{\text{PA}}(\mathbf{q}_r, \mathbf{q}') + b_m^{\text{PA}}$$

which is to be fused with the score of the GO SVM.

## 4. FUSION OF GO AND PROFILE ALIGNMENT

Fig 1 illustrates the fusion of GO and profile alignment methods. The GO and profile alignment scores produced by the GO and profile alignment SVMs are normalized by Z-norm:

$$\tilde{s}_m^{\text{GO}}(\mathbf{p}') = \frac{s_m^{\text{GO}}(\mathbf{p}') - \mu_m^{\text{GO}}}{\sigma_m^{\text{GO}}} \text{ and } \tilde{s}_m^{\text{PA}}(\mathbf{q}') = \frac{s_m^{\text{PA}}(\mathbf{q}') - \mu_m^{\text{PA}}}{\sigma_m^{\text{PA}}}$$

where  $(\mu_m^{\text{GO}}, \sigma_m^{\text{GO}})$  and  $(\mu_m^{\text{PA}}, \sigma_m^{\text{PA}})$  are respectively the mean and standard derivation of the GO and profile alignment SVM scores derived from the training sequences. The normalized GO and profile-alignment SVM scores are fused:

$$\tilde{s}_m^{\text{Fuse}}(\mathbf{p}', \mathbf{q}') = w^{\text{GO}} \tilde{s}_m^{\text{GO}}(\mathbf{p}') + w^{\text{PA}} \tilde{s}_m^{\text{PA}}(\mathbf{q}')$$

where  $w^{\text{GO}} + w^{\text{PA}} = 1$ . Finally, the predicted class of the test sequence is given by

$$m^* = \arg \max_{m=1}^M \tilde{s}_m^{\text{Fuse}}(\mathbf{p}', \mathbf{q}')$$

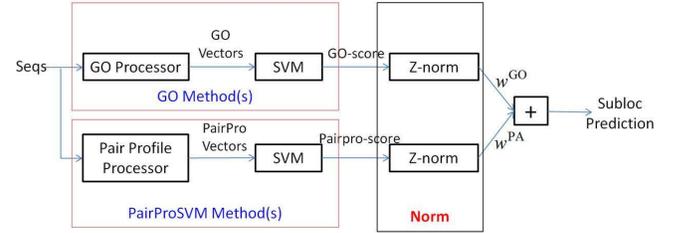


Fig. 1. Fusion of GO and profile alignment SVMs

## 5. EXPERIMENTS AND RESULTS

### 5.1. Datasets and Performance Metric

The performance was evaluated on Huang and Li's dataset [18], which was created by selecting all eukaryotic proteins with annotated subcellular locations from Swiss-Prot 41.0. The dataset comprises 3572 proteins with 11 classes (622 cytoplasm, 1188 nuclear, 424 mitochondria, 915 extracellular, 26 Golgi apparatus, 225 chloroplast, 45 endoplasmic reticulum, 7 cytoskeleton, 29 vacuole, 47 peroxisome, and 44 lysosome). The sequence similarity is cut off at 50%. Among the 3572 sequences, only 3120 sequences have valid GO vectors (with at least one non-zero element). For the remaining 452 sequences, InterProScan cannot find any GO terms. Therefore, we only used sequences with valid GO vectors in our experiments and reduced the dataset size to 3120 sequences.

Five-fold cross validation was used for performance evaluation. This ensures that every sequence in the dataset will be tested. The performance measures include the accuracy (ACC) and Mathew's correlation coefficient (MCC) [19]. The latter has the advantage of avoiding the performance to be dominated by the majority classes.

| Classifier | Feature             | Post-processing | Overall Acc.  | MCC           |
|------------|---------------------|-----------------|---------------|---------------|
| RBF-SVM    | AA                  | Vector Norm     | 54.29%        | 0.4972        |
| RBF-SVM    | AA+PairAA           | Vector Norm     | 56.47%        | 0.5212        |
| Linear SVM | AA+PairAA+GapAA(59) | Vector Norm     | 61.44%        | 0.5759        |
| RBF-SVM    | AA+PseAA            | Vector Norm     | 57.98%        | 0.5378        |
| Linear SVM | Profile Alignment   | Geometric Mean  | <b>77.05%</b> | <b>0.7476</b> |

**Table 1.** Performance obtained by using amino acid composition (AA) [1], amino-acid pair composition (PairAA) [1], AA composition with gap (length = 59) (GapAA) [3], pseudo AA composition (PseAA) [4], and profile alignment scores as feature vectors and different SVMs as classifiers. The last row corresponds to the PairProSVM proposed in [10].

| Method ID | GO Vector Construction Method | Post-processing Method | Overall Acc.  | MCC           |
|-----------|-------------------------------|------------------------|---------------|---------------|
| GO_1      | 1-0 value                     | None                   | <b>72.21%</b> | <b>0.6943</b> |
| GO_2      | ISF                           | None                   | 71.89%        | 0.6908        |
| GO_3      | TF                            | None                   | 71.99%        | 0.6919        |
| GO_4      | TF-ISF                        | None                   | 71.15%        | 0.6827        |
| GO_5      | 1-0 value                     | Vector Norm            | 71.25%        | 0.6837        |
| GO_6      | ISF                           | Vector Norm            | 72.02%        | 0.6922        |
| GO_7      | TF                            | Vector Norm            | 70.96%        | 0.6806        |
| GO_8      | TF-ISF                        | Vector Norm            | 71.73%        | 0.6890        |
| GO_9      | 1-0 value                     | Geometric Mean         | 70.51%        | 0.6756        |
| GO_10     | ISF                           | Geometric Mean         | 72.08%        | 0.6929        |
| GO_11     | TF                            | Geometric Mean         | 70.64%        | 0.6771        |
| GO_12     | TF-ISF                        | Geometric Mean         | 71.03%        | 0.6813        |

**Table 2.** Performance of GO methods using different approaches to constructing the raw GO vectors and different post-processing approaches to normalizing the raw GO vectors. ‘None’ in Post-processing means that the raw GO vectors  $\mathbf{p}_i$  are used as input to the SVMs. *ISF*: inverse sequence-frequency; *TF*: term-frequency; *TF-ISF*: term-frequency inverse sequence frequency.

| Method I | Optimal $w^{GO}$ | Overall Acc.  | MCC           |
|----------|------------------|---------------|---------------|
| GO_1     | 0.4490           | 78.91%        | 0.7680        |
| GO_2     | 0.2643           | 78.56%        | 0.7641        |
| GO_3     | 0.3970           | 78.75%        | 0.7662        |
| GO_4     | 0.3693           | 78.72%        | 0.7659        |
| GO_5     | 0.3711           | 78.78%        | 0.7666        |
| GO_6     | 0.3428           | 78.78%        | 0.7666        |
| GO_7     | 0.4263           | 78.81%        | 0.7670        |
| GO_8     | 0.2947           | 78.40%        | 0.7624        |
| GO_9     | 0.4186           | 78.97%        | 0.7687        |
| GO_10    | 0.4515           | <b>79.04%</b> | <b>0.7694</b> |
| GO_11    | 0.3993           | 78.37%        | 0.7620        |
| GO_12    | 0.3670           | 78.62%        | 0.7648        |

**Table 3.** Performance of the fusion of GO Methods and PairProSVM .

## 5.2. Performance of Individual Predictors

Table 1 shows the performance of different SVMs using various features extracted from the protein sequences. The features include amino acid composition (AA) [1], amino-acid pair composition (PairAA) [1], AA composition with the maximum gap length equal to 59 (the minimum length of all of the 3120 sequences is 61) [3], pseudo AA composition [4], and profile alignment scores. The penalty factor for training the SVMs was set to 1 for both linear SVM and RBF-SVM. For RBF-SVMs the kernel parameter was set to 1. As AA and PairAA produce low-dimensional feature vectors, the performance achieved by RBF-SVM is better than that of the linear SVM. So, we just present the performance of RBF-SVM.

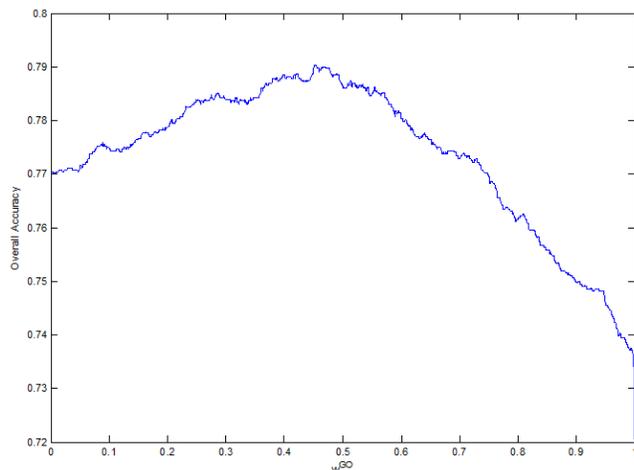
Table 1 shows that amino-acid composition and its variant

are not good features for subcellular localization. The highest accuracy is only 61.44%. On the other hand, the homology-based method that exploits the homologous sequences in protein databases (via PSI-BLAST) achieves a significant better performance. This suggests that the information pertaining to the amino acid sequences is limited.

Table 2 shows the performance of 12 GO methods. For ease of reference, we label these methods as GO\_1, GO\_2, ..., GO\_12. Linear SVMs were used in all cases. When using vector norm or geometric mean to post-process the GO vectors, the inverse sequence-frequency can produce more discriminated GO vectors, as evident in the higher accuracy and MCC corresponding to GO\_6 and GO\_10. Except for ISF, using the raw GO vectors as the SVM input achieves the best performance, as evident in the higher accuracy and MCC corresponding to GO\_1, GO\_3, and GO\_4. This suggests that post-processing could remove some of the subcellular localization information pertaining to the raw GO vectors.

## 5.3. Performance of Fusion Predictor

Table 3 shows the performance of fusing the GO methods and PairProSVM. The performance was obtained by optimizing the fusion weights  $w^{GO}$  (based on the test dataset). The results show that the combination of PairProSVM and GO\_10 (ISF with geometric mean) achieves the highest accuracy—79.04%, which is significant better than PairProSVM (77.05%) and the GO method (72.21%) alone. The results also suggest that fusion of PairProSVM and any of the GO methods can outperform the individual methods. This



**Fig. 2.** Performance of fusing of GO\_10 and PairProSVM varying with respect to the fusion weight  $w^{GO}$

is mainly because the information obtained from homology search and from functional domain databases has different perspectives and is therefore complementary to each other. Surprisingly, fusing the best performing GO method and profile-alignment method does not give the best performance.

Fig 2 shows the performance of fusing GO\_10 and PairProSVM by varying  $w^{GO}$  from 0 to 1. As can be seen, the performance changes steadily with the change of  $w^{GO}$ . Further, the p-value between the accuracy of the fusion system (GO\_10 and PairProSVM) and the PairProSVM system is 0.0055, which suggests that the performance of the fusion predictor is significantly better than that of the PairProSVM predictor.

## 6. CONCLUSIONS

This paper proposes fusing homology-based methods (PairProSVM) and functional-domain based methods to predict protein's subcellular locations. Gene ontology (GO) vectors are produced by presenting protein sequences to InterProScan and considering the GO terms as the axes of a high-dimensional Euclidean space and the existence or number of occurrences of GO terms as coordinates. The GO vectors are further post-processed by normalizing with their vector norm or by the geometric mean of the pairwise dot products. Results show that homology-based methods that exploit sequence and profile similarities and functional-domain based methods that exploit the GO annotations consider the subcellular localization problem from different perspectives, thus providing significant complementary information for enhancing classification performance. This paper also demonstrates that these two types of methods are far more advantageous than the amino-acid composition based methods.

## 7. REFERENCES

[1] H. Nakashima and K. Nishikawa, "Discrimination of intracellular and extracellular proteins using amino acid composition and residue-pair

- frequencies," *J. Mol. Biol.*, pp. 54–61, 1994, 238.
- [2] K.C. Chou and Y.D. Cai, "Predicting protein localization in budding yeast," *Bioinformatics*, pp. 944–950, 2005, 21.
- [3] K.J. Park and M. Kanehisa, "Prediction of protein subcellular locations by support vector machines using compositions of amino acid and amino acid pairs," *Bioinformatics*, pp. 1656–1663, 2003, 19.
- [4] K.C. Chou, "Prediction of protein cellular attributes using pseudo-amino-acid-composition," *Proteins*, pp. 246–255, 2001, 43.
- [5] K. Nakai, "Protein sorting signals and prediction of subcellular localization," *Advances in Protein Chemistry*, vol. 54, no. 1, pp. 277–344, 2000.
- [6] K. Nakai and M. Kanehisa, "Expert system for predicting protein localization sites in gram-negative bacteria," *Proteins: Structure, Function, and Genetics*, vol. 11, no. 2, pp. 95–110, 1991.
- [7] P. Horton, K. J. Park, T. Obayashi, and K. Nakai, "Protein subcellular localization prediction with WOLF PSORT," in *Proc. 4th Annual Asia Pacific Bioinformatics Conference (APBC06)*, 2006, pp. 39–48.
- [8] R. Nair and B. Rost, "Sequence conserved for subcellular localization," *Protein Science*, vol. 11, pp. 2836–2847, 2002.
- [9] Z. Lu, D. Szafron, R. Greiner, P. Lu, D. S. Wishart, B. Poulin, J. Anvik, C. Macdonell, and R. Eisner, "Predicting subcellular localization of proteins using machine-learned classifiers," *Bioinformatics*, vol. 20, no. 4, pp. 547–556, 2004.
- [10] M.W. Mak, J. Guo, and S.Y. Kung, "PairProSVM: Protein subcellular localization based on local pairwise profile alignment and SVM," *IEEE/ACM Trans. on Computational Biology and Bioinformatics*, vol. 5, no. 3, pp. 416–422, 2008.
- [11] K.C. Chou and H.B. Shen, "Hum-PLoc: A novel ensemble classifier for predicting human protein subcellular localization," *Biochemical and Biophysical Research Communications*, pp. 150–157, 2006, 347.
- [12] W.L. Huang, C.W. Tung, S.W. Ho, S.F. Hwang, and S.Y. Ho, "ProLoc-GO: Utilizing informative Gene Ontology terms for sequence-based prediction of protein subcellular localization," *BMC Bioinformatics*, 2008.
- [13] O. Emanuelsson, H. Nielsen, S. Brunak, and G. von Heijne, "Predicting subcellular localization of proteins based on their N-terminal amino acid sequence," *J. Mol. Biol.*, vol. 300, no. 4, pp. 1005–1016, 2000.
- [14] O. Emanuelsson, S. Brunak, G. von Heijne, and H. Nielsen, "Locating proteins in the cell using TargetP, SignalP, and related tools," *Nature Protocols*, vol. 2, no. 4, pp. 953–971, 2007.
- [15] W. Wang, M. W. Mak, and S. Y. Kung, "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)*, 2010, pp. 147–154.
- [16] K.C. Chou and H.B. Shen, "Euk-mPloc: A fusion classifier for large-scale eukaryotic protein subcellular location prediction by incorporating multiple sites," *Journal of Proteome Research*, pp. 1728–1734, 2007, 6.
- [17] S.F. Altschul, T.L. Madden, A.A. Schafer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman, "Gapped BLAST and PSI-BLAST: A new generation of protein database search programs," *Nucleic Acids Res.*, vol. 25, pp. 3389–3402, 1997.
- [18] Y. Huang and Y. D. Li, "Prediction of protein subcellular locations using fuzzy K-NN method," *Bioinformatics*, vol. 20, no. 1, pp. 21–28, 2004.
- [19] B.W. Matthews, "Comparison of predicted and observed secondary structure of t4 phage lysozyme," *Biochem. Biophys. Acta*, vol. 405, pp. 442–451, 1975.