Journal of Theoretical Biology 313 (2012) 61–67

Contents lists available at SciVerse ScienceDirect

Journal of Theoretical Biology

journal homepage: www.elsevier.com/locate/yjtbi

Subcellular localization prediction for human internal and organelle membrane proteins with projected gene ontology scores

Pufeng Du*, Yang Tian, Yan Yan

School of Computer Science and Technology, Tianjin University, Tianjin 300072, China

HIGHLIGHTS

- ▸ We developed a novel method for predicting subcellular localizations for membrane proteins.
- ▸ This method performs better than the existing methods.
- ▸ This method can provide more than one prediction result for a protein.

ARTICLE INFO

Article history:
Received 30 March 2012
Received in revised form
5 July 2012
Accepted 15 August 2012
Available online 23 August 2012

Keywords:
Multiple-location prediction

ABSTRACT

The membrane proteins make up more than a third of all known human proteins. The subcellular localizations play a key role to elucidate the potential biological functions of these membrane proteins. Although the experimental approaches for determining protein subcellular localizations exist, they are usually costly and time consuming. Thus, computational predictions provided an alternative approach for determining the protein subcellular localizations. However, current subcellular location predictors are generally developed for globular proteins. They did not perform well for membrane proteins. In this paper, we proposed a novel prediction algorithm, namely Projected Gene Ontology Score, which introduces the Gene Ontology annotation as a descriptor of the protein. This algorithm could significantly improve the prediction accuracy for the subcellular localizations of membrane proteins. It can designate each protein to one of the eight different locations, while the existing algorithm only covers three locations. Actually, the biological problem considered by our algorithm goes one level deeper than the existing algorithms. In addition, our algorithm can provide more than one location for the testing protein, which could be very useful in practical studies. Our algorithm is expected to be a good complement to the existing algorithms and has the potential to be extended to solve other problems.

1. Introduction

In human genome, a large number of genes encode membrane proteins. According to the Uniprot database release 2011_08, over 6000 reviewed human proteins have the word “membrane” in their annotations. This number makes up over 30% of all reviewed human proteins. The membrane proteins can be classified mainly into two categories, the integral membrane proteins and the peripheral membrane proteins. The integral membrane proteins contain trans-membrane domain that spans the lipid bilayer with a structure of alpha-helix or beta-barrel. The peripheral membrane proteins can interact with membranes through different mechanisms, mainly various types of anchors (Ghosh et al., 2008).

A cell is deemed to be the most basic structure and functional unit of almost every living creature, including human. Every living cell is composed by a number of different sub-compartments, which are usually separated with each other by one or two layers of membranes. These sub-compartments are known as the subcellular organelles. The subcellular localization of a protein is the subcellular compartment where the protein can perform its function within. The knowledge of protein subcellular localization is important for unraveling its cellular functions. The membrane proteins play crucial roles in mediating exchange of matter, energy and information among different subcellular compartments. The localization of membrane proteins can be very helpful in elucidating their biological functions (Casadio et al., 2008).

Although the subcellular locations of a given protein can be determined by experimental approaches, like fluorescent protein tagging and immunofluorescence, these approaches are usually costly and time consuming (Sadowski et al., 2008). With the

* Corresponding author. Tel/Fax: +86 22 23689450.
E-mail address: PufengDu@gmail.com (P. Du).

0022-5193/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jtbi.2012.08.016

explosion increment of protein sequences, the *in silico* methods for predicting protein subcellular locations provide an alternative choice. These methods are of particular interest as they are fast, cheap and can save hours of laborious experiments. Actually, in the last decade, many different computational methods were introduced, like BaCello (Pierleoni et al., 2006), pTARGET (Guda, 2006), KnowPred (Lin et al., 2009), YLoc (Briesemeister et al., 2010), WegoLoc (Chi and Nam, 2012), Hum-mPLoc (Shen and Chou, 2009), Sort-PLoc (Wan et al., 2011) and Cell-PLoc (Chou and Shen, 2008). Most of these methods included advanced machine learning algorithms, including SVM (Li et al., 2012), ensemble classifier (Li et al., 2012; Nanni and Lumini, 2006; Nanni and Lumini, 2007), mRMR (Cai et al., 2010; Cai et al., 2012; Niu et al., 2010), Genetic Algorithm (Mazzucato et al., 2008), SFFS (Nanni and Lumini, 2006; Yuan et al., 2010) and many others. However, none of these methods were developed specifically to identify the membrane protein localizations. They are general purpose predictors, which are trained and tested mainly on globular proteins. Although a most recent study concerning the comparison of trans-membrane domains provided a method for predicting membrane protein localizations (Sharpe et al., 2010), the ability of this method was restricted to only single-span trans-membrane proteins from ER, Golgi and plasma membranes (Sharpe et al., 2010).

As far as we know, the only available method, which is specifically suited for predicting membrane protein localizations, is the MemLoci algorithm (Pierleoni et al., 2011). The MemLoci algorithm can designate a protein to one of the three subcellular locations, including Plasma Membrane, Internal Membrane and Organelle Membrane. With the sequence based descriptors and the SVM classifiers, MemLoci achieved better performance on the benchmarking dataset than those general purpose predictors (Pierleoni et al., 2011). Although MemLoci reported that the Internal Membrane and Organelle Membrane compartments can be further divided into 10 more accurate subcellular locations, it did not perform prediction at this more accurate level. These accurate locations were not provided in the MemLoci benchmarking dataset.

In this article, we would try to improve the performance in predicting membrane protein localizations not only in term of prediction accuracy, but also the prediction quality and application range. To be more precise, we would improve the prediction performance in the following aspects. (1) We would carry out one level deeper prediction than the MemLoci, providing eight different accurate subcellular locations as prediction results for the internal and organelle membrane proteins. Because the MemLoci algorithm has provided good predictions to the plasma membrane proteins, which provide a very successful preprocessing step to our algorithm in identifying plasma membrane proteins, we did not put the plasma membrane proteins into our consideration. (2) We would provide higher prediction accuracy. (3) Our algorithm would consider the situation that a protein can localize simultaneously to more than one subcellular location, as there are over 30% of known proteins with multiple subcellular locations (Du et al., 2011; Shen and Chou, 2007). (4) We would carry out species-specific predictions, where only human membrane proteins were considered, as the species-specific prediction had been proved to be more useful in practical studies (Shen and Chou, 2009).

Since the MemLoci has demonstrated the performance boundary when only the protein sequence information was applied, we have to introduce novel and informative protein descriptors to achieve our goal. Gene Ontology annotations are very commonly applied in predicting protein subcellular locations (Lei and Dai, 2006; Fyshe et al., 2008; Mei et al., 2011; Chi, 2010). However, most of the GO based studies suffer from two problems. Firstly, the reliable Gene Ontology annotations of a newly sequenced protein usually do not exist, while many algorithms require experimentally determined GO annotations. The users cannot achieve a result with only a protein sequence. Secondly, the Gene Ontology annotation databases, like GOA, update frequently, the computation procedures may be affected by the updates of Gene Ontology databases. This problem makes it difficult to maintain a stable implementation of the algorithm. By introducing the Projected Gene Ontology Scores algorithm, we successfully avoid the above two problems. Additionally, the dataset scarcity problem, which was described by the MemLoci (Pierleoni et al., 2011), still exists in our study. But, our algorithm could tackle this problem by its stability and strong generalization ability with very limited data. Our algorithm could prompt the development of tools for predicting membrane protein localizations and provides a useful complementary to the existing algorithms.

2. Materials and methods

2.1. Dataset construction

We extracted the raw data from the Uniprot database (UniProt, 2010) release 2011_08 with the online data retrieval system. This raw dataset contained not only the protein sequences, but also the comment lines and all related GO terms. To further construct a high quality working dataset, this raw dataset was filtered strictly according to the following steps.

(1) Only reviewed non-fragment human proteins were included.
(2) All subcellular location annotations that were marked as “Possible”, “Probable”, “Potential” or “By Similarity” were excluded.
(3) The sequences without any internal or organelle membrane associated subcellular location annotations were excluded.
(4) All non-membrane proteins were excluded. The Uniprot database provides a controlled vocabulary to describe the topology of membrane proteins. This vocabulary includes the following 11 terms: GPI-anchor, GPI-like-anchor, Lipid-anchor, Multi-pass membrane protein, Peptidoglycan-anchor, Peripheral membrane protein, Single-pass membrane protein, Single-pass type I membrane protein, Single-pass type II membrane protein, Single-pass type III membrane protein and Single-pass type IV membrane protein. If a protein record had at least one of these 11 terms in its comments lines, it would be recognized as a membrane protein.
(5) The remaining protein sequences were processed with the CD-HIT program (Li and Godzik, 2006). The sequence similarity in the processed dataset was controlled to less than 60%.
(6) The proteins with single and multiple subcellular locations were separated. For the proteins with single subcellular location annotation, if a subcellular location contained less than 20 proteins, all proteins in this location would be excluded, as they cannot provide enough information for further analysis. The remaining proteins with single subcellular location annotation composed the dataset MP60S. The remaining proteins with multiple subcellular locations provided the dataset MP60M.

The dataset MP60S contains 981 proteins from 8 different subcellular locations. Table 1 gives the summary of MP60S dataset. The proteins in MP60M can have multiple subcellular locations. At least one location of each protein in MP60M was covered by the 8 locations of MP60S dataset. The distribution of subcellular location numbers can be found in Fig. 1. Since the number of proteins with multiple locations is almost half of that with single locations, they should not be ignored. In this study, we

Table 1
MP60S dataset summary.

Compartments	Count[a]
Endosome	96
ER	295
Golgi	240
Lysosome	31
Mitochondrion	202
Nucleus	36
Peroxisome	26
Vesicle	55
Overall	981

[a] These numbers were counted only in MP60S dataset.

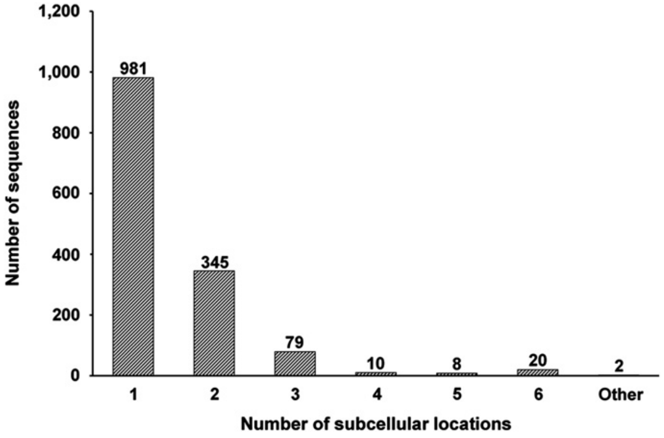

Fig. 1. The distribution of subcellular location numbers. The proteins in the benchmarking datasets have different number of annotated subcellular locations. Two third of all proteins have only one subcellular location, while the other one third has more than one subcellular location annotations.

would use MP60S dataset to train and cross validate our algorithm. The MP60M dataset would be used only as an independent testing dataset to see the behavior of our algorithm if we allow it to provide more than one subcellular locations. This testing strategy is similar to the pioneer study in subnuclear location prediction (Lei and Dai 2005).

2.2. Projected gene ontology Scores

We proposed a novel algorithm, namely Projected Gene Ontology Scores, for predicting the subcellular localizations of human membrane proteins. This algorithm is relatively simpler for implementation and easier for interpretation than other machine learning algorithms.

Without losing generality, we denoted the membrane proteins of every subcellular location as a subset (i.e. S_1, S_2, …, S_8) of the whole training dataset T.

$$T = \bigcup_{j=1}^{8} S_j \tag{1}$$

The ith protein in the jth location was denoted as P_{ij}, while the number of proteins in the jth location was M_j (i.e. $P_{ij} \in S_j$, $i=1, 2, \ldots, M_j$, $j=1, 2, \ldots, 8$). The gene ontology terms, which had been annotated to the protein P_{ij}, composed the gene ontology annotation set of the protein P_{ij}. We denoted this gene ontology annotation set as $G(P_{ij})$. The number of gene ontology terms in $G(P_{ij})$ was N_{ij}.

If we denoted the number of all gene ontology terms in the gene ontology annotation database as N_g, the gene ontology annotations of P_{ij} can be represented as a N_g-dimension binary vector, which can be denoted as $\mathbf{V}(P_{ij})$. Every dimension of this vector corresponds to a single gene ontology term in the gene ontology annotation database. If the protein P_{ij} was annotated with the kth gene ontology term in the gene ontology annotation database, the kth dimension of $\mathbf{V}(P_{ij})$ was set to one, otherwise zero. If we use $g_1, g_2, \ldots, g_{Ng}$ to denote the gene ontology terms in the database, the kth dimension of $\mathbf{V}(P_{ij})$ can be constructed using

$$V_k(P_{ij}) = \begin{cases} 1 & g_k \in G(P_{ij}) \\ 0 & g_k \notin G(P_{ij}) \end{cases}, k = 1, 2, \cdots, N_g \tag{2}$$

Since there are N_{ij} terms in $G(P_{ij})$ and $\mathbf{V}(P_{ij})$ is a binary vector, the Euclidean length of $\mathbf{V}(P_{ij})$ in N_g dimension space should be calculated as Eq. (3).

$$\|\mathbf{V}(P_{ij})\| = \sqrt{\sum_{k=1}^{N_g} V_k^2(P_{ij})} = \sqrt{N_{ij}} \tag{3}$$

Given a testing protein P, we used BLASTP program to find up to two most similar sequences in the entire Uniprot database. The gene ontology annotations of these similar sequences were used as the estimation of gene ontology annotations of P. This estimated gene ontology term set was denoted as $G_E(P)$, which contained N gene ontology terms.

Similarly, we constructed the $G_E(P)$ in the same N_g dimension space as the training samples. The N_g dimension binary vector of P is denoted as $\mathbf{V}(P)$.

The basic idea of Projected Gene Ontology Scores is to project $\mathbf{V}(P)$ on every $\mathbf{V}(P_{ij})$ in subset S_j and sum up the projection length to see whether P belongs to some subsets more than other subsets.

The projection length of $\mathbf{V}(P)$ on every single $\mathbf{V}(P_{ij})$ can be calculated as

$$m(P,P_{ij}) = \mathbf{V}(P) \cdot \mathbf{V}(P_{ij}) / \|\mathbf{V}(P_{ij})\| = \mathbf{V}(P) \cdot \mathbf{V}(P_{ij}) / \sqrt{N_{ij}} \tag{4}$$

We sum up all projections in one subset S_j to calculate the primary projected gene ontology score as Eq. 5.

$$R_j(P) = \sum_{i=1}^{M_j} m(P,P_{ij}) = \mathbf{V}(P) \cdot \sum_{i=1}^{M_j} \left[\mathbf{V}(P_{ij}) / \sqrt{N_{ij}} \right] \tag{5}$$

If we define the representative vector of subset S_j as Eq. (6)

$$\mathbf{C}_j = \sum_{i=1}^{M_j} \left[\mathbf{V}(P_{ij}) / \sqrt{N_{ij}} \right] \tag{6}$$

the primary score can be written as

$$R_j(P) = \mathbf{V}(P) \cdot \mathbf{C}_j \tag{7}$$

Since the number of proteins in each subset is different, the length of $\mathbf{C}_j$ could be significantly different. The primary projection score could bias on those subset containing more training samples. Thus, we need to normalize the primary score to get the final projected gene ontology score as Eq. (8).

$$F_j(P) = \mathbf{V}(P) \cdot \mathbf{C}_j / \|\mathbf{C}_j\| \tag{8}$$

From Eqs. (6)–(8), we can see that the $F_j(P)$ did not rely on the value of N_g. Since the N_g is the number of all gene ontology terms in the database, $F_j(P)$ would not be affected if new gene ontology terms are introduced in the database updates. This would make the algorithm more stable to the update of gene ontology system.

The locations would be sorted according to the descending order of the final projected gene ontology scores. If we need to provide Y locations as the prediction results, the first Y locations will be reported. In the case that we only need to provide a single location, the protein would be designated to the location that provides the maximum value of final projected gene ontology score.

In the above procedures, the computations can be separated into two parts. The first part is to calculate the $\mathbf{C}_j$ vectors from the training dataset. We call this part the training procedure of the algorithm. The other part of the computation is to calculate $F_j(P)$ with the testing sample P. We call this part the testing procedure of the algorithm. The terms training and testing are similar to what we use in machine learning studies. The training procedure creates the model, which is the representative vector $\mathbf{C}_j$ in this paper. The testing procedure used the trained model to make the prediction results to the testing samples.

2.3. Evaluation methods

Jackknife test, which deemed to be the most objective and rigorous method for evaluating predictive bioinformatics methods, was applied to estimate the prediction performance of this algorithm on MP60S dataset. When performing the jackknife test, all the GO terms were estimated using the GO supporting sets, regardless of whether the protein has its own GO annotations. Especially, when finding the similar sequences, the BLASTP procedure would ignore the query protein itself in the Uniprot database, assuming the query protein has not been annotated with any GO terms. This actually simulated the worst case in practical studies, which resulted in a very conservative estimation of prediction performance.

Four statistics, including accuracy (ACC), positive predictive value (PPV), overall accuracy ($ACC_{Overall}$) and Matthew's Correlation Coefficients (MCC), were applied to measure the prediction performance of this method on the MP60S dataset. These statistics can be calculated according to Eqs. (9)–(12).

$$ACC_j = TP_j/(TP_j+FN_j) \tag{9}$$

$$PPV_j = TP_j/(TP_j+FP_j) \tag{10}$$

$$ACC_{Overall} = \left(\sum_{j=1}^{8} TP_j\right) / \left(\sum_{j=1}^{8} M_j\right) \tag{11}$$

$$MCC_j = (TP_jTN_j - FP_jFN_j) / \sqrt{(TP_j+FP_j)(TP_j+FN_j)(TN_j+FP_j)(TN_j+FN_j)} \tag{12}$$

The TP_j, FP_j, TN_j and FN_j are the number of true positives, false positives, true negatives and false negatives of the jth location. The M_j is the total number of proteins in the jth location.

When we tested our algorithm on the MP60M dataset, we configured our algorithm to provide more than one subcellular locations. The above performance measures could not be well established as one protein can have more than one subcellular location. The Top-K measure, which has been applied in other studies (Chou and Cai, 2005; Lee et al., 2008), was used to measure the prediction performance of our algorithm on MP60M dataset. The Top-K measure is defined as the fraction of correctly predicted samples, in which the prediction is considered correct if at least one of the known locations is included in the K predicted locations (Chou and Cai, 2005; Lee et al., 2008).

3. Results and discussions

3.1. Prediction performance

We estimate the prediction performance of our algorithm on the MP60S dataset with the jackknife test method. The performance detail can be found in Table 2. The prediction accuracy of every subcellular location is higher than 85%. The overall prediction accuracy is over 90%. This indicates our algorithm performs well in jackknife test. Although the PPV is not as high as the accuracy, it is still acceptable.

Table 2
Jackknife test result on MP60S dataset.

Compartments	PPV[a](%)	ACC[b](%)	MCC[c]
Endosome	77.4	85.4	0.79
ER	93.2	93.6	0.91
Golgi	96.0	89.6	0.90
Lysosome	77.8	90.3	0.83
Mitochondrion	95.9	93.6	0.93
Nucleus	70.8	94.4	0.81
Peroxisome	88.5	88.5	0.88
Vesicle	91.7	80.0	0.85
Overall		90.8	

[a] PPV: Positive Predictive Value.
[b] ACC: Accuracy.
[c] MCC: Matthew's correlation coefficients.

Table 3
Independent dataset test results on MP60S dataset.

Test Proportion[a]	ACC[b](%)	STDEV[c](%)
0.3	89.8	1.6
0.5	90.4	1.4
0.7	89.7	0.7
0.9	87.0	1.7

[a] The proportion of dataset that were used as testing dataset.
[b] ACC: Overall prediction accuracy in independent dataset test. For each test proportion, the validations were repeated for 10 times. The average value of these 10 times repeats were reported in the table.
[c] STDEV: Stand deviation of the ACC in 10 times repeats.

To further evaluate the prediction performance of our algorithm, we carried out a serial of independent dataset test. In every independent dataset test, a fixed proportion of samples were randomly selected as the testing dataset. These selected samples were removed from the original dataset. The remaining samples were used to retrain the algorithm. The prediction performance would be estimated on the testing dataset. These procedures would be repeated ten times for every different testing proportion to see if the performance would vary significantly for different random selections. The average and the standard deviation of the performance in ten repeats would be reported. The testing proportions were set to 30%, 50%, 70% and 90%. Altogether 40 different independent dataset test were carried out. We find that even we use only 10% samples to train our algorithm and 90% samples as the testing dataset, the overall accuracy can still be over 87%. In the meantime, the variations of the performances are very small, less than 2%. Table 3 gives the performance estimated by independent dataset test. These results not only indicate that our algorithm is not over-optimized, but also implied that our algorithm is robust to different testing dataset and possess good generalization ability.

To further demonstrate this advantage, we carried out a simple comparison between our algorithm and the widely applied SVM classifier. We use the same features and the same independent dataset test method to evaluate the performance of an SVM classifier trained with a Gaussian kernel. We applied the most commonly used SVM software libSVM (Chang and Lin, 2011) and optimized the SVM parameters with a grid search strategy using the script within the software package. If 30% samples were used as the testing dataset, the SVM classifier could achieve over 94% prediction accuracy, which is much better than our algorithm. However, when 90% of the samples were used as the testing dataset, and only 10% samples were used as the training dataset, the performance of the SVM classifier dropped to about 76%. This observation not only indicates that the SVM classifier is easy to be over-optimized, but also implies that the prediction performance of an SVM classifier could become worse when the known training samples are much less than the unknown testing samples. Considering the development of proteome science, the unknown samples could be much more than the currently known ones in future. Thus, it would be better to use our algorithm than the SVM classifier for newly sequenced proteins.

To further demonstrate the performance of our algorithm, comparisons to several existing GO based subcellular location predictors, which are not designed for membrane proteins, were carried out. The comparison details are provided in the online supplementary materials A.

3.2. Parameter analysis

Although this algorithm relies on the gene ontology annotations, when applying this algorithm in practical studies, the users only need to provide the protein sequences. As we have described in the Method section, given a testing protein sequence, the BLASTP program would be applied to find up to two most similar sequences in the Uniprot database. The gene ontology annotations of these similar sequences would be used to estimate the gene ontology annotations of the testing protein. This procedure actually introduced the only parameter in our algorithm, which is the number of similar sequences that would be found in the database. To determine the default value of this parameter, different numbers of similar sequences (from 1 to 9) were applied. Fig. 2 gives the overall accuracy in jackknife test with different parameters. Actually, the prediction performance was not affected significantly by this parameter. When this parameter was set to two, we can achieve the best performance. Thus, the default value of this parameter was set to two.

In Fig. 2, we have a dot for zero similar sequences. This dot was depicted with the gene ontology terms, which have been annotated to the protein sequences in the UniProt database. We call these gene ontology annotations as the real gene ontology annotations. The prediction performance for using the real gene ontology annotations is 3% lower than using our gene ontology estimations. Since gene ontology annotations in the UniProt database could be incomplete for some proteins, our estimation method may provide more comprehensive information than currently available annotations. This is also consistent with the fact that the homology search is a main strategy to electronically annotate proteins with gene ontology. Thus, requiring sequences as input and estimating the gene ontology annotations not only give the user more convenience, but also improve the performance of the algorithm.

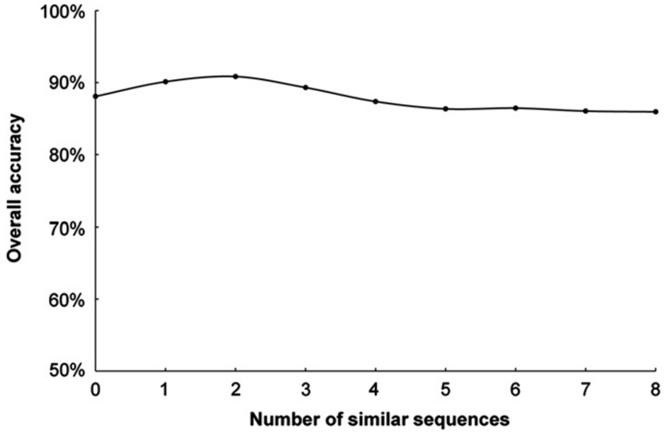

Fig. 2. The parameter analysis result. The only parameter in this algorithm is the number of similar sequences that need to be found in the Uniprot database. The value of this parameter only affects the performance slightly. The default value of this parameter is two, as there is a slight peak of the overall accuracy when it is two. The “zero” dot in the figure indicates the performance using the gene ontology annotations which have been annotated in the UniProt database.

3.3. Predicting multiple locations for one protein

As we have mentioned, over 30% of known proteins have multiple locations. Thus, we have to consider providing more than one subcellular location for each protein as the prediction result. This is necessary for all proteins as the protein currently with only one location could have other subcellular localizations in future.

We trained our algorithm with MP60S dataset and tested the trained model with MP60M dataset. We used our algorithm to provide one, two, three and four locations for every protein in MP60M dataset. We used the Top-K accuracy to measure the prediction performance under this condition. Fig. 3A gives the performance of our algorithm with different numbers of predicting locations on MP60M dataset. It is easy to understand the increment of performance when the number of predicted locations was increasing. Especially, when predicting two locations for every protein, the prediction accuracy was over 94%. Considering the total number of proteins in MP60M, our algorithm missed only 28 proteins under this condition.

In addition, we carried out a test for providing more than one locations for the MP60S dataset in jackknife test. Since there may be undiscovered localization of currently single localized proteins, providing more than one locations for these proteins could be very useful. We provided one, two, three and four locations for every protein in MP60S dataset in a jackknife test. The prediction performance increased significantly when more locations were provided (Fig. 3B). When we provided four locations for every protein, the prediction performance was over 98%.

In general, if we provided two locations for every protein, our algorithm can predict at least one correct location for over 95% of all proteins, regardless of whether they localized in one or more locations.

3.4. Comparison with MemLoci

Although we have emphasized that our work dedicated in solving the problem one level deeper than the MemLoci, it is interesting to see whether our algorithm could work for the same problem that MemLoci has tackled. We obtained the dataset of MemLoci from its official website. According to MemLoci, all 10634 proteins were categorized into three different groups, the Cellular Membrane group, the Internal Membrane group and the Organelle Membrane group. The MemLoci algorithm was evaluated by 10-fold cross validation. To provide a comparable result with MemLoci, we re-trained our algorithm using the MemLoci dataset and used 10-fold cross validation to estimate the prediction performance of our algorithm on the MemLoci dataset. The performances of both algorithms were measured using the statistics defined by MemLoci (Table 4). Our algorithm performed significantly better than the MemLoci algorithm, not only in terms of higher accuracy, but also in terms of higher MCC and lower FPR.

3.5. Availability

We have implemented our algorithm as an online service called SubMem, which allows the readers to use the most basic function of our algorithm. The SubMem service can be accessed at http://59.67.33.228/biosrv/submem. The input data should be

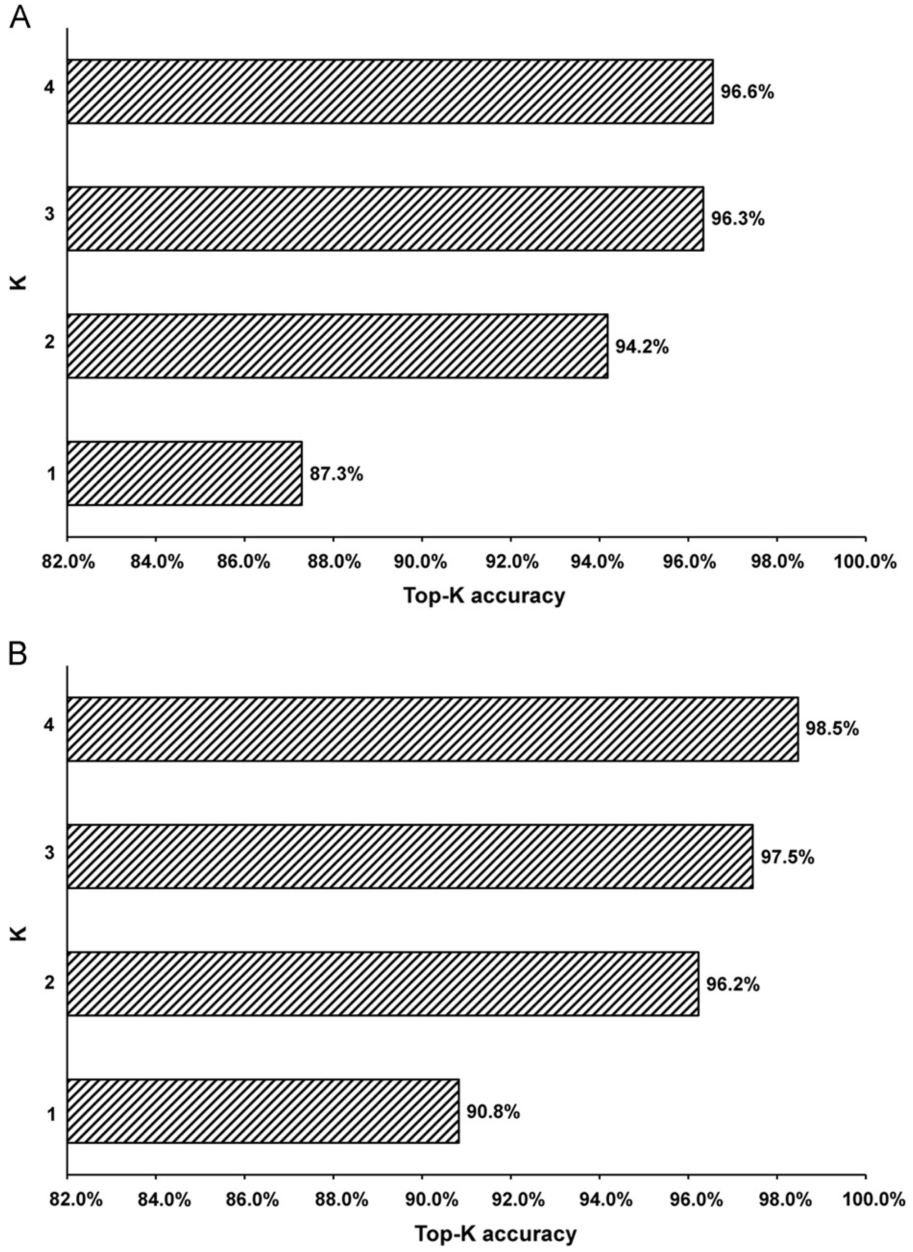

Fig. 3. Performance analysis when predicting multiple locations for every protein. (A) The overall Top-K accuracy on MP60M dataset. The training dataset was the MP60S dataset. The vertical axis is the value of K in Top-K accuracy, indicating the algorithm provided K locations for every protein. (B) The overall Top-K accuracy for the jackknife test on MP60S dataset. The vertical axis is the value of K in Top-K accuracy, indicating the algorithm provided K locations for every protein.

Table 4
Performance comparison with MemLoci.

Compartments	This algorithm			MemLoci		
	ACC[a](%)	FPR[b](%)	MCC[c]	ACC(%)	FPR(%)	MCC
Plasma membrane	88.3	2.0	0.88	56.0	15.0	0.43
Internal membrane	95.9	8.4	0.87	72.0	30.0	0.42
Organelle membrane	94.8	1.2	0.94	70.0	9.0	0.60
Overall	92.8			66.0		

[a] ACC: Accuracy.

[b] FPR: False Positive Rate. This was defined in the MemLoci study to indicate the performance of identifying negative samples for each location. FPR=FP/(TN+FP) (Pierleoni et al., 2011).

[c] MCC: Matthew's correlation coefficients.

FASTA format sequences. The prediction results would be annotated to the FASTA sequences. The benchmarking datasets of this algorithm are provided as the online supplementary material B.

4. Conclusion

We proposed a novel algorithm, Projected Gene Ontology Score, for predicting human membrane protein subcellular localization. This algorithm can serve as the complement to the existing algorithms, as our algorithm goes one level deeper than the existing algorithms. Since the performance of our algorithm is better than the existing algorithms and the commonly used machine learning algorithms, it is hopeful to be applied in other studies.

Acknowledgments

We would thank Dr. Bo Zhang and Dr. Tingting Li for helpful suggestions in writing the manuscript. This work was supported by National Science Foundation of China (NSFC 61005041), the Ph.D. Programs Foundation of Ministry of Education of China (no. 20100032120039), Tianjin Natural Science Foundation (no. 12JCQNJC02300) and the Seed Foundation of Tianjin University (nos. 60302006 and 60302024).

Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.jtbi.2012.08.016.

References

Briesemeister, S., Rahnenfuhrer, J., Kohlbacher, O., 2010. Going from where to why–interpretable prediction of protein subcellular localization. Bioinformatics 26, 1232–1238.

Cai, Y., Huang, T., Hu, L., Shi, X., Xie, L., Li, Y., 2012. Prediction of lysine ubiquitination with mRMR feature selection and analysis. Amino Acids 42, 1387–1395.

Cai, Y.D., Lu, L., Chen, L., He, J.F., 2010. Predicting subcellular location of proteins using integrated-algorithm method. Mol. Divers 14, 551–558.

Casadio, R., Martelli, P.L., Pierleoni, A., 2008. The prediction of protein subcellular localization from sequence: a shortcut to functional genome annotation. Brief Funct. Genomic Proteomic 7, 63–73.

Chang, C.-C., Lin, C.-J., 2011. LIBSVM: a library for support vector machines. ACM Trans. Intelligent Sys. Technol. 2, 27.

Chi, S.M., 2010. Prediction of protein subcellular localization by weighted gene ontology terms. Biochem. Biophys. Res. Commun. 399, 402–405.

Chi, S.M., Nam, D., 2012. WegoLoc: accurate prediction of protein subcellular localization using weighted Gene Ontology terms. Bioinformatics 28, 1028–1030.

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

Chou, K.C., Shen, H.B., 2008. Cell-PLoc: a package of Web servers for predicting subcellular localization of proteins in various organisms. Nat. Protoc. 3, 153–162.

Du, P., Li, T., Wang, X., 2011. Recent progress in predicting protein sub-subcellular locations. Expert Rev. Proteomics 8, 391–404.

Fyshe, A., Liu, Y., Szafron, D., Greiner, R., Lu, P., 2008. Improving subcellular localization prediction using text classification and the gene ontology. Bioinformatics 24, 2512–2517.

Ghosh, D., Beavis, R.C., Wilkins, J.A., 2008. The identification and characterization of membranome components. J. Proteome Res. 7, 1572–1583.

Guda, C., 2006. pTARGET: a web server for predicting protein subcellular localization. Nucleic Acids Res. 34, W210–W213.

Lee, K., Chuang, H.Y., Beyer, A., Sung, M.K., Huh, W.K., Lee, B., Ideker, T., 2008. Protein networks markedly improve prediction of subcellular localization in multiple eukaryotic species. Nucleic Acids Res. 36, e136.

Lei, Z., Dai, Y., 2005. An SVM-based system for predicting protein subnuclear localizations. BMC Bioinformatics 6, 291.

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

Li, L., Zhang, Y., Zou, L., Li, C., Yu, B., Zheng, X., Zhou, Y., 2012. An ensemble classifier for eukaryotic protein subcellular location prediction using gene ontology categories and amino acid hydrophobicity. PLoS One 7, e31057.

Li, W., Godzik, A., 2006. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics 22, 1658–1659.

Lin, H.N., Chen, C.T., Sung, T.Y., Ho, S.Y., Hsu, W.L., 2009. Protein subcellular localization prediction of eukaryotes using a knowledge-based approach. BMC Bioinformatics 10 (Suppl 15), S8.

Mazzucato, A., Papa, R., Bitocchi, E., Mosconi, P., Nanni, L., Negri, V., Picarella, M.E., Siligato, F., Soressi, G.P., Tiranti, B., Veronesi, F., 2008. Genetic diversity, structure and marker-trait associations in a collection of Italian tomato (Solanum lycopersicum L.) landraces. Theor. Appl. Genet. 116, 657–669.

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

Nanni, L., Lumini, A., 2006. An ensemble of K-local hyperplanes for predicting protein–protein interactions. Bioinformatics 22, 1207–1210.

Nanni, L., Lumini, A., 2007. Ensemblator: An ensemble of classifiers for reliable classification of biological data. Pattern Recognition Lett. 28, 622–630.

Niu, S., Huang, T., Feng, K., Cai, Y., Li, Y., 2010. Prediction of tyrosine sulfation with mRMR feature selection and analysis. J. Proteome Res. 9, 6490–6497.

Pierleoni, A., Martelli, P.L., Fariselli, P., Casadio, R., 2006. BaCelLo: a balanced subcellular localization predictor. Bioinformatics 22, e408–e416.

Pierleoni, A., Martelli, P.L., Casadio, R., 2011. MemLoci: predicting subcellular localization of membrane proteins in eukaryotes. Bioinformatics 27, 1224–1230.

Sadowski, P.G., Groen, A.J., Dupree, P., Lilley, K.S., 2008. Sub-cellular localization of membrane proteins. Proteomics 8, 3991–4011.

Sharpe, H.J., Stevens, T.J., Munro, S., 2010. A comprehensive comparison of transmembrane domains reveals organelle-specific properties. Cell 142, 158–169.

Shen, H.B., Chou, K.C., 2007. Hum-mPLoc: an ensemble classifier for large-scale human protein subcellular location prediction by incorporating samples with multiple sites. Biochem. Biophys. Res. Commun. 355, 1006–1011.

Shen, H.B., Chou, K.C., 2009. A top-down approach to enhance the power of predicting human protein subcellular localization: Hum-mPLoc 2.0. Anal. Biochem. 394, 269–274.

UniProt.Consortium., 2010. The Universal Protein Resource (UniProt) in 2010. Nucleic Acids Research, vol. 38, pp. D142-8.

Wan, S.-B., Hu, L.-L., Niu, S., Wang, K., Cai, Y.-D., Lu, W.-C., Chou, K.-C., 2011. Identification of multiple subcellular locations for proteins in budding yeast. Curr. Bioinform. 6, 71–80.

Yuan, Y., Shi, X., Li, X., Lu, W., Cai, Y., Gu, L., Liu, L., Li, M., Kong, X., Xing, M., 2010. Prediction of interactiveness of proteins and nucleic acids based on feature selections. Mol. Divers 14, 627–633.