The following publication S. Wan, M. -W. Mak and S. -Y. Kung, "Mem-mEN: Predicting Multi-Functional Types of Membrane Proteins by Interpretable Elastic Nets," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 13, no. 4, pp. 706-718, July-Aug. 2016 is available at https://doi.org/10.1109/TCBB.2015.2474407.

Mem-mEN: Predicting Multi-Functional Types of Membrane Proteins by Interpretable Elastic Nets

Shibiao Wan, Man-Wai Mak, *Senior Member, IEEE* and Sun-Yuan Kung, *Fellow, IEEE*

Abstract—Membrane proteins play important roles in various biological processes within organisms. Predicting the functional types of membrane proteins is indispensable to the characterization of membrane proteins. Recent studies have extended to predicting single- and multi-type membrane proteins. However, existing predictors perform poorly and more importantly, they are often lack of interpretability. To address these problems, this paper proposes an efficient predictor, namely Mem-mEN, which can produce sparse and interpretable solutions for predicting membrane proteins with single- and multi-label functional types. Given a query membrane protein, its associated gene ontology (GO) information is retrieved by searching a compact GO-term database with its homologous accession number, which is subsequently classified by a multi-label elastic net (EN) classifier. Experimental results show that Mem-mEN significantly outperforms existing state-of-the-art membrane-protein predictors. Moreover, by using Mem-mEN, 338 out of more than 7,900 GO terms are found to play more essential roles in determining the functional types. Based on these 338 essential GO terms, Mem-mEN can not only predict the functional type of a membrane protein, but also explain why it belongs to that type. For reader's convenience, the Mem-mEN server is available online at http://bioinfo.eie.polyu.edu.hk/MemmENServer/.

Index Terms—membrane protein type prediction; multi-label classification; interpretable predictor; elastic net; gene ontology.

1 Introduction

Membrane proteins, which interact with the membranes of a cell or an organelle, play essential roles in a variety of vital biological processes [1]. Because membrane proteins mediate many interactions between cells and extracellular surroundings as well as between the cytosol and membrane-bound organelles, almost half of all drug targets contain a membrane domain [2]. Although membrane proteins are located at the membrane and often have the same basic phospholipid bilayer structure [3], they perform various and diversified functions. This diversity is manifested by the remarkably different functional types of membrane proteins.

Traditionally, depending upon the interactions between membrane proteins and the membrane, some studies [3] broadly classified membrane proteins into two categories, namely integral (or intrinsic) membrane proteins and peripheral (or extrinsic) membrane proteins. Other studies [4] grouped membrane proteins into three distinct classes: integral, peripheral and lipid-anchored. Integral membrane proteins are permanently bound to the biological membrane. Peripheral membrane proteins are temporarily attached to a membrane or integral membrane proteins. Lipid-anchored membrane proteins are covalently linked to a lipid molecule and serve to anchor them to either the cytoplasmic or extracellular surface of a biological membrane.

With the avalanche of protein sequences generated in the post-genomic era, these three groups of membrane proteins are further divided into eight types [5]: (1) single-pass type I; (2) single-pass type II; (3) single-pass type III; (4) single-pass type IV; (5) multi-pass; (6) lipid-anchor; (7) GPI-anchor and (8) peripheral. The hierarchical relationships between these eight types and the former three groups are shown in Fig. 1. As can be seen, the former five types belong to integral membrane proteins, Types 6 and 7 belong to lipid-anchored proteins and the last type belongs to peripheral proteins. GPI-anchored proteins is a kind of special lipid-anchored proteins.[1] Due to the fact that GPI-anchored proteins ubiquitously exist in many species and have been intensively studied for their unique functions [6], Type 7 (GPI-anchored proteins) is singled out from Type 6. The definitions of these eight functional types are detailed in Table 1. As can be seen, the integral membrane proteins are more complicated, containing four different types of single-pass membrane proteins and one type of multi-pass membrane proteins.

The functional types of membrane proteins can be directly used to infer the biological functions of membrane proteins. For example, phospholipases [7], belonging to Type 8, are a group of water-soluble enzymes that are temporarily associated with the polar head groups of membrane phospholipids. Their major functions are lipid signaling. This is achieved by hydrolizing various bonds linking phospholipases with the lipid layer with which they are temporarily associated. Due to the nature

- *Shibiao Wan and Man-Wai Mak are with the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong SAR, China.*
  E-mail: shibiao.wan@connect.polyu.hk and enmwmak@polyu.edu.hk
- *Sun-Yuan Kung is with the Department of Electrical Engineering, Princeton University, New Jersey, USA.*
  E-mail: kung@princeton.edu

1. http://www.uniprot.org/locations/SL-9902

of their fluidity, membrane proteins can freely move within the lipid bilayer to the place where their functions are required. Knowing the type of a membrane protein can help reveal the mechanisms of this kind of biological activities. Moreover, although about 20%∼35% of genes encode membrane proteins, the structurally annotated membrane proteins only account for less than 1% of the proteins with known structures [8]. Knowing the functional types of membrane proteins can accelerate the process of annotating their structures. Therefore, it is highly required to develop computational methods for fast and accurate prediction of membrane protein functional types.

Recent decades have witnessed remarkable progress in predicting functional types of membrane proteins [5], [8]–[14]. Some other studies [15], [16] are focusing on predicting membrane proteins in particular subcellular locations, such as lysosome [15] or Golgi [16]. However, these predictors are limited to the prediction of membrane proteins with single-label functional types. They are based on the assumption that a membrane protein rarely belongs to more than one functional type. Actually, there exist many membrane proteins that simultaneously belong to multiple functional types. For example, the envelope glycoprotein p57 [17], [18] is reported to belong to single-pass type I (Type 1) when locating in the host endoplasmic reticulum membrane, and simultaneously it belongs to peripheral (Type 8) when locating in the host cell membrane. To the best of our knowledge, only two predictors, namely iMem-Seq [19] and Mem-PseAA [20],[2] are able to predict multi-label membrane proteins. iMem-Seq extracts the information from position-specific score matrices and physical-chemical property matrices, whereas Mem-PseAA extracts feature information from pseudo-amino acid compositions. It has been shown [19] that the former performs better than the latter. However, both of these predictors still perform poorly. More importantly, while these predictors can determine the functional type(s) of a query protein, they fail to provide biological reasons on why the query protein belongs to the predicted type(s).

To tackle these problems, this paper proposes an efficient multi-label predictor, namely Mem-mEN, which leverages a multi-label elastic net (EN) classifier for predicting membrane proteins with single- and multi-label functional types. Unlike the previous studies [19], [20] in which features are exclusively extracted from the amino acid sequences, Mem-mEN extracts features by exploiting the gene ontology (GO) information retrieved from a GO-term database. By using a multi-label EN classifier, 338 out of 7,900+ GO terms were selected as features. With the selected GO terms, the original high-dimensional feature vectors are converted into low-dimensional vectors, which were subsequently classified by another multi-label elastic net classifier.

2. For ease of reference, we named Mem-PseAA for the predictor proposed in [20].

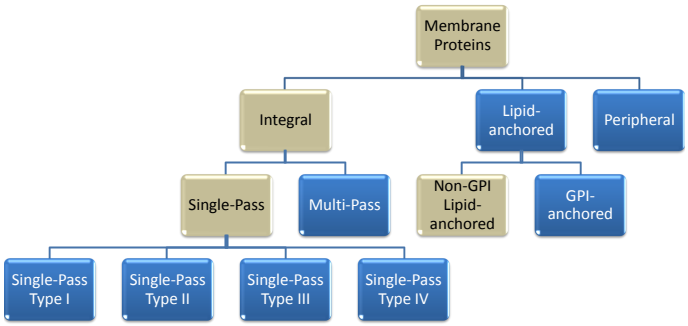


Fig. 1. Hierarchical relationships between different types of membrane proteins. The blue boxes are the eight types of membrane proteins studied in this paper.

Experimental results on two recent benchmark datasets demonstrate the superiority of Mem-mEN over existing state-of-the-art predictors. More importantly, based on the 338 selected GO terms, not only can Mem-mEN decide the functional type(s) of a query membrane protein, but it also provides reasons on why the protein belongs to the predicted type(s). This work also found that in addition to cellular-component GO terms, GO terms from the other two categories also play important roles in the predictions.

2 Feature Extraction

Before introducing the proposed predictor, for readers' convenience, the definitions of all symbols used in this paper are summarized in Table 2.

2.1 GO Terms as Features

In the past decade, GO-based methods have been successively applied to protein subcellular localization prediction [21]–[26]. Extensive analyses and comparisons among different GO-based predictors have been reported in a recent book [26]. One of the challenges that GO-based approaches are facing is how to deal with query proteins whose GO information is not available in the gene ontology annotation (GOA) database.[3] This situation is especially prevalent in novel proteins. Traditionally, if the accession number (AC) of a query protein does not associate with any GO terms in the GOA database, BLAST search [27] is used to find the AC of the top homolog of the query protein. Then, the homologous AC is searched against the GOA database to find a set of GO terms and a GO vector (see Section 2.2 below) can be constructed. This strategy effectively transfers the homologous GO information to the query protein. This strategy, however will still lead to null GO vectors when the top homologous protein has not been annotated in the GOA database, i.e., its AC does not associate with any GO terms. To address this problem, some predictors [28], [29] use the AC of lower-rank homologs as a

3. http://www.ebi.ac.uk/GOA

TABLE 1
Definitions of different membrane protein types.

Supertype	Type Number	Membrane Type	Definition
Integral	1	Single-pass type I	Spanning the membrane once, with its N-terminus on the extracellular side.
	2	Single-pass type II	Spanning the membrane once, with its N-terminus on the cytoplasmic side.
	3	Single-pass type III	Similar to Single-pass Type I except no signal sequence.
	4	Single-pass type IV	Similar to Single-pass Type II except transmembrane domain located close to the C-terminus.
	5	Multi-pass	Spanning the membrane more than once.
Lipid-anchor	6	Lipid-anchor	Bound to the lipid bilayer through a post-translational modification (PTM).
	7	GPI-anchor	Bound to the lipid bilayer by a GPI-anchor.
Peripheral	8	Peripheral	Temporarily bound to the lipid layer or integral membrane proteins.

replacement until a non-null GO vector can be found. Some others give up using GO information and apply back-up methods that rely on other features such as pseudo-amino-acid composition [30] and sorting signals [31]. Nevertheless, the backup methods usually lead to poor prediction accuracy for these special proteins.

Another issue of GO-based methods is to minimize prediction time. In this case, it is necessary to store the mapping between the ACs and their GO terms as a hash map or hash table in memory. Given the large number of ACs in the GOA database, the hash map will easily occupy tens of gigabytes of memory. Given the rapid increase in the number of entries in the GOA database, the memory consumption will be further increased in the future. To overcome this storage complexity and to avoid null-GO vectors, we have previously proposed to filter the Swiss-Prot and GOA database to form two compact yet efficient databases called ProSeq and ProSeq-GO [32]. The former is a subset of Swiss-Prot whereas the latter is a subset of the GOA database. Null-GO vectors can be avoided because the filtering process guarantees that ProSeq will only keep the sequences whose ACs have at least one GO term in ProSeq-GO. As a result, all of the homologous ACs will be associated with at least one GO terms and the GO vectors will have at least one non-null entry. The ProSeq-GO also reduces memory consumption from tens of gigabytes to several hundred megabytes because the number of ACs in ProSeq-GO is substantially smaller than that in the GOA database.

In this work, we use ProSeq and ProSeq-GO databases to construct the GO vectors, which will be elaborated in Section 2.2 below.

TABLE 2
Definitions of symbols used in this paper.

Symbol	Definition
N	Number of training proteins
T	Number of distinct GO terms
M	Number of classes (membrane types)
i	Protein index ($i = 1, \ldots, N$)
j	GO-term index ($j = 1, \ldots, T$)
m	Class index ($m = 1, \ldots, M$)
$\mathbb{Q}_i$	The i-th training protein
$\mathbf{q}_i$, $\mathbf{x}_i$	The i-th GO vector
$f_{i,j}$	The j-th element of $\mathbf{q}_i$
y_i	Label of $\mathbf{x}_i$ for binary classification
$\boldsymbol{\beta}$	Weight vector
β_j	The j-th element of $\boldsymbol{\beta}$
γ	Ridge regression penalty
λ	LASSO penalty
$\mathcal{Y}_i$	Label set of $\mathbf{x}_i$ for multi-label classification
$y_{i,m}$	The i-th transformed label for the m-th class
$\boldsymbol{\beta}_m$	Weight vector for the m-th class
$\hat{\boldsymbol{\beta}}_m$	The optimized weight vector for the m-th class
$\beta_{j,m}$	The j-th element of $\boldsymbol{\beta}_m$
γ_m	Ridge regression penalty for the m-th class
λ_m	LASSO penalty for the m-th class
S	Number of selected GO terms
$\mathbf{x}_t$	The t-th test vector
$\mathbf{x}_t^s$	The t-th test vector after feature selection
$\boldsymbol{\alpha}_m$	Same as $\boldsymbol{\beta}_m$ except after feature selection
$\hat{\boldsymbol{\alpha}}_m$	Same as $\hat{\boldsymbol{\beta}}_m$ except after feature selection
$\alpha_{j,m}$	Same as $\beta_{j,m}$ except after feature selection
$\mathcal{M}^*(\mathbb{Q}_t)$	Predicted functional type(s) of the t-th protein

2.2 Construction of GO Frequency Vectors

Two steps are needed for constructing GO frequency vectors: (1) retrieval of GO terms; and (2) construction of GO vectors.

For retrieving GO terms, given a query protein, its amino acid sequence is presented to BLAST [27] to find its homologs in the ProSeq database. The homologous ACs are then used as keys to search against the ProSeq-GO database. We used the default parameter setting for BLAST in our experiments.

For constructing GO vectors, given a dataset, the GO terms of all of its proteins are retrieved by the procedures described above. Because term-frequency (TF) based GO vectors [28], [29] were found to perform better than the conventional 1-0 vectors, we adopted the TF method to construct GO vectors. Let $\mathbb{T}$ denotes a set of distinct GO terms corresponding to a dataset of interest. $\mathbb{T}$ is constructed in two steps: (1) identifying all of the GO terms in the dataset and (2) removing the repetitive GO terms. Suppose T distinct GO terms are found, i.e., $|\mathbb{T}| = T$; these GO terms form a GO Euclidean space with T dimensions. For each protein sequence in the dataset, a GO vector is constructed by matching its GO

terms against $\mathbb{T}$, using the number of occurrences of individual GO terms in $\mathbb{T}$ as the coordinates. Specifically, the frequency GO vector $\mathbf{q}_i$ of the i-th protein $\mathbb{Q}_i$ is defined as:

$$\mathbf{q}_i = [f_{i,1}, \ldots, f_{i,j}, \ldots, f_{i,T}]^{\mathsf{T}}, \quad (1)$$

where $f_{i,j}$ is the number of occurrences of the j-th GO term (term-frequency) in the i-th protein sequence. Detailed information about GO vectors can be found in [28], [29].

3 Multi-label Elastic Net Classifier

The elastic net (EN) [33] is a linear sparse regression model. It can produce "parsimonious" solutions that enable us to find a set of features that are the most relevant to the problem (target variables) being addressed. Learning in EN is achieved by imposing $(L_1 + L_2)$-regularized constraints on the weights associated with the features. A similar L_1-regularized linear regression model is LASSO [34] (Least Absolute Shrinkage and Selection Operator). The L_1 constraint in LASSO forces the weights of some features to exactly zero [35], and hence LASSO can automatically select relevant features. However, LASSO tends to force many weights to zeros in order to produce a sparse solution, causing some important information to be missed. EN can overcome this disadvantage. The convex combination of L_1 and L_2 penalties in EN can yield sparse representations similar to LASSO, while encouraging correlated features to be selected or deselected together [33]. Actually, LASSO can be regarded as a special case of EN. EN has been extensively used in various bioinformatics domains, such as single nucleotide polymorphism (SNP) selection [36], genetic trait prediction [37], ICU mortality risk detection [38], etc.

3.1 Objective Function of Elastic Net

Although EN is a kind of sparse regression models, it is applicable to both feature selection and classification. Suppose for a two-class single-label problem, we are given a set of training data $\{\mathbf{x}_i, y_i\}_{i=1}^N$, where $\mathbf{x}_i \in \mathcal{R}^T$ and $y_i \in \{-1, 1\}$. In our case, $\mathbf{x}_i = \mathbf{q}_i$, where $\mathbf{q}_i$ is defined in Eq. 1.

Specifically, an ordinary least squares (OLS) model is written as:

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{N}(y_i - f(\mathbf{x}_i))^2 = \sum_{i=1}^{N}\left(y_i - \varepsilon_0 - \sum_{j=1}^{T}\beta_j x_{i,j}\right)^2, \quad (2)$$

where $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_j, \ldots, \beta_T]^{\mathsf{T}}$ is the weight vector to be optimized, ε_0 is a bias,[4] and $x_{i,j}$ is the j-th element of $\mathbf{x}_i$.

4. For ease of presentation, we omitted the bias in equations thereafter.

EN is to impose an $(L_1 + L_2)$-style regularization on Eq. 2. Thus, the object function of EN is:

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{N}\left(y_i - \boldsymbol{\beta}^{\mathsf{T}}\mathbf{x}_i\right)^2 + \lambda\sum_{j=1}^{T}|\beta_j| + \gamma\sum_{j=1}^{T}{\beta_j}^2, \quad (3)$$

where $\gamma > 0$ and $\lambda > 0$ are the penalty parameters controlling the ridge regression penalty and LASSO penalty, respectively.

In Eq. 3, when $\lambda = 0$, Eq. 3 becomes simple ridge regression, namely an L_2-regularized linear model; when $\gamma = 0$, Eq. 3 becomes a LASSO model, namely an L_1-regularized linear model. LASSO is a convex optimization problem, which can be efficiently solved by the famous least angle regression (LARS) [39] method. By a simple transformation, Eq. 3 can be converted to an equivalent LASSO-style problem on augmented data [33]. Because of this property, Eq. 3 can be solved by the same way as LASSO by absorbing the L_2-norm term into the objective function. Detailed descriptions of the solutions can be found in [33]. In terms of choosing the penalty parameters (λ, γ), we used a two-dimensional grid-search method similar to [33], except that we used five-fold cross-validation instead of ten-fold cross-validation for computational simplicity. Only the results with the optimized (λ, γ) are reported in this paper. Details of parameter optimization and runtime analysis can be found in the supplementary materials in the Mem-mEN web-server.

3.2 Multi-label EN for Feature Selection

In an M-class multi-label problem, the training data set is written as $\{\mathbf{x}_i, \mathcal{Y}_i\}_{i=1}^N$, where $\mathbf{x}_i \in \mathcal{R}^T$ and $\mathcal{Y}_i \subset \{1, 2, \ldots, M\}$ is a set which may contain one or more labels.

For the multi-label EN, M independent binary one-vs-rest ENs are trained, one for each class. The labels $\{\mathcal{Y}_i\}_{i=1}^N$ are converted to *transformed labels* [29] $y_{i,m} \in \{-1, 1\}$, where $i = 1, \ldots, N$, and $m = 1, \ldots, M$. Then, the optimal weight vector for the m-th class is given by:

$$\hat{\boldsymbol{\beta}}_m = \arg\min_{\boldsymbol{\beta}_m}\left\{\sum_{i=1}^{N}\left(y_{i,m} - \boldsymbol{\beta}_m^{\mathsf{T}}\mathbf{x}_i\right)^2 + \lambda_m\sum_{j=1}^{T}|\beta_{j,m}| + \gamma_m\sum_{j=1}^{T}{\beta_{j,m}}^2\right\}, \quad (4)$$

where $m = 1, \ldots, M$, $\{y_{i,m}\}_{i=1}^N \in \{-1, 1\}$, λ_m and γ_m are the L_1 penalized parameter and the L_2 penalized parameter for the m-th class, respectively. Since L_1 regularization tends to force some weights $\{\beta_{j,m}\}_{j=1}^T$ for the m-th class to exactly zero, EN can be used for feature selection.

Specifically, the GO vectors obtained from Eq. 1 are used for training multi-label one-vs-rest EN classifiers. For an M-class problem (here M is the number of subcellular locations), M independent binary EN classifiers

are trained, one for each class. After training, the union of those GO terms whose weights are not zero in any one of the M classes constitute the selected features. EN can significantly remove those irrelevant features (or GO terms). Suppose S out of the T weights are nonzero. They are defined as $\{\beta_{s,m}\}_{s=\{1,\ldots,S\},m=\{1,\ldots,M\}}$ and their corresponding GO terms are called *essential GO terms*. In fact, in our experiments, through our proposed multi-label EN classifiers, 338 out of 7954 GO terms were selected. This means that only around 4.25% of the GO terms are *essential GO terms* and the weights for about 95.75% of the 7954 GO terms are exactly zero.

3.3 Multi-label EN for Classification

Besides feature selection, EN can also be used for classification. Specifically, given the t-th query protein $\mathbb{Q}_t$, the feature vector $\mathbf{x}_t \in \mathcal{R}^T$ defined in Eq. 1 is obtained. Then, the elements of $\mathbf{x}_t$ with non-zero weights $\beta_{j,m}$ in Eq. 4 for EN are selected to form a low-dimensional feature vector represented by $\mathbf{x}_t^s \in \mathcal{R}^S$, where $S < T$ is the number of essential GO terms. Similar to the EN described in Section 3.2, for an M-class problem, M independent binary EN classifiers are trained, one for each class. Then, the score of the m-th EN is:

$$s_m(\mathbb{Q}_t) = \hat{\boldsymbol{\alpha}}_m^{\mathsf{T}} \mathbf{x}_t^s, \tag{5}$$

where $\hat{\boldsymbol{\alpha}}_m$ is given by

$$\hat{\boldsymbol{\alpha}}_m = \arg\min_{\boldsymbol{\alpha}_m} \left\{ \sum_{i=1}^{N} \left(y_{i,m} - \boldsymbol{\alpha}_m^{\mathsf{T}} \mathbf{x}_i^s \right)^2 + \lambda_m \sum_{j=1}^{S} |\alpha_{j,m}| + \gamma_m \sum_{j=1}^{S} {\alpha_{j,m}}^2 \right\}, \tag{6}$$

where $\boldsymbol{\alpha}_m = [\alpha_{1,m}, \ldots, \alpha_{j,m}, \ldots, \alpha_{S,m}]^{\mathsf{T}}$ is the weight vector to be optimized and $\mathbf{x}_i^s \in \mathcal{R}^S$ is the feature vector for the i-th training protein. Note that $\hat{\boldsymbol{\alpha}}_m$'s are obtained based on the training dataset only.

To predict membrane proteins with both single- and multi-label functional types, a decision scheme for multi-label EN classifiers should be used. Unlike the single-label problem where each protein has one predicted label only, a multi-label protein should have more than one predicted labels. This paper uses the decision scheme described in mGOASVM [29]. In this scheme, the predicted functional type(s) of the i-th query protein are given by:

$$\mathcal{M}^*(\mathbb{Q}_t) = \begin{cases} \bigcup_{m=1}^{M} \{m : s_m(\mathbb{Q}_t) > 0\}, \text{ where } \exists\ s_m(\mathbb{Q}_t) > 0\ ; \\ \arg\max_{m=1}^{M} s_m(\mathbb{Q}_t), \text{ otherwise.} \end{cases} \tag{7}$$

For ease of presentation, we refer to the proposed predictors as Mem-mEN.

4 Datasets and Performance Metrics

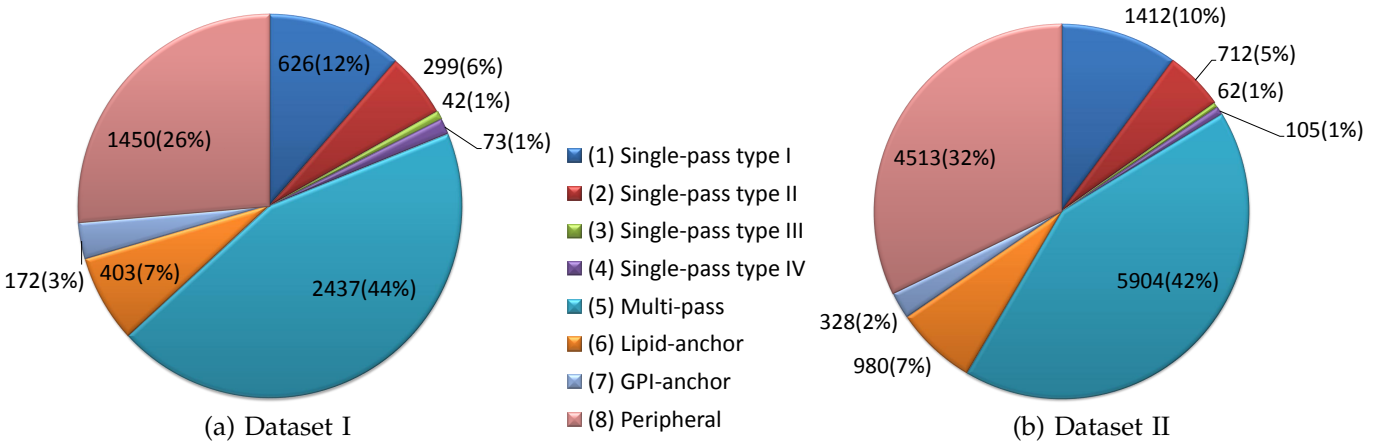
Two benchmark datasets [19], [20] were used to evaluate the performance of Mem-mEN. The breakdown of these two datasets are shown in Fig. 2. Datasets I [19] and II [20] were extracted from Swiss-Prot released in March 2013 and June 2012, respectively. In Dataset I, there are 5502 virtual proteins [19] corresponding to 5307 actual proteins, of which 5117 belong to one type, 185 to two types and 5 to three types. In Dataset II, there are 14,016 virtual proteins corresponding to 13,659 actual proteins, of which 13,313 belong to one type, 335 to two types and 11 to three types. The concept of *virtual proteins* is as follows: If a protein belongs to two functional types, then it will be counted as two virtual proteins; if a protein belongs to three types, then it will be counted as three virtual proteins; and so forth. As can be seen from Fig. 2, the majority (70%/74%) of membrane proteins in both datasets belong to multi-pass type and peripheral type, while proteins in other 6 types totally account for no more than 30% in both datasets. This means that both datasets are very imbalanced. The sequence identity of Dataset I was cut off at 25%, and that of Dataset II was cut off at 80%. Because Dataset I is much more stringent and recent than Dataset II, unless stated otherwise results and analyses reported below are based on Dataset I.

Performance metrics for multi-label classification are more sophisticated than those for single-label classification. This paper uses some popular multi-label evaluation metrics [19], [20], [40], [41], including *Hamming loss*, *Ranking loss*, *One-error*, *Coverage*, *Average precisions*, *Accuracy*, *Precision*, *Recall* and *Absolute true*. For the first four metrics, the smaller the better, and for the remaining metrics, the larger the better. Among these performance metrics, *Absolute true* (equivalent to *overall actual accuracy* in [42]) is the most objective and stringent [42]. The definitions of these metrics can be found in supplementary materials on the Mem-mEN server.

5 Results and Analysis

5.1 Statistical Analyses of Essential GO Terms

Fig. 3 shows the statistical analyses of type-specific essential GO terms, including (a) type-specific statistics of essential GO terms and (b) categorical breakdown of essential GO terms in each functional type. In Fig. 3(a), we can see that about 111∼185 essential GO terms were selected from 7954 GO terms by Mem-mEN for each functional type, which determine the type(s) of a query protein. This suggests that the number of essential GO terms for each type is not significantly different. Besides, the union of the unique essential GO terms of these eight types comprises 338 GO terms, which suggests that some of the essential GO terms coexist in several functional types. Fig. 3(b) shows the categorical breakdown of essential GO terms in each functional type. As can be seen from Fig. 3(b), the percentage of GO terms for the three categories vary significantly for different types. For



(a) Dataset I (b) Dataset II

Fig. 2. Breakdown of the virtual proteins [19] in the benchmark datasets. (a) Dataset I [19]; (b) dataset II [20]. In (a), there are 5,502 virtual proteins corresponding to 5,307 actual proteins, of which 5117 belong to one type, 185 to two types and 5 to three types. In (b), there are 14,016 virtual proteins corresponding to 13,659 actual proteins, of which 13,313 belong to one type, 335 to two types and 11 to three types.

example, in Type 1 (single-pass type I), the numbers of essential GO terms in the cellular-component (CC), molecular-function (MF) and biological-process (BP) are about the same; whereas in Type 8 (peripheral), the number of CC GO terms is more than twice than that of BP GO terms. This indicates that CC GO terms may contribute more to the prediction for some functional types (e.g., peripheral), while for other types (e.g., single-pass type I), GO terms from the other two categories may contribute more to the prediction. Besides, as shown in Fig. 3(b), when the functional types are ignored (the ring with the label "All"), the numbers of unique essential GO terms in the three categories are 114 (CC), 95 (MF) and 129 (BP). Contrary to individual types, the number of BP GO terms is larger than that of CC GO terms, suggesting that there are more overlapping GO terms in the CC category than in the BP category.

5.2 Significance of Type-Specific GO Terms

Fig. 4 shows the boxplots of categorical significance of essential GO terms for different membrane types. For simplicity, $\{\beta_{s,m}\}_{s=\{1,\ldots,S\},m=\{1,\ldots,M\}}$ in Eq. 6 is abbreviated as β in the figures. In Fig. 4(a), we can see that for Type 1 (single-pass type I), the weights of CC GO terms have narrower range than those of GO terms in other two categories. Besides, the median and maximum weights of the former are smaller than the respective weights of the latter. These results suggest that the CC GO terms play less significant roles in the prediction than MF and BP GO terms. Similar conclusions can be drawn for Fig. 4(b), where CC GO terms possess weights with a narrower range and smaller median and maximum weights than the other two categories. However, the scenario is the opposite in Fig. 4(h), where the weights of CC GO terms have a wider range and their maximum is also larger than that of the other two categories. This suggests that CC GO terms are more important for predicting Type 8's proteins than MF and BP GO terms.

Fig. 5 shows the overall significance of essential GO terms for different membrane types. As can be seen, the weights for SP1, MP and PE have a larger range than the other five types. Besides, the median and maximum weight of PE is larger than those in the other seven types. This suggests that Mem-mEN can predict peripheral proteins with a higher confidence than the other types.

5.3 Significance of Ranking Essential GO Terms

Figs. 6(a)–(h) show the significance (weights) of the GO terms corresponding to Type 1 to Type 8. In each sub-figure, the GO terms are ranked according to their significance in the three categories (CC, MF and BP). As can be seen from Fig. 6(a), the GO terms with large weights are from the MF and BP categories, suggesting that GO terms from MF and BP categories outweigh CC GO terms for predicting Type 1 membrane proteins. However, the opposite phenomena occur for Type 6 and Type 7, where GO terms with top weights are from the CC category, suggesting that the CC GO terms play a dominant role in the prediction of lipid-anchor and GPI-anchor membrane proteins. Similar analysis can also be applied to other types.

To have in-depth understanding on which GO term has the largest contribution for each functional type, we show the representative GO term for each membrane type in Table 3. The representative GO term for each type is the GO term whose weight is the largest for the corresponding type. For example, the weight of GO:0004896 (MF, cytokine receptor activity) is the largest (0.4984) among all of the essential GO terms for Type 1. Interestingly, we found that not all of the representative GO terms belong to the CC category. For example, the representative GO terms for Types 1 and 3 are from the MF and BP categories. This suggests that GO terms from

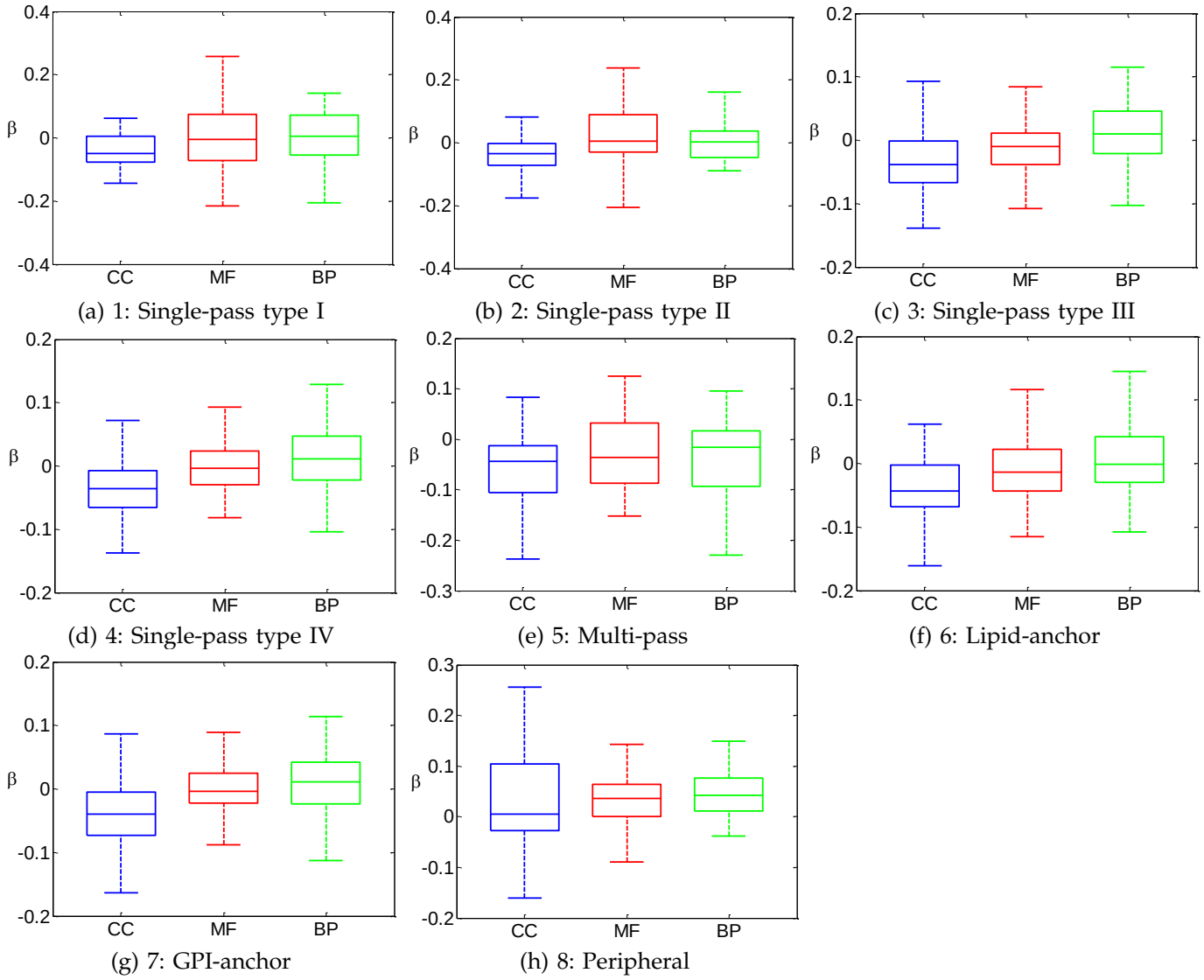


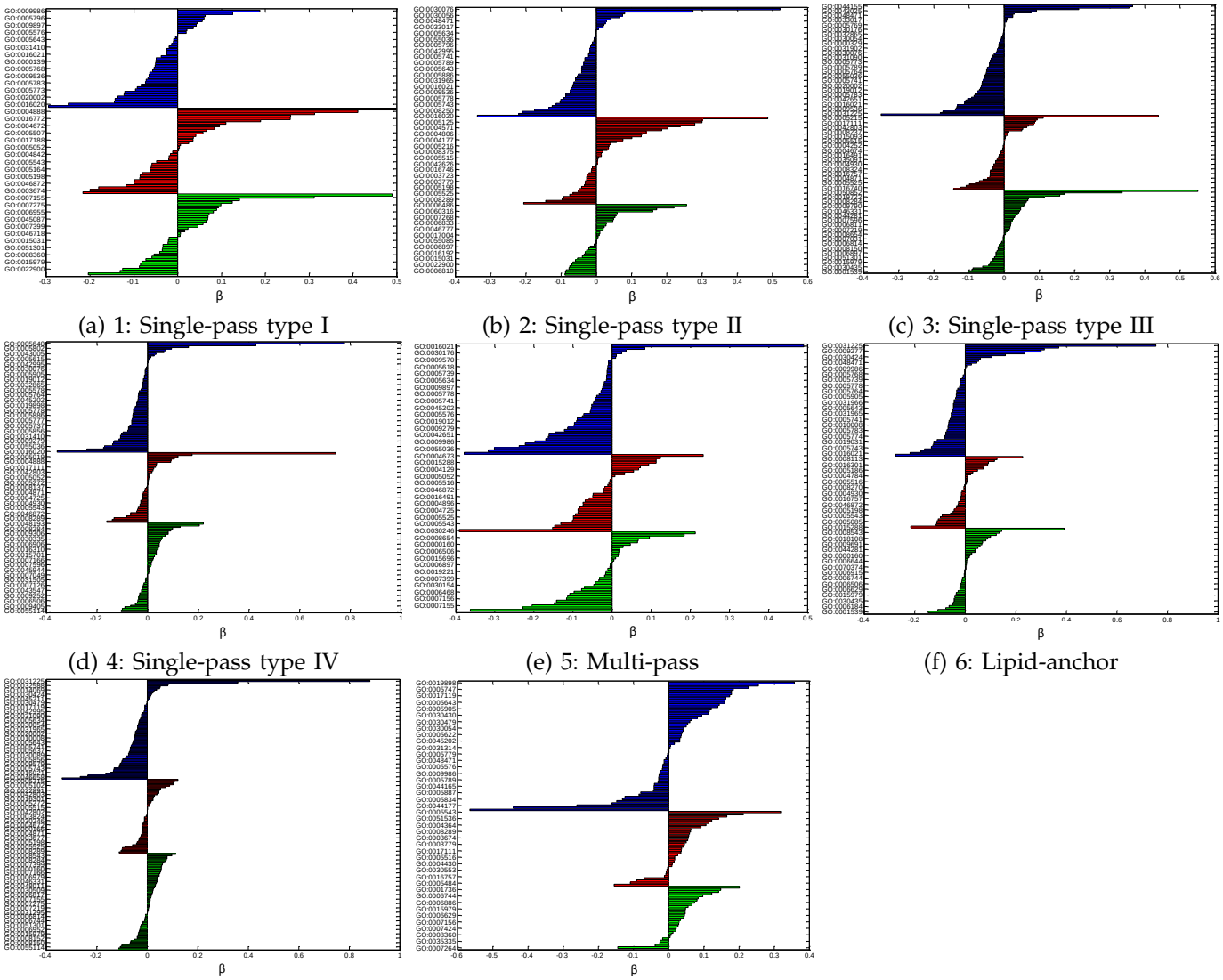
Fig. 4. Categorical significance of essential GO terms for different membrane types, including (a) single-pass type I, (b) single-pass type II, (c) single-pass type III, (d) single-pass type IV, (e) multi-pass, (f) lipid-anchor, (g) GPI-anchor and (h) peripheral. *CC*: cellular component; *MF*: molecular function; *BP*: biological function.

the MF and BP categories are also contributive to the predictions.

We can see from Table 3 that the annotations of the representative GO terms for some types are directly related to the corresponding functional types. For example, the representative GO term for Type 8 is GO:0019898, whose annotation (CC, extrinsic component of membrane) directly associates with the peripheral type. However, for some other types, even if their representative GO terms are from the CC category, there is no relationship between the annotations and the membrane types. For example, the representative GO term for Type 2 is GO:0030076 (CC, light-harvesting complex), which has no direct linkage with any kind of membrane types. These results indicate that some CC GO terms, which do not associate with any membrane types, are helpful in determining the functional types of membrane proteins. Besides, because GPI-anchor type is a special kind of lipid-anchor type, it is reasonable that the representative GO terms for these two types are the same (GO:0031225). Nevertheless, it by no means indicates that all of the essential GO terms and the corresponding weights are the same for these two types. This is clearly demonstrated in Fig. 6(f) and Fig. 6(h), where some of the essential GO terms as well as their weights for Types 6 and 7 are different.

5.4 Circular Network for Essential GO Terms and Membrane Types

To have a comprehensive understanding of the relationships between essential GO terms selected by Mem-mEN and the eight membrane types, Fig. 7 shows a circular network connecting the essential GO terms and the membrane types. Small green dots on the right represent



(a) 1: Single-pass type I (b) 2: Single-pass type II (c) 3: Single-pass type III

(d) 4: Single-pass type IV (e) 5: Multi-pass (f) 6: Lipid-anchor

(g) 7: GPI-anchor (h) 8: Peripheral

Fig. 6. Ranking GO terms according to their categorical significance for different membrane types, including (a) single-pass type I, (b) single-pass type II, (c) single-pass type III, (d) single-pass type IV, (e) multi-pass, (f) lipid-anchor, (g) GPI-anchor and (h) peripheral. The blue, red and green colors represent the categories of cellular component, molecular function and biological process.

TABLE 3
Representative essential GO terms for different membrane types. *CC*: cellular component; *MF*: molecular function; *BP*: biological function.

Type Number	Membrane Type	Representative GO Term	Category	Weight	Name in GO*
1	Single-pass type I	GO:0004896	MF	0.4984	Cytokine receptor activity
2	Single-pass type II	GO:0030076	CC	0.5218	Light-harvesting complex
3	Single-pass type III	GO:0038095	BP	0.5514	Fc-epsilon receptor signaling pathway
4	Single-pass type IV	GO:0005640	CC	0.7777	Nuclear outer membrane
5	Multi-pass	GO:0016021	CC	0.4908	Integral component of membrane
6	Lipid-anchor	GO:0031225	CC	0.7547	Anchored component of membrane
7	GPI-anchor	GO:0031225	CC	0.8825	Anchored component of membrane
8	Peripheral	GO:0019898	CC	0.4984	Extrinsic component of membrane

*: http://geneontology.org/.

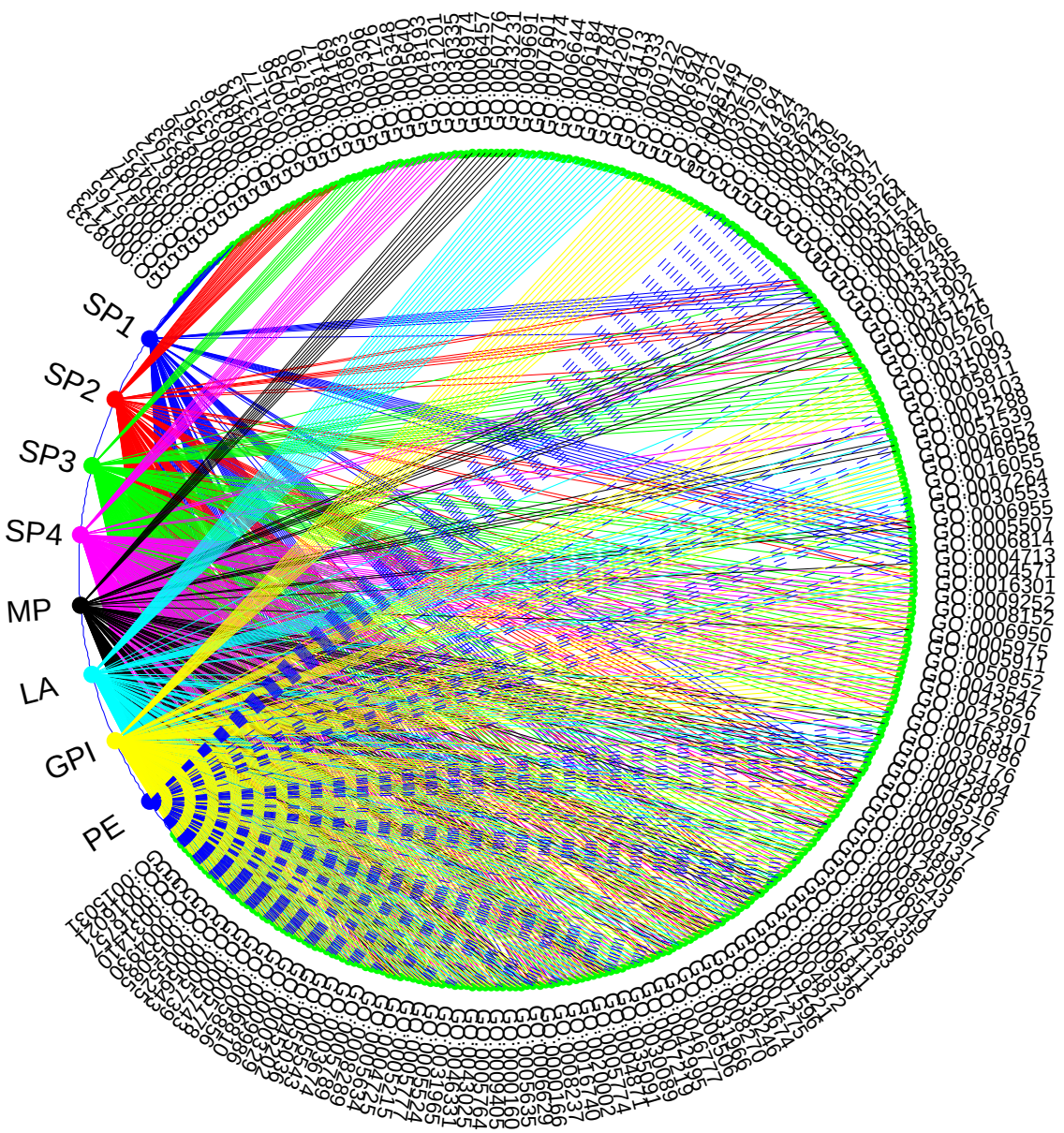
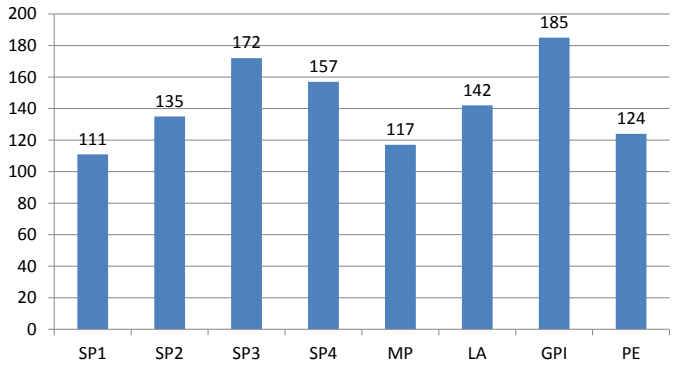


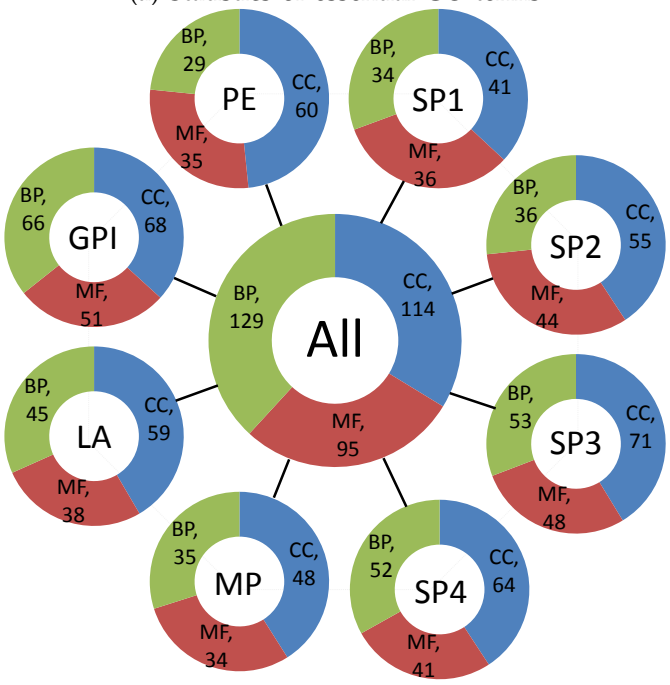
Fig. 7. A network showing the relationship between the essential GO terms and each membrane type. Small green dots on the right represent the GO terms and the large dots in different colors on the left represent the 8 membrane protein types. A line connecting an essential GO term and a membrane type denotes that the GO term contributes to the prediction of the membrane protein type. On the contrary, if there is no line connecting an essential GO term with a particular membrane protein type, then this GO term does not provide any information about whether a protein belongs to the particular functional type or not. *SP1*: single-pass type I; *SP2*: single-pass type II; *SP3*: single-pass type III; *SP4*: single-pass type IV; *MP*: multi-pass; *LA*: lipid-anchor; *GPI*: GPI-anchor; *PE*: peripheral.

the GO terms and the large dots in different colors on the left represent the 8 membrane protein types. A line connecting an essential GO term and a membrane type denotes that the GO term contributes to the prediction of the membrane protein type. On the contrary, if there is no line connecting an essential GO term with a particular membrane protein type, then this GO term does not provide any information about the particular functional type.

Starting from the top-left green dots to the bottom-left green dots in clockwise direction, the degree of overlapping among the lines gradually increases, denoting that the number of membrane types to which an essential GO term contributes also gradually increases. For example, the first top-left GO term, GO:0008233, only contributes to the prediction of SP1 (single-pass type I), indicating whether a protein belongs to SP1 or not. On the contrary, the last bottom-left GO term, GO:0015031, is indicative of all of the eight functional types. In other words, these essential GO terms are indicators of whether a query



(a) Statistics of essential GO terms



(b) Breakdown of the essential GO terms

Fig. 3. Type-specific information of essential GO terms, including (a) type-specific statistics of essential GO terms and (b) categorical breakdown of essential GO terms in each functional type. *SP1*: single-pass type I; *SP2*: single-pass type II; *SP3*: single-pass type III; *SP4*: single-pass type IV; *MP*: multi-pass; *LA*: lipid-anchor; *GPI*: GPI-anchor; *PE*: peripheral. *CC*: cellular component; *MF*: molecular function; *BP*: biological function.

protein belongs to one or multiple membrane types or not.

5.5 Comparing Mem-mEN with State-of-the-art Predictors

Table 4 and Table 5 compare Mem-mEN with two state-of-the-art multi-label predictors on Datasets I and II, respectively. To the best of our knowledge, there are only two predictors, namely iMem-Seq [19] and Mem-PseAA [20], which can predict membrane proteins with both single- and multi-label functional types. iMem-Seq uses position-specific score matrices to construct feature vectors, while Mem-PseAA uses pseudo-amino acid composition features. Both predictors use multi-label kNN classifier to deal with the multi-label classification problem. Our proposed Mem-mEN extracts homologous GO frequency information from ProSeq and ProSeq-GO databases and then uses multi-label elastic net classifiers for both feature selection and classification.

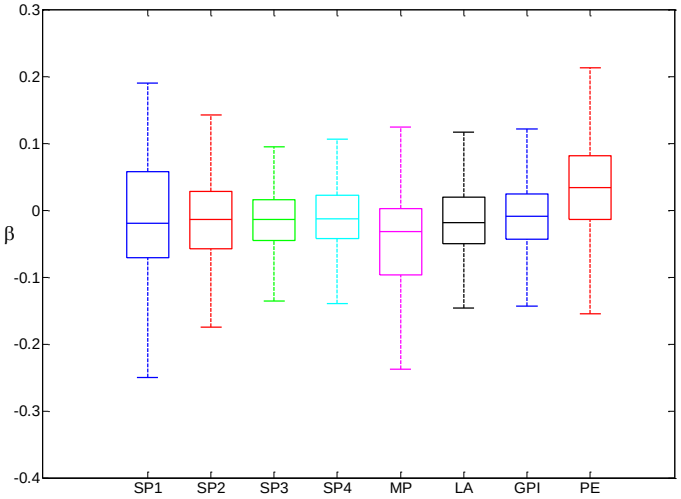


Fig. 5. Overall type-specific significance of essential GO terms. *SP1*: single-pass type I; *SP2*: single-pass type II; *SP3*: single-pass type III; *SP4*: single-pass type IV; *MP*: multi-pass; *LA*: lipid-anchor; *GPI*: GPI-anchor; *PE*: peripheral.

TABLE 4
Comparing Mem-mEN with a state-of-the-art predictor based on leave-one-out cross-validation tests on Dataset I. ↓ means the lower the better; ↑ denotes the higher the better.

Evaluation Criteria	Predictors	
	iMem-Seq [19]	Mem-mEN
Hamming loss ↓	0.0635	**0.0493**
Ranking loss ↓	0.0902	**0.0521**
One-error ↓	0.2572	**0.1892**
Coverage ↓	0.6735	**0.4046**
Average precision ↑	0.8335	**0.8881**
Accuracy ↑	0.6804	**0.8056**
Precision ↑	0.6825	**0.8085**
Recall ↑	0.6813	**0.8135**
Absolute-true ↑	0.6774	**0.7948**

The results in Table 4 were based on leave-one-out cross-validation tests on Dataset I, where the sequence identity was cut off at 25%. As shown in Table 4, Mem-mEN performs much better than iMem-Seq in terms of all performance metrics. In particular, for the most stringent and object criteria *absolute-true*, Mem-mEN outperforms iMem-Seq by more than 11% (absolute).

Similar conclusions can be drawn for Dataset II in Table 5. The results in Table 5 were based on the average

TABLE 5
Comparing Mem-mEN with state-of-the-art predictors based on five 5-fold cross-validation tests on Dataset II. ↓ means the lower the better; ↑ denotes the higher the better.

Evaluation Criteria	Predictors		
	Mem-PseAA [20]	iMem-Seq [19]	Mem-mEN
Hamming loss ↓	0.0495±0.0019	0.0317±0.0013	**0.0303±0.0008**
Ranking loss ↓	0.0600±0.0025	0.0425±0.0008	**0.0339±0.0006**
One-error ↓	0.1964±0.0033	0.1192±0.0011	**0.1154±0.0013**
Coverage ↓	0.4470±0.0215	0.3266±0.0031	**0.2636±0.0029**
Average precision ↑	0.8780±0.0025	0.9211±0.0007	**0.9225±0.0009**

of five 5-fold cross-validation tests on Dataset II, where the sequence identity was cut off at 80%.[5] As can be seen, the *average precision* of Mem-mEN is higher than that of Mem-PseAA and iMem-Seq, and the former also performs better than the latter in terms of the first four metrics. In Table 5, the superiority of Mem-mEN over iMem-Seq is less apparent than that in Table 4. This is possibly because the sequence identity in Dataset II is much higher than that in Dataset I (80% vs. 25%), which increases the bias on the prediction performance. In this case, the performance comparison in Table 4 is more trustworthy than that in Table 5.

6 Predicting and Interpreting Membrane Proteins

Fig. 8 demonstrates how researchers can use Mem-mEN to predict and interpret the functional types of query membrane proteins. Fig. 8(a) shows the scores produced by Eq. 7 in descending order using the query protein P26952 as input, where (C), (F) and (P) stand for cellular component, molecular function and biological process categories, respectively. Also, the columns "Weight" and "Term-Freq" represent non-zero elements of $\hat{\boldsymbol{\alpha}}_m$ in Eq. 6 and $\mathbf{x}_i^s$ in Eq. 5, and the column "Feature Score" represents the product of Weight and Term-Freq. The higher the feature score, the more contribution is the corresponding GO term to the prediction result. Since only one of the 8 scores is positive, the number of functional types is predicted to be 1 and this protein is predicted to belong to SP1 (Type 1). The scores and weights for the essential GO terms in SP1 and SP2 are also shown in the right panel of Fig. 8(a).[6] As can be seen, 5 essential GO terms contribute to the score of SP1, while 4 GO terms contribute to the score of SP2. The maximum and minimum scores correspond to SP1 and SP2, respectively, which suggests that P26952 is likely to be an SP1 protein but unlikely to be an SP2 protein. Besides, for SP1, the top two essential GO terms (GO:0004896 and GO:0019221) belong to molecular function (F) and biological process (P), respectively, while the remaining 3 belong to the cellular-component category. More interestingly, the scores of the top two GO terms from non-CC categories are positive whereas those of the remaining CC GO terms are negative. This suggests that GO terms from the categories of molecular function and biological process play key roles in determining the functional types of the query membrane protein.

The essential GO terms that lead to large positive feature scores enable us to *interpret* the prediction decision. For example, as indicated in Table 3, GO:0004896 is a representative essential GO term for Type 1 proteins, and its definition in the GO database is "Combining with a cytokine and transmitting the signal from one side of the membrane to the other to initiate a change in cell activity". This information boosts our confidence in the prediction decisions of Mem-mEN and enables us to explain why P26952 is an SP1 protein.

Fig. 8(b) shows the case for a multi-label membrane protein (P06015). Evidently, there are two positive scores, both determined by 3 essential GO terms. Thus, P06015 is predicted to co-locate in GPI (Type 7) and LA (Type 6). This demonstrates that Mem-mEN can predict membrane proteins with multi-functional types. Again, the GO terms (GO:0031225) that leads to the largest feature score is a representative essential GO term in Table 3 and its definition in the GO database is "The component of a membrane consisting of the gene products that are tethered to the membrane only by a covalently attached anchor, such as a lipid group that is embedded in the membrane." Evidently, this definition together with the weights found by the EN-based feature selector clearly indicate that the prediction decision is correct.

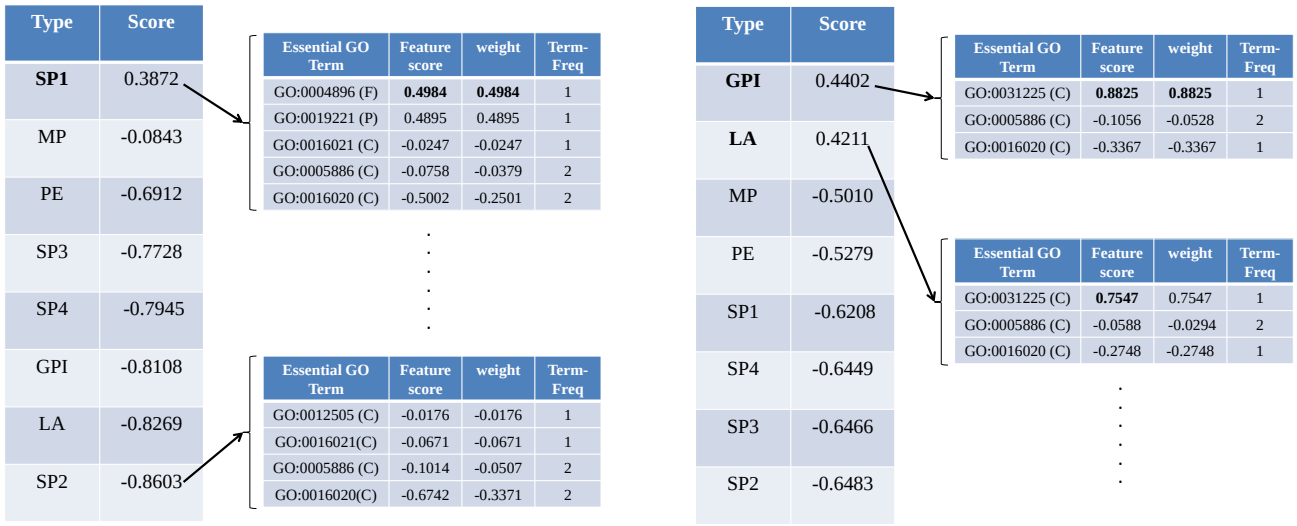
7 Discussion

7.1 Why Two-Stage EN Training?

One may wonder the advantages of two-stage EN training (one for feature selection in Section 3.2, and one for classification in Section 3.3). To clarify this, we have compared the proposed two-stage training against a single-stage one based on Dataset I. The results are shown in Table 6. As can be seen, the predictor based on two-stage training performs slightly better than the one based

5. Because [20] and [19] do not report the *Precision, Recall* and *Absolute-True* on Dataset II, for consistency, we do not report these results.

6. The scores and weights for the essential GO terms for all of the 8 functional types can be seen by inputing the query protein sequence to our Mem-mEN web-server.



(a) Predicting Protein P26952 (b) Predicting Protein P06015

Fig. 8. Examples showing how Mem-mEN predicts and interprets functional types of (a) a single-label membrane protein (P26952) and (b) a multi-label membrane protein (P06015). *Type*: functional type; *Score*: the score determined by Eq. 7; *Feature Score*: the score that each essential GO term contributes to the final prediction; *Term-freq*: the frequency of occurrences of an essential GO term; *C*: cellular component; *F*: molecular function; *P*: biological process; *SP1*: single-pass type I; *SP2*: single-pass type II; *SP3*: single-pass type III; *SP4*: single-pass type IV; *MP*: multi-pass; *LA*: lipid-anchor; *GPI*: GPI-anchor; *PE*: peripheral.

on single-stage training in terms of all performance metrics. This means that while the features selected in the first stage of training are optimal (in terms of *Absolute True*) for the feature-selection EN, the GO vectors formed by using these features as the bases require another set of weights to achieve the best classification performance. This observation is reasonable because the feature-selection EN and the classification EN work on two different feature spaces.

More importantly, the two-stage approach facilitates us to construct flexible application-oriented predictors. For example, in some applications, it is better to classify the selected features by nonlinear classifiers such as neural networks. In other situations, wrapper approaches such as RFE-SVM [43] may be more appropriate for selecting the features. By dividing feature selection and pattern classification into two separated stages, it is possible to use any feature selection methods and pattern classifiers for these two stages. The advantages of the two-stage approaches have also been demonstrated in our recent work on subcellular localization prediction [25].

7.2 EN vs LASSO

Note that LASSO is a special case of elastic net (EN). Specifically, when $\gamma = 0$ in Eq. 3, EN becomes LASSO. As detailed in Section S1 (**"Parameter Optimization Implementation"**) of the supplementary materials in the Mem-mEN web-server[7], $\gamma = 0$ is included in the grid search for optimizing the ridge regression penalty (γ) and L_1 penalty (λ) parameters of EN during the feature selection stage. The grid search suggests that $\lambda = 10$ and $\gamma = 0.001$ achieve the best performance on the EN-based feature selector, which means that LASSO ($\gamma = 0$) is inferior to EN in terms of feature selection. Therefore, we adopted EN with $(\lambda = 10, \gamma = 10^{-3})$ instead of LASSO ($\lambda > 0, \gamma = 0$) to construct the feature selector.

Given a set of features selected by an elastic net, we may use different types of classifiers for the classification stage. While we propose using an EN-based classifier for this purpose, it is of interest to compare its performance with a LASSO-based classifier. The experimental results are shown in Table 7, which shows that EN performs

TABLE 6
Comparing two-stage training against one-stage training based on Dataset I. ↓ means the lower the better; ↑ denotes the higher the better.

Evaluation Criteria	Predictors	
	One-Stage	Two-Stage
Hamming loss ↓	0.0518	**0.0493**
Ranking loss ↓	0.0542	**0.0521**
One-error ↓	0.2005	**0.1892**
Coverage ↓	0.4196	**0.4046**
Average precision ↑	0.8811	**0.8881**
Accuracy ↑	0.7954	**0.8056**
Precision ↑	0.7987	**0.8085**
Recall ↑	0.8035	**0.8135**
Absolute-true ↑	0.7839	**0.7948**

7. http://bioinfo.eie.polyu.edu.hk/MemmENServer/suppl.html

TABLE 7
Comparing EN with LASSO as classifiers after using EN feature selection on Dataset I. ↓ means the lower the better; ↑ denotes the higher the better.

Evaluation Criteria	Classifiers	
	LASSO	EN
Hamming loss ↓	0.0525	**0.0493**
Ranking loss ↓	0.0552	**0.0521**
One-error ↓	0.2043	**0.1892**
Coverage ↓	0.4262	**0.4046**
Average precision ↑	0.8796	**0.8881**
Accuracy ↑	0.7923	**0.8056**
Precision ↑	0.7956	**0.8085**
Recall ↑	0.7992	**0.8135**
Absolute-true ↑	0.7822	**0.7948**

better than LASSO in all performance metrics. Moreover, among the 338 essential GO terms, LASSO finds 203 non-zero weights only; that is, only 203 out of 338 GO terms are useful for LASSO classification. On the other hand, EN finds 328 non-zero weights; in other words, 328 out of 338 GO terms are used in EN classification. This is probably because GO terms from the same category are not independent with each other; instead they are correlated with some hierarchical relationships, such as 'is_a' and 'part_of'. Compared to LASSO, EN will select correlated features together, thus causing more essential GO terms to be selected. In fact, the results are consistent with the claims in [33].

8 Conclusion

This paper proposes an efficient and interpretable predictor, namely Mem-mEN, for predicting membrane proteins with single- and multi-label functional types. By using a one-vs-rest EN classifier, 338 out of 7,900+ GO terms were found to play more important roles in determining to which type(s) the query protein belongs. Based on these selected essential GO terms, users of Mem-mEN can not only predict to which type(s) a query protein belongs, but also why it belongs to that type.

Experimental results show that Mem-mEN performs significantly better than state-of-the-art multi-label membrane-protein predictors. Besides, this paper also found that GO terms from all of the three categories contribute to prediction of membrane protein functional types. And, GO terms from different categories contribute diversely to different functional types. Furthermore, the significance of contributions of an essential GO term depends on the functional type, with major contribution on one functional type while with minor contribution on some other types.

Acknowledgments

This work was in part supported by the RGC of Hong Kong SAR Grant No. PolyU152117114E and Hong Kong PolyU Grant No. GYN18.

References

[1] M. S. Almén, K. J. V. Nordström, R. Fredriksson, and H. B. Schiöth, "Mapping the human membrane proteome: a majority of the human membrane proteins can be classified according to function and evolutionary origin," *BMC Biology*, vol. 7, no. 1, pp. 50, 2009.

[2] T. M. Bakheet and A. J. Doig, "Properties and identification of human protein drug targets," *Bioinformatics*, vol. 25, no. 4, pp. 451–457, 2009.

[3] H. F. Lodish, A. Berk, S. L. Zipursky, P. Matsudaira, D. Baltimore, and J. Darnell, *Molecular cell biology*, W. H. Freeman New York, 4 edition, 2000.

[4] K. Gerald, *Cell and molecular biology: concepts and experiments*, John Wiley and Sons, Hoboken, NJ, 7 edition, 2013.

[5] K. C. Chou and H. B. Shen, "MemType-2L: a web server for predicting membrane proteins and their types by incorporating evolution information through Pse-PSSM," *Biochemical and Biophysical Research Communications*, vol. 360, no. 2, pp. 339–345, 2007.

[6] H. Ikezawa, "Glycosylphosphatidylinositol (GPI)-anchored proteins," *Biological and Pharmaceutical Bulletin*, vol. 25, no. 4, pp. 409–417, 2002.

[7] P. S. Tappia and N. S. Dhalla, *Phospholipases in health and disease*, Springer, 1 edition, 2014.

[8] L. Nanni and A. Lumini, "An ensemble of support vector machines for predicting the membrane protein type directly from the amino acid sequence," *Amino Acids*, vol. 35, no. 3, pp. 573–580, 2008.

[9] H. L. Zou, "A multi-label classifier for prediction membrane protein functional types in animal," *The Journal of Membrane Biology*, vol. 247, no. 11, pp. 1141–1148, 2014.

[10] M. Hayat, A. Khan, and M. Yeasin, "Prediction of membrane proteins using split amino acid and ensemble classification," *Amino acids*, vol. 42, no. 6, pp. 2447–2460, 2012.

[11] C. Ding, L. F. Yuan, S. H. Guo, H. Lin, and W. Chen, "Identification of mycobacterial membrane proteins and their types using over-represented tripeptide compositions," *Journal of Proteomics*, vol. 77, pp. 321–328, 2012.

[12] T. Wang, T. Xia, and X. M. Hu, "Geometry preserving projections algorithm for predicting membrane protein types," *Journal of Theoretical Biology*, vol. 262, no. 2, pp. 208–213, 2010.

[13] K. C. Chou and Y. D. Cai, "Using GO-PseAA predictor to identify membrane proteins and their types," *Biochemical and Biophysical Research Communications*, vol. 327, no. 3, pp. 845–847, 2005.

[14] Y. D. Cai, G. P. Zhou, and K. C. Chou, "Support vector machines for predicting membrane protein types by using functional domain composition," *Biophysical Journal*, vol. 84, no. 5, pp. 3257–3263, 2003.

[15] V. Tripathi and D. K. Gupta, "Discriminating lysosomal membrane protein types using dynamic neural network," *Journal of Biomolecular Structure and Dynamics*, vol. 32, no. 10, pp. 1575–1582, 2014.

[16] Z. Yuan and R. D. Teasdale, "Prediction of Golgi Type II membrane proteins based on their transmembrane domains," *Bioinformatics*, vol. 18, no. 8, pp. 1109–1115, 2002.

[17] R. Clemente and C. Juan, "Cell entry of Borna disease virus follows a clathrin-mediated endocytosis pathway that requires Rab5 and microtubules," *Journal of Virology*, vol. 83, no. 20, pp. 10406–10416, 2009.

[18] T. W. Vahlenkamp, A. Konrath, M. Weber, and H. Müller, "Persistence of Borna disease virus in naturally infected sheep," *Journal of Virology*, vol. 76, no. 19, pp. 9735–9743, 2002.

[19] X. Xiao, H. L. Zou, and W. Z. Lin, "iMem-Seq: A multi-label learning classifier for predicting membrane proteins types," *The Journal of Membrane Biology*, pp. 1–8, 2015.

[20] C. Huang and J. Q. Yuan, "A multilabel model based on Chou's pseudo–amino acid composition for identifying membrane proteins with both single and multiple functional types," *The Journal of Membrane Biology*, vol. 246, no. 4, pp. 327–334, 2013.

[21] S. Wan, M. W. Mak, and S. Y. Kung, "HybridGO-Loc: Mining hybrid features on gene ontology for predicting subcellular localization of multi-location proteins," *PLoS ONE*, vol. 9, no. 3, pp. e89545, 2014.

[22] K. C. Chou, Z. C. Wu, and X. Xiao, "iLoc-Euk: A multi-label classifier for predicting the subcellular localization of singleplex and multiplex eukaryotic proteins," *PLoS ONE*, vol. 6, no. 3, pp. e18258, 2011.
[23] S. Wan, M. W. Mak, and S. Y. Kung, "Semantic similarity over gene ontology for multi-label protein subcellular localization," *Engineering*, vol. 5, pp. 68–72, 2013.
[24] S. Mei, "Multi-label multi-kernel transfer learning for human protein subcellular localization," *PLoS ONE*, vol. 7, no. 6, pp. e37716, 2012.
[25] S. Wan, M. W. Mak, and S. Y. Kung, "mLASSO-Hum: A LASSO-based interpretable human-protein subcellular localization predictor," *Journal of Theoretical Biology*, vol. 382, pp. 223–234, 2015.
[26] S. Wan and M. W. Mak, *Machine learning for protein subcellular localization prediction*, De Gruyter, 2015.
[27] S. F. Altschul, T. L. Madden, A. A. Schaffer, 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.
[28] S. Wan, M. W. Mak, and S. Y. Kung, "GOASVM: A subcellular location predictor by incorporating term-frequency gene ontology into the general form of Chou's pseudo-amino acid composition," *Journal of Theoretical Biology*, vol. 323, pp. 40–48, 2013.
[29] S. Wan, M. W. Mak, and S. Y. Kung, "mGOASVM: Multi-label protein subcellular localization based on gene ontology and support vector machines," *BMC Bioinformatics*, vol. 13, pp. 290, 2012.
[30] K. C. Chou, "Prediction of protein cellular attributes using pseudo amino acid composition," *Proteins: Structure, Function, and Genetics*, vol. 43, pp. 246–255, 2001.
[31] K. Nakai, "Protein sorting signals and prediction of subcellular localization," *Advances in Protein Chemistry*, vol. 54, no. 1, pp. 277–344, 2000.
[32] S. Wan, M. W. Mak, and S. Y. Kung, "R3P-Loc: A compact multi-label predictor using ridge regression and random projection for protein subcellular localization," *Journal of Theoretical Biology*, vol. 360, pp. 34–45, 2014.
[33] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.
[34] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.
[35] B. Zhang, H. Li, R. B. Riggins, M. Zhan, J. Xuan, Z. Zhang, E. P. Hoffman, R. Clarke, and Y. Wang, "Differential dependency network analysis to identify condition-specific topological changes in biological networks," *Bioinformatics*, vol. 25, no. 4, pp. 526–532, 2009.
[36] K. L. Ayers and H. J. Cordell, "SNP selection in genome-wide and candidate gene studies via penalized logistic regression," *Genetic Epidemiology*, vol. 34, no. 8, pp. 879–891, 2010.
[37] D. He, Z. Wang, and L. Parida, "Data-driven encoding for quantitative genetic trait prediction," *BMC Bioinformatics*, vol. 16, no. Suppl 1, pp. S10, 2015.
[38] B. J. Marafino, W. J. Boscardin, and R. A. Dudley, "Efficient and sparse feature selection for biomedical text classification via the elastic net: Application to ICU risk stratification from nursing notes," *Journal of Biomedical Informatics*, vol. 54, pp. 114–120, 2015.
[39] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.
[40] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," in *Data Mining and Knowledge Discovery Handbook, O. Maimon, l. Rokach (Ed.). Springer, 2nd edition*, 2010, pp. 667–685.
[41] R. E. Schapire and Y. Singer, "Boostexter: A boosting-based system for text categorization," *Machine Learning*, vol. 39, no. 2/3, pp. 135–168, 2000.
[42] S. Wan, M. W. Mak, and S. Y. Kung, "mPLR-Loc: An adaptive decision multi-label classifier based on penalized logistic regression for protein subcellular localization prediction," *Analytical Biochemistry*, vol. 473, pp. 14–27, 2015.
[43] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, pp. 389–422, 2002.

Shibiao Wan is currently a Postdoctoral Fellow of Department of Electronic and Information Engineering at The Hong Kong Polytechnic University. He obtained his BEng degree in Telecommunication Engineering from Wuhan University, China in 2010 and his PhD degree in Bioinformatics from The Hong Kong Polytechnic University in 2014. He was a visiting scholar in the Virginia Tech and The Johns Hopkins School of Medicine from Spring 2013 to Summer 2013. His current research interests include bioinformatics, computational biology and machine learning. He is the leading author of the book *Machine Learning for Protein Subcellular Localization Prediction, De Gruyter, 2015*. He has published a number of technical articles on top bioinformatics journals such as BMC Bioinformatics, PLoS ONE, Journal of Theoretical Biology, Analytical Biochemistry, etc, and key international conferences on signal processing, bioinformatics and machine learning such as ICASSP, BIBM, MLSP, etc. He serves as a reviewer for a number of journals, such as IEEE Trans. on Nanobioscience, Analytical Biochemistry, AMC, JAM, IJBI and JMLC.

Man-Wai Mak (M'93–SM'15) received a PhD in Electronic Engineering from the University of Northumbria in 1993. He joined the Department of Electronic and Information Engineering at The Hong Kong Polytechnic University in 1993 and is currently an Associate Professor in the same department. He has authored more than 150 technical articles in speaker recognition, machine learning, and bioinformatics. Dr. Mak also coauthored a postgraduate textbook *Biometric Authentication: A Machine Learning Approach, Prentice Hall, 2005* and a research monograph *Machine Learning for Protein Subcellular Localization Prediction, De Gruyter, 2015*. He served as a member of the IEEE Machine Learning for Signal Processing Technical Committee in 2005-2007. He has served as an associate editor of IEEE Trans. on Audio, Speech and Language Processing. He is currently an editorial board member of Journal of Signal Processing Systems and Advances in Artificial Neural Systems. He also served as Technical Committee Members of a number of international conferences, including ICASSP and Interspeech. Dr. Mak's research interests include speaker recognition, machine learning, and bioinformatics.

Sun-Yuan Kung is a Professor at Department of Electrical Engineering in Princeton University. His research areas include VLSI array processors, system modeling and identification, neural networks, wireless communication, sensor array processing, multimedia signal processing, bioinformatic data mining and biometric authentication. He was a founding member of several Technical Committees (TC) of the IEEE Signal Processing Society, and was appointed as the first Associate Editor in VLSI Area (1984) and later the first Associate Editor in Neural Network (1991) for the IEEE Transactions on Signal Processing. He has been a Fellow of IEEE since 1988. He served as a Member of the Board of Governors of the IEEE Signal Processing Society (1989-1991). Since 1990, he has been the Editor-In-Chief of the Journal of VLSI Signal Processing Systems. He has authored and co-authored more than 400 technical publications and numerous textbooks including "VLSI and Modern Signal Processing", Prentice-Hall (1985), "VLSI Array Processors", Prentice-Hall (1988); "Digital Neural Networks", Prentice-Hall (1993) ; "Principal Component Neural Networks", John-Wiley (1996); and "Biometric Authentication: A Machine Learning Approach", Prentice-Hall (2004).