research articles Journal of proteome research

Predicting Eukaryotic Protein Subcellular Location by Fusing Optimized Evidence-Theoretic K-Nearest Neighbor Classifiers

Kuo-Chen Chou*,[†,‡] and Hong-Bin Shen[‡]

Gordon Life Science Institute, 13784 Torrey Del Mar Drive, San Diego, California 92130, and Institute of Image Processing & Pattern Recognition, Shanghai Jiaotong University, 1954 Hua-Shan Road, Shanghai 200030, People's Republic of China

Received April 12, 2006

Facing the explosion of newly generated protein sequences in the post genomic era, we are challenged to develop an automated method for fast and reliably annotating their subcellular locations. Knowledge of subcellular locations of proteins can provide useful hints for revealing their functions and understanding how they interact with each other in cellular networking. Unfortunately, it is both expensive and time-consuming to determine the localization of an uncharacterized protein in a living cell purely based on experiments. To tackle the challenge, a novel hybridization classifier was developed by fusing many basic individual classifiers through a voting system. The "engine" of these basic classifiers was operated by the OET-KNN (Optimized Evidence-Theoretic K-Nearest Neighbor) rule. As a demonstration, predictions were performed with the fusion classifier for proteins among the following 16 localizations: (1) cell wall, (2) centriole, (3) chloroplast, (4) cyanelle, (5) cytoplasm, (6) cytoskeleton, (7) endoplasmic reticulum, (8) extracell, (9) Golgi apparatus, (10) lysosome, (11) mitochondria, (12) nucleus, (13) peroxisome, (14) plasma membrane, (15) plastid, and (16) vacuole. To get rid of redundancy and homology bias, none of the proteins investigated here had ≥25% sequence identity to any other in a same subcellular location. The overall success rates thus obtained via the jack-knife cross-validation test and independent dataset test were 81.6% and 83.7%, respectively, which were 46∼63% higher than those performed by the other existing methods on the same benchmark datasets. Also, it is clearly elucidated that the overwhelmingly high success rates obtained by the fusion classifier is by no means a trivial utilization of the GO annotations as prone to be misinterpreted because there is a huge number of proteins with given accession numbers and the corresponding GO numbers, but their subcellular locations are still unknown, and that the percentage of proteins with GO annotations indicating their subcellular components is even less than the percentage of proteins with known subcellular location annotation in the Swiss-Prot database. It is anticipated that the powerful fusion classifier may also become a very useful high throughput tool in characterizing other attributes of proteins according to their sequences, such as enzyme class, membrane protein type, and nuclear receptor subfamily, among many others. A web server, called "Euk-OET-PLoc", has been designed at *http://202.120.37.186/bioinf/euk-oet* for public to predict subcellular locations of eukaryotic proteins by the fusion OET-KNN classifier.

Keywords: cellular networking • organelle • gene ontology • amphiphilic pseudo amino acid composition • OET-KNN • fusion classifier • 25% sequence identity cutoff

I. Introduction

One of the fundamental goals in cell biology and proteomics is to identify the functions of proteins in the context of compartments that organize them in the cellular environment. Existing as independent units of life as in monads or forming colonies or tissues as in higher plants and animals, cells are the basic structural and functional units of all organisms. Every cell contains many different compartments or organelles (Figure 1). The organelles are specialized to carry out different tasks. Most of these functions, which are critical to the cell's survival, are performed by the proteins therein.[1,2] Accordingly, the significance to identify the subcellular localization of a newly found protein has become self-evident.

Although the information about protein subcellular localization can be determined by conducting various experiments, that is both expensive and time-consuming. Particularly, the number of newly found protein sequences has increased explosively in the post genomic era. For instance, in 1986 Swiss-

[*] To whom correspondence should be addressed. E-mail: kchou@san.rr.com.

[†] Gordon Life Science Institute.

[‡] Shanghai Jiaotong University.

10.1021/pr060167c CCC: $33.50

Published on Web 07/14/2006

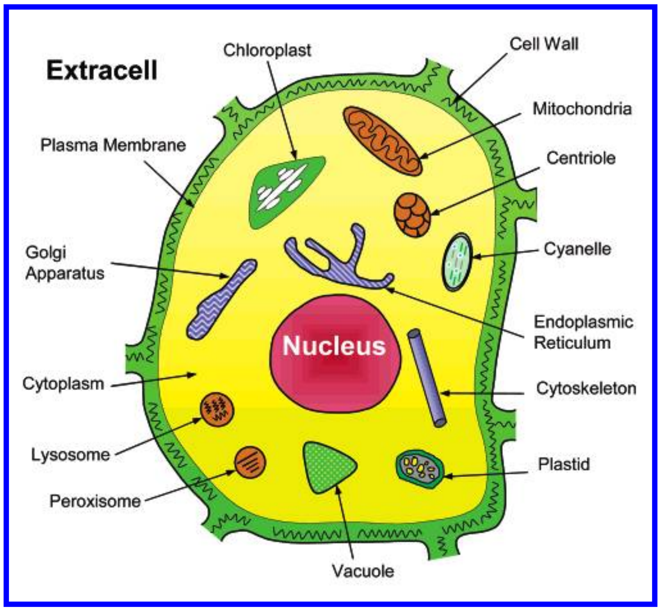

Figure 1. Schematic illustration to show the 16 subcellular locations of proteins: (1) cell wall, (2) centriole, (3) chloroplast, (4) cyanelle, (5) cytoplasm, (6) cytoskeleton, (7) endoplasmic reticulum, (8) extracell, (9) Golgi apparatus, (10) lysosome, (11) mitochondria, (12) nucleus, (13) peroxisome, (14) plasma membrane, (15) plastid, and (16) vacuole.

Prot[3] contained only 3939 protein sequence entries, but now the number has jumped to 216 380 according to version 49.5 of the UniProtKB/Swiss-Prot Release as of 18-Apr-2006, implying that the number of protein sequences has increased by about 55 times in about two decades. Facing such a “protein sequence explosion”, it is both challenging and indispensable to develop an automated method for fast and reliably annotating the subcellular attributes of uncharacterized proteins. The knowledge thus obtained can help us timely utilize these newly found protein sequences for both basic research and drug discovery.[4]

Many efforts have been made in this regard.[5–18] However, all these prediction methods were established basically on a single classifier derived from a single learning process regardless of whether the operation was engineered with the covariant discriminant algorithm, or SVM (support vector machine), or neural network. Obviously, the prediction quality might be limited by only using a single classifier to deal with complicated protein sequences with extreme variation in both sequence order and length.

Besides, the datasets constructed to train the existing predictors cover very limited cellular locations. For instance, the datasets constructed by Nakashima et al.[7] only cover two locations, those by Reinhardt and Hubbard[9] 3 or 4 locations, those by Matsuda et al.[18] and by Garg et al.[17] 4 locations, those by Cedano et al.[8] and Gardy et al.[19] 5 locations. Although the datasets constructed by the authors in[16] extended the coverage to 12 subcellular locations, these datasets were constructed with a very tolerant criterion that allowed inclusion of those homologous proteins with up to 80% sequence identity to each other. To avoid homology bias, a much stricter criterion should be adopted for constructing the benchmark datasets. Also, to make the predictor practically more useful, more subcellular locations should be covered.

To enlarge the scope of practical application and reduce the homology bias, new working datasets covering 16 subcellular locations were constructed and screened with a cutoff of 25% sequence identity to guarantee that none of proteins included has ≥25% sequence identity to any other within a same subcellular location. As is well-known, the more subcellular locations the classification involves, or the less homologous sequences the benchmark dataset contains, the harder it would be in getting a higher success prediction rate. To overcome such a difficulty, the samples of proteins were formulated by hybridizing the information derived from gene ontology[20] and amphiphilic pseudo amino acid composition.[21] On the basis of the hybridization representation, a novel classifier was formed by fusing many individual basic classifiers through a voting system. The success rates obtained by the fusion classifier in predicting protein subcellular location were significantly improved.

II. Materials

Protein sequences were collected from the Swiss−Prot database[3] release 48.2 at *http://www.ebi.ac.uk/swissprot/* according to their experimentally annotated subcellular locations (-!-SUBCELLULAR LOCATION). Because a same subcellular location might be annotated with different terms, to collect as much desired information as possible, several key words might be used for a same subcellular location. For example, in search for centriole proteins, the key words “centriole”, “centrosome”, and “centromer” were used; in search for cytoskeleton proteins, the key words “cytoskeleton”, “filament”, and “microtubule” were used; in search for extracell proteins, the key words “extracell”, “extracellular, and “secreted” were used; in search for peroxisome proteins, the key words “peroxisome”, “microsome”, “glyoxysomal”, and “glycosomal” were used; in search for plasma membrane proteins, the key words “plasma membrane” and “integral membrane” were used; and so forth. To obtain high-quality, well-defined working datasets, the data were screened strictly according to the following criteria. (**1**) Sequences annotated with “prokaryotic” were excluded because the current study was focused on eukaryotic proteins only. (**2**) Sequences annotated with ambiguous or uncertain words, such as “potential”, “probable”, “probably′”, “maybe”, or “by similarity”, were excluded. (**3**) Sequences annotated by two or more locations were not included because of lack of the uniqueness. (**4**) Sequences annotated with “fragment” were excluded; also, sequences with less than 50 amino acid residues were removed because they might just be fragments. (**5**) To avoid any homology bias, a redundancy cutoff was operated by a culling program[22] to winnow those sequences which have ≥25% sequence identity to any other in a same subcellular location. (**6**) Those subcellular locations (subsets) which contain less than 20 protein sequences were left out because of lacking statistical significance.

After strictly following the above procedures, we finally obtained 4150 protein sequences of which 25 belonged to cell wall, 21 to centriole, 258 to chloroplast, 97 to cyanelle, 718 to cytoplasm, 25 to cytoskeleton, 113 to endoplasmic reticulum, 806 to extracell, 85 to Golgi apparatus, 46 to lysosome, 228 to mitochondrion, 1169 to nucleus, 64 to peroxisome, 413 to plasma membrane, 38 to plastid, and 44 to vacuole (Figure 1). Thus, we have a dataset $\mathbb{S}^0$ which is a union of the following 16 subsets; i.e.,

$$\mathbb{S}^0 = S_1{}^0 \cup S_2{}^0 \cup S_3{}^0 \cup \cdots \cup S_{16}{}^0 \tag{1}$$

On the basis of dataset $\mathbb{S}^0$, two working datasets, i.e., a learning

Table 1. Number of Proteins in Each of the 16 Subcellular Locations for the Learning and Testing Datasets, Respectively

subcellular location	learning dataset $\mathbb{S}^{L}$	testing dataset $\mathbb{S}^{T}$
(1) cell wall	20	5
(2) centriole	17	4
(3) chloroplast	207	51
(4) cyanelle	78	19
(5) cytoplasm	384	334
(6) cytoskeleton	20	5
(7) endoplasmic reticulum	91	22
(8) extracell	402	404
(9) Golgi apparatus	68	17
(10) lysosome	37	9
(11) mitochondrion	183	45
(12) nucleus	474	695
(13) peroxisome	52	12
(14) plasma membrane	323	90
(15) plastid	31	7
(16) vacuole	36	8
total	2423	1727

(training) dataset $\mathbb{S}^{L}$ and an independent testing dataset $\mathbb{S}^{T}$, were constructed. To fully use the data in $\mathbb{S}^{0}$ and meanwhile guarantee that $\mathbb{S}^{L}$ and $\mathbb{S}^{T}$ be completely independent of each other, the following condition was imposed:

$$\mathbb{S}^{L} \cup \mathbb{S}^{T} = \mathbb{S}^{0} \text{ and } \mathbb{S}^{L} \cap \mathbb{S}^{T} = \varnothing \tag{2}$$

where $\cup$, $\cap$, and $\varnothing$ represent the symbols for "union", "intersection", and "empty set" in the set theory, respectively. Protein samples in the corresponding subsets of $\mathbb{S}^{L}$ and $\mathbb{S}^{T}$ are randomly assigned according to the following "bracket percentage distribution" criterion:

$$\begin{cases} n_i^{L} = 300 + \mathrm{INT}\{(n_i^{0} - 300) \times 0.2\}, & \text{if } n_i^{0} \geq 300 \\ n_i^{L} = \mathrm{INT}\{n_i^{0} \times 0.8\}, & \text{if } n_i^{0} < 300 \\ n_i^{T} = n_i^{0} - n_i^{L} \end{cases} \qquad (i = 1, 2, \cdots, 16) \tag{3}$$

where n_i^0 is the number of protein samples in the ith subset, S_i^0, of the original dataset $\mathbb{S}^0$, n_i^{L} that in the ith subset, S_i^{L}, of the learning dataset $\mathbb{S}^{L}$, and n_i^{T} that in the ith subset, S_i^{T}, of the testing dataset $\mathbb{S}^{T}$, whereas the symbol {INT} means taking the integer part for the number in the brackets right after it. The numbers of proteins thus obtained for the 16 subcellular locations in the learning dataset $\mathbb{S}^{L}$ and testing dataset $\mathbb{S}^{T}$are given in Table 1. The accession numbers and sequences for the corresponding proteins in the learning and testing datasets are given in the Online Supporting Information A and B, respectively.

III. Method

The key to enhance the prediction quality for protein subcellular location is to grasp the core features of a protein that are intimately related to its localization in a cell. Accordingly, the source of gene ontology consortium[20] can be used as a vehicle to formulate the samples of proteins. The rationale is as follows. The gene ontology, or GO, is a controlled vocabulary used to describe the biology of a gene product in any organism. The GO database was established based on the following three species-independent principles: molecular function, biological process, and cellular component.[23,24]

However, how to effectively use the GO database to improve the prediction quality for protein subcellular location is by no means a trivial problem. It is prone to ask: Can we just use the cellular component annotations in the GO database to annotate the location-unknown proteins in the Swiss-Prot database?[3] The answer is absolutely no with the reasons given below. As clearly indicated via a statistical analysis, the percentage of proteins with GO annotations to indicate their subcellular components is 97092/216380 ≃ 44.9% (Table 2); while the percentage of proteins with subcellular location annotation in the Swiss-Prot database is 113416/216380 ≃ 52.4%. In other words, the former is even less than the latter. Actually, for those proteins with "subcellular location unknown" annotation in Swiss-Prot database, most (more than 99%) of their corresponding GO numbers in GO database are also annotated with "cellular component unknown" (see, e.g., the protein with accession number Q25513 in Table 3). Even for those proteins whose subcellular locations are clearly annotated in Swiss-Prot database, their corresponding GO numbers in GO database are not always directly indicating their corresponding subcellular locations. In some cases they are actually annotated with "cellular component unknown". For example, for the protein with accession number Q9BYI3 in Table 3, its subcellular location is annotated with "cytoplasm" in Swiss-Prot database, but none of its GO numbers indicates its subcellular location. Similar situations also occur for the proteins with accession numbers O60477, O75897, and O43303 (Table 3). Therefore, the key is how to find an effective approach to incorporate the information on GO into the prediction algorithm. This can be realized through the formulation given below.

Mapping UniProtKB/Swiss-Prot protein entries[25] to the GO database, one can get a list of data called "gene_association.goa_uniprot", where each UniProtKB/Swiss-Port protein entry corresponds to one or several GO numbers. In this study, such a data file was directly downloaded from ftp://ftp.ebi.ac.uk/pub/databases/GO/goa/UNIPROT/ (released on 21-Nov-2005). The relationships between the UniProtKB/Swiss-Port protein entries and the GO numbers may be one-to-many (cf. Table 3), "reflecting the biological reality that a particular protein may function in several processes, contain domains that carry out diverse molecular functions, and participate in multiple alternative interactions with other proteins, organelles or locations in the cell".[20] Also, because the current GO database is not complete yet, some protein entries (such as "Q8K2Y9", "Q61900", and "P32034") have no corresponding GO numbers, i.e., no mapping records at all in the GO database, and hence are not included in gene_association.goa_uniprot.

The GO numbers do not increase successively and orderly. For easier handling, some reorganization and compression procedure was taken to renumber them. The GO database obtained through such a treatment is called GO_compress database, whose dimensions were reduced to 9567 from 51675 in the original GO database. Each of the 9567 entities in the GO_compress database served as a base to define a protein sample. Unfortunately, the current GO numbers failed to completely cover the proteins concerned, i.e., some proteins might not belong to any of the GO numbers. Although the problem would gradually become trivial or no longer exist with the continuous developing of GO database, to cope with such a problem right now, a hybridization approach was introduced by fusing the GO representation and the amphiphilic pseudo amino acid composition (PseAA) representation,[21] as formulated below.

1. Search a protein sample in the GO_compress database, if there is a hit corresponding to the ith GO_compress number,

Table 2. Breakdown of the 216 380 Protein Sequence Entries from Swiss-Prot Database (version 49.5, released 18-April-2006) According to the Nature of Their Subcellular Location Annotation and Their Expression in GO

item	description	no.	percentage
(1)	proteins with subcellular locations annotated in Swiss-Prot database	113416	113416/216380 = 52.4%
(2)	proteins in (1) with uncertain labels, such as “potential” and “probable”	65458	65458/216380 = 30.3%
(3)	proteins that can be represented in the GO space (see eq 4)	202857	202857/216380 = 93.8%
(4)	proteins with subcellular component annotations in the GO database	97092	97092/216380 = 44.9%

Table 3. Examples to Show the Subcellular Location Annotations for Some Proteins in the Swiss-Prot Database and the Annotations for the Corresponding GO Numbers in the GO Database

Swiss-Prot database		GO database	
accession no.	Swiss-Prot annotation	GO no.	GO annotation
P0A2X0	no subcellular location annotated	GO:0004040	amidase activity
		GO:0009851	auxin biosynthesis
		GO:0016787	hydrolase activity
Q25513	no subcellular location annotated	GO:0000004	biological process unknown
		GO:0005554	molecular function unknown
		GO:0008372	cellular component unknown
Q9BYI3	cytoplasm	GO:0000004	biological process unknown
		GO:0004871	signal transducer activity
		GO:0008372	cellular component unknown
O60477	extracellular	GO:0005515	protein binding
		GO:0007049	cell cycle
		GO:0008219	cell death
		GO:0008372	cellular component unknown
O75897	cytoplasm	GO:0000004	biological process unknown
		GO:0008146	sulfotransferase activity
		GO:0008372	cellular component unknown
		GO:0016740	transferase activity
O43303	centrosome	GO:0000004	biological process unknown
		GO:0005554	molecular function unknown
		GO:0008372	cellular component unknown

then the *i*th component of the protein in the 9567-D (dimensional) GO_compress space is assigned 1; otherwise, 0. Thus, the protein can be formulated as

$$\mathbf{P} = [G_1 \;\; G_2 \;\; \cdots \;\; G_i \;\; \cdots \;\; G_{9567}]^{\mathbf{T}} \tag{4}$$

where **T** is the transpose operator, and

$$G_i = \begin{cases} 1, & \text{hit found in GO\_compress} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

2. If no hit (i.e., no record in the GO_compress database) is found at all, then the protein should be defined in the (20 + 2λ) - D amphiphilic PseAA space,[21] as given below

$$\mathbf{P} = [F_1 \;\; \cdots \;\; F_{20} \;\; F_{20+1} \;\; \cdots \;\; F_{20+2\lambda}]^{\mathbf{T}} \tag{6}$$

where F_1, F_2,..., F_{20} are associated with the amino acid composition reflecting the occurrence frequencies of the 20 native amino acids in the protein,[7,26] and F_{20+1}, F_{20+2}, ..., $F_{20+2\lambda}$ are the 2λ correlation factors that reflect its sequence-order pattern through the amphiphilic feature. The protein representation as defined by eq 6 is called the “amphiphilic pseudo amino acid composition” or PseAA, which has the same form as the conventional amino acid composition but contains more components and information. For reader’s convenience, a brief introduction about the PseAA and the key equations for deriving its components are provided in Appendix A.

Suppose there are N proteins ($\mathbf{P}_1$, $\mathbf{P}_2$, ..., $\mathbf{P}_N$) which have been classified into 16 subsets (subcellular locations). Now, for a query protein **P**, how can we identify which subset it belongs to? To deal with this problem, various classifiers have been proposed, such as the Least Euclidean distance classifier,[7] ProtLoc classifier,[8] SVM (Support Vector Machine) classifier,[14,27,28] NN (Nearest Neighbor) classifier,[29] and KNN (K-Nearest Neighbor) classifier.[30] The KNN classifier is quite popular in pattern recognition community owing to its good performance and simple-to-use feature. According to the KNN rule,[30−32] also named as the “voting KNN rule, the query protein should be assigned to the subset represented by a majority of its K-Nearest Neighbors. Since the inception of KNN classifier, some modified versions have been proposed in order to further improve its performance. One of these modified versions is the OET-KNN (Optimized Evidence-Theoretic K-Nearest Neighbors) classifier, which has been proved very powerful in statistical prediction,[30,33,34] and hence is adopted here as the basic operation engine during the prediction process. For reader’s convenience, a brief introduction about OET-KNN classifier and its key equations are given in Appendix B.

There are two parameters that may directly affect the predicted result of an OET-KNN classifier. One is K, the number of the nearest proteins counted against the query protein during the prediction process; the other is the number of components used to represent the protein samples. When a protein is represented in the GO_compress space, its dimension (number of components) is fixed at 9567 (eq 4), but in the PseAA space (eq 6), its dimension (20 + 2λ) is allowed to vary.

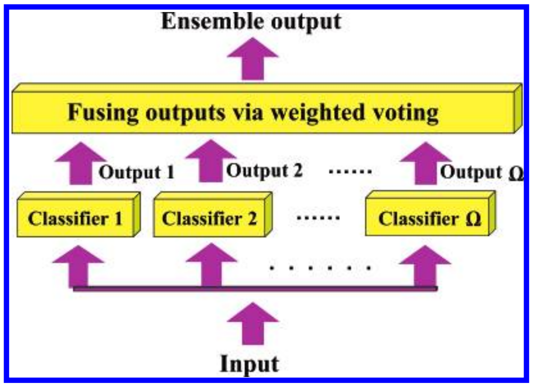

Figure 2. Flowchart to show how the ensemble classifiers $\mathbb{C}^{GO}$ (eq 10) and $\mathbb{C}^{Pse}$ (eq 16) are formed by fusing Ω individual classifiers, where Ω = 18 and 360 for the cases of $\mathbb{C}^{GO}$ and $\mathbb{C}^{Pse}$, respectively.

Therefore, the OET-KNN classifier can be formulated as

$$\text{OET-KNN} = \begin{cases} \mathbb{C}(K), & \text{when protein samples are represented by eq 4} \\ \mathbb{C}(K,\, 20+2\lambda), & \text{when protein samples are represented by eq 6} \end{cases} \tag{7}$$

where the classifier $\mathbb{C}$(K) is the function of K, while the classifier $\mathbb{C}(K, 20 + 2\lambda)$ the function of both K and λ.

During the course of prediction, the following self-consistency principle should be strictly followed. If a query protein could be defined in the 9567-D GO_compress space (eq 4), then the prediction should be carried out based on those proteins in the training dataset that could be defined in the same 9567-D space. If the query protein in the 9567-D GO_compress space was a naught vector and hence must be defined instead in the $(20 + 2\lambda)$-D space (eq 6), then the prediction should be conducted according to the principle that all the proteins in the training dataset be defined in the same $(20 + 2\lambda)$-D space as well. Accordingly, the current hybridization predictor actually consists of two subpredictors: **(1)** the $\mathbb{C}(K)$ predictor that operates in the 9567-D GO_compress space, and **(2)** the $\mathbb{C}(K, 20 + 2\lambda)$ predictor that operates in the $(20 + 2\lambda)$-D amphiphilic PseAA space. For different learning datasets, the selection of K and λ would be different in order to get the optimal result. It is time-consuming and tedious to test the results by using different numbers of K and λ one by one. To solve such a problem, two different ensemble classifiers were introduced for the $\mathbb{C}(K)$ and $\mathbb{C}(K,20 + 2\lambda)$ predictors, as formulated below.

For the $\mathbb{C}(K)$ predictor, the ensemble classifier was formed by fusing many single classifiers each having a different specified value for K, as described below.

Preliminary tests indicated that the success rates obtained by the $\mathbb{C}(K)$ predictor were lower when $K = 1, 2$, or > 20, and hence these numbers can be excluded during the fusion process. Suppose

$$K \in \{3, 4, \cdots, 20\} \tag{8}$$

where ∈ is a symbol in the set theory meaning "member of", then we have a set of corresponding classifiers as formulated by

$$\mathbb{C}(K),\ (K = 3, 4, \cdots, 20) \tag{9}$$

where $\mathbb{C}(3)$ is the OET-KNN classifier trained with 3 nearest neighbors in the 9567-D GO_compress space, $\mathbb{C}(4)$ is the one trained with 4 nearest neighbors, and so forth. The ensemble classifier formed by fusing such a set of individual classifiers is formulated by

$$\mathbb{C}^{GO} = \mathbb{C}(3) \forall \mathbb{C}(4) \forall \cdots \forall \mathbb{C}(20) \tag{10}$$

where the symbol ∀ denotes the fusion operator, and $\mathbb{C}^{GO}$ the ensemble classifier formed by fusing $\mathbb{C}(3)$, $\mathbb{C}(4)$, ..., and $\mathbb{C}(20)$ according to the flowchart of Figure 2.

The process of how the ensemble classifier $\mathbb{C}^{GO}$ works is as follows. Suppose the predicted classification results for the query protein **P** by the $(20 - 3 + 1) = 18$ individual classifiers in eq 9 are

$$C_K \in \{S_1, S_2, \cdots, S_{16}\},\ (K = 3, 4, \cdots, 20) \tag{11}$$

where $S_1, S_2, \cdots, S_{16}$ represent the 16 subsets defined by the 16 subcellular locations studied here (Figure 1), and the voting score for the protein **P** belonging to the jth subset is defined by

$$Y_j^{GO} = \sum_{K=3}^{20} w_K \Delta(C_K, S_j),\ (j = 1, 2, \cdots, 16) \tag{12}$$

where w_K is the weight and was set at 1 for simplicity, and the delta function in eq 12 is given by

$$\Delta(C_K, S_j) = \begin{cases} 1, & \text{if } C_K \in S_j \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

thus the query protein **P** is predicted belonging to the subset (subcellular location) with which its score of eq 12 is the highest.

The above fusion process can be straightforwardly extended to the case of the $\mathbb{C}(K,20 + 2\lambda)$ predictor as well. However, since it has two parameters, the ensemble classifier for $\mathbb{C}(K,20 + 2\lambda)$ should be formed by fusing many individual classifiers with different K or λ, respectively; i.e., the fusion process should involve a two-dimensional process, as formulated below.

For the similar reason as mentioned above regarding eq 8, let us suppose

$$K \in \{3, 4, \cdots, 20\};\ (20 + 2\lambda) \in \{22, 24, \cdots, 58, 60\} \tag{14}$$

then we have a set of 18 × 20 = 360 individual classifiers as formulated by

$$\mathbb{C}(K,\, 20 + 2\lambda),\ (K = 3, 4, \cdots, 20;\ \lambda = 1, 2, \cdots, 20) \tag{15}$$

where $\mathbb{C}(3, 22)$ is the OET-KNN classifier trained with 3 nearest neighbor in the 22-D PseAA space, $\mathbb{C}(4, 24)$ is the one trained with 4 nearest neighbors in the 24-D PseAA space, and so forth. The ensemble classifier formed by fusing such 360 individual classifiers is formulated by

$$\mathbb{C}^{Pse} = \mathbb{C}(3,22) \forall \mathbb{C}(3,24) \forall \cdots \forall \mathbb{C}(20,58) \forall \mathbb{C}(20,60) \tag{16}$$

where the fusion operator ∀ has the same meaning as that of eq 10, and the fusion flowchart is shown in Figure 2.

The detailed process of how the ensemble classifier $\mathbb{C}^{Pse}$ works is as follows. Suppose the predicted classification results for the query protein **P** by the 360 individual classifiers in eq 15 are

$$C_{K,20+2\lambda} \in \{S_1, S_2, \cdots, S_{16}\},\ (K = 3, 4, \cdots, 20;\ \lambda = 1, 2, \cdots, 20) \tag{17}$$

Thus, the voting score for the protein **P** belonging to the *k*th subset is defined by

$$Y_j^{\mathrm{Pse}} = \sum_{K=3}^{20}\sum_{\lambda=1}^{20} w_{\mathrm{K},20+2\lambda}\,\Delta(C_{K,20+2\lambda},S_j),\ (j = 1, 2, \cdots, 16) \quad (18)$$

where $w_{K,20+2\lambda}$ is the weight and was set at 1 for simplicity, the delta function in eq 18 is given by

$$\Delta(C_{K,20+2\lambda},S_j) = \begin{cases} 1, & \text{if } C_{K,20+2\lambda} \in S_j \\ 0, & \text{otherwise} \end{cases} \quad (19)$$

thus the query protein **P** is predicted belonging to the subset (subcellular location) with which its score of eq 18 is the highest.

IV. Results and Discussion

For the proteins listed in the Online Supporting Information A and B, we obtained the following results according to Steps 1−2 of Methods: (**1**) of the 2423 proteins in the learning dataset, 2380 got hits in the GO_compress database, and hence were defined in the 9567-D GO_compress space (eqs 4−5), and the remainder defined in the $(20 + 2\lambda)$-DPseAA space (eq 6); (**2**) of the 1727 proteins in the testing dataset, 1697 got hits and were defined in the 9567-D GO_compress space, and the remainder defined in the $(20 + 2\lambda)$-D PseAA space. Therefore, if the protein samples were represented only based on the GO_compress database, $2423 - 2380 = 43$ proteins in the training dataset and $1727 - 1697 = 30$ proteins in the independent dataset would have no definition, immediately leading to a failure of identifying their subcellular locations. Although most of proteins studied could be defined in the 9567-D GO_compress space, it would be better to hybridize with the PseAA approach, by which not only a protein can always be defined but also a considerable amount of sequence-order information can be incorporated. Thus, the prediction process was operated according to the following procedures: if a query protein was defined in the 9567-D GO_compress space, then the ensemble classifier $\mathbb{C}^{\mathrm{GO}}$ (eq 10) was used to predict its subcellular location; otherwise, the ensemble classifier $\mathbb{C}^{\mathrm{Pse}}$ (eq 16) was used to predict its subcellular location. The prediction quality was examined by two standard test methods in statistics: the jackknife test and the independent dataset test.

1. Jack-Knife Test. In the jackknife test, each protein in the learning dataset was singled out in turn as a "test protein" and all the rule parameters were calculated from the remaining $(N - 1)$ proteins. In other words, the subcellular location of each protein was predicted by the rules derived using all the other proteins except the one that was being predicted. During the jackknifing process, both the learning and testing dataset were actually open, and a protein was in turn moving from one to the other. The jack-knife test result on the dataset of Online Supporting Information A obtained by the fusion classifier is given in Table 4, where for facilitating comparison, the corresponding results by various other methods are also listed. It can be seen from Table 4 that the overall jack-knife success rate by the current approach is 46.5−55.8% higher than those by the other existing approaches.

2. Independent Dataset Test. In the independent dataset test, the rule parameters were derived from the proteins only in the learning dataset S^{L} (Online Supporting Information A), and the prediction was made for proteins in an independent dataset S^{T} (Online Supporting Information B). The predicted results thus obtained are also given in Table 4, from which we can see that the current fusion classifier outperformed the other methods by 49.2−63.3%.

It should be pointed out that the independent dataset test performed here was just for a demonstration of practical application. Because the selection of independent dataset often bears some sort of arbitrariness,[35] the jack-knife test is deemed more objective than the independent dataset test, and have been used by more and more investigators[12,15,36−38] in examining the power of various prediction methods. Therefore, the power of a predictor should be measured by the success rate of jack-knife test.

An interesting question might be raised: Why the Ploc method[16] and the HSLPred[17] method as originally reported could yield an overall success rate higher than 70% and 80%, but here only within the range of 32% and 36%? To address this question, let us consider the following facts. **(1)** The benchmark datasets originally used in those methods contained many homologous sequences in a same subcellular location. For example, the dataset used in ref 17 contained homologous proteins with up to 90% sequence identity; and the dataset in ref 16 up to 80% sequence identity. In contrast to these, none of protein in the current dataset has ≥25% sequence identity to any other in a same subcellular location. When tests were made by those methods on such a stringent dataset, the success rates would of course decrease significantly because, as mentioned above, the more stringent the benchmark dataset in excluding homologous sequences, the harder it becomes to get a higher success rate for cross validation test. **(2)** In addition to using a high homologous benchmark dataset, the original high success rate reported in ref 17 was derived for a case in which prediction was made only among 4 subcellular locations. Now the prediction must be made among 16 locations, which

Table 4. Overall Success Rates for the 16 Subcellular Locations (Figure 1) of Proteins by Different Classifiers and Test Methods

		test method	
classifier	input form	jack-knife[a]	independent dataset[b]
least Euclidean distance [7]	amino acid composition	(624/2423) = 25.8%	(353/1727) = 20.4%
ProtLock [8]	amino acid composition	(696/2423) = 28.7%	(437/1727) = 25.3%
SVM (Ploc)[16]	amino acid composition and amino acid pairs[16]	(851/2423) = 35.1%	(566/1727) = 32.8%
SVM (HSLPred)[17]	amino acid composition and dipeptide composition[17]	(802/2423) = 33.1%	(595/1727) = 34.5%
fusion classifier	hybridization of GO (eq 1) and amphiphilic PseAA(eq 3)	(1976/2423) = **81.6%**	(1445/1727) = **83.7%**

[a] Jack-knife cross-validation test was performed for the 2423 proteins in the Online Supporting Information A, where none of the proteins has ≥25% sequence identity to any other in the same subcellular location. [b] Prediction was performed for the 1727 independent proteins in the Online Supporting Information B; none of proteins in the Online Supporting Information A and B has ≥25% sequence identity to any others in the same subcellular location.

will of course bring in additional difficulty since, as mentioned above, the more subcellular locations the classification involves, the harder it is in getting a higher success prediction rate. **(3)** The success rates originally reported in refs 16 and 17 were obtained by the sub-sampling cross-validation test. When tested by the jack-knife cross-validation, the corresponding rates would naturally further diminish because, as mentioned above, the jackknife cross-validation is more strictly.

V. Conclusion

Prediction of protein subcellular location is a very challenging and complicated problem. The more the subcellular locations covered, the lower the odds are in getting a correct prediction. Also, the more strictly the working dataset in excluding homologous sequences, the harder it becomes to get a higher success rate for cross validation test. That is why **(1)** most of the existing prediction methods only cover 2 to 5 subcellular locations, and **(2)** the benchmark datasets used by the existing methods contain homologous proteins with sequence identity up to 80%, 90%, or even higher, leading to an overestimate of success rates.

In this study, we constructed a broadly extensive and meanwhile much more stringent dataset, which covers 16 subcellular locations and in which none of proteins has ≥25% sequence identity to any others in a same subcellular location. To improve the prediction quality, we adopted the strategy of **(1)** representing protein samples by hybridizing GO (eq 4) and PseAA (eq 6), and **(2)** introducing the ensemble classifier that were formed by fusing many basic individual classifiers operated by the engine of the OET-KNN rule.

Using GO to represent the sample of a protein could effectively grasp its core features that might be correlated with the subcellular location. However, how to use the information in the GO database is by no means a trivial work. The reasons are as follows. (**1**) For those proteins with “subcellular location unknown” annotation in Swiss-Prot database, most (more than 99%) of their corresponding GO numbers in GO database are also annotated with “cellular component unknown”. (**2**) Even for those proteins whose subcellular locations are clearly annotated in Swiss-Prot database, their corresponding GO numbers in GO database may be in some cases still annotated with “cellular component unknown”. (**3**) The percentage (44.9%) of proteins with GO annotations to indicate their subcellular components is even less than the percentage (52.4%) of proteins with subcellular location annotation in the Swiss-Prot database. Therefore, the high success rate obtained by the current fusion classifier was by no means due to a trivial utilization of subcellular location annotation from one database (GO) for the other (Swiss-Prot).

There is a huge number of proteins with given accession numbers and the corresponding GO numbers (Table 2), but their subcellular locations are still unknown. Therefore, in addition to the limited number of known subcellular location annotations, the information and clues useful for predicting the subcellular location of unknown query proteins are actually buried into a series of tedious GO numbers, just like they are buried into a pile of complicated sequences. To dig out the knowledge about their locations, a sophisticated operation engine is needed. The current fusion classifier is one of theses kinds, and has proved to be very powerful, as reflected by the overwhelmingly high success rates compared with those obtained by the other existing predictors.

Appendix A. Amphiphilic Pseudo Amino Acid Composition

Given a protein **P** with L amino acid residues

$$R_1R_2R_3R_4R_5R_6R_7 \cdots R_L \qquad \text{(A1)}$$

where R_1 represents the residue at the sequence position 1, R_2 at position 2, and so forth, its amino acid composition is given by[26]

$$\mathbf{P}_{AA} = [f_1 \;\; f_2 \;\; \cdots \;\; f_{20}]^{\mathbf{T}} \qquad \text{(A2)}$$

where f_1 is the normalized occurrence frequency of amino acid A in the protein, f_2 that of amino acid C, and so forth. Here, without loss of generality, the single codes of the 20 native amino acids are used according to their alphabetical order. As we can see from eq A1, if a protein is represented by such a set of discrete numbers, all its sequence information would be lost. To keep the representation of a protein sample with a discrete mode but without completely losing its sequence-order information, we can define a pseudo amino acid composition by merging a series of sequence-order-correlated factors into the conventional amino acid composition. As is well-known, the hydrophobicity and hydrophilicity play a very important role to the folding of a protein as well as its microenvironment and interior packing (see, e.g., refs 39−42). For instance, many helices in proteins are amphiphilic that are formed by the hydrophobic and hydrophilic amino acids according to a special order along the helix chain, as illustrated by the “wenxiang” diagram.[43] Therefore, these two indices may be one of the optimal choices to reflect the sequence order effects. In view of this, the sequence-order effects can be indirectly and partially, but quite effectively, reflected through the following equations (see Figure A1):

$$\begin{cases} \tau_1 = \dfrac{1}{L-1}\displaystyle\sum_{i=1}^{L-1} H_{i,i+1}^{1} \\ \tau_2 = \dfrac{1}{L-1}\displaystyle\sum_{i=1}^{L-1} H_{i,i+1}^{2} \\ \tau_3 = \dfrac{1}{L-2}\displaystyle\sum_{i=1}^{L-2} H_{i,i+2}^{1} \\ \tau_4 = \dfrac{1}{L-2}\displaystyle\sum_{i=1}^{L-2} H_{i,i+2}^{2} \qquad (\lambda < L) \\ \cdots\cdots\cdots\cdots \\ \tau_{2\lambda} - 1 = \dfrac{1}{L-\lambda}\displaystyle\sum_{i=1}^{L-\lambda} H_{i,i+\lambda}^{1} \\ \tau_{2\lambda} = \dfrac{1}{L-\lambda}\displaystyle\sum_{i=1}^{L-\lambda} H_{i,i+\lambda}^{2} \end{cases} \qquad \text{(A3)}$$

where $H_{i,j}{}^1$ and $H_{i,j}{}^2$ are the hydrophobicity and hydrophilicity correlation functions given by

$$\begin{cases} H_{i,j}{}^{1} = w[h^1(R_i)h^1(R_j)] \\ H_{i,j}{}^{2} = w[h^2(R_i)h^2(R_j)] \end{cases} \qquad \text{(A4)}$$

where $h^1(R_i)$ and $h^2(R_i)$ are, respectively, the hydrophobicity and hydrophilicity values for the ith ($i = 1, 2, ..., L$) amino acid in eq A1, and w is the weight factor. In the current study, we chose $w = 0.5$ to make the data within the range easier to be handled (w can be of course assigned with other values, but this would not have a big impact to the final results). In eq A3

τ_1 and τ_2 are called the 1st-rank correlation factors that reflect the sequence-order correlation between all the most contiguous residues along a protein chain through hydrophobicity and hydrophilicity, respectively [Figure A1(a1),(a2)], τ_3 and τ_4 are the corresponding second-rank correlation factors that reflect the sequence-order correlation between all the 2nd most contiguous residues [Figure A1(b1),(b2)], and so forth. Note that before substituting the values of hydrophobicity and hydrophilicity into eq A4, they are standardized:

$$\begin{cases} h_1(\mathrm{R}_i) = \dfrac{h_1{}^0(\mathrm{R}_i) - <h_1{}^0>}{\mathrm{SD}(h_1{}^0)} \\ h_2(\mathrm{R}_i) = \dfrac{h_2{}^0(\mathrm{R}_i) - <h_2{}^0>}{\mathrm{SD}(h_2{}^0)} \end{cases} \tag{A5}$$

where the symbols $h_1{}^0(\mathrm{R}_i)$ and $h_2{}^0(\mathrm{R}_i)$ represent the original hydrophobicity value[44] and hydrophilicity value[45] for amino acid R_i, respectively (Table A1); $<h_1{}^0>$ and $<h_2{}^0>$ their means over 20 native amino acids; $\mathrm{SD}(h_1{}^0)$ and $\mathrm{SD}(h_2{}^0)$ their standard deviations. The converted hydrophobicity and hydrophilicity values obtained by eq A5 will have a zero mean value over the 20 native amino acids, and will remain unchanged if going through the same conversion procedure again. After merging the sequence-order-correlated factors from eq A3 into the classical 20D (dimensional) amino acid composition (eq A2), we obtain a pseudo amino acid composition with $20 + 2\lambda$ components. In other words, the representation for the protein sequence of eq A1 is now formulated as

$$\mathbf{P}_{\mathrm{PseAA}} = [f_1 \ \ f_2 \ \cdots \ f_{20} \ \ \tau_1 \ \ \tau_2 \ \cdots \ \tau_{2\lambda}]^{\mathbf{T}} \tag{A6}$$

which can be easily converted to eq 6 by performing a normalization procedure according to the following equation:

$$F_u = \begin{cases} \dfrac{f_u}{\sum_{i=1}^{20} f_i + \sum_{j=1}^{2\lambda} \tau_j}, & (1 \le u \le 20) \\ \dfrac{\tau_u}{\sum_{i=1}^{20} f_i + \sum_{j=1}^{2\lambda} \tau_j}, & (20 + 1 \le u \le 20 + 2\lambda) \end{cases} \tag{A7}$$

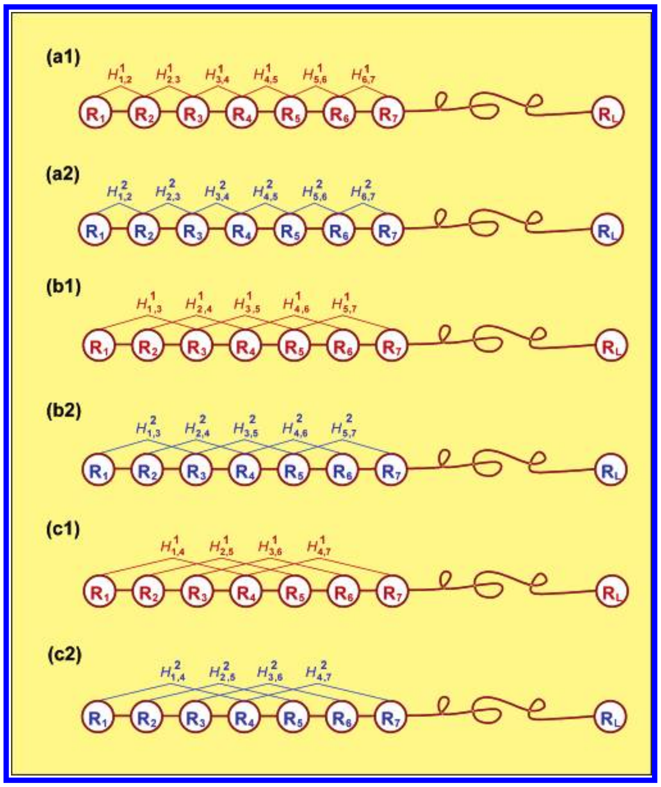
(a1)
$H^1_{1,2}$ $H^1_{2,3}$ $H^1_{3,4}$ $H^1_{4,5}$ $H^1_{5,6}$ $H^1_{6,7}$
(a2)
$H^2_{1,2}$ $H^2_{2,3}$ $H^2_{3,4}$ $H^2_{4,5}$ $H^2_{5,6}$ $H^2_{6,7}$
(b1)
$H^1_{1,3}$ $H^1_{2,4}$ $H^1_{3,5}$ $H^1_{4,6}$ $H^1_{5,7}$
(b2)
$H^2_{1,3}$ $H^2_{2,4}$ $H^2_{3,5}$ $H^2_{4,6}$ $H^2_{5,7}$
(c1)
$H^1_{1,4}$ $H^1_{2,5}$ $H^1_{3,6}$ $H^1_{4,7}$
(c2)
$H^2_{1,4}$ $H^2_{2,5}$ $H^2_{3,6}$ $H^2_{4,7}$
R_1 R_2 R_3 R_4 R_5 R_6 R_7 R_L

Figure A1. Schematic drawing to show the amphiphilic correlation along a protein chain, where the values of $H^1_{i,j}$ and $H^2_{i,j}$ are given by eqs A4–A5 and Table A1. The correlation via hydrophobicity is shown in red, while the correlation via hydrophilicity in blue. Panel (a1/a2) reflects the coupling mode between all the most contiguous residues, panel (b1/b2) that between all the 2nd most contiguous residues, and panel (c1/c2) that between all the 3rd most contiguous residues.

Table A1. Amino Acid Parameters Used for Deriving the Amphiphilic Pseudo Amino Acid Components (cf. eq A5)

code	hydrophobicity[a] $h_1{}^0$	hydrophilicity[b] $h_2{}^0$
A	0.62	−0.5
C	0.29	−1.0
D	−0.90	3.0
E	−0.74	3.0
F	1.19	−2.5
G	0.48	0.0
H	−0.40	−0.5
I	1.38	−1.8
K	−1.50	3.0
L	1.06	−1.8
M	0.64	−1.3
N	−0.78	2.0
P	0.12	0.0
Q	−0.85	0.2
R	−2.53	3.0
S	−0.18	0.3
T	−0.05	−0.4
V	1.08	−1.5
W	0.81	−3.4
Y	0.26	−2.3

[a] The hydrophobicity values were taken from ref 44. [b] The hydrophilicity values were taken from ref 45.

For reader's convenience, the $20 + 2\lambda = 60$ components of $\mathbf{P}_{\mathrm{PseAA}}$ in eq A6 for the proteins in the training and testing datasets studied here are given in the Online Supporting Information C and D, respectively, from which the user can easily generate the normalized pseudo amino acid composition $\mathbf{P}$ of eq 6 with any dimension of $(20 + 2\lambda) \le 60$ through eq A7. For instance, to generate the $\mathbf{P}$ with $(20 + 2\lambda) = 22$, just read the first 22 data for each protein in the Online Supporting Information followed by substituting them into eq A7; to generate the $\mathbf{P}$ with $(20 + 2\lambda) = 24$, just read the first 24 data followed by the same procedure; and so forth. Actually, suppose the length of the shortest protein sequence studied here is $L_{\min}$, by following the above procedures one can always generate the $\mathbf{P}$ with a dimension of $(20 + 2\lambda) \le [20 + 2(L_{\min} - 1)]$ (see eq A3 and Figure A1). It should be pointed out that, according to the definition of the classical amino acid composition, all its components must be ≥ 0; it is not always true, however, for the pseudo amino acid composition: the components derived from the sequence correlation factors (cf. eq A3) may also be <0.

Appendix B

Optimized Evidence-Theoretic K-Nearest Neighbors (OET-KNN) Classifier. For reader's convenience, a brief introduction of the OET-KNN classifier is given below. For further explanation, refer to refs 30, 33, and 34. Let us consider a problem of classifying N entities (proteins) ($\in\mathbb{S}$) into 16 subsets (subcellular locations), which can be formulated as

$$\mathbb{S} = S_1 \cup S_2 \cup \cdots \cup S_\mu \cup \cdots \cup S_{16} \tag{B1}$$

The available information is assumed to consist in a learning (training) dataset

$$\mathbb{S}^{\mathrm{L}} = \{(\mathbf{P}_1,C_1), (\mathbf{P}_2,C_2) \cdots , (\mathbf{P}_N,C_N)\} \tag{B2}$$

where the N entities $\mathbf{P}_i$ ($i = 1, 2, ..., N$) and their corresponding attribute (subcellular location) labels C_i ($i = 1, 2, ..., N$) are defined in $\mathbb{S}$ of eq B1. According to the KNN (K-Nearest Neighbors) rule,[31] an unclassified query entity P is assigned to the class represented by a majority of its K nearest neighbors of P.

The OET-KNN classifier was developed from the ET-KNN (Evidence Theoretic K-nearest Neighbors) classifier, a pattern classification method established on the basis of the Dempster-Shafer theory of belief functions.[30] During the process of classification, each neighbor of a pattern to be classified is considered as an item of evidence supporting certain hypotheses concerning the class membership of that pattern. On the basis of this evidence, basic belief masses are assigned to each subset concerned. Such masses are obtained for each of the K nearest neighbors of the pattern under consideration and aggregated using the Dempster's rule of combination.[33] A decision is made by assigning a pattern to the class with the maximum credibility.

In the current case, the pattern to be classified is the subcellular location of the query protein $\mathbf{P}$. Suppose $S_{\mathrm{K}}^{\mathbf{P}}$ is the set of K-nearest neighbors to $\mathbf{P}$ in the learning dataset $\mathbb{S}^{\mathrm{L}}$ of eq B2. Thus, for any $\mathbf{P}_i \in S_{\mathrm{K}}^{\mathbf{P}}$, the knowledge that $\mathbf{P}_i$ belongs to class $S_\mu \subset \mathbb{S}$ can be considered as a piece of evidence that increases our belief that $\mathbf{P}$ also belongs to S_μ. According to the basic belief assignment mapping theory,[33] this item of evidence can be formulated by

$$\mathscr{M}(\mathbf{P}_i,S_\mu) = \alpha_0 \exp[-\gamma_\mu^2 D^2(\mathbf{P}_i,\mathbf{P})],\ (\mu = 1, 2, \cdots, 16\ ) \tag{B3}$$

where $\alpha_0 = 0.95$ is a fixed parameter, γ_μ is a parameter associated with class S_μ, and $D^2(\mathbf{P}_i,\mathbf{P})$ is the square distance between the query protein $\mathbf{P}$ and the ith protein $\mathbf{P}_i$ in the learning dataset $\mathbb{S}^{\mathrm{L}}$. Various distance metrics, such as Hamming distance, Euclidean distance, and Mahalanobis distance,[46–48] could be used to measure the nearness between $\mathbf{P}$ and $\mathbf{P}_i$. In this study, if the protein samples were expressed in the PseAA space (eq 3), the Euclidean distance would be used to define $D(\mathbf{P}_i,\mathbf{P})$; if, however, they were expressed in the 9567-D GO_compress space (eq 1), the dissimilarity of $1 - \cos(\mathbf{P},\mathbf{P}_i)$, i.e.

$$D(\mathrm{P},\mathrm{P}_i) = 1 - \frac{\mathrm{P} \cdot \mathrm{P}_i}{\|\mathrm{P}\|\,\|\mathrm{P}_i\|} \tag{B4}$$

would be adopted.

In the ET-KNN rule, it was not addressed how to optimally select the parameters in eq B3. In 1998, an optimization procedure to determine the optimal or near-optimal parameter values for γ_μ ($\mu = 1,2, \cdots, 16$) was proposed from the learning dataset $\mathbb{S}^{\mathrm{L}}$ by minimizing an error function.[34] After such an optimization treatment, the ET-KNN would become the OET-KNN classifier, leading to a substantial improvement in classification accuracy.

The belief function of $\mathbf{P}$ belonging to class S_μ is a combination of its K-Nearest Neighbors, and can be formulated as

$$\mathscr{M}(\mathbf{P},S_\mu) = \oplus_{i=1}^{\mathrm{K}}\ \mathscr{M}(\mathbf{P}_i,S_\mu) \tag{B5}$$

where the symbol $\oplus_{i=1}^{\mathrm{K}}$ represents the orthogonal sum from $i = 1$ to K. According to Dempster's rule,[33] the belief function of eq B5 can be expressed as

$$\mathscr{M}(\mathbf{P},S_\mu) = \frac{\sum_{S_{\mathrm{K},i}^{\mathbf{P}} \subseteq S_{\mathrm{K}}^{\mathbf{P}},\ S_{\mathrm{K},j}^{\mathbf{P}} \subseteq S_{\mathrm{K}}^{\mathbf{P}},\ S_{\mathrm{K}}^{\mathbf{P}} \cap S_{\mathrm{K},j}^{\mathbf{P}} = S\mu}\ \mathscr{M}(\mathbf{P},S_{\mathrm{K},i}^{\mathbf{P}})\mathscr{M}(\mathbf{P},S_{\mathrm{K},j}^{\mathbf{P}})}{1 - \sum_{S_{\mathrm{K},i}^{\mathbf{P}} \subseteq S_{\mathrm{K}}^{\mathbf{P}},\ S_{K,j}^{\mathbf{P}} \subseteq S_{\mathrm{K}}^{\mathbf{P}},\ S_{\mathrm{K},i}^{\mathbf{P}} \cap S_{\mathrm{K},j}^{\mathbf{P}} = \varnothing}\ \mathscr{M}(\mathbf{P},S_{\mathrm{K},i}^{\mathbf{P}})\mathscr{M}(\mathbf{P},S_{\mathrm{K},j}^{\mathbf{P}})} \tag{B6}$$

where $S_{\mathrm{K},i}^{\mathbf{P}}$ is the ith possible subset of $S_{\mathrm{K}}^{\mathbf{P}}$, while $\subseteq$ and $\cap$ are the symbols in the set theory, representing "contained in" and "intersection", respectively.

A decision is made by assigning the query protein $\mathbf{P}$ to the class μ with which the belief or credibility function of eq B6 has the maximum value; i.e.,

$$\mu = \mathbf{Arg\ Max}_i\{\mathscr{M}(\mathbf{P},S_i)\},\ (i = 1, 2, \cdots, 16) \tag{B7}$$

where the operator **Arg Max**$_i$ means taking the subscript with which $\mathscr{M}(\mathbf{P},S_i)$ is the maximum. If there was a tie among two or more subsets, then the query protein would be randomly assigned to one of their corresponding subcellular locations although this kind of tie case rarely happened.

Acknowledgment. We would like to express our gratitude to the two anonymous Reviewers, whose suggestions are very helpful in strengthening the presentation of this paper.

Supporting Information Available: The accession numbers and sequences for the 2423 proteins in the learning dataset (Online Supporting Information A); those for the 1727 proteins in the independent dataset (Online Supporting Information B); the 60 PseAA components for the 2423 proteins in the learning dataset (Online Supporting Information C); and those for the 1727 proteins in the independent dataset (Online Supporting Information D). These materials are available free of charge via the Internet at http://pubs.acs.org.

References

(1) Alberts, B.; Bray, D.; Lewis, J.; Raff, M.; Roberts, K.; Watson, J. D. *Mol. Biol. Cell, Chapter 1*, 3rd ed.; Garland Publishing: New York & London, 1994.

(2) Lodish, H.; Baltimore, D.; Berk, A.; Zipursky, S. L.; Matsudaira, P.; Darnell, J. *Mol. Cell. Biol., Chapter 3*, 3rd ed.; Scientific American Books: New York, 1995.

(3) Bairoch, A.; Apweiler, R. The SWISS–PROT protein sequence data bank and its supplement TrEMBL. *Nucleic Acids Res.* **2000**, *25*, 31–36.

(4) Chou, K. C. Review: Structural bioinformatics and its impact to biomedical science. *Curr. Med. Chem.* **2004**, *11*, 2105–2134.

(5) Nakai, K.; Horton, P. PSORT: A program for detecting sorting signals in proteins and predicting their subcellular localization. *Trends Biochem. Sci.* **1999**, *24*, 34–36.

(6) Nakai, K. Protein sorting signals and prediction of subcellular localization. *Adv. Protein Chem.* **2000**, *54*, 277–344.

(7) Nakashima, H.; Nishikawa, K. Discrimination of intracellular and extracellular proteins using amino acid composition and residue-pair frequencies. *J. Mol. Biol.* **1994**, *238*, 54–61.

(8) Cedano, J.; Aloy, P.; Pérez-Pons, J. A.; Querol, E. Relation between amino acid composition and cellular location of proteins. *J. Mol. Biol.* **1997**, *266*, 594–600.

(9) Reinhardt, A.; Hubbard, T. Using neural networks for prediction of the subcellular location of proteins. *Nucleic Acids Res.* **1998**, *26*, 2230−2236.

(10) Chou, K. C.; Elrod, D. W. Protein subcellular location prediction. *Protein Eng.* **1999**, *12*, 107−118.

(11) Chou, K. C. Prediction of protein cellular attributes using pseudo amino acid composition. *Proteins: Struct. Funct. Genet. (Erratum: ibid., 2001, Vol. 44, 60)* **2001**, *43*, 246−255.

(12) Feng, Z. P. Prediction of the subcellular location of prokaryotic proteins based on a new representation of the amino acid composition. *Biopolymers* **2001**, *58*, 491−499.

(13) Feng, Z. P. An overview on predicting the subcellular location of a protein. *In Silico Biol.* **2002**, *2*, 291−303.

(14) Chou, K. C.; Cai, Y. D. Using functional domain composition and support vector machines for prediction of protein subcellular location. *J. Biol. Chem.* **2002**, *277*, 45765−45769.

(15) Zhou, G. P.; Doctor, K. Subcellular location prediction of apoptosis proteins. *Proteins: Struct. Funct. Genet.* **2003**, *50*, 44−48.

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

(17) Garg, A.; Bhasin, M.; Raghava, G. P. Support vector machine-based method for subcellular localization of human proteins using amino acid compositions, their order, and similarity search. *J. Biol. Chem.* **2005**, *280*, 14427−14432.

(18) Matsuda, S.; Vert, J. P.; Saigo, H.; Ueda, N.; Toh, H.; Akutsu, T. A novel representation of protein sequences for prediction of subcellular location using support vector machines. *Protein Sci.* **2005**, *14*, 2804−2813.

(19) Gardy, J. L.; Spencer, C.; Wang, K.; Ester, M.; Tusnady, G. E.; Simon, I.; Hua, S.; deFays, K.; Lambert, C.; Nakai, K.; Brinkman, F. S. PSORT-B: Improving protein subcellular localization prediction for Gram-negative bacteria. *Nucleic Acids Res.* **2003**, *31*, 3613−3617.

(20) Ashburner, M.; Ball, C. A.; Blake, J. A.; Botstein, D.; Butler, H.; Cherry, J. M.; Davis, A. P.; Dolinski, K.; Dwight, S. S.; Eppig, J. T.; Harris, M. A.; Hill, D. P.; Issel-Tarver, L.; Kasarskis, A.; Lewis, S.; Matese, J. C.; Richardson, J. E.; Ringwald, M.; Rubin, G. M.; Sherlock, G. Gene ontology: tool for the unification of biology. *Nat. Genet.* **2000**, *25*, 25−29.

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

(22) Wang, G. L.; Dunbrack Jr., R. L. PISCES: A protein sequence culling server. *Bioinformatics* **2003**, *19*, 1589−1591.

(23) Camon, E.; Magrane, M.; Barrell, D.; Lee, V.; Dimmer, E.; Maslen, J.; Binns, D.; Harte, N.; Lopez, R.; Apweiler, R. The Gene Ontology Annotation (GOA) Database: sharing knowledge in Uniprot with Gene Ontology. *Nucleic Acids Res.* **2004**, *32*, D262−266.

(24) Lee, V.; Camon, E.; Dimmer, E.; Barrell, D.; Apweiler, R. Who tangos with GOA?-Use of Gene Ontology Annotation (GOA) for biological interpretation of '-omics' data and for validation of automatic annotation. tools *In Silico Biol.* **2005**, *5*, 5−8.

(25) Apweiler, R.; Bairoch, A.; Wu, C. H.; Barker, W. C.; Boeckmann, B.; Ferro, S.; Gasteiger, E.; Huang, H.; Lopez, R.; Magrane, M.; Martin, M. J.; Natale, D. A.; O'Donovan, C.; Redaschi, N.; Yeh, L. S. UniProt: the Universal Protein knowledgebase *Nucleic Acids Res.* **2004**, *32*, D115−119.

(26) Chou, K. C.; Zhang, C. T. Predicting protein folding types by distance functions that make allowances for amino acid interactions *J. Biol. Chem.* **1994**, *269*, 22014−22020.

(27) Vapnik, V. N. *The Nature of Statistical Learning Theory*; Springer-Verlag: New York; 1995.

(28) Cortes, C.; Vapnik, V. Support vector networks. Machine Learning *Machine Learning* **1995**, *20*, 273−293.

(29) Friedman, J. H.; Baskett, F.; Shustek, L. J. An algorithm for finding nearest neighbors *IEEE Trans. Inform. Theory* **1975**, *C-24*, 1000−1006.

(30) Denoeux, T. A k-nearest neighbor classification rule based on Dempster-Shafer theory. *IEEE Trans. Syst. Man. Cybernet.* **1995**, *25*, 804−813.

(31) Cover, T. M.; Hart, P. E. Nearest neighbour pattern classification *IEEE Trans. Inform. Theory* **1967**, *IT-13*, 21−27.

(32) Keller, J. M.; Gray, M. R.; Givens, J. A. A fuzzy k-nearest neighbours algorithm. *IEEE Trans. Syst. Man. Cybernet.* **1985**, *15*, 580−585.

(33) Shafer, G. *A Mathematical Theory of Evidence*; Princeton University Press: Princeton, New Jersey, 1976.

(34) Zouhal, L. M.; Denoeux, T. An evidence-theoretic K−NN rule with parameter optimization. *IEEE Trans. Syst. Man. Cybernet.* **1998**, *28*, 263−271.

(35) Chou, K. C.; Zhang, C. T. Review: Prediction of protein structural classes *Crit. Rev. Biochem. Mol. Biol.* **1995**, *30*, 275−349.

(36) Zhou, G. P.; Assa-Munt, N. Some insights into protein structural class prediction. *Proteins: Struct. Funct. Genet.* **2001**, *44*, 57−59.

(37) Luo, R. Y.; Feng, Z. P.; Liu, J. K. Prediction of protein strctural class by amino acid and polypeptide composition. *Eur. J. Biochem.* **2002**, *269*, 4219−4225.

(38) Zhou, G. P.; Cai, Y. D. Predicting protease types by hybridizing gene ontology and pseudo amino acid composition. *Proteins* **2006**, *63*, 681−684.

(39) Ptitsyn, O. B.; Finkelstein, A. V. Similarities of protein topologies: evolutionary divergence, functional convergence or principles of folding? *Q. Rev. Biophys.* **1980**, *13*, 339−386.

(40) Chou, K. C.; Nemethy, G.; Scheraga, H. A. Review: Energetics of interactions of regular structural elements in proteins. *Acc. Chem. Res.* **1990**, *23*, 134−141.

(41) Creighton, T. E. Protein folding. *Biochem. J.* **1990**, *270*, 1−16.

(42) Creighton, T. E. Protein folding. An unfolding story. *Curr. Biol.* **1995**, *5*, 353−356.

(43) Chou, K. C.; Zhang, C. T.; Maggiora, G. M. Disposition of amphiphilic helices in heteropolar environments. *Proteins: Struct. Funct. Genet.* **1997**, *28*, 99−108.

(44) Tanford, C. Contribution of hydrophobic interactions to the stability of the globular conformation of proteins *J. Am. Chem. Soc.* **1962**, *84*, 4240−4274.

(45) Hopp, T. P.; Woods, K. R. Prediction of protein antigenic determinants from amino acid sequences *Proc. Natl. Acad. Sci. U.S.A.* **1981**, *78*, 3824−3828.

(46) Mahalanobis, P. C. On the generalized distance in statistics *Proc. Natl. Inst. Sci. India* **1936**, *2*, 49−55.

(47) Pillai, K. C. S. In *Encyclopedia of Statistical Sciences*; Kotz, S., Johnson, N. L., Eds.; John Wiley & Sons: New York, 1985; Vol. 5, pp 176−181. This reference also presents a brief biography of Mahalanobis who was a man of great originality and who made considerable contributions to statistics.

(48) Chou, K. C. A novel approach to predicting protein structural classes in a (20−1)-D amino acid composition space *Proteins: Struct. Funct. Genet.* **1995**, *21*, 319−344.

PR060167C