Available online at www.sciencedirect.com
ScienceDirect
Analytical Biochemistry 370 (2007) 1–16
ANALYTICAL BIOCHEMISTRY
www.elsevier.com/locate/yabio

Review

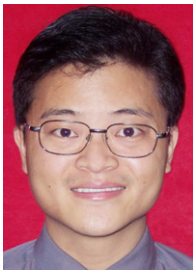
Recent progress in protein subcellular location prediction

Kuo-Chen Chou [a,*], Hong-Bin Shen [b]

[a] *Gordon Life Science Institute, San Diego, CA 92130, USA*
[b] *Department of Biological Chemistry and Molecular Pharmacology, Harvard Medical School, Boston, MA 02115, USA*

Received 20 May 2007
Available online 12 July 2007

Kuo-Chen Chou



Hong-Bin Shen

With a typical size of 10 μm and a typical mass of 1 ng, the cell is deemed to be the most basic structural and functional unit of all living organisms and often is called a "building block of life." An adult human is made up of approximately 100 trillion (or 10^{14}) cells [1].

According to the cellular anatomy, a cell is constituted by many different components, compartments, or organelles (Fig. 1) that are specialized to carry out different tasks such as the following. The cytoskeleton functions as a cell's scaffold, organizing and maintaining the cell's shape as well as anchoring organelles in place. Cytoplasm, a jelly-like material, takes up most of the cell volume, filling the cell and serving as a "molecular soup" in which all of the cell's organelles are suspended. The cell membrane functions as a boundary layer to contain the cytoplasm, whereas the cell wall provides protection from physical injury. The cell nucleus contains the genetic material (DNA) governing all functions of the cell. The mitochondrion is the "power generator," playing a critical role in generating energy in the eukaryotic cell. The chloroplast is the site of photosynthesis. The centriole forms spindle fibers to separate chromosomes during cell division. The endoplasmic reticulum transports chemicals between and within cells. The Golgi apparatus modifies chemicals to make them functional. The lysosome breaks large molecules into small molecules by inserting a molecule of water into the chemical bond. The endoplasmic reticulum is, together with the ribosome, responsible for synthesizing proteins. Vacuoles store food and waste.

However, most of these functions, which are critical to a cell's survival, are performed by the proteins in the cell [2,3]. A typical cell contains approximately 1 billion (or 10^9) protein molecules that reside in many different compartments or organelles (Fig. 1), usually termed "subcellular locations." Therefore, one of the fundamental goals in cell biology and proteomics is to identify the subcellular locations and functions of these proteins, the cell's primary machinery. Information of the subcellular locations of proteins can provide useful clues about their functions. For understanding the intricate pathways that regulate the biological processes at the cellular level, we also need to know the subcellular distributions of proteins.

Although the information about protein subcellular localization can be determined by conducting various biochemical experiments, the approach of purely doing experiments is both time-consuming and costly. In particular, the number of newly found protein sequences has increased greatly in the post genomic era. For instance, according to version 52.0 of the Swiss–Prot database released in March 2007 (www.ebi.ac.uk/swissprot), the number of total pro-

* Corresponding author.
E-mail address: kcchou@gordonlifescience.org (K.-C. Chou).

0003-2697/$ - see front matter
doi:10.1016/j.ab.2007.07.006

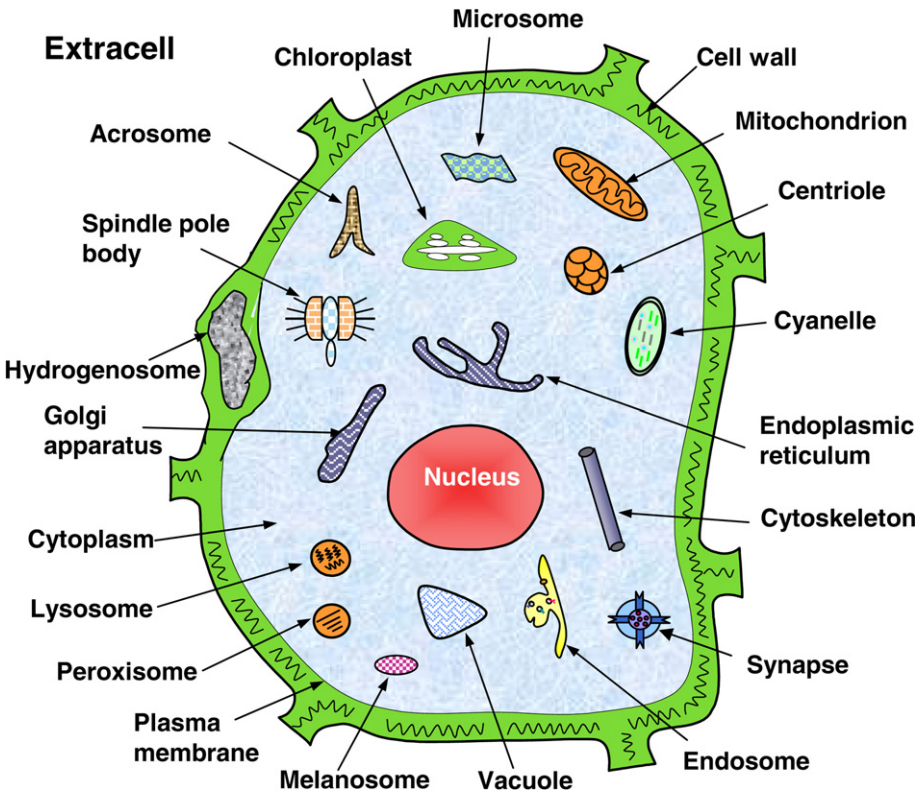


Fig. 1. Schematic illustration showing many different components or organelles in a eukaryotic cell.

tein entries was 260,175. After excluding those annotated as "fragments" or containing less than 50 amino acid residues, the number is reduced to 247,262, of which 133,652 are with subcellular location annotations (item 1 of Table 1). However, of the 133,652 proteins, only 49,367 are annotated with experimental observations (item 2 of Table 1) and 84,285 are annotated with uncertain labels such as "probable," "potential," "perhaps," and "by similarity" (item 3 of Table 1). The uncertain annotations cannot be used as robust data for training a solid predictor. In actuality, proteins with uncertain annotations also belong to the targets of identification either by newly developed predictors or by further experiments.

A similar gap also exists in the gene ontology (GO)[1] database [4] that was established according to the molecular function, biological process, and cellular component. As shown in item 5 of Table 1, of the 247,262 proteins, only 116,593 have GO annotations to indicate their subcellular components. In other words, the percentage of the protein entries with subcellular annotations in the GO database (47.2%) is even lower than that in the Swiss–Prot database (54.1%). Moreover, it is instructive to point out that the GO database was derived from other more basic databases, including the Swiss–Prot database. Therefore, the GO annotations might be contaminated by the uncertain information from the 84,285 entries as indicated in item 3 of Table 1.

Therefore, the number of proteins that have reliable subcellular location annotations is 49,367 (item 2 of Table 1), which is approximately 20% of all protein entries concerned. In other words, there are 247,262–49,367 = 197,895 proteins for which the subcellular localization needs to be identified or further confirmed.

With the deluge of gene products in the postgenomic age, it is expected that the gap between the newly found protein sequences and the knowledge of their subcellular localization will grow continuously larger. To use these newly found proteins for basic research and drug discovery in a timely manner [5,6], it is highly desired to develop an effective method to bridge such a gap. During the past 15 years, a variety of predictors have been developed to deal with the challenge [7–39].

The current review is focused on those predictors that distinguish themselves by having one or more of the following features:

1. *Wide coverage scopes.* Compared with the earlier stage predictors that cover the scope from only 2 to 5 subcellular locations, some predictors developed recently can cover up to 22 subcellular locations.
2. *Incorporation of multiplex proteins.* Proteins with multiple subcellular locations are particularly interesting because they may have some special biological functions

[1] *Abbreviations used:* GO, gene ontology; AA, amino acid; PseAA, pseudo amino acid; FunD, functional domain; -D, -dimensional; CD, covariant discriminant; KNN, *K* nearest neighbor; OET-KNN, optimized evidence-theoretic *K* nearest neighbor; ET-KNN, evidence-theoretic *K* nearest neighbor.

Table 1
Breakdown of 247,262 protein entries from Swiss–Prot database (version 52.0) according to the nature of their subcellular location annotation and their expression in GO database

Item	Description	Number	Percentage
1	Proteins with subcellular location annotations in Swiss–Prot database	133,652	$\frac{133652}{247262} = 54.1\%$
2	Proteins in item 1 with experimentally observed subcellular locations	49,367	$\frac{49367}{247262} = 20.0\%$
3	Proteins in item 1 with uncertain terms such as "potential", "probable", and "by similarity"	84,285	$\frac{84285}{247262} = 34.1\%$
4	Proteins that can be represented in the GO space (cf. Eq. (3))	226,596	$\frac{226596}{247262} = 91.6\%$
5	Proteins with subcellular component annotations in GO database	116,593	$\frac{116593}{247262} = 47.2\%$

Note. The original number of protein entries was 260,175. Of these, 12,913 were annotated as "fragments" or contained less than 50 amino acid residues and, hence, were removed from further consideration.

[40]. However, most existing predictors can be used only on the assumption that each protein has one, and only one, subcellular location, the so-called "single site" case.
3. *Rigorous datasets.* To avoid homology bias, a stringent criterion was imposed to construct the benchmark datasets where none of protein samples included has equal to or higher than 25% sequence identity to any other in a same subcellular location. In contrast, many datasets constructed for the earlier stage predictors allow inclusion of protein samples with 80% or even higher sequence identity.
4. *Organism-specific approach.* Some recent predictors were developed according to different specific organisms such as human, plant, and bacterium. This will make the prediction more accurate [29] without reducing the value of practical application because the source of a query protein usually is known.
5. *State-of-the-art technique.* Some powerful tools, such as ensemble classifier and fusion approach, were introduced to enhance prediction accuracy.
6. *Availability.* The Web servers for these predictors are freely accessible to the public.

Of these six features, features 1 to 3 will make the predictors more useful in a practical sense, features 4 and 5 will enhance the success rates of prediction, and feature 6 will provide readers with a user-friendly means to apply these predictors.

Construction of benchmark datasets

A benchmark dataset usually consists of a learning (or training) dataset and an independent testing dataset [41]. The learning dataset is one of the important components for a statistical predictor because it is used for training the predictor's "engine," whereas the testing dataset is used for examining the predictor's accuracy via a cross-validation. However, as will be illustrated later, if the cross-validation is performed by the subsampling or jackknife approach, one dataset can serve both the training and testing purposes.

The benchmark datasets constructed for most of the existing predictors have the following problems. First, the coverage was too narrow, particularly for the datasets constructed in the earlier stage (see, e.g., Refs. [7–9,11]) that covered the scope from only two to five location sites. Second, there was not a clear cutoff criterion to remove proteins with high sequence identity from the datasets. To avoid the redundancy and homology bias, such a cutoff criterion is necessary. Although some datasets were constructed with a cutoff to exclude those proteins that had greater than 80% (see, e.g., Ref. [24]) or 90% (see, e.g., Refs. [11,29]) sequence identity to any other in a same subcellular location, cutoff with thresholds like these is too tolerable to exclude the homologous effects because many proteins with just about 40% sequence identity might be homologous to each other [5]. Third, the datasets were constructed by mixing proteins from many different organisms, and this would be inconsistent with the organism-specific approach, as mentioned above. Fourth, most of existing predictors were established on the assumption that a protein has one, and only one, subcellular location. As a consequence, in constructing the benchmark datasets for these predictors, an arbitrary criterion was imposed to exclude those proteins that had more than one subcellular location. Although most proteins are of a single location, some may simultaneously exist at, or move between, two or more different subcellular locations [40]. Proteins with multiple locations or a dynamic feature of this kind are particularly interesting because they may have some very special biological functions intriguing to investigators in both basic research and drug discovery [35,42].

To cope with the aforementioned four problems, the benchmark datasets constructed recently were assembled strictly according to the following criteria. First, protein sequences were collected from the Swiss–Prot database according to their experimentally annotated subcellular locations (-!-SUBCELLULAR LOCATION) using the most recent version possible. Second, to construct the benchmark dataset for an organism-specific predictor, only those entries related to the organism concerned are included. For example, when constructing the dataset for predicting the human protein subcellular location, only those entries annotated with "human" in the ID (identification) fields were collected [43], and when constructing the dataset for predicting the plant protein subcellular location, only those entries annotated with "viridiplantae" in the OC (organism classification) fields were collected [38].

Third, because a same subcellular location might be annotated with different terms, to collect as much desired information as possible, several "keywords" might be used for a same subcellular location. For example, in search for centriole proteins, the keywords "centriole," "centrosome," and "centromer" were used; in search for cytoskeleton proteins, the keywords "cytoskeleton," "filament," and "microtubule" were used; in search for extracell proteins, the keywords "extracell," "extracellular," and "secreted" were used; in search for peroxisome proteins, the keywords "peroxisome," "microsome," "glyoxysomal," and "glycosomal" were used; and in search for plasma membrane proteins, the keywords "plasma membrane" and "integral membrane" were used [44]. Fourth, sequences annotated with ambiguous or uncertain words, such as "potential," "probable," "probably," "maybe," and "by similarity," were excluded. Fifth, sequences annotated with "fragment" were excluded, and sequences with less than 50 amino acid residues were removed because they might be just fragments (see, e.g., Ref. [43]). Sixth, to avoid any homology bias, a redundancy cutoff was operated by a culling program to winnow those sequences that have equal to or higher than 25% sequence identity to any other in a same subcellular location. Seventh, for the datasets constructed for the single location predictor, sequences annotated by two or more locations were not included; however, for the datasets constructed for the multiple-location predictor [39,42], no such constraint was imposed.

According to the above procedures, eight benchmark datasets were constructed for predicting the subcellular locations of eukaryotic proteins, human proteins, plant proteins, Gram-positive proteins, Gram-negative proteins, virus proteins, eukaryotic proteins with both single and multiple location sites, and human proteins with both single and multiple location sites. All of these datasets can be freely downloaded from the Web sites mentioned below.

Representation of protein samples

In developing a method for predicting protein subcellular location, the first problem we face is how to represent the sample of a protein. Two kinds of representations were generally used in this regard: the sequential representation and the nonsequential representation.

Sequential representation

The most typical sequential representation for a protein sample is its entire amino acid sequence, which can contain the most complete information of the protein. This is an obvious advantage of the sequential representation. To get the desired results, the sequence similarity search-based tools, such as BLAST [45,46], usually are used to conduct the prediction. However, this kind of approach failed to work when the query protein did not have significant homology to proteins of known location. Thus, various nonsequential representation models were proposed, as illustrated below.

Nonsequential representation

Rather than a series of successive amino acid codes according a certain order, the nonsequential representation for a protein sample is expressed by a set of discrete numbers and, hence, is also called the discrete representation. Various discrete models for this kind of representation were developed and can be briefly formulated as follows.

AA composition discrete model

The simplest discrete representation was based on the amino acid (AA) composition. The AA composition discrete model can be formulated as follows. Given a protein sequence $\mathbf{P}$ with L amino acid residues,

$$\mathbf{P} = \mathrm{R}_1\mathrm{R}_2\mathrm{R}_3\mathrm{R}_4\mathrm{R}_5\mathrm{R}_6\mathrm{R}_7\cdots\mathrm{R}_L, \tag{1}$$

where R_1 represents the first residue of the protein $\mathbf{P}$, R_2 represents the second residue, and so forth. According to the AA composition discrete model, the protein $\mathbf{P}$ of Eq. (1) can be expressed by

$$\mathbf{P} = [f_1\ f_2\ \cdots\ f_{20}]^{\mathbf{T}}, \tag{2}$$

where f_u ($u = 1, 2, \ldots, 20$) are the normalized occurrence frequencies of the 20 native amino acids [47,48] in protein $\mathbf{P}$ and $\mathbf{T}$ is the transposing operator. Many methods for predicting protein subcellular location were based on the AA composition discrete model (see, e.g., Refs. [8,9,12,22]). The AA composition discrete model was also widely used for predicting the structural class of proteins (see, e.g., Refs. [47–55]) and their other attributes. However, as can be seen from Eq. (2), all of the sequence order effects are lost by using the AA composition discrete model. This is the main shortcoming of the AA composition discrete model.

PseAA composition

To avoid losing the sequence order information completely, the concept of pseudo amino acid composition (PseAA composition) was proposed [56]. According to the PseAA composition discrete model, the protein $\mathbf{P}$ of Eq. (1) can be formulated as

$$\mathbf{P} = \left[p_1, p_2, \ldots, p_{20}, p_{20+1}, \ldots, p_{20+\lambda}\right]^{\mathbf{T}}, \quad (\lambda < L), \tag{3}$$

where the $20 + \lambda$ components are given by

$$p_u = \begin{cases} \dfrac{f_u}{\sum_{i=1}^{20} f_i + w\sum_{k=1}^{\lambda}\tau_k}, & (1 \leqslant u \leqslant 20) \\ \dfrac{w\tau_{u-20}}{\sum_{i=1}^{20} f_i + w\sum_{k=1}^{\lambda}\tau_k}, & (20+1 \leqslant u \leqslant 20+\lambda) \end{cases}, \tag{4}$$

where w is the weight factor, which was set at 0.05 in Ref. [56], and τ_k is the kth tier correlation factor, which reflects the sequence order correlation between all of the kth most contiguous residues (Fig. 2) as formulated by

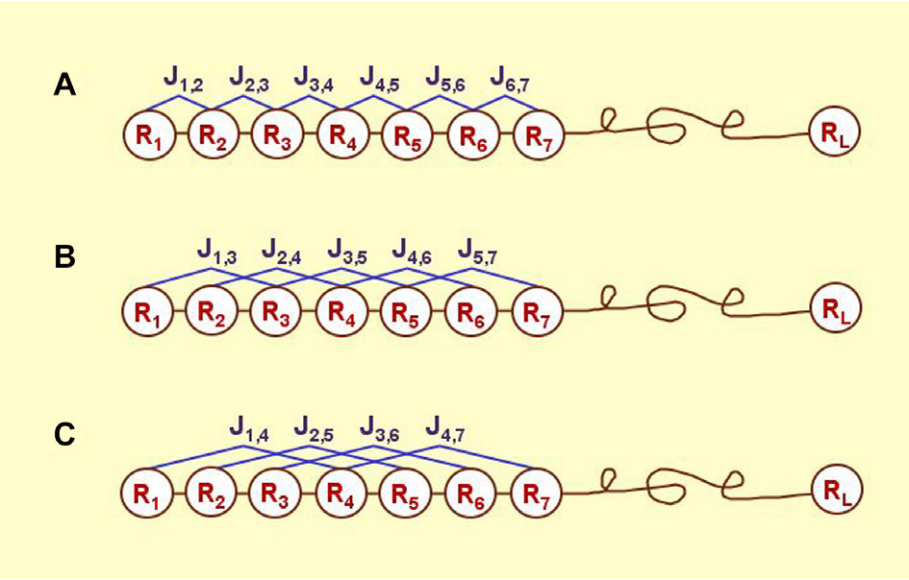


Fig. 2. Schematic drawing showing first-tier (A), second-tier (B), and third-tier (C) sequence order correlation mode along a protein sequence, where R_1 represents the amino acid residue at the sequence position 1, R_2 represents the amino acid residue at the sequence position 2, and so forth, and the coupling factors $J_{i,j}$ are given by Eq. (6). Panel A reflects the correlation mode between all of the most contiguous residues, panel B reflects the correlation mode between all of the second most contiguous residues, and panel C reflects the correlation mode between all of the third most contiguous residues. (Adapted from Ref. [56] with permission.)

$$\tau_k = \frac{1}{L-k}\sum_{i=1}^{L-k} J_{i,i+k}, \quad (k < L) \tag{5}$$

with

$$J_{i,i+k} = \frac{1}{3}\Big\{[H_1(\mathrm{R}_{i+k}) - H_1(\mathrm{R}_i)]^2 + [H_2(\mathrm{R}_{i+k}) - H_2(\mathrm{R}_i)]^2 + [M(\mathrm{R}_{i+k}) - M(\mathrm{R}_i)]^2\Big\}, \tag{6}$$

where $H_1(\mathrm{R}_i)$, $H_2(\mathrm{R}_i)$, and $M(\mathrm{R}_i)$ are the hydrophobicity value, hydrophilicity value, and side chain mass for the amino acid R_i, respectively. Note that before substituting the values of hydrophobicity, hydrophilicity, and side chain mass into Eq. (6), they all are subjected to a standard conversion, as described by the following equation:

$$\begin{cases} H_1(\mathrm{R}_i) = \frac{H_1^0(\mathrm{R}_i) - \langle H_1^0 \rangle}{\mathrm{SD}(H_1^0)} \\ H_2(\mathrm{R}_i) = \frac{H_2^0(\mathrm{R}_i) - \langle H_2^0 \rangle}{\mathrm{SD}(H_2^0)} \\ M(\mathrm{R}_i) = \frac{M^0(\mathrm{R}_i) - \langle M^0 \rangle}{\mathrm{SD}(M^0)} \end{cases}, \tag{7}$$

where the symbols $H_1^0(\mathrm{R}_i)$ and $H_2^0(\mathrm{R}_i)$ are the original hydrophobicity and hydrophilicity values for R_i, which can be obtained from Tanford [57] and Hopp and Woods [58], respectively, and $M^0(\mathrm{R}_i)$ is the mass of the side chain for R_i, which can be found in any biochemistry textbook. In Eq. (7) the symbol $< >$ means taking the average of the quantity therein over 20 native amino acids, and SD means the corresponding standard deviation. The converted values obtained by Eq. (7) will have a zero mean value over the 20 native amino acids, and will remain unchanged if they go through the same conversion procedure again. As can be seen, the first 20 components in Eq. (3) (i.e., $p_1, p_2, \ldots, p_{20}$) are associated with the conventional amino acid composition of **P**, whereas the remaining components ($p_{20+1}, p_{20+2}, \ldots, p_{20+\lambda}$) are the λ correlation factors that reflect the first tier, second tier, and so forth up to the λth tier sequence order correlation patterns (Fig. 2). It is these additional λ factors that approximately incorporate the sequence order effects. Note that λ is a parameter of integer (see Fig. 1 and Eq. (3)) and that choosing a different integer for λ will lead to a dimension different Pse-ACC, as will be further discussed later. Also note that using Eq. (6) is just one of the modes for deriving the correlation factors or PseAA components. The others, such as the physicochemical distance mode [59] and the amphiphilic pattern mode [60], can also be used to derive different types of PseAA composition. For the convenience of users, a free server called PseAA is provided at the Web site http://chou.med.harvard.edu/bioinf/PseAA. By using the Web server, users can generate the PseAA composition for any given protein sequence by selecting the mode they wish. Since the concept of PseAA composition was proposed [56], varieties of PseAA composition approaches have been stimulated to improve the quality of predicting subcellular localization of proteins and their other attributes (see, e.g., Refs. [23,37,61–67]).

FunD discrete model

The representation for a protein sample in this model is expressed by its functional domain (FunD), as originally introduced in Refs. [21,68] based on the rationale that the function of a protein usually is correlated with its subcellular location. The integrated domain and motif database

[69], or InterPro database, consists of many sequences with well-known FunD types. For example, InterPro release 6.2 (April 2003) contains 7785 entries that are available from the Web site www.ebi.ac.uk/interpro. Using each of the 7785 FunDs as a vector base, a sequence can be defined as a 7785-D (dimensional) vector according to the following steps. First, use the program IPRSCAN [69] to search the InterPro database for a given protein sequence (e.g., the sequence in **P** of Eq. (1)). If there is a hit (e.g., IPR001970, meaning that the protein contains a sequence segment very similar to that of the 1970th domain of the InterPro database), then the 1970th component of the protein in the 7785-D FunD space is assigned 1; otherwise, it is assigned 0. Second, the protein can be explicitly formulated as follows:

$$\mathbf{P} = [d_1\ d_2 \cdots d_j \cdots\ d_{7785}]^{\mathbf{T}}, \tag{8}$$

where

$$d_j = \begin{cases} 1, & \text{hit found in InterPro database} \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

Defined in this way, a protein will correspond to a 7785-D vector with each of the 7785 FunD patterns as a base for the vector space. With the continuous expansion of the FunD database, the dimension of the FunD discrete model will become larger as well. Because the function of a protein is correlated with its structure, the FunD discrete model is particularly effective for predicting protein structural class [70].

GO discrete model

Instead of the FunD database, the representation for a protein sample in this discrete model is defined in a GO database space [4], as originally introduced in Ref. [71]. Doing so was based on an assumption that proteins mapped onto the GO database space would be clustered in a way better reflecting their subcellular locations, although the percentage of proteins with known subcellular location annotations in the GO database is even lower than that in the Swiss–Prot database, as mentioned in Introduction and illustrated in Table 1. The procedures to represent a protein sample are as follows. First, mapping UniProtKB protein entries [72] onto the GO database, one can get a list of data called gene_association.goa_uniprot, where each UniProtKB protein entry (accession number) corresponds to one or several GO numbers. Such a data file can be downloaded directly from ftp://ftp.ebi.ac.uk/pub/databases/GO/goa/UNIPROT. The number of relationships between the UniProtKB protein entries and the GO numbers may be one or many, "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," as described in Ref. [4] and as illustrated in Refs. [38,43,73] through some examples for different organisms. Second, the GO numbers in the existing GO database do not increase successively and orderly. For easier handling, some reorganization and compression procedure was taken to renumber them. For example, for the GO database released on 30 May 2006, after such a procedure the original GO numbers GO:0000001, GO:0000002, GO:0000003, GO:0000004, GO:0000006, ..., GO:0051990 would become GO_compress:0000001, GO_compress:0000002, GO_compress:0000003, GO_compress:0000004, GO_compress:0000005, ..., GO_compress:0010173, respectively. The GO database obtained in this way is called the GO_compress database, whose maximum number was reduced from 51,990 in the original GO database to 10,173. Each of the 10,173 entities in the GO_compress database serves as a base to define a protein sample. Third, search the GO_compress database for a protein sample. If there is a hit corresponding to the ith GO_compress number, then the ith component of the protein in the 10,173-D GO_compress space is assigned 1; otherwise, it is assigned 0. Thus, the protein can be formulated as

$$\mathbf{P} = [g_1\ g_2\ \cdots\ g_i\ \cdots\ g_{10173}]^{\mathbf{T}}, \tag{10}$$

where

$$g_i = \begin{cases} 1, & \text{hit found in GO\_compress} \\ 0, & \text{otherwise} \end{cases}. \tag{11}$$

Hybridization discrete model

Unfortunately, the current GO database may fail to give complete coverage of a benchmark dataset in the sense that some proteins might not belong to any of the GO numbers and, hence, Eq. (10) will become a naught vector. Although the problem gradually will become trivial or eventually will be solved with the GO database developing continuously, to tackle such a problem right now a hybridization approach has been introduced by fusing the GO discrete model Eq. (10) with the PseAA composition discrete model Eq. (3). The discrete model obtained in this way is called the GO-PseAA model (see, e.g., Ref. [44]), where proteins corresponding to a naught vector in the GO model Eq. (10) will be redefined by the PseAA composition model Eq. (3). Likewise, the hybridization approach can be used to deal with the situation where the representation for a protein in the FunD model Eq. (8) is a naught vector because no hit whatsoever can be found in searching the current FunD database for the protein. The discrete model obtained by hybridizing FunD with Pse-AAC is called the FunD-PseAA model (see, e.g., Ref. [74]). Also, we can deal with this kind of situation by hybridizing the GO, FunD, and PseAA approaches, and the model obtained in this way is called the GO-FunD-PseAA model (see, e.g., Refs. [71,75]).

The self-consistency principle

Here the so-called self-consistency principle means that, regardless of which kind of representation model is

adopted for protein samples, the query proteins and the proteins used to train the prediction engine must be defined in a same infrastructural frame with exactly the same dimensions. For instance, if a query protein is defined in the 10,173-D GO space see Eq. (10), then the prediction should be carried out based on those proteins in the training set that can be defined in the exactly same 10,173-D GO space as well. If the query protein in the 10,173-D GO space is a naught vector and, hence, must be defined instead in the $(20+\lambda)$-D PseAA composition space as shown in Eq. (3), then all of the proteins used to train the prediction engine must also be formulated in the same $(20+\lambda)$-D PseAA composition space. It is particularly important to follow such a self-consistency principle when using the hybridization discrete model [44] or when building an ensemble classifier by fusing many individual classifiers [32].

Prediction algorithms

In general, the problem of predicting protein subcellular localization can be formulated as follows. Consider a system containing N proteins ($\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_N$) that have been classified into M subsets (subcellular locations),

$$\mathbb{S} = S_1 \cup S_2 \cup S_3 \cup \cdots \cup S_M, \tag{12}$$

where each subset S_m $(m = 1, 2, \ldots, M)$ is composed of proteins with the same subcellular location and its size (the number of proteins therein) is N_m. Obviously, we have $N = N_1 + N_2 + \cdots + N_M$. Now for a query protein $\mathbf{P}$, how can we identify the subset to which it belongs? Many different prediction algorithms have been developed to address this problem. Here we focus on the covariant discriminant (CD) algorithm, the K nearest neighbor (KNN) algorithm, and the optimized evidence-theoretic K nearest neighbor (OET-KNN) algorithm as well as on how to generate an ensemble classifier by fusing many individual prediction engines characterized by different control parameters to enhance the prediction accuracy.

CD classifier

The CD classifier originally was introduced to predict the structural classification of proteins [76]. Subsequently, it was used to predict protein subcellular location first based on the AA composition discrete model [12] and later based on the PseAA composition discrete model [56]. According to Eqs. (2) and (3), we can suppose without losing generality that the uth protein in the subset S_m (see Eq. (12)) is formulated by

$$\mathbf{P}_m^u = \left[\, p_{m,1}^u \quad p_{m,2}^u \quad \cdots \quad p_{m,20}^u \quad \cdots \quad p_{m,\Lambda}^u \,\right]^{\mathbf{T}}, \tag{13}$$

where $p_{m,j}^u (j = 1, 2, \ldots, \Lambda)$ is the jth component of the uth protein in S_m and the standard vector for the subset S_m is defined by

$$\bar{\mathbf{P}}_m = \left[\, \bar{p}_{m,1} \quad \bar{p}_{m,2} \quad \cdots \quad \bar{p}_{m,20} \quad \cdots \quad \bar{p}_{m,\Lambda} \,\right]^{\mathbf{T}}, \tag{14}$$

where

$$\bar{p}_{m,i} = \frac{1}{N_m} \sum_{u=1}^{N_m} p_{m,i}^u, \quad (i = 1, 2, \ldots, \Lambda). \tag{15}$$

$\bar{\mathbf{P}}_m$ as defined above can be deemed to be a standard protein for the subset S_m. According to the self-consistency principle, the sample of a query protein should be represented by

$$\mathbf{P} = \left[\, p_1 \quad p_2 \quad \cdots \quad p_{20} \quad \cdots \quad p_\Lambda \,\right]^{\mathbf{T}}. \tag{16}$$

For the PseAA composition discrete model, the components in Eqs. (13) and (16) can be derived by Eq. (4) from the sequences of the corresponding proteins or by simply using the Web server PseAA. Thus, the similarity between a query protein $\mathbf{P}$ and $\bar{\mathbf{P}}_m$ is defined by the following CD function:

$$\mathbb{F}(\mathbf{P}, \bar{\mathbf{P}}_m) = D_{\mathrm{Mah}}^2(\mathbf{P}, \bar{\mathbf{P}}_m) + \ln|\mathbf{C}_m|, \quad (m = 1, 2, \ldots, M), \tag{17}$$

where

$$D_{\mathrm{Mah}}^2(\mathbf{P}, \bar{\mathbf{P}}_m) = (\mathbf{P} - \bar{\mathbf{P}}_m)^{\mathbf{T}} \mathbf{C}_m^{-1} (\mathbf{P} - \bar{\mathbf{P}}_m) \tag{18}$$

is the squared Mahalanobis distance [53,77,78] between $\mathbf{P}$ and $\bar{\mathbf{P}}_m$,

$$\mathbf{C}_m = \begin{bmatrix} c_{1,1}^m & c_{1,2}^m & \cdots & c_{1,\Lambda}^m \\ c_{2,1}^m & c_{2,2}^m & \cdots & c_{2,\Lambda}^m \\ \vdots & \vdots & \ddots & \vdots \\ c_{\Lambda,1}^m & c_{\Lambda,2}^m & \cdots & c_{\Lambda,\Lambda}^m \end{bmatrix} \tag{19}$$

is the covariance matrix for the subset S_m, the $\Lambda \times \Lambda$ elements in $\mathbf{C}_{\mathrm{m}}$ are given by

$$c_{i,j}^m = \frac{1}{N_m - 1} \sum_{u=1}^{N_m} \left(p_{m,i}^u - \bar{p}_{m,i}\right)\left(p_{m,j}^u - \bar{p}_{m,j}\right), \quad (i, j = 1, 2, \ldots, \Lambda), \tag{20}$$

$\mathbf{C}_m{}^{-1}$ is the inverse matrix of $\mathbf{C}$, and $|\mathbf{C}_m|$ is the determinant of the matrix $\mathbf{C}_m$. The smaller the value of $\mathbb{F}(\mathbf{P}, \bar{\mathbf{P}}_m)$, the higher the similarity between $\mathbf{P}$ and $\bar{\mathbf{P}}_m$. Therefore, the query protein is predicted to belong to the subset S_μ or the μth subcellular location if

$$\mu = \arg\ \min_m \left\{\mathbb{F}(\mathbf{P}, \bar{\mathbf{P}}_m)\right\}, \quad (m = 1, 2, \ldots, M), \tag{21}$$

where μ is the argument of m that minimizes $\mathbb{F}(\mathbf{P}, \bar{\mathbf{P}}_m)$. If there are more than one argument leading to a same minimum value for $\mathbb{F}(\mathbf{P}, \bar{\mathbf{P}}_m)$, then the query protein will be randomly assigned to one of the subcellular locations associated with these arguments, although this kind of tie case rarely happens. Note that owing to the normalization condition imposed by Eq. (4), of the Λ components in Eq. (13), only $\Lambda - 1$ are independent; hence, the covariance matrix $\mathbf{C}_m$ as defined by Eq. (19) must be a singular one [53]. This would lead the Mahalanobis distance defined by Eq. (18) and the CD function defined by Eq. (17) to be divergent and meaningless. To cope with such a situation, the

dimension-reducing procedure [48] was adopted in practical calculations; that is, instead of Λ-D space, a protein sample is defined in a $(\Lambda-1)$-D space by leaving out one of its Λ components. The remaining $(\Lambda-1)$ components would be completely independent; therefore, the corresponding covariance matrix $\mathbf{C}_m$ no longer would be singular. In other words, the Mahalanobis distance Eq. (18) and the CD function Eq. (17) based on such a $(\Lambda-1)$-D space can be uniquely defined without any trouble. However, a question might be raised as to which one of the Λ components can be left out. The answer is that any one of them can be left out. Will leaving out a different component lead to a different predicted result? The answer is no. According to the invariance theorem given in Chou's [48] Appendix A, both the value of the Mahalanobis distance and the value of the determinant of $\mathbf{C}_m$ will remain exactly the same regardless of which one of the Λ components is left out. Accordingly, the final value of the CD function Eq. (17) can be uniquely defined through such a dimension-reducing procedure.

KNN classifier

The KNN classifier is quite popular in the pattern recognition community owing to its good performance and ease of use. According to the KNN rule [79–81], also called the voting KNN rule, the query protein should be assigned to the subset represented by a majority of its K nearest neighbors. There are many different definitions to measure the "nearness" for the KNN classifier, including Euclidean distance, Hamming distance [82], and Mahalanobis distance [48,77,78]. Usually, the following equation has been adopted to measure the nearness between proteins $\mathbf{P}$ and $\mathbf{P}_m^u$ (cf. Eqs. (13) and (16)):

$$D(\mathbf{P},\mathbf{P}_m^u)=1-\frac{\mathbf{P}\cdot\mathbf{P}_m^u}{\|\mathbf{P}\|\|\mathbf{P}_m^u\|}, \tag{22}$$

where $\mathbf{P}\cdot\mathbf{P}_m^u$ is the dot product of the two vectors and $\|\mathbf{P}\|$ and $\|\mathbf{P}_m^u\|$ are their respective moduluses. According to Eq. (22), when $\mathbf{P}\equiv\mathbf{P}_m^u$, we have $D(\mathbf{P},\mathbf{P}_m^u)=0$, indicating that the "distance" between these two proteins is zero and, hence, they have perfect or 100% similarity. In using the KNN rule, the predicted result will depend on the selection of the parameter K, that is, the number of the nearest neighbors to the query protein $\mathbf{P}$. Below, let us consider the following two cases. First, when $K=1$, the protein $\mathbf{P}$ will be predicted to belong to the same subcellular location of the protein in the learning dataset that has the shortest distance to $\mathbf{P}$; that is, the query protein will be classified in the μth subcellular location if

$$\mu=\mathbf{arg}\ \mathbf{min}_m\left\{\mathbf{min}_{\mathbf{P}_m^u\in S_m}\left[D(\mathbf{P},\mathbf{P}_m^u)\right]\right\},\quad (m=1,2,\ldots,M), \tag{23}$$

where $\mathbf{min}_{\mathbf{P}_m^u\in S_m}$ means taking the minimum value of $D(\mathbf{P},\mathbf{P}_m^u)$ for the proteins in the subset S_m (cf. Eqs. (12) and (13)), and $\mathbf{arg}\ \mathbf{min}_m$ has the same meaning as in Eq. (21). Also, the same treatment as mentioned above for Eq. (21) is used to deal with the tie case, although it rarely happens. Second, when $K>1$, the subcellular location of the query protein $\mathbf{P}$ will be determined by the majority of its K nearest neighbors via a vote, as can be formulated as follows. Suppose that $(\mathbf{P}_1^*,\mathbf{P}_2^*,\ldots,\mathbf{P}_K^*)$ are the K proteins in $\mathbb{S}$ that have the closest distances to $\mathbf{P}$. The query protein will be predicted to belong to the μth subcellular location if

$$\mu=\mathbf{arg}\ \mathbf{max}_m\left\{\sum_{i=1}^{K}\Delta(\mathbf{P}_i^*,S_m)\right\},\quad (m=1,2,\ldots,M), \tag{24}$$

where μ is the argument of m that maximizes $\{\sum_{i=1}^{K}\Delta(\mathbf{P}_i^*,S_m)\}$, and

$$\Delta(\mathbf{P}_i^*,S_m)=\begin{cases}1, & \text{if } \mathbf{P}_i^*\in S_m\\ 0, & \text{otherwise}\end{cases}, \tag{25}$$

where $\in$ is a symbol in the set theory meaning "member of." If there is a tie for the voting results, the query protein will be randomly assigned to one of the locations associated with the tie case. In general, the greater the K (the number of the nearest neighbors counted), the less likely the tie case occurs.

OET-KNN classifier

The OET-KNN classifier was developed from the evidence-theoretic K nearest neighbor (ET-KNN) classifier, a pattern classification method established on the basis of the Dempster–Shafer theory of belief functions [80]. During the process of classification, each neighbor of a protein sample to be classified is considered as an item of evidence supporting certain hypotheses concerning the class membership of that protein. Based on this kind of evidence, the basic belief masses are assigned to each subset concerned. Such masses are obtained from the K nearest neighbors of the query protein and are aggregated using Dempster's rule of combination [83]. Thus, a decision is made by assigning the query protein 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 that $S_K^{\mathbf{P}}$ is the set formed by the K nearest neighbors to $\mathbf{P}$ in $\mathbb{S}$ of Eq. (12). Thus, for any $\mathbf{P}_i^*\in S_K^{\mathbf{P}}$, the knowledge that $\mathbf{P}_i^*$ belongs to class $S_m\subset\mathbb{S}$ can be considered as a piece of evidence that increases our belief that $\mathbf{P}$ also belongs to S_m. According to the basic belief assignment mapping theory [83], this item of evidence can be formulated by

$$\mathrm{B}\left(\mathbf{P}_i^*,S_m\right)=\alpha_0\exp\left[-\gamma_m^2D^2(\mathbf{P}_i^*,\mathbf{P})\right],\quad (m=1,2,\ldots,M), \tag{26}$$

where $\alpha_0=0.95$ is a fixed parameter, γ_m is a parameter associated with class S_m, and $D^2(\mathbf{P}_i^*,\mathbf{P})$ is the square distance or dissimilarity between the query protein $\mathbf{P}$ and its ith nearest protein $\mathbf{P}_i^*\in S_K^{\mathbf{P}}$. Various distance metrics (e.g., Hamming distance, Euclidean distance, Mahalanobis distance [48,77,78]) or dissimilarity scales could be used to

measure the nearness between $\mathbf{P}$ and $\mathbf{P}_i^*$. For instance, according to the practice [44,73], when the protein samples were expressed in the PseAA composition space (cf. Eq. (3)), the Euclidean distance was used to define $D(\mathbf{P}_i^*, \mathbf{P})$, whereas when they were expressed in the GO space (cf. Eq. (10)), the dissimilarity of $1 - \cos(\mathbf{P}, \mathbf{P}_i^*)$ as defined in Eq. (22) was adopted. In the ET-KNN rule, how to optimally select the parameters in Eq. (26) was not addressed. In 1998, an optimization procedure to determine the optimal or near optimal parameter values for γ_μ ($\mu = 1, 2, \ldots, M$) was proposed from the learning dataset by minimizing an error function [84]. 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 subset S_m is a combination of its K nearest neighbors and can be formulated as

$$\mathrm{B}(\mathbf{P}, S_m, K) = \oplus_{i=1}^{K} \mathrm{B}(\mathbf{P}_i, S_m), \tag{27}$$

where the symbol $\oplus_{i=1}^{K}$ represents the orthogonal sum from $i = 1$ to K. According to Dempster's rule [83], the belief function of Eq. (27) can be expressed as

$$\mathrm{B}(\mathbf{P}, S_m, K) = \frac{\sum_{S_{K,i}^{\mathbf{P}} \subseteq S_K^{\mathbf{P}}, S_{K,j}^{\mathbf{P}} \subseteq S_K^{\mathbf{P}}, S_{K,i}^{\mathbf{P}} \cap S_{K,j}^{\mathbf{P}} = S_m} \mathrm{B}(\mathbf{P}, S_{K,i}^{\mathbf{P}}) \mathrm{B}(\mathbf{P}, S_{K,j}^{\mathbf{P}})}{1 - \sum_{S_{K,i}^{\mathbf{P}} \subseteq S_K^{\mathbf{P}}, S_{K,j}^{\mathbf{P}} \subseteq S_K^{\mathbf{P}}, S_{K,i}^{\mathbf{P}} \cap S_{K,j}^{\mathbf{P}} = \emptyset} \mathrm{B}(\mathbf{P}, S_{K,i}^{\mathbf{P}}) \mathrm{B}(\mathbf{P}, S_{K,j}^{\mathbf{P}})}, \tag{28}$$

where $S_{K,i}^{\mathbf{P}}$ is the ith possible subset of $S_K^{\mathbf{P}}$, whereas $\subseteq$ and $\cap$ are the symbols in the set theory representing contained in and intersection, respectively, and $\emptyset$ represents the empty set. A decision is made by assigning the query protein $\mathbf{P}$ to subset S_μ or the μth subcellular location with which the belief or credibility function of Eq. (28) has the maximum value; that is,

$$\mu = \mathbf{arg\ max}_m \{\mathrm{B}(\mathbf{P}, S_m, K)\}, \quad (m = 1, 2, \ldots, M), \tag{29}$$

where μ is the argument of m that maximizes the belief function $\mathrm{B}(\mathrm{P}, S_m, K)$. If there are more than one argument leading to a same maximum value for $\mathrm{B}(\mathrm{P}, S_m, K)$, then the query protein will be randomly assigned to one of the subcellular locations associated with these arguments, although this kind of tie case rarely happens.

Ensemble classifier

As mentioned above, the PseAA composition discrete model contains a parameter λ, which is associated with the number of components in a protein representation Eq. (3). In general, the larger the λ, the more components the PseAA composition contains and, hence, the more information the representation bears. However, λ must be smaller than the number of amino acids in the protein concerned (cf. Eq. (3) and Fig. 2). Also, it will reduce the cluster-tolerant capacity [76] if the PseAA composition contains too many components so as to lower the success rate of cross-validation. Accordingly, for a given training dataset, there is an optimal number for λ. It would be time-consuming and tedious to find the optimal λ by changing its value and doing tests one at a time.

Likewise, the KNN classifier Eq. (24) or the OET-KNN classifier Eq. (28) also contains a parameter K, that is, the number of the nearest neighbors to a query protein. It will affect the predicted result by choosing a different value for K. In other words, for a given training dataset, there is an optimal value for K as well.

Parameters such as λ and K are called uncertain parameters. The number of uncertain parameters depends on which model is used to represent the protein samples and what classifier is used for the prediction engine, as summarized in Table 2. It can be seen from the table that no uncertain parameter needs to be considered if using the CD classifier to predict protein subcellular location based on the AA composition discrete model; that one uncertain parameter, λ, needs to be determined if using the CD classifier based on the PseAA composition discrete model or using the KNN (or OET-KNN) classifier based on the FunD (or GO) model; and that two uncertain parameters, λ and K, need to be determined if using the KNN (or OET-KNN) classifier based on the PseAA composition discrete model. It would be much more tedious and time-consuming to determine the optimal values for two uncertain parameters. To solve the problem, let us introduce a two-dimensional fusion approach. Once the problem with two uncertain parameters has been solved, the one with one uncertain parameter will be automatically solved. In general, the shortest protein sequence considered is 50 amino acids [43]; hence, we can set the maximum value for λ at 49 Eq. (3). Also, for most datasets, when $K > 20$, the success rate by the KNN (or OET-KNN) classifier would decrease markedly. Therefore, the basic individual classifiers to be considered can be generally expressed by

$$\mathbb{C}(\lambda, K), (\lambda = 0, 1, 2, \ldots, 49; \quad K = 1, 2, \ldots, 20). \tag{30}$$

Now let us introduce an ensemble classifier $\mathbb{C}^{\mathrm{E}}$, which is formed by fusing $50 \times 20 = 1000$ basic individual classifiers of Eq. (30) as formulated by

$$\mathbb{C}^{\mathrm{E}} = \mathbb{C}(0,1) \forall \mathbb{C}(0,2) \forall \cdots \forall \mathbb{C}(49,19) \forall \mathbb{C}(49,20), \tag{31}$$

where the symbol $\forall$ denotes the fusing operator. A flowchart to illustrate how these individual classifiers are fused

Table 2
Uncertain parameters with respect to representation model and prediction engine

Representation model	Prediction engine	Uncertain parameter
AA composition	CD	None
PseAA composition	CD	λ
	KNN	λ and K
	OET-KNN	λ and K
FunD	KNN	K
	OET-KNN	K
GO	KNN	K
	OET-KNN	K

into the ensemble classifier $\mathbb{C}^{\mathrm{E}}$ is given in Fig. 3, and the corresponding mathematical equations are as follows. Suppose that the predicted classification results for a query protein **P** by each of the 1000 individual classifiers in Eq. (30) are

$$\mathbb{C}(\lambda, K) \triangleright \mathbf{P} = C_{\lambda,K}(\mathbf{P}) \in \mathbb{S} \quad (\lambda = 0, 1, 2, \ldots, 49; \; K = 1, 2, \ldots, 20), \tag{32}$$

where the symbol $\triangleright$ is an action operator meaning using $\mathbb{C}(\lambda, K)$ to classify **P**, $\mathbb{S}$ is defined by Eq. (12) (representing the union of the M subsets defined by the M subcellular locations), and the voting score for the protein **P** belonging to the mth subset $S_m \in \mathbb{S}$ is defined by

$$Y_m(\mathbf{P}) = \sum_{\lambda=0}^{49} \sum_{K=1}^{20} w_{\lambda,K} \Delta[C_{\lambda,K}(\mathbf{P}), S_m], \quad (m = 1, 2, \ldots, M), \tag{33}$$

where $w_{\lambda,K}$ is the weight and was set at 1 for simplicity. The delta function in Eq. (33) is given by

$$\Delta[C_{\lambda,K}(\mathbf{P}), S_m] = \begin{cases} 1, & \text{if } C_{\lambda,K}(\mathbf{P}) \in S_m \\ 0, & \text{otherwise} \end{cases}; \tag{34}$$

thus, the query protein **P** is predicted to belong to the subset (subcellular location) with which its score of Eq. (33) is the highest; that is, the query protein **P** is classified as the μth subcellular location if

$$\mu = \mathbf{arg\ max}_m\{Y_m(\mathbf{P})\}, \quad (m = 1, 2, \ldots, M), \tag{35}$$

where μ is the argument of m that maximizes the score function Y_m of Eq. (33). If there are more than one argument leading to a same maximum value, then the query protein will be randomly assigned to one of the subcellular locations associated with these arguments, although this kind of tie case rarely happens.

Modifications to incorporate multiple sites

All of the above algorithms were developed based on the assumption that each of the proteins concerned is located at one, and only one, subcellular location. In actuality, proteins may simultaneously exist at, or move between, two or more different subcellular locations. For instance, according to the Swiss–Prot database (version 50.7, released September 2006), among the 33,925 eukaryotic protein entries that have experimentally observed subcellular location annotations, 2715 have multiple-location sites, meaning that approximately 8% bear the multiplex feature, and among the 6408 human protein entries that have experimentally observed subcellular location annotations, 973 (~15%) have multiple-location sites. Proteins with multiple locations or dynamic features of this kind are particularly interesting because they may have some very special biological functions intriguing to investigators in both basic research and drug discovery.

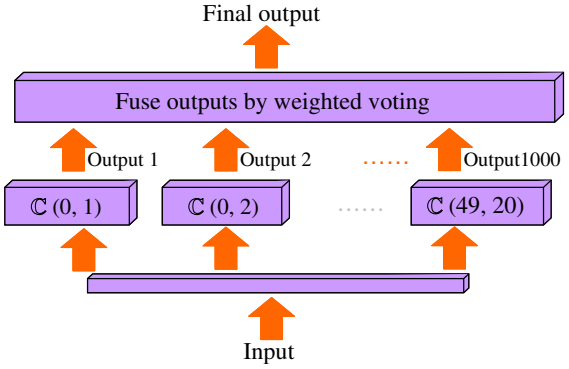


Fig. 3. Flowchart illustrating how the 1000 individual classifiers $C(\lambda,K)$ in Eq. (30) are fused into the ensemble classifier $\mathbb{C}^{\mathrm{E}}$ of Eq. (31) through a voting system. See text for further explanation.

Recently, two predictors were developed to deal with the system containing both single and multiple location proteins. One is for the human protein system [39], and the other is for the eukaryotic protein system [42].

To make the aforementioned classifiers also able to deal with the multiplex case, let us consider the following procedures. First, let us introduce the concept of "locative protein" according to the following identity. Given a same protein coexisting at two different subcellular locations, it will be counted as two locative proteins; if it coexists at three locations, it will be counted as three locative proteins, and so forth. Thus, the number of total locative proteins, $\tilde{N}$, can be expressed as

$$\tilde{N} = \tilde{n}_1 + \tilde{n}_2 + \cdots + \tilde{n}_m = \tilde{N}_1 + 2\tilde{N}_2 + \cdots + m\tilde{N}_m = \sum_{\tau=1}^{m} \tau \tilde{N}_\tau, \tag{36}$$

where m is the number of total subcellular locations; $\tilde{n}_1$ is the number of locative proteins in the first subcellular location, $\tilde{n}_2$ is the number of locative proteins in the second subcellular location, and so forth; $\tilde{N}_1$ is the number of proteins with one subcellular location, $\tilde{N}_2$ is the number of proteins with two subcellular locations, and so forth. Suppose that N is the total number of different proteins; it can be expressed by

$$N = \tilde{N}_1 + \tilde{N}_2 + \cdots + \tilde{N}_m = \sum_{\tau=1}^{m} \tilde{N}_\tau. \tag{37}$$

Subtracting Eq. (37) from Eq. (36), we obtain the relation between N and $\tilde{N}$ as given by

$$\tilde{N} = N + \tilde{N}_2 + \cdots + (m-1)\tilde{N}_m = N + \sum_{\tau=2}^{m} (\tau - 1)\tilde{N}_\tau. \tag{38}$$

Therefore, the number of locative proteins is greater than the number of different proteins (i.e., $\tilde{N} > N$); when, and only when, none of the proteins could exist in more than one subcellular location should we have $\tilde{N} = N$. Also, instead of Eq. (35), we should have

$$\{\mu\} = \mathbf{which}\{Y_{m2}(\mathbf{P}) \geqslant \max\,[Y_{m1}(\mathbf{P})] - \theta\}, \quad (m1 = 1, 2, \ldots, M; \; m2 = 1, 2, \ldots, M), \tag{39}$$

where "**which**" is a function often used in the R language (www.r-project.org) that returns the indexes satisfying the condition stated in the braces and θ is the threshold for the allowable deviation in determining the optimal score for $Y_{m2}(\mathbf{P})$. This means that when any subset, say $\mu 2$, has a score of $Y_{\mu 2}(\mathbf{P})$ that is within a deviation of θ from the highest score, say $Y_{\mu 1}(\mathbf{P})$ (i.e., $Y_{\mu 1}(\mathbf{P}) - Y_{\mu 2}(\mathbf{P}) \leqslant \theta$), the query protein $\mathbf{P}$ will be assigned to the subcellular location $\mu 2$ as well. Accordingly, rather than only a single index as in Eq. (36), $\{\mu\}$ in Eq. (39) may also represent one or more than one index corresponding to one or more than one subcellular location predicted. Also, because now the number of proteins predicted with multiple locations will depend on the value of θ (see Eq. (39)), $\mathbb{C}^{\mathrm{E}}$ of Eq. (31) should be converted to $\mathbb{C}^{\mathrm{E}}(\theta)$, that is,

$$\mathbb{C}^{\mathrm{E}} \Rightarrow \mathbb{C}^{\mathrm{E}}(\theta). \tag{40}$$

The larger the value of θ, the more the proteins will be predicted to have multiple locations. If θ is too large, this will lead to an overprediction of multiplex proteins; if θ is too small, it will lead to an underprediction. Enlightened by the approach to predicting the HIV protease cleavage sites in proteins [85,86], the optimal value for the threshold θ can be derived as follows.

Because the score functions, $Y_m(\mathbf{P})$, generated by the fusion classifier for different m are integers (see Eqs. (33) and (34)), the θ can also be reduced to the scope of nonnegative integers. Thus, we can assign θ in Eq. (40) with a set of consecutive integers, such as 0, 1, 2, 3, and 4 and find the optimal value for θ through the following procedure. Suppose that the predicted subcellular locations for a query protein $\mathbf{P}$ by the ensemble classifier $\mathbb{C}^{\mathrm{E}}(\theta)$ form the set $\mathbb{E}(\mathbf{P},\theta)$, that is,

$$\mathbb{E}(\mathbf{P},\theta) = \{E_1(\mathbf{P},\theta), E_2(\mathbf{P},\theta), \ldots, E_{m(\mathbf{P},\theta)}(\mathbf{P},\theta)\} \in \mathbb{S}, \tag{41}$$

where $E_1(\mathbf{P},\theta), E_2(\mathbf{P},\theta), \ldots, E_{m(\mathbf{P},\theta)}(\mathbf{P},\theta)$ represent $m(\mathbf{P},\theta)$ different subcellular locations predicted by $\mathbb{C}^{\mathrm{E}}(\theta)$ on $\mathbf{P}$, whereas the true subcellular locations to which the protein $\mathbf{P}$ belongs form the set $\mathbb{R}(\mathbf{P})$, that is,

$$\mathbb{R}(\mathbf{P}) = \{R_1(\mathbf{P}), R_2(\mathbf{P}), \ldots, R_{n(\mathbf{P})}(\mathbf{P})\} \in \mathbb{S}, \tag{42}$$

where $R_1(\mathbf{P}), R_2(\mathbf{P}), \ldots, R_{n(\mathbf{P})}(\mathbf{P})$ represent $n(\mathbf{P})$ different subcellular locations observed by experiments on $\mathbf{P}$. Now let us define the following quality control function for an individual protein $\mathbf{P}$ in a given benchmark dataset S when it is predicted by $\mathbb{C}^{\mathrm{E}}(\theta)$:

$$Q(\mathbf{P},\theta) = H^*(\mathbf{P},\theta) - H^{\mathrm{over}}_{\mathrm{miss}}(\mathbf{P},\theta), \tag{43}$$

where $H^*(\mathbf{P},\theta)$ represents the number of successful hits and $H^{\mathrm{over}}_{\mathrm{miss}}(\mathbf{P},\theta)$ represents the number of miss-hits and over-hits in using $\mathbb{C}^{\mathrm{E}}(\theta)$ to predict the query protein $\mathbf{P}$, and they can be formulated as (see Eqs. (41) and (42) and Fig. 4)

$$\begin{cases} H^*(\mathbf{P},\theta) = \|\mathbb{R}(\mathbf{P}) \cap \mathbb{E}(\mathbf{P},\theta)\| \\ H^{\mathrm{over}}_{\mathrm{miss}}(\mathbf{P},\theta) = \|\mathbb{R}(\mathbf{P}) \cup \mathbb{E}(\mathbf{P},\theta)\| - \|\mathbb{R}(\mathbf{P}) \cap \mathbb{E}(\mathbf{P},\theta)\| \end{cases}, \tag{44}$$

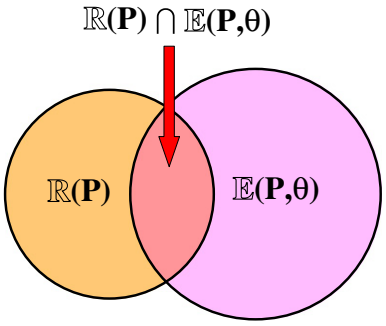


Fig. 4. Schematic drawing showing $\mathrm{R}(\mathbf{P}) \cap \mathrm{E}(\mathbf{P},\theta)$, the intersection region (pink) between the set $\mathrm{R}(\mathbf{P})$ (see Eq. (42)) and the set $\mathrm{E}(\mathbf{P},\theta)$ (see Eq. (41)), with the number of elements in such a region (i.e., $H^*(\mathbf{P},\theta) = \|\mathbb{R}(\mathbf{P}) \cap \mathbb{E}(\mathbf{P},\theta)\|$ [see Eq. (44)]) representing the number of successful hits obtained by the ensemble fusion classifier $\mathrm{C}(\theta)$, and showing $\mathrm{R}(\mathbf{P}) \cup \mathrm{E}(\mathbf{P},\theta) - \mathrm{R}(\mathbf{P}) \cap \mathrm{E}(\mathbf{P},\theta)$, the remaining region (orange or purple), with the number of elements (i.e., $H^{\mathrm{over}}_{\mathrm{miss}}(\mathbf{P},\theta) = \|\mathbb{R}(\mathbf{P}) \cup \mathbb{E}(\mathbf{P},\theta)\| - \|\mathbb{R}(\mathbf{P}) \cap \mathbb{E}(\mathbf{P},\theta)\|$ [see Eq. (44)]) representing the sum of the over-hits and miss-hits. See text for further explanation. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

where $\cup$ and $\cap$ represent the symbols of union and intersection, respectively, in the set theory, and the symbol $\|\ \|$ is an operator acting on the set therein to count the number of its elements. Hence, it follows (see Eqs. (41) and (42) and Fig. 4) that

$$\begin{cases} \|\mathbb{R}(\mathbf{P}) \cap \mathbb{E}(\mathbf{P},\theta)\| = \sum_{i=1}^{m(\mathbf{P},\theta)} \Delta_i[\mathbb{R}(\mathbf{P}), E_i(\mathbf{P},\theta)] \\ \|\mathbb{R}(\mathbf{P}) \cup \mathbb{E}(\mathbf{P},\theta)\| = n(\mathbf{P}) + m(\mathbf{P},\theta) - \sum_{i=1}^{m(\mathbf{P},\theta)} \Delta_i[\mathbb{R}(\mathbf{P}), E_i(\mathbf{P},\theta)] \end{cases}, \tag{45}$$

with the delta function given by

$$\Delta_i[\mathbb{R}(\mathbf{P}), E_i(\mathbf{P},\theta)] = \begin{cases} 1, & \text{if } E_i(\mathbf{P},\theta) \in \mathbb{R}(\mathbf{P}) \\ 0, & \text{if } E_i(\mathbf{P},\theta) \notin \mathbb{R}(\mathbf{P}) \end{cases}, \tag{46}$$

where $\notin$ is a symbol in the set theory meaning "not member of."

According to Eq. (43), the overall quality control function for $\mathbb{C}^{\mathrm{E}}(\theta)$ is given by

$$\mathbb{Q}(\theta) = \sum_{\mathbf{P} \in \mathbb{S}} Q(\mathbf{P},\theta) \tag{47}$$

and the optimal value for θ is given by

$$\theta^* = \mathbf{arg\ max}_\theta\{\mathbb{Q}(\theta)\}, \tag{48}$$

where $\mathbf{arg\ max}_\theta$ has the same meaning as that of Eq. (35), meaning that it takes the value of argument θ that maximizes $\mathbb{Q}(\theta)$ of Eq. (47). Thus, the ensemble classifier in Eq. (40) is uniquely defined as $\mathbb{C}^{\mathrm{E}}(\theta^*)$.

Testing methods

A statistical predictor usually contains two constituent parts; one is a benchmark dataset, and the other is a prediction algorithm (or engine). The benchmark dataset usually consists of a learning (or training) dataset and an independent testing dataset [41]. To test the quality of a predictor, the following examination procedures often are adopted.

Self-consistency examination

Before it can be tested, a statistical predictor needs to be trained. In the self-consistency examination, the predictor is trained and tested by using a same dataset; hence, such a test procedure is also called resubstitution examination. When applied to the current case, the subcellular location of each protein in the dataset is in turn identified using the rule parameters derived from the same dataset, the so-called learning dataset $\mathbb{S}^{\mathrm{L}}$. Because the same proteins are used to derive the rule parameters and to test themselves, the rule parameters must bear the information of the query protein later plugged back for testing. This kind of memory information certainly will lead to an overestimation of the success rate and, hence, cannot represent an objective estimation [48]. Yet the result obtained in this way can be used to check the self-consistency of a predictor, especially for its algorithm part. A predictor certainly cannot be deemed to be a good one if its self-consistency rate is poor. In other words, the resubstitution test is useful but not sufficient for evaluating a predictor effectively. To examine the quality of a predictor more rigorously and objectively, a cross-validation is absolutely needed, as is illustrated below.

Cross-validation examination

To check a predictor for its effectiveness in practical application, one should conduct a cross-validation by using an independent testing dataset $\mathbb{S}^{\mathrm{T}}$ in which none of the proteins occurs in the learning dataset $\mathbb{S}^{\mathrm{L}}$, as can be formulated by

$$\begin{cases} \mathbb{S}^{\mathrm{L}} \cup \mathbb{S}^{\mathrm{T}} = \mathbb{S} \\ \mathbb{S}^{\mathrm{L}} \cap \mathbb{S}^{\mathrm{T}} = \emptyset \end{cases}, \tag{49}$$

where the symbol $\emptyset$ represents the empty set. In general, the following three different kinds of cross-validation examinations often are used in the literature.

Single independent dataset examination

Although none of the proteins to be tested in $\mathbb{S}^{\mathrm{T}}$ occurs in $\mathbb{S}^{\mathrm{L}}$ used to train the predictor, the selection of proteins for the testing dataset $\mathbb{S}^{\mathrm{T}}$ could be quite arbitrary. This kind of arbitrariness may directly affect the conclusion. For instance, a predictor yielding a higher success rate than the other predictors for a testing dataset might fail to do so when applied to another testing dataset.

Subsampling examination

The benchmark dataset $\mathbb{S}$ is divided into a learning dataset $\mathbb{S}^{\mathrm{L}}$ and a testing dataset $\mathbb{S}^{\mathrm{T}}$ according to Eq. (49); the former is used to train a predictor, and the latter is used to test its accuracy. The practical procedure often used in the literature is the fivefold cross-validation, where the proteins in $\mathbb{S}$ are divided into five groups by splitting each of their subsets into five approximately equal-sized subgroups. Each of these five groups is in turn used for $\mathbb{S}^{\mathrm{T}}$ and the rest are used for $\mathbb{S}^{\mathrm{L}}$, thereby generating five different success rates, with their average representing the success rate by the fivefold cross-validation. The problem with the subsampling examination as such is that the number of possible selections in dividing the dataset $\mathbb{S}$ will be extremely large, as shown below. For a simplifying illustration, let us consider a hypothetical benchmark dataset consisting of 165 proteins classified into three subsets, with subset 1 containing 50 proteins, subset 2 containing 60 proteins, and subset 3 containing 55 proteins. For such a benchmark dataset, the number of possible fivefold divisions will be

$$\begin{aligned} \Pi &= \Pi_1 \cdot \Pi_2 \cdot \Pi_3 \\ &= \frac{50!}{(50-10)!10!} \cdot \frac{60!}{(60-12)!12!} \cdot \frac{55!}{(55-11)!11!} \\ &> 1.71 \times 10^{33}, \end{aligned} \tag{50}$$

where Π_1 is the number of possible different ways of splitting 50 proteins of subset 1 into five equal-sized groups, Π_2 is the number of possible different ways of splitting 60 proteins of subset 2, and Π_3 is the number of possible different ways of splitting 55 proteins of subset 3. It can be seen from Eq. (50) that even for such a simple and small dataset, the number of possible fivefold divisions is astronomical. The actual benchmark datasets for subcellular location prediction are much more complicated in composition and large in size; hence, the number of possible divisions will be even larger. Accordingly, any practical results from subsampling tests as such represent those derived from only a tiny fraction of the possibilities and, hence, cannot avoid arbitrariness.

Jackknife examination

Each of the protein samples in the benchmark dataset is in turn singled out as a tested protein, and the predictor is trained by the remaining proteins. Therefore, the examination is also termed the "leave one out" test, where the subcellular location of each protein is identified by the rule parameters derived using all of the other proteins except the one that is being identified. During the jackknifing process, both the training dataset and the testing dataset are actually open, and a protein will in turn move from one dataset to the other. The jackknife test can exclude the memory effects that exist in the resubstitution test, and the results obtained in this way always are unique for a given benchmark dataset.

Therefore, of the above examination methods, the jackknife test is deemed to be the most objective [41] and has been used increasingly by investigators to examine the accuracy of various predictors (see, e.g., Refs. [22,23,26,31,33,36,37,54,61–67,87–91]).

Web servers

The purpose of developing the methods for predicting protein subcellular localization was to get timely desired information that would otherwise be difficult or costly to acquire by experimental approaches. To maximize their use, some of these methods have been put on the World

Wide Web and are freely accessible to the public. Listed in Table 3 are the names and Web site addresses of the eight Web servers developed recently for various organisms and their coverage scopes. Two of the eight Web servers, Hum-mPLoc and Euk-mPLoc, can be used to deal with the proteins with multiple location sites as well.

To help users to use these Web servers effectively, let us take Euk-mPLoc as an example for illustration. It was specialized for predicting the subcellular locations of eukaryotic proteins, including those with multiple sites. Shown in Fig. 5 is the top page of the Web server. By clicking the relevant button, one can browse the desired information. For example, clicking the "Read Me" button will produce a screen showing the "Caveat" in using the predictor and its coverage scope. The current version of Euk-mPLoc can cover 22 subcellular locations, as shown in Fig. 1. By clicking "Citation," one can find the relevant articles that document the detailed development of Euk-mPLoc. By clicking "Data." one can find the benchmark dataset. By clicking "Download," one can download the results predicted by Euk-mPLoc for all of the eukaryotic protein entries in the Swiss–Prot database that do not have subcellular location annotations or are annotated as being uncertain.

The central empty window in Fig. 5 is for users to type or paste the sequence of a query protein for prediction. The sequence should be in Fasta format, as can be seen by clicking "Example." For speeding up the computation and getting a more accurate predicted result, it is important to enter the exact accession number right above the protein sequence according to the Fasta format. The accession number must be preceded by the symbol >; must have a first letter of A, O, P, or Q; and must be six characters long. An accession number is assigned to each protein sequence once it is included into UniProtKB database, which is composed of Swiss-Prot database and TrEMBL database and hosts 4,949,164 protein sequences according to version

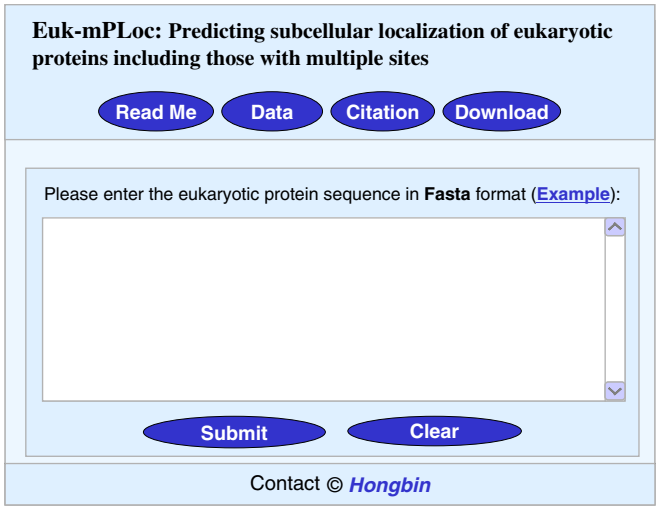


Fig. 5. Illustration showing the top page of the Web server Euk-mPLoc at http://202.120.37.186/bioinf/hum-multi or http://chou.med.harvard.edu/bioinf/euk-multi. See text and Table 3 for further explanation.

12.0 of UniProtKB released on 24-Jul-2007. This means that if a query protein can be found in UniProtKB, it must be one of the 4,949,164 proteins, and users should input its sequence and its exact accession number into the input box for prediction.

In case no true accession numbers are available for some proteins, such as synthetic and hypothetical proteins [6], Euk-mPLoc can still be used to predict their subcellular locations based on their PseAA composition (cf. Eq. (3)). To make a prediction go through by the Web server under such circumstances, users can just add a dummy accession number PPPPPP right after the symbol > and above the query protein sequence.

Note that the large-scale results predicted by Euk-mPLoc will be updated in a timely manner to include new entries of eukaryotic proteins and reflect the continu-

Table 3
Some Web servers developed recently for predicting subcellular locations of proteins in various organisms

Predictor name	Web site address	Organism	Number of subcellular locations to be covered	Reference
Hum-PLoc	http://202.120.37.186/bioinf/hum or http://chou.med.harvard.edu/bioinf/hum	Human	12	[43]
Hum-mPLoc	http://202.120.37.186/bioinf/hum-multi or http://chou.med.harvard.edu/bioinf/hum-multi	Human	14 (including those for proteins with multiple sites)	[39]
Plant-PLoc	http://202.120.37.186/bioinf/plant or http://chou.med.harvard.edu/bioinf/plant	Plant	11	[38]
Euk-OET-PLoc	http://202.120.37.186/bioinf/euk-oet or http://chou.med.harvard.edu/bioinf/euk-oet	Eukaryotic	16	[44]
Euk-mPLoc	http://202.120.37.186/bioinf/euk-multi or http://chou.med.harvard.edu/bioinf/euk-multi	Eukaryotic	22 (including those for proteins with multiple sites)	[42]
Gneg-PLoc	http://202.120.37.186/bioinf/Gneg or http://chou.med.harvard.edu/bioinf/Gneg	Gram negative	8	[93]
Gpos-PLoc	http://202.120.37.186/bioinf/Gpos or http://chou.med.harvard.edu/bioinf/Gpos	Gram positive	5	[73]
Virus-PLoc	http://202.120.37.186/bioinf/virus or http://chou.med.harvard.edu/bioinf/virus	Virus	7	[94]

ous development of Euk-mPLoc. The same is true for the other seven Web servers in Table 3. For users' convenience, these web-servers have been recently collected into a web-server package called Cell-PLoc at http://chou.med.harvard.edu/bioinf/Cell-PLoc/, which was designed for predicting subcellular localization of proteins in various different organisms and is freely accessible to the public.

Conclusions and perspectives

Significant progress has been achieved during the past 15 years in predicting protein subcellular localization, as reflected by the following aspects.

Benchmark datasets

The scope of coverage has been enlarged from the original 2 to 5 sites to 22 sites. The quality has also been improved by tightening the cutoff threshold to winnow those sequences that have equal to or higher than 25% sequence identity to any other in a same subset. Meanwhile, various organism-specific benchmark datasets have been established. Also, for developing predictors that can be used to deal with proteins with multiple subcellular locations, benchmark datasets, including multiplex proteins, have been constructed.

Sample representation

Compared with the early stage, when the sample of a protein was defined in a 20-D AA composition space and, hence, all of the sequence order information was lost, the $(20 + \lambda)$-D PseAA composition space and the GO space have been developed to represent protein samples. The essential advantage of using the GO space is that proteins defined in the GO space can be clustered in a way that better reflects their subcellular locations, thereby significantly enhancing the success rate of prediction.

Prediction engine

Instead of a single prediction algorithm, the hybridization approach of combining two different algorithms and the ensemble classifier formed by fusing many basic individual classifiers have been developed, significantly enhancing the power of the prediction engine.

Web servers

To support people working in the relevant areas, various user-friendly Web servers freely accessible to the public have been established and are particularly useful for those experimental scientists who use the predictors as a tool without the need to understand the detailed mathematics. In addition, the eight Web servers listed in Table 3 have a common feature in that they all were established based on the GO-PseAA hybridization model. When the query proteins have the corresponding GO numbers (as most do, as shown by the statistical data in Table 1), the success rates predicted by these Web servers will be overwhelmingly higher than those predicted by the developed methods not based on the GO space. When some of the query proteins do not have the corresponding GO numbers, such as synthesized or hypothetical protein sequences [6], their subcellular localization will be identified automatically by the state-of-the-art prediction engine based on the PseAA composition model, and the success rates obtained in this way are at least comparable to those obtained by the best of the other "ab initio" sequence-based methods [92] developed with very powerful algorithms such as CD, support vector machine (SVM), and neural network (NN). Accordingly, the overall success rates of the Web servers based on the GO-PseAA hybridization model are generally much higher; hence, these Web servers are particularly useful to experimental scientists for getting more reliable results easily and quickly. For the convenience of experimental scientists, at most of these Web sites a large-scale downloadable file for the organism concerned is provided to list the subcellular locations for all of the protein entries in the Swiss–Prot database that do not have subcellular location annotations or are annotated as being uncertain. These large-scale files will be updated in a timely manner to include new protein entries and reflect the continuous development of these Web server predictors.

As shown in Table 3, only two predictors, Hum-PLoc and Euk-mPLoc, can be used to deal with both single and multiple location proteins. Of the human proteins with experimental location annotations, 15% are those with multiple locations [39], and of the eukaryotic proteins with experimental location annotations, 8% are those with multiple locations [42]. For the proteins in other organisms, to this point such percentages are still less than 5%. It is anticipated that with more experimental annotation data available in the future, the benchmark datasets will be improved in both coverage scope and quality, further stimulating the development of protein subcellular location prediction.

Acknowledgment

We express our gratitude to the two anonymous reviewers, whose constructive comments were very helpful in strengthening the presentation of this review article.

References

[1] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, P. Walter, Molecular Biology of the Cell, fourth ed., Garland, New York, 2002.
[2] B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, J.D. Watson, Molecular Biology of the Cell, Garland, New York, 1994.
[3] H. Lodish, D. Baltimore, A. Berk, S.L. Zipursky, P. Matsudaira, J. Darnell, Molecular Cell Biology, Scientific American Books, New York, 1995.
[4] M. Ashburner, C.A. Ball, J.A. Blake, D. Botstein, H. Butler, J.M. Cherry, A.P. Davis, K. Dolinski, S.S. Dwight, J.T. Eppig, M.A. Harris,

D.P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J.C. Matese, J.E. Richardson, M. Ringwald, G.M. Rubin, G. Sherlock, Gene ontology: Tool for the unification of biology, Nat. Genet. 25 (2000) 25–29.
[5] K.C. Chou, Review: Structural bioinformatics and its impact to biomedical science, Curr. Med. Chem. 11 (2004) 2105–2134.
[6] G. Lubec, L. Afjehi-Sadat, J.W. Yang, J.P. John, Searching for hypothetical proteins: Theory and practice based upon original data and literature, Prog. Neurobiol. 77 (2005) 90–127.
[7] K. Nakai, M. Kanehisa, A knowledge base for predicting protein localization sites in eukaryotic cells, Genomics 14 (1992) 897–911.
[8] H. Nakashima, K. Nishikawa, Discrimination of intracellular and extracellular proteins using amino acid composition and residue-pair frequencies, J. Mol. Biol. 238 (1994) 54–61.
[9] J. Cedano, P. Aloy, J.A. P'erez-Pons, E. Querol, Relation between amino acid composition and cellular location of proteins, J. Mol. Biol. 266 (1997) 594–600.
[10] K. Nakai, P. Horton, PSORT: A program for detecting sorting signals in proteins and predicting their subcellular localization, Trends Biochem. Sci. 24 (1999) 34–36.
[11] A. Reinhardt, T. Hubbard, Using neural networks for prediction of the subcellular location of proteins, Nucleic Acids Res. 26 (1998) 2230–2236.
[12] K.C. Chou, D.W. Elrod, Protein subcellular location prediction, Protein Eng. 12 (1999) 107–118.
[13] Z. Yuan, Prediction of protein subcellular locations using Markov chain models, FEBS Lett. 451 (1999) 23–26.
[14] K. Nakai, Protein sorting signals and prediction of subcellular localization, Adv. Protein Chem. 54 (2000) 277–344.
[15] R.F. Murphy, M.V. Boland, M. Velliste, Towards a systematics for protein subcellular location: Quantitative description of protein localization patterns and automated analysis of fluorescence microscope images, Proc. Intl. Conf. Intell. Syst. Mol. Biol. 8 (2000) 251–259.
[16] O. Emanuelsson, H. Nielsen, S. Brunak, G. von Heijne, Predicting subcellular localization of proteins based on their N-terminal amino acid sequence, J. Mol. Biol. 300 (2000) 1005–1016.
[17] Z.P. Feng, Prediction of the subcellular location of prokaryotic proteins based on a new representation of the amino acid composition, Biopolymers 58 (2001) 491–499.
[18] S. Hua, Z. Sun, Support vector machine approach for protein subcellular localization prediction, Bioinformatics 17 (2001) 721–728.
[19] Z.P. Feng, C.T. Zhang, Prediction of the subcellular location of prokaryotic proteins based on the hydrophobicity index of amino acids, Intl. J. Biol. Macromol. 28 (2001) 255–261.
[20] Z.P. Feng, An overview on predicting the subcellular location of a protein, In Silico Biol. 2 (2002) 291–303.
[21] K.C. Chou, Y.D. Cai, Using functional domain composition and support vector machines for prediction of protein subcellular location, J. Biol. Chem. 277 (2002) 45765–45769.
[22] G.P. Zhou, K. Doctor, Subcellular location prediction of apoptosis proteins, Proteins Struct. Funct. Genet. 50 (2003) 44–48.
[23] Y.X. Pan, Z.Z. Zhang, Z.M. Guo, G.Y. Feng, Z.D. Huang, L. He, Application of pseudo amino acid composition for predicting protein subcellular location: Stochastic signal processing approach, J. Protein Chem. 22 (2003) 395–402.
[24] K.J. Park, M. Kanehisa, Prediction of protein subcellular locations by support vector machines using compositions of amino acid and amino acid pairs, Bioinformatics 19 (2003) 1656–1663.
[25] J.L. Gardy, C. Spencer, K. Wang, M. Ester, G.E. Tusnady, I. Simon, S. Hua, K. deFays, C. Lambert, K. Nakai, F.S. Brinkman, PSORT-B: Improving protein subcellular localization prediction for gram-negative bacteria, Nucleic Acids Res. 31 (2003) 3613–3617.
[26] Y. Huang, Y. Li, Prediction of protein subcellular locations using fuzzy k-NN method, Bioinformatics 20 (2004) 21–28.
[27] X. Xiao, S. Shao, Y. Ding, Z. Huang, Y. Huang, K.C. Chou, Using complexity measure factor to predict protein subcellular location, Amino Acids 28 (2005) 57–61.
[28] Z. Lei, Y. Dai, An SVM-based system for predicting protein subnuclear localizations, BMC Bioinformatics 6 (2005) 291.
[29] A. Garg, M. Bhasin, G.P. Raghava, Support vector machine-based method for subcellular localization of human proteins using amino acid compositions, their order, and similarity search, J. Biol. Chem. 280 (2005) 14427–14432.
[30] S. Matsuda, J.P. Vert, H. Saigo, N. Ueda, H. Toh, T. Akutsu, A novel representation of protein sequences for prediction of subcellular location using support vector machines, Protein Sci. 14 (2005) 2804–2813.
[31] Q.B. Gao, Z.Z. Wang, C. Yan, Y.H. Du, Prediction of protein subcellular location using a combined feature of sequence, FEBS Lett. 579 (2005) 3444–3448.
[32] K.C. Chou, H.B. Shen, Predicting protein subcellular location by fusing multiple classifiers, J. Cell. Biochem. 99 (2006) 517–527.
[33] J. Guo, Y. Lin, X. Liu, GNBSL: A new integrative system to predict the subcellular location for gram-negative bacteria proteins, Proteomics 6 (2006) 5099–5105.
[34] A. Hoglund, P. Donnes, T. Blum, H.W. Adolph, O. Kohlbacher, MultiLoc: Prediction of protein subcellular localization using N-terminal targeting sequences, sequence motifs, and amino acid composition, Bioinformatics 22 (2006) 1158–1165.
[35] K. Lee, D.W. Kim, D. Na, K.H. Lee, D. Lee, PLPD: Reliable protein localization prediction from imbalanced and overlapped datasets, Nucleic Acids Res. 34 (2006) 4655–4666.
[36] Z.H. Zhang, Z.H. Wang, Z.R. Zhang, Y.X. Wang, A novel method for apoptosis protein subcellular localization prediction combining encoding based on grouped weight and support vector machine, FEBS Lett. 580 (2006) 6169–6174.
[37] J.Y. Shi, S.W. Zhang, Q. Pan, Y.-M. Cheng, J. Xie, Prediction of protein subcellular localization by support vector machines using multi-scale energy and pseudo amino acid composition, Amino Acids 33 (2007) 69–74.
[38] K.C. Chou, H.B. Shen, Large-scale plant protein subcellular location prediction, J. Cell. Biochem. 100 (2007) 665–678.
[39] H.B. Shen, K.C. Chou, Hum–mPLoc: An ensemble classifier for large-scale human protein subcellular location prediction by incorporating samples with multiple sites, Biochem. Biophys. Res. Commun. 355 (2007) 1006–1011.
[40] E. Glory, R.F. Murphy, Automated subcellular location determination and high-throughput microscopy, Dev. Cell 12 (2007) 7–16.
[41] K.C. Chou, C.T. Zhang, Review: Prediction of protein structural classes, Crit. Rev. Biochem. Mol. Biol. 30 (1995) 275–349.
[42] K.C. Chou, H.B. Shen, Euk-mPLoc: A fusion classifier for large-scale eukaryotic protein subcellular location prediction by incorporating multiple sites, J. Proteome Res. 6 (2007) 1728–1734.
[43] K.C. Chou, H.B. Shen, Hum-PLoc: A novel ensemble classifier for predicting human protein subcellular localization, Biochem. Biophys. Res. Commun. 347 (2006) 150–157.
[44] K.C. Chou, H.B. Shen, Predicting eukaryotic protein subcellular location by fusing optimized evidence-theoretic K-nearest neighbor classifiers, J. Proteome Res. 5 (2006) 1888–1897.
[45] S.F. Altschul, Evaluating the statistical significance of multiple distinct local alignments, in: S. Suhai (Ed.), Theoretical and Computational Methods in Genome Research, Plenum, New York, 1997, pp. 1–14.
[46] J.C. Wootton, S. Federhen, Statistics of local complexity in amino acid sequences and sequence databases, Comput. Chem. 17 (1993) 149–163.
[47] H. Nakashima, K. Nishikawa, T. Ooi, The folding type of a protein is relevant to the amino acid composition, J. Biochem. 99 (1986) 152–162.
[48] K.C. Chou, A novel approach to predicting protein structural classes in a (20–1)-D amino acid composition space, Proteins Struct. Funct. Genet. 21 (1995) 319–344.
[49] P. Klein, C. Delisi, Prediction of protein structural class from amino acid sequence, Biopolymers 25 (1986) 1659–1672.
[50] P. Klein, Prediction of protein structural class by discriminant analysis, Biochim. Biophys. Acta 874 (1986) 205–215.
[51] P.Y. Chou, Prediction of protein structural classes from amino acid composition, in: G.D. Fasman (Ed.), Prediction of Protein Structure

and the Principles of Protein Conformation, Plenum, New York, pp. 549–586.
[52] B.A. Metfessel, P.N. Saurugger, D.P. Connelly, S.T. Rich, Cross-validation of protein structural class prediction using statistical clustering and neural networks, Protein Sci. 2 (1993) 1171–1182.
[53] K.C. Chou, C.T. Zhang, Predicting protein folding types by distance functions that make allowances for amino acid interactions, J. Biol. Chem. 269 (1994) 22014–22020.
[54] G.P. Zhou, An intriguing controversy over protein structural class prediction, J. Protein Chem. 17 (1998) 729–738.
[55] G.P. Zhou, N. Assa-Munt, Some insights into protein structural class prediction, Proteins Struct. Funct. Genet. 44 (2001) 57–59.
[56] K.C. Chou, Prediction of protein cellular attributes using pseudo amino acid composition, Proteins Struct. Funct. Genet. 43 (2001) 246–255 (Erratum: 2001, vol. 44, p. 60).
[57] C. Tanford, Contribution of hydrophobic interactions to the stability of the globular conformation of proteins, J. Am. Chem. Soc. 84 (1962) 4240–4274.
[58] T.P. Hopp, K.R. Woods, Prediction of protein antigenic determinants from amino acid sequences, Proc. Natl. Acad. Sci. USA 78 (1981) 3824–3828.
[59] K.C. Chou, Prediction of protein subcellular locations by incorporating quasi-sequence-order effect, Biochem. Biophys. Res. Commun. 278 (2000) 477–483.
[60] K.C. Chou, Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes, Bioinformatics 21 (2005) 10–19.
[61] C. Chen, X. Zhou, Y. Tian, X. Zou, P. Cai, Predicting protein structural class with pseudo-amino acid composition and support vector machine fusion network, Anal. Biochem. 357 (2006) 116–121.
[62] C. Chen, Y.X. Tian, X.Y. Zou, P.X. Cai, J.Y. Mo, Using pseudo-amino acid composition and support vector machine to predict protein structural class, J. Theor. Biol. 243 (2006) 444–448.
[63] S.W. Zhang, Q. Pan, H.C. Zhang, Z.C. Shao, J.Y. Shi, Prediction protein homo-oligomer types by pseudo amino acid composition: Approached with an improved feature extraction and naive Bayes feature fusion, Amino Acids 30 (2006) 461–468.
[64] P. Du, Y. Li, Prediction of protein submitochondria locations by hybridizing pseudo-amino acid composition with various physicochemical features of segmented sequence, BMC Bioinformatics 7 (2006) 518.
[65] S. Mondal, R. Bhavna, R. Mohan Babu, S. Ramakumar, Pseudo amino acid composition and multi-class support vector machines approach for conotoxin superfamily classification, J. Theor. Biol. 243 (2006) 252–260.
[66] H. Lin, Q.Z. Li, Predicting conotoxin superfamily and family by using pseudo amino acid composition and modified Mahalanobis discriminant, Biochem. Biophys. Res. Commun. 354 (2007) 548–551.
[67] H. Lin, Q.Z. Li, Using pseudo amino acid composition to predict protein structural class: Approached by incorporating 400 dipeptide components, J. Comput. Chem. 28 (2007) 1463–1466.
[68] Y.D. Cai, G.P. Zhou, K.C. Chou, Support vector machines for predicting membrane protein types by using functional domain composition, Biophys. J. 84 (2003) 3257–3263.
[69] R. Apweiler, T.K. Attwood, A. Bairoch, A. Bateman, E. Birney, M. Biswas, P. Bucher, L. Cerutti, F. Corpet, M.D.R. Croning, R. Durbin, L. Falquet, W. Fleischmann, L. Gouzy, H. Hermjakob, N. Hulo, I. Jonassen, D. Kahn, A. Kanapin, Y. Karavidopoulou, R. Lopez, B. Marx, N.J. Mulder, T.M. Oinn, M. Pagni, F. Servant, C.J.A. Sigrist, E.M. Zdobnov, The InterPro database, an integrated documentation resource for protein families, domains, and functional sites, Nucleic Acids Res. 29 (2001) 37–40.
[70] K.C. Chou, Y.D. Cai, Predicting protein structural class by functional domain composition, Biochem. Biophys. Res. Commun. 321 (2004) 1007–1009 (Corrigendum: 2005, vol. 329, p. 1362).
[71] K.C. Chou, Y.D. Cai, Prediction of protein subcellular locations by GO–FunD–PseAA predictor, Biochem. Biophys. Res. Commun. 320 (2004) 1236–1239.
[72] R. Apweiler, A. Bairoch, C.H. Wu, W.C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, M.J. Martin, D.A. Natale, C. O'Donovan, N. Redaschi, L.S. Yeh, UniProt: The Universal Protein Knowledgebase, Nucleic Acids Res. 32 (2004) D115–D119.
[73] H.B. Shen, K.C. Chou, Gpos-PLoc: An ensemble classifier for predicting subcellular localization of gram-positive bacterial proteins, Protein Eng. Design Select. 20 (2007) 39–46.
[74] Y.D. Cai, K.C. Chou, Nearest neighbour algorithm for predicting protein subcellular location by combining functional domain composition and pseudo-amino acid composition, Biochem. Biophys. Res. Commun. 305 (2003) 407–411.
[75] K.C. Chou, Y.D. Cai, Predicting protein localization in budding yeast, Bioinformatics 21 (2005) 944–950.
[76] K.C. Chou, A key driving force in determination of protein structural classes, Biochem. Biophys. Res. Commun. 264 (1999) 216–224.
[77] P.C. Mahalanobis, On the generalized distance in statistics, Proc. Natl. Inst. Sci. India 2 (1936) 49–55.
[78] K.C.S. Pillai, Mahalanobis D2, in: S. Kotz, N.L. Johnson (Eds.), Encyclopedia of Statistical Sciences, Wiley, New York, 1985, pp. 176–181.
[79] T.M. Cover, P.E. Hart, Nearest neighbour pattern classification, IEEE Trans. Inform. Theory IT-13 (1967) 21–27.
[80] T. Denoeux, A k-nearest neighbor classification rule based on Dempster–Shafer theory, IEEE Trans. Syst. Man Cybernetics 25 (1995) 804–813.
[81] J.M. Keller, M.R. Gray, J.A. Givens, A fuzzy k-nearest neighbours algorithm, IEEE Trans. Syst. Man Cybernetics 15 (1985) 580–585.
[82] K.V. Mardia, J.T. Kent, J.M. Bibby, Multivariate Analysis, Academic Press, London, 1979.
[83] G. Shafer, A Mathematical Theory of Evidence, Princeton University Press, Princeton, NJ, 1976.
[84] L.M. Zouhal, T. Denoeux, An evidence-theoretic K-NN rule with parameter optimization, IEEE Trans. Syst. Man Cybernetics 28 (1998) 263–271.
[85] K.C. Chou, A vectorized sequence-coupling model for predicting HIV protease cleavage sites in proteins, J. Biol. Chem. 268 (1993) 16938–16948.
[86] K.C. Chou, Review: Prediction of HIV protease cleavage sites in proteins, Anal. Biochem. 233 (1996) 1–14.
[87] Y. Cao, S. Liu, L. Zhang, J. Qin, J. Wang, K. Tang, Prediction of protein structural class with Rough Sets, BMC Bioinformatics 7 (2006) 20.
[88] Q.B. Gao, Z.Z. Wang, Classification of G-protein coupled receptors at four levels, Protein Eng. Design Select. 19 (2006) 511–516.
[89] G.P. Zhou, Y.D. Cai, Predicting protease types by hybridizing gene ontology and pseudo amino acid composition, Proteins Struct. Funct. Bioinformatics 63 (2006) 681–684.
[90] K.C. Chou, H.B. Shen, Signal-CF: A subsite-coupled and window-fusing approach for predicting signal peptides, Biochem. Biophys. Res. Commun. 357 (2007) 633–640.
[91] K.C. Chou, H.B. Shen, MemType-2L: A Web server for predicting membrane proteins and their types by incorporating evolution information through Pse-PSSM, Biochem. Biophys. Res. Commun. 360 (2007) 339–345.
[92] O. Emanuelsson, S. Brunak, G. von Heijne, H. Nielsen, Locating proteins in the cell using TargetP, SignalP, and related tools, Nat. Protocols 2 (2007) 953–971.
[93] K.C. Chou, H.B. Shen, Large-scale predictions of gram-negative bacterial protein subcellular locations, J. Proteome Res. 5 (2006) 3420–3428.
[94] H.B. Shen, K.C. Chou, Virus-PLoc: A fusion classifier for predicting the subcellular localization of viral proteins within host and virus-infected cells, Biopolymers 85 (2007) 233–240.