Molecular BioSystems

Cite this: *Mol. BioSyst.*, 2012, **8**, 629–641

www.rsc.org/molecularbiosystems

PAPER

iLoc-Hum: using the accumulation-label scale to predict subcellular locations of human proteins with both single and multiple sites†

Kuo-Chen Chou,[a] Zhi-Cheng Wu[b] and Xuan Xiao[ab]

Received 10th October 2011, Accepted 4th November 2011
DOI: 10.1039/c1mb05420a

Although numerous efforts have been made for predicting the subcellular locations of proteins based on their sequence information, it still remains as a challenging problem, particularly when query proteins may have the multiplex character, *i.e.*, they simultaneously exist, or move between, two or more different subcellular location sites. Most of the existing methods were established on the assumption: a protein has one, and only one, subcellular location. Actually, recent evidence has indicated an increasing number of human proteins having multiple subcellular locations. This kind of multiplex proteins should not be ignored because they may bear some special biological functions worthy of our attention. Based on the accumulation-label scale, a new predictor, called iLoc-Hum, was developed for identifying the subcellular localization of human proteins with both single and multiple location sites. As a demonstration, the jackknife cross-validation was performed with iLoc-Hum on a benchmark dataset of human proteins that covers the following 14 location sites: centrosome, cytoplasm, cytoskeleton, endoplasmic reticulum, endosome, extracellular, Golgi apparatus, lysosome, microsome, mitochondrion, nucleus, peroxisome, plasma membrane, and synapse, where some proteins belong to two, three or four locations but none has 25% or higher pairwise sequence identity to any other in the same subset. For such a complicated and stringent system, the overall success rate achieved by iLoc-Hum was 76%, which is remarkably higher than that by any of the existing predictors that also have the capacity to deal with this kind of system. Further comparisons were also made *via* two independent datasets; all indicated that the success rates by iLoc-Hum were even more significantly higher than its counterparts. As a user-friendly web-server, iLoc-Hum is freely accessible to the public at http://icpr.jci.edu.cn/bioinfo/iLoc-Hum or http://www.jci-bioinfo.cn/iLoc-Hum. For the convenience of most experimental scientists, a step-by-step guide is provided on how to use the web-server to get the desired results by choosing either a straightforward submission or a batch submission, without the need to follow the complicated mathematical equations involved.

I. Introduction

In order to in-depth study cell biology, or to study life science at the molecular, cellular and systems levels, such as analyzing protein–protein interaction networks within a cell,[1,2] understanding the intricate pathways that regulate biological processes at the cellular level,[3,4] and revealing the mechanism of how the functions of a cell are carried out by the proteins therein,[5,6] the information of proteins subcellular localization is indispensable. It is also very important for identifying and prioritizing drug targets[7] during the process of drug development.

Although the knowledge of protein subcellular localization can be acquired by carrying out varieties of experiments, it is both time-consuming and expensive to completely rely on experimental observations alone. Nowadays, the speed of discovering new protein sequences has become increasingly fast. In 1986, for example, the Swiss-Prot[8] database only contained 3939 protein sequence entries, but the number now has jumped to 532 792 according to the release 2011_10 on 19-Oct-11 of UniProtKB/Swiss-Prot at http://www.expasy.org/sprot/relnotes/relstat.html; indicating that the number of protein sequence entries now is more than 135 times the number about 25 years ago.

With the avalanche of protein sequences generated in the post-genomic age, we are facing the challenge of developing automated methods for fast and effective identification of the

[a] *Gordon Life Science Institute, 13784 Torrey Del Mar Drive, San Diego, California 92130, USA. E-mail: kcchou@gordonlifescience.org*
[b] *Computer Department, Jing-De-Zhen Ceramic Institute, Jing-De-Zhen 333046, China. E-mail: xiaoxuan0326@yahoo.com.cn*

† Electronic supplementary information (ESI) available. See DOI: 10.1039/c1mb05420a

subcellular locations of the newly found proteins according to their sequence information alone.

Actually, many computational methods have been developed in this regard (see, *e.g.*, ref. 9–22 as well as a long list of references cited in two comprehensive review articles[23,24]).

However, for the practical application in drug development, it is more important and urgent to timely identify the subcellular locations of human proteins. Unfortunately, relatively much fewer predictors have been developed that are specialized for the identification of the subcellular localization of human proteins.

Among the existing predictors, the HSLPred developed by Garg *et al.*[25] is the one specialized for human proteins. However, it can only cover four subcellular location sites: cytoplasm, mitochondria, nucleus, and plasma membrane. If a query protein is located outside the four location sites, such as Golgi apparatus and peroxisome, the predictor would fail to work, or the results thus obtained would be meaningless. Obviously, the fewer location sites a predictor covers, the more limits it will have for practical applications.

To improve this kind of coverage limitation, the predictor called Hum-PLoc[26] was developed by extending the coverage scope for human proteins from the aforementioned four location sites to twelve, *i.e.*, by adding the following eight sites: centrosome, cytoskeleton, endoplasmic reticulum, extracellular, Golgi apparatus, lysosome, microsome, and peroxisome. However, Hum-PLoc[26] can only be used to deal with the single-location or "singleplex" proteins but not multiple-locations or "multiplex" proteins. The latter may simultaneously reside at, or move between, two or more different subcellular locations. Proteins with multiple location sites or dynamic feature of this kind are particularly interesting because they may have some unique biological functions worthy of our special notice.[4,7] Particularly, as pointed out by Millar *et al.*,[27] recent evidence has indicated that an increasing number of proteins have multiple locations in the cell.

To make Hum-PLoc be able to predict the multiplex protein locations as well, a predictor called Hum-mPLoc[28] and its updated and improved version called Hum-mPLoc 2.0[29] were developed, where the character "m" in front of "PLoc" stands for "multiple", meaning that it can be also used to deal with human proteins with multiple locations. Meanwhile, the scope covered by Hum-mPLoc 2.0 was further extended from 12 location sites to 14 by adding two more sites: endosome and synapse.

However, Hum-mPLoc 2.0 has the following shortcomings. (1) In formulating the protein samples, only the integer numbers 0 and 1 were used to reflect the GO (gene ontology) information.[30,31] Such an over-simplified expression might lose some important information and hence affect the prediction quality. (2) The number of subcellular location sites a protein may have was determined through an optimal threshold factor θ^* (see eqn (48) of ref. 24). Obviously, it would be more instructive or illuminative if we could find a different approach to determine this in a more natural and intuitive manner. (3) Although a web-server for Hum-mPLoc 2.0 has been established at http://www.csbio.sjtu.edu.cn/bioinf/hum-multi-2/, only one query protein sequence at a time is allowed for each input when using the web-server. It would be much more convenient for the users if such a rigid limit could be improved. (4) Particularly, for those users who need to identify the subcellular locations for a large number of uncharacterized human proteins, it is important for the web-server to have the function of supporting batch job submissions; unfortunately, Hum-mPLoc 2.0 lacks such a function.

The present study was initiated in an attempt to develop a new and more powerful predictor, called iLoc-Hum, for predicting human protein subcellular localization by addressing the above four problems.

According to a recent review,[32] to develop a useful predictor for identifying subcellular locations of proteins based on their sequence information, the following things usually need to be considered: (1) benchmark dataset construction or selection; (2) formulate the protein samples with an effective mathematical expression that can truly reflect the intrinsic correlation with their subcellular locations; (3) introduce or develop a powerful algorithm (or engine) to operate the prediction; (4) properly perform cross-validation tests to objectively evaluate the anticipated accuracy of the predictor; (5) establish a user-friendly web-server for the predictor that is accessible to the public. Below, let us elaborate how to deal with these procedures.

II. Materials

In this study, we adopted the benchmark dataset as constructed in ref. 29 for developing Hum-mPLoc 2.0. The reasons to do so are as follows. (1) The dataset was constructed specially for human proteins, which are the focus of this study. Also, it can cover 14 subcellular location sites; compared with the other datasets in this area that only covered four subcellular locations (see, *e.g.*, ref. 25), the coverage scope of the current dataset is much wider. (2) None of the proteins included in the dataset has $\geq 25\%$ pairwise sequence identity to any other in the same subcellular location; compared with most of the other benchmark datasets, such a cutoff threshold is much more stringent. For example, in constructing the benchmark dataset for developing HSLPred by Garg *et al.*[25] or MultiLoc by Hoglund *et al.*[20] for predicting protein subcellular localization, the corresponding cutoff threshold was set at 90% or 80%, meaning that their dataset would allow inclusion of those proteins that had up to 90% or 80% pairwise sequence identity to each other. Therefore, the benchmark dataset used in the current study is much more strict and harsh in excluding homology bias and redundancy. (3) It contains both singleplex and multiplex proteins and hence can be used to train and test a predictor developed with an aim to be able to deal with proteins with both single and multiple location sites. (4) The use of the current benchmark dataset will also make it more fair and convincing to compare the new predictor with the existing one because the success rates by the rigorous jackknife test for Hum-mPLoc 2.0 on the current benchmark dataset have been well documented and reported in a recent paper.[29]

To study a protein system with some proteins simultaneously occurring in two or more locations, it is instructive to introduce the concept of "locative protein" as briefed below. If a protein coexists at two different subcellular location sites, it will be counted as two locative proteins; if it coexists at three location

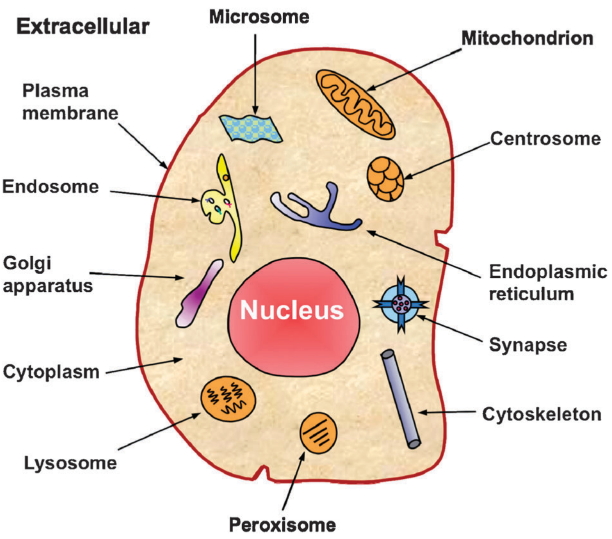


Fig. 1 Illustration to show the 14 subcellular locations of human proteins. The 14 locations are: (1) centrosome, (2) cytoplasm, (3) cytoskeleton, (4) endoplasmic reticulum, (5) endosome, (6) extracellular, (7) Golgi apparatus, (8) lysosome, (9) microsome, (10) mitochondrion, (11) nucleus, (12) peroxisome, (13) plasma membrane, and (14) synapse. Adapted from Analytical Biochemistry, 2009, 394, 269–274, with permission.

sites, it will be counted as three locative proteins; and so forth. Thus, the number of total locative proteins can be expressed as

$$N(\mathrm{loc}) = N(\mathrm{seq}) + \sum_{m=1}^{M} (m-1)N(m) \qquad (1)$$

where N(loc) is the number of total locative proteins, N(seq) the number of total different protein sequences, N(1) the number of proteins with one location, N(2) the number of proteins with two locations, and so forth; while M is the number of total subcellular location sites investigated. As we can see from eqn (1), the number of total locative proteins is generally greater than that of total different protein sequences. When, and only when, all the proteins have a single location site, can the two be the same.

The benchmark dataset used in this study covers 14 subcellular locations (Fig. 1) with a total of 3106 different human protein sequences, of which 2580 belong to one subcellular location, 480 to two locations, 43 to three locations, 3 to four locations, and none to five and more locations.[29] Substituting these data into eqn (1), we obtain

$$\begin{aligned} N(\mathrm{loc}) = {} & N(\mathrm{seq}) + (1-1)\times 2580 + (2-1)\times 480 + (3-1) \\ & \times 43 + (4-1)\times 3 + \textstyle\sum_{m=5}^{14}(m-1) \\ & \times 0 = 3106 + 480 + 86 = 3681 \end{aligned} \qquad (2)$$

which is fully consistent with the figures in Table 1 of ref. 29.

For readers' convenience, each of the protein sequences in the current benchmark dataset and its accession number are given in ESI† S1. Also, for the convenience of the mathematical description later, let us use $\mathbb{S}$ to denote the benchmark dataset, which can be formulated as

$$\mathbb{S} = \mathbb{S}_1 \cup \mathbb{S}_2 \cup \mathbb{S}_3 \cup \mathbb{S}_4 \cup \mathbb{S}_5 \cup \mathbb{S}_6 \cup \cdots \cup \mathbb{S}_{14} \qquad (3)$$

where $\mathbb{S}_1$ represents the subset for the subcellular location of "centrosome", $\mathbb{S}_2$ for "cytoplasm", $\mathbb{S}_3$ for "cytoskeleton", and so forth (*cf.* ESI† S1); while ∪ represents the symbol for "union" in the set theory. For convenience, hereafter let us just use the subscripts of eqn (3) as the codes of the 14 location sites; *i.e.*, "1" for "centrosome", "2" for "cytoplasm", "3" for "cytoskeleton", and so forth (Table 1).

Table 1 Success rates by the jackknife test for iLoc-Hum[a] on the benchmark dataset $\mathbb{S}$ given in the ESI S1. The dataset contains 3106 different human protein sequences covering 14 location sites where none of the proteins included has ≥ 25% pairwise sequence identity to any other in the same location. Of the 3106 proteins, 2580 belong to one subcellular location, 480 to two locations, 43 to three locations, and 3 to four locations

Subcellular location	Subset	Size	Sensitivity (%)	Specificity (%)	Accuracy (%)
Centrosome	$\mathbb{S}_1$	77	77.78	99.31	72.73
Cytoplasm	$\mathbb{S}_2$	817	66.08	88.70	68.67
Cytoskeleton	$\mathbb{S}_3$	79	52.94	98.30	34.18
Endoplasmic reticulum	$\mathbb{S}_4$	229	33.33	99.26	72.49
Endosome	$\mathbb{S}_5$	24	80.98	97.86	4.17
Extracellular	$\mathbb{S}_6$	385	90.53	97.82	84.42
Golgi apparatus	$\mathbb{S}_7$	161	74.44	97.91	61.49
Lysosome	$\mathbb{S}_8$	77	83.58	99.31	72.73
Microsome[b]	$\mathbb{S}_9$	24	77.78	99.45	29.17
Mitochondrion	$\mathbb{S}_{10}$	364	88.20	97.13	78.02
Nucleus	$\mathbb{S}_{11}$	1021	84.30	94.89	89.91
Peroxisome	$\mathbb{S}_{12}$	47	95.24	99.12	42.55
Plasma membrane	$\mathbb{S}_{13}$	354	81.47	97.22	78.25
Synapse	$\mathbb{S}_{14}$	22	80.00	99.68	54.55
Overall		3681[c]	79.30	94.99	**76.31**

[a] The parameter K for the KNN classifier in the current iLoc-Hum was 10, which was derived by optimizing the overall jackknife success rate obtained by iLoc-Hum on the benchmark dataset $\mathbb{S}$. [b] It should be pointed out that "microsomes" are artifactual vesicle/membrane particles formed during disruption of cells. They do not indicate any specific structure inside the living cells. [c] Note that instead of 3106 (the number of total different proteins), here we have 3681 (the number of total different locative proteins). This is because some proteins may have two or more location sites. See eqn (1) and (2) and the relevant text for more explanation about the concept of locative proteins.

III. Methods

To develop a powerful method for statistically predicting protein subcellular localization according to the sequence information, one of the most important things is to formulate the protein sequences with an effective mathematical expression that can truly reflect the intrinsic correlation with their subcellular localization.[32] However, it is by no means an easy job to realize this because this kind of correlation is usually deeply hidden or "buried" in piles of complicated sequences.

The most straightforward method to formulate the sample of a query protein **P** is just using its entire amino acid sequence, as can be generally described by

$$\mathbf{P} = \mathrm{R}_1\mathrm{R}_2\mathrm{R}_3\mathrm{R}_4\mathrm{R}_5\mathrm{R}_6\mathrm{R}_7\ldots\mathrm{R}_L \qquad (4)$$

where R_1 represents the 1st residue of the protein **P**, R_2 the 2nd residue, . . ., R_L the Lth residue, and each of them belongs to one of the 20 native amino acids. In order to identify its subcellular location(s), sequence-similarity-search-based tools, such as BLAST,[33,34] was utilized to search protein database for those proteins that have high sequence similarity to the query protein **P**. Subsequently, the subcellular location annotations of the proteins thus found were used to deduce the subcellular location(s) for **P**. Unfortunately, this kind of straightforward sequential model, although quite intuitive and able to contain the entire information of a protein sequence, failed to work when the query protein **P** did not have significant sequence similarity to any location-known protein.

Thus, various non-sequential or discrete models to formulate protein samples were proposed in hope of establishing some sort of correlation or cluster manner by which the prediction power can be enhanced.

Among the discrete models for a protein sample, the simplest one is its amino acid (AA) composition or AAC.[35] According to the AAC-discrete model, the protein **P** of eqn (4) can be formulated by[36]

$$\mathbf{P} = [f_1 \quad f_2 \quad \cdots \quad f_{20}]^{\mathbf{T}} \tag{5}$$

where f_i (i = 1, 2, …, 20) are the normalized occurrence frequencies of the 20 native amino acids in protein **P**, and **T** the transposing operator. Many methods for predicting protein subcellular localization were based on the AAC-discrete model (see, *e.g.*, ref. 9–12 and 37). However, as we can see from eqn (5), if the ACC model is used to represent the protein **P**, all its sequence-order effects would be lost, and hence the prediction quality might be considerably limited.

To avoid the complete loss of the sequence-order information, the pseudo amino acid composition (PseAAC) was proposed to represent the sample of a protein, as formulated by[38]

$$\mathbf{P} = [p_1 \quad p_2 \quad \ldots \quad p_{20} \quad p_{20+1} \quad \ldots \quad p_{20+\lambda}]^{\mathbf{T}} \tag{6}$$

where the first 20 elements are associated with the 20 elements in eqn (5) or the 20 amino acid components of the protein **P**, while the additional λ factors are used to incorporate some sequence-order information *via* a series of rank-different correlation factors along a protein chain.

According to ref. 32, the PseAAC for the protein **P** can be generally formulated as

$$\mathbf{P} = [\psi_1 \quad \psi_2 \quad \ldots \quad \psi_u \quad \ldots \quad \psi_\Omega]^{\mathbf{T}} \tag{7}$$

where the subscript Ω is an integer, and its value as well as the components ψ_1, ψ_2, … will depend on how the desired information is extracted from the amino acid sequence of **P** (*cf.* eqn (4)). As a general form, eqn (7) can cover various different modes of PseAAC. For example, when its elements are given by

$$\psi_u = \begin{cases} \dfrac{f_u}{\sum\limits_{i=1}^{20} f_i + w \sum\limits_{j=1}^{\lambda} \theta_j}, & (1 \le u \le 20) \\ \dfrac{w\theta_{u-20}}{\sum\limits_{i=1}^{20} f_i + w \sum\limits_{j=1}^{\lambda} \theta_j}, & (20 + 1 \le u \le 20 + \lambda = \Omega;\ \lambda < L) \end{cases} \tag{8}$$

we immediately obtain the formulation of PseAAC as originally introduced in ref. 38, where w is the weight factor for the sequence order effect, θ_j is the jth tier correlation factor reflecting the sequence order correlation between all the jth most contiguous residues along a protein chain, and λ is an integer parameter for the maximum number of correlation tiers to be considered. Readers can also find a brief description of eqn (8) as well as the definition for each of the symbols therein by clicking the link at http://en.wikipedia.org/wiki/Pseudo_amino_acid_composition to see a Wikipedia article about the pseudo amino acid composition.

Now let us consider how to use the general form of PseAAC (eqn (7)) to find a formulation for protein samples that can grasp the core and essential features correlated with their subcellular localization.

1. Formulation *via* GO (gene ontology) approach

GO is a controlled vocabulary used to describe the biology of a gene product in any organism.[39,40] GO database[30] was established according to the molecular function, biological process, and cellular component. Therefore, protein samples defined in a GO database space would be clustered in a manner that better reflects their subcellular locations.[41] However, in order to incorporate more information, instead of only using 0 and 1 elements as done in ref. 29, here let us use a different approach to formulate protein samples, as described below.

Step 1: compression and reorganization of the existing GO numbers. The GO database (version 74.0, released 30 July 2009) contains numerous GO numbers. However, owing to the lack of sufficient information, the existing GO numbers are not increasing successively according to a certain order. For easier handling, some reorganization and compression procedure was taken to renumber them. For example, after such a procedure, the original GO numbers GO:0000001, GO:0000002, GO:0000003, GO:0000009, GO:00000011, GO:0000012, GO:0000015,. . ., GO:0090204 would become GO_compress: 00001, GO_compress: 00002, GO_compress: 00003, GO_compress: 00004, GO_compress: 00005, GO_compress: 00006, GO_compress: 00007,. . . . . ., GO_compress: 11118, respectively. The GO database obtained through such a treatment is called GO_compress database, which contains 11 118 numbers increasing successively from 1 to the last one.

Step 2: using eqn (7) with Ω = 11 118, the protein **P** can now be formulated by

$$\mathbf{P}_{\mathrm{GO}} = [\psi_1^{\mathrm{G}} \quad \psi_2^{\mathrm{G}} \quad \ldots \quad \psi_u^{\mathrm{G}} \quad \ldots \quad \psi_{11118}^{\mathrm{G}}]^{\mathbf{T}} \tag{9}$$

where ψ_u^{G} (u = 1, 2,. . .,11 118) are defined according to the following steps.

Step 3: use BLAST[42] to search the Swiss-Prot database (version 55.3) for the homologous proteins of the protein **P** (with the E-value set to 0.001).

Step 4: those proteins which have ≥ 60% pairwise sequence identity with the protein **P** are collected into a set denoted by $\mathbb{S}_{\mathrm{homo}}^{\mathbf{P}}$ and called the "homology set" of **P**. All the elements in $\mathbb{S}_{\mathrm{homo}}^{\mathbf{P}}$ can be regarded as the "representative proteins" of **P**, sharing some similar attributes such as structural conformations and biological functions.[43–45] Because they were retrieved from the Swiss-Prot database, each of these representative proteins must have their own accession numbers.

Step 5: search the GO database at http://www.ebi.ac.uk/GOA/ to find the corresponding GO number(s)[39] for each of the accession numbers collected in Step 4, followed by converting the GO numbers thus obtained to their GO_compress numbers as described in Step 1. Note that the relationships between the UniProtKB/Swiss-Port protein entries and the GO numbers may be one-to-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".[30] For example, the Uni-ProtKB/Swiss-Prot protein entry "P01040" corresponds to three GO numbers, *i.e.*, "GO:0004866", "GO:0004869", and "GO:0005622".

Step 6: based on the results obtained in Step 5, the elements in eqn (9) can be formulated as

$$\psi_u^{\mathrm{G}} = \frac{\sum_{k=1}^{\mathbb{N}_{\mathrm{homo}}^{\mathbf{P}}} \delta(u,k)}{\mathbb{N}_{\mathrm{homo}}^{\mathbf{P}}} \quad (u = 1,2,\ldots,11\,118) \tag{10}$$

where $\mathbb{N}_{\mathrm{homo}}^{\mathbf{P}}$ is the number of representative proteins in $\mathbb{S}_{\mathrm{homo}}^{\mathbf{P}}$, and

$$\delta(u,k) = \begin{cases} 1, & \text{if the } k\text{th representative protein hits} \\ & \text{the } u\text{th GO\_compress number} \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

As we can see from eqn (9), the GO formulation derived from the above steps consists of 11 118 real numbers rather than the elements 0 and 1 only as in the GO formulation defined in ref. 29.

Note that the GO formulation of eqn (9) may become a naught vector or meaningless under any of the following situations: (1) the protein **P** does not have significant homology to any protein in the Swiss-Prot database, *i.e.*, $\mathbb{S}_{\mathrm{homo}}^{\mathbf{P}} = \varnothing$, meaning that the homology set $\mathbb{S}_{\mathrm{homo}}^{\mathbf{P}}$ is an empty one; (2) its representative proteins do not contain any useful GO information because the current GO database is not complete yet.

Under such a circumstance, we are to use the sequential evolution formulation to represent the protein **P**, as described below.

2. Formulation *via* a SeqEvo (sequential evolution) approach

The evolution of protein sequences involves changes of single residues, insertions and deletions of several residues, gene doubling, and gene fusion. With these changes accumulated for a long period of time, many similarities between initial and resultant amino acid sequences are gradually eliminated, but the corresponding proteins may still share many common attributes, such as assuming roughly the same folding topology, having basically the same biological function and residing in the same subcellular location.

To incorporate this kind of sequential evolution information into the general form of PseAAC (eqn (7)), let us utilize the data derived from the PSSM (Position-Specific Scoring Matrix),[42] as described below.

Step 1: according to ref. 42, the sequential evolution of a protein **P** consisting of L amino acids (*cf.* eqn (4)) can be reflected by a 20 × L matrix as given by

$$\mathbb{PSSM} = \begin{bmatrix} E_{1\to1}^{0} & E_{2\to1}^{0} & \cdots & E_{L\to1}^{0} \\ E_{1\to2}^{0} & E_{2\to2}^{0} & \cdots & E_{L\to2}^{0} \\ \vdots & \vdots & \vdots & \vdots \\ E_{1\to20}^{0} & E_{2\to20}^{0} & \cdots & E_{L\to20}^{0} \end{bmatrix} \tag{12}$$

where $E_{i\to j}^{0}$ represents the score of the amino acid residue in the ith position of the protein sequence that is being changed to amino acid type j during the evolutionary process. Here, the numerical codes 1, 2, …, 20 are used to denote the 20 native amino acid types according to the alphabetical order of their single character codes. The 20 × L scores in eqn (12) were generated by using PSI-BLAST[42] to search the UniProtKB/Swiss-Prot database (Release 2010_04 of 23-Mar-2010) through three iterations with 0.001 as the E-value cutoff for multiple sequence alignment against the sequence of the protein **P**.

Step 2: use the elements in the above equation for $\mathbb{PSSM}$ to define a new matrix **M** as formulated by

$$\mathbf{M} = \begin{bmatrix} E_{1\to1} & E_{2\to1} & \cdots & E_{L\to1} \\ E_{1\to2} & E_{2\to2} & \cdots & E_{L\to2} \\ \vdots & \vdots & \vdots & \vdots \\ E_{1\to20} & E_{2\to20} & \cdots & E_{L\to20} \end{bmatrix} \tag{13}$$

where

$$E_{i\to j} = \frac{E_{i\to j}^{0} - \bar{E}_{j}^{0}}{\mathrm{SD}(\bar{E}_{j}^{0})} \quad (i = 1,2,\ldots,L;\ \ j = 1,2,\ldots,20) \tag{14}$$

and

$$\bar{E}_{j}^{0} = \frac{1}{L}\sum_{i=1}^{L} E_{i\to j}^{0} \quad (j = 1,2,\ldots,20) \tag{15}$$

is the mean for $E_{i\to j}^{0}$ (i = 1, 2, …, L), while

$$\mathrm{SD}(\bar{E}_{j}^{0}) = \sqrt{\sum_{i=1}^{L} [E_{i\to j}^{0} - \bar{E}_{j}^{0}]^2 / L} \tag{16}$$

is the corresponding standard deviation.

Step 3: introduce a new matrix generated by multiplying **M** of eqn (13) with its own transpose matrix $\mathbf{M}^{\mathbf{T}}$; *i.e.*,

$$\mathbf{S_M} = \mathbf{MM^T} \\ = \begin{bmatrix} \sum_{i=1}^{L} (E_{i\to1})^2 & \sum_{i=1}^{L} E_{i\to1}E_{i\to2} & \cdots & \sum_{i=1}^{L} E_{i\to1}E_{i\to20} \\ \sum_{i=1}^{L} E_{i\to2}E_{i\to1} & \sum_{i=1}^{L} (E_{i\to2})^2 & \cdots & \sum_{i=1}^{L} E_{i\to2}E_{i\to20} \\ \vdots & \vdots & \vdots & \vdots \\ \sum_{i=1}^{L} E_{i\to20}E_{i\to1} & \sum_{i=1}^{L} E_{i\to20}E_{i\to2} & \cdots & \sum_{i=1}^{L} (E_{i\to20})^2 \end{bmatrix} \tag{17}$$

which contains 20 × 20 = 400 elements. Since $\mathbf{S_M}$ is a symmetric matrix, its upper triangular elements are identical with its lower ones. Accordingly, we only need to use its 20 diagonal elements and 190 upper triangular elements to formulate the protein **P**; *i.e.*, according to eqn (7) we now have

$$\mathbf{P}_{\mathrm{Evo}} = [\psi_1^{\mathrm{E}}\ \ \psi_2^{\mathrm{E}}\ \ \ldots\ \ \psi_u^{\mathrm{E}}\ \ \ldots\ \ \psi_{210}^{\mathrm{E}}]^{\mathbf{T}} \tag{18}$$

where the components ψ_u^{E} (u = 1,2,…,210) are, respectively, taken from the 210 diagonal and upper triangular elements of eqn (17) by following a given order as marked in the following equation

$$\begin{bmatrix} [1] & [2] & [3] & \cdots & [20] \\ & [21] & [22] & \cdots & [39] \\ & & [40] & \cdots & [57] \\ & & & \ddots & \vdots \\ & & & & [210] \end{bmatrix} \tag{19}$$

where the numbers in brackets indicate the order of elements taken from eqn (17) for eqn (18).

3. The self-consistency principle in formulating protein samples

No matter using which formulation to represent protein samples, the following self-consistency principle must be complied during the course of prediction: if the query protein **P** was defined in the form of $\mathbf{P}_{\mathrm{GO}}$ (see eqn (9)), then all the protein samples used to train the prediction engine should also be formulated in the GO form; if the query protein was defined in the form of $\mathbf{P}_{\mathrm{Evo}}$ (see eqn (18)), then all the training data should be defined in that form as well.

Below, let us consider the algorithm or operation engine for conducting the prediction.

4. Accumulation-label K-nearest neighbor (KNN) classifier

Here let us introduce a novel classifier, called the accumulation-label KNN or abbreviated as AL-KNN classifier, to predict the subcellular localization for the systems that contain both singleplex and multiplex proteins.

Suppose the *m*th subset $\mathbb{S}_m$ of $\mathbb{S}$ (eqn (3)) contains N_m proteins, and **P**(*m*,*j*) is the *j*th protein in that subset, we have

$$\mathbf{P}(m,j) = \begin{cases} \mathbf{P}_{\mathrm{GO}}(m,j), & \text{in GO space} \\ \mathbf{P}_{\mathrm{Evo}}(m,j), & \text{in SeqEvo space} \end{cases} \qquad (20)$$

$$(m = 1, 2, \ldots, 14; \;\; j = 1, 2, \ldots, N_m)$$

where $\mathbf{P}_{\mathrm{GO}}(m, j)$ has the same form as $\mathbf{P}_{\mathrm{GO}}$ of eqn (9); while $\mathbf{P}_{\mathrm{Evo}}(m, j)$ has the same form as $\mathbf{P}_{\mathrm{Evo}}$ of eqn (18). The only difference is that the corresponding constituent elements are derived from the amino acid sequence of **P**(*m*, *j*) instead of **P**.

In sequence analysis, there are many different metrics to define the similarity between two protein sequences, such as Euclidean distance, Hamming distance,[46] and Mahalanobis distance.[36,47,48] Usually, the similarity between **P**(*m*, *j*) and **P** was defined by $1 - \cos^{-1}[\mathbf{P}, \mathbf{P}(m, j)]$ (see, *e.g.*, ref. 24 and 49). However, we found that when the GO descriptor was formulated with real numbers, better results would be obtained by using the Euclidean metric; *i.e.*, the similarity between **P** and **P**(*m*, *j*) is defined here by

$$D\{\mathbf{P}, \mathbf{P}(m, j)\} = \|\mathbf{P} - \mathbf{P}(m, j)\| \qquad (21)$$

where $\|\mathbf{P} - \mathbf{P}(m, j)\|$ represents the module of the vector difference between **P** and **P**(*m*, *j*) in the Euclidean space. According to eqn (21), when $\mathbf{P} \equiv \mathbf{P}(m, j)$ we have $D\{\mathbf{P}, \mathbf{P}(m, j)\} = 0$, indicating that the distance between these two protein sequences is zero and hence they have perfect or 100% similarity.

Suppose $\mathbf{P}_1^{\#}, \mathbf{P}_2^{\#}, \ldots, \mathbf{P}_K^{\#}$ are the *K* nearest neighbor proteins to the protein **P**. They form a set denoted by $\mathbb{S}_K^{\mathbf{P}}$, which is a subset of $\mathbb{S}$ (*cf.* eqn (3)); *i.e.*, $\mathbb{S}_K^{\mathbf{P}} \subseteq \mathbb{S}$. Based on the *K* nearest neighbor proteins in $\mathbb{S}_K^{\mathbf{P}}$, let us define a novel scale, called the "accumulation-label" (AL) scale as given by

$$\mathbb{Q}(\mathbf{P}, K) = \{\rho_1^K \quad \rho_2^K \quad \ldots \quad \rho_m^K \quad \ldots \quad \rho_{14}^K\} \qquad (22)$$

where

$$\rho_m = \frac{\sum_{i=1}^{K} \delta(\mathbf{P}_i^{\#}, m)}{\mathbb{N}_K^{\#}} \quad (m = 1, 2, \ldots, 14) \qquad (23)$$

where

$$\delta(\mathbf{P}_i^{\#}, m) = \begin{cases} 1, & \text{if } \mathbf{P}_i^{\#} \text{ belongs to the } m\text{th location} \\ 0, & \text{otherwise} \end{cases} \qquad (24)$$

and

$$\mathbb{N}_K^{\#} = \sum_{m=1}^{14} \sum_{i=1}^{K} \delta(\mathbf{P}_i^{\#}, m) \qquad (25)$$

Note that $\mathbb{N}_K^{\#} \geq K$ because a protein may belong to more than one subcellular location site in the current system.

Now, for a query protein **P**, its subcellular location(s) will be predicted according to the following steps.

Step 1: the number of how many different subcellular locations it belongs to will be determined by its nearest neighbor protein in $\mathbb{S}$ (*cf.* eqn (3)). For example, suppose $\mathbf{P}^{\#}$ is the nearest protein to **P** in $\mathbb{S}$. If $\mathbf{P}^{\#}$ belongs to only one subcellular location, then **P** will be predicted to be belonging to only one location; if $\mathbf{P}^{\#}$ belongs to two subcellular locations, then **P** also belongs to two locations; and so forth. Therefore, in general we have

$$\mathfrak{M}(\mathbf{P}) = \mathfrak{M}(\mathbf{P}^{\#}) \qquad (26)$$

where $\mathfrak{M}(\mathbf{P}^{\#})$ is an integer (≤ 14) representing the number of different subcellular locations to which $\mathbf{P}^{\#}$ belongs, and $\mathfrak{M}(\mathbf{P})$ represents the number of different subcellular locations to which **P** belongs.

Step 2: the actual location site(s) where **P** resides will not be determined by the location site(s) of $\mathbf{P}^{\#}$, but by the element(s) of the AL (accumulation-label) scale (*cf.* eqn (22)) that has (have) the highest score(s), as can be expressed by $\{\ell\}$, the subscript(s) in eqn (3). For example, if **P** is found to belong to only one location in Step 1, *i.e.*, $\mathfrak{M}(\mathbf{P}) = 1$, and the highest score in eqn (22) is ρ_3^K, then **P** will be predicted as $\{\ell\} = \{3\}$, meaning that it belongs to $\mathbb{S}_3$ or resides at "Cytoskeleton" (*cf.* Table 1). If **P** is found to belong to three locations, *i.e.*, $\mathfrak{M}(\mathbf{P}) = 3$, and the first three highest scores in eqn (22) are ρ_1^K, ρ_5^K, and ρ_{14}^K, then **P** will be predicted as $\{\ell\} = \{1,5,14\}$, meaning that it belongs to $\mathbb{S}_1$, $\mathbb{S}_5$ and $\mathbb{S}_{14}$ or resides simultaneously at "Centrosome", "Endosome", and "Synapse". And so forth. In other words, the actual predicted subcellular location(s) for **P** can be formulated as

$$\{\ell\} = \mathrm{Max} \triangleright_{\mathrm{Sub}}^{\mathfrak{M}(\mathbf{P})} \{\rho_1^K \quad \rho_2^K \quad \cdots \quad \rho_m^K \quad \cdots \quad \rho_{14}^K\} \quad (\mathfrak{M}(\mathbf{P}) \leq 14) \qquad (27)$$

where the operator "$\mathrm{Max} \triangleright_{\mathrm{Sub}}^{\mathfrak{M}(\mathbf{P})}$" means identifying the $\mathfrak{M}(\mathbf{P})$ highest scores for the elements in the brackets right after it, followed by taking their $\mathfrak{M}(\mathbf{P})$ subscripts. The value for the parameter *K* in eqn (27) will be determined by optimizing the overall jackknife success rate on the benchmark dataset $\mathbb{S}$ (ESI† S1) as will be further discussed later.

The entire classifier thus established is called iLoc-Hum, which can be used to predict the subcellular localization of both singleplex and multiplex human proteins. To provide an intuitive picture, a flowchart is provided in Fig. 2 to illustrate the prediction process of iLoc-Hum.

5. Web-server guide

For enhancing the value of its practical applications, a web-server for iLoc-Hum was established. Moreover, for the convenience of

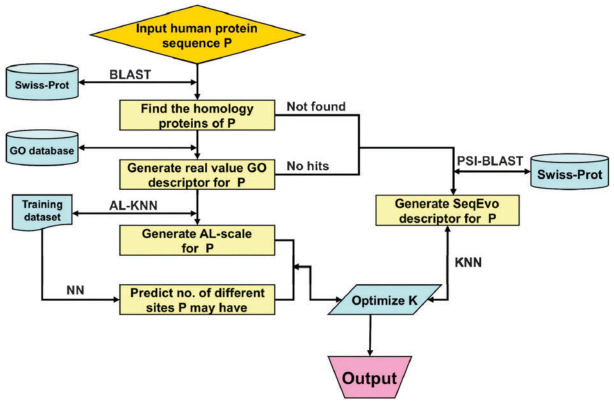

Input human protein sequence P
Swiss-Prot
BLAST
Find the homology proteins of P
Not found
GO database
Generate real value GO descriptor for P
No hits
PSI-BLAST
Swiss-Prot
Training dataset
AL-KNN
Generate SeqEvo descriptor for P
Generate AL-scale for P
KNN
NN
Predict no. of different sites P may have
Optimize K
Output

Fig. 2 A flowchart to show the prediction process of iLoc-Hum.

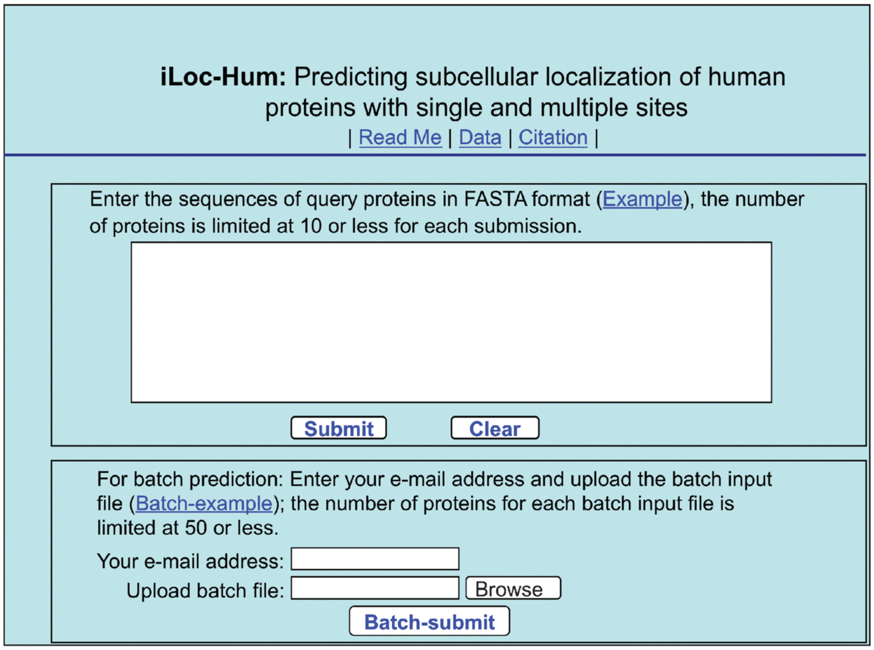
the vast majority of experimental scientists, below let us give a step-by-step guide on how to use the web-server to get their desired results without the need to follow the complicated mathematical equations presented above for its integrity.

Step 1: open the web server at the site of either http://www.jci-bioinfo.cn/iLoc-Hum or http://icpr.jci.edu.cn/bioinfo/iLoc-Hum and you will see the top page of the predictor on your computer screen, as shown in Fig. 3. Click on the ReadMe button to see a brief introduction about the iLoc-Hum predictor and the caveat when using it.

Step 2: either type or copy and paste the query protein sequence into the input box at the center in Fig. 3. The input sequence should be in the FASTA format. A sequence in the FASTA format consists of a single initial line beginning with a greater than symbol ("$>$") in the first column, followed by lines of sequence data. The words right after the "$>$" symbol in the single initial line are optional and only used for the purpose of identification and description. All lines should be no longer than 120 characters and usually do not exceed 80 characters. The sequence ends if another line starting with a "$>$" appears; this indicates the start of another sequence. Example sequences in the FASTA format can be seen by clicking on the Example button right above the input box. For more information about the FASTA format, visit http://en.wikipedia.org/wiki/Fasta_format. Different from Hum-mPLoc 2.0,[29] where only one query protein sequence at a time is allowed as an input for each submission, now the maximum number of query proteins for each submission can be 10.

Step 3: click on the Submit button to see the predicted result. For example, if you use the three query protein sequences in the Example window as the input, after clicking the Submit button, you will see Fig. 4 on your screen, indicating that the predicted result for the 1st query protein is "Mitochondrion", that for the 2nd one is "Golgi apparatus; Nucleus", and that for the 3rd one is "Centrosome; Cytoplasm; Mitochondrion". In other words, the 1st query protein (P82932) is a single-location one residing at "mitochondrion" only, the 2nd one (Q9Y3M2) can simultaneously occur in two different sites ("Golgi apparatus" and "nucleus"), and the 3rd one (Q9Y2Y0) can simultaneously occur in three different sites ("centrosome", "cytoplasm", and "mitochondrion"). All these results are fully consistent with the experimental observation as summarized in the ESI† S1. It takes about 10 seconds for the above computation before the predicted result appears on your computer screen; the more number of query proteins and longer each sequence, the more time it is usually needed.

Step 4: as shown in the lower panel of Fig. 3, you may also choose the batch prediction by entering your e-mail address and your desired batch input file (in FASTA format) *via* the "Browse" button. To see the sample of the batch input file, click on the button Batch – example. The maximum number of query proteins for each batch input file is 50. After clicking the button Batch – submit, you will see "Your batch job is



iLoc-Hum: Predicting subcellular localization of human proteins with single and multiple sites

| Read Me | Data | Citation |

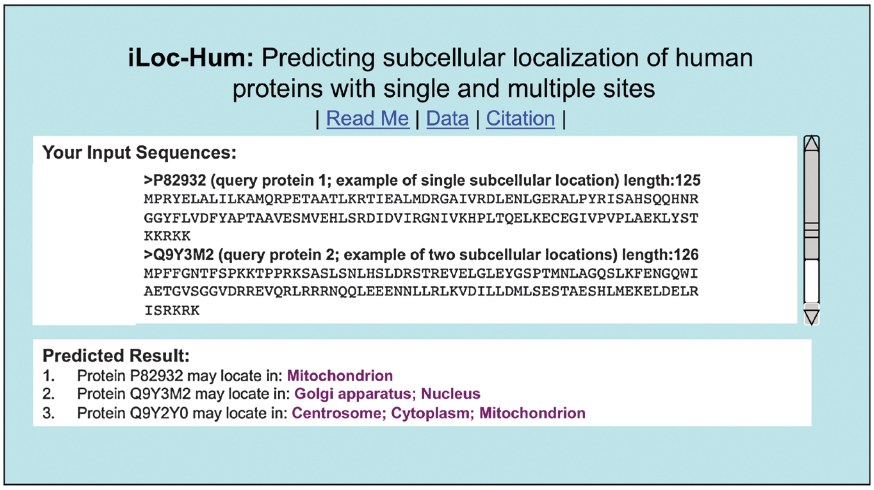
Enter the sequences of query proteins in FASTA format (Example), the number of proteins is limited at 10 or less for each submission.

For batch prediction: Enter your e-mail address and upload the batch input file (Batch-example); the number of proteins for each batch input file is limited at 50 or less.

Your e-mail address:

Upload batch file:

Fig. 3 A semi-screenshot to show the top page of the iLoc-Hum web-server. Its website address is at http://www.jci-bioinfo.cn/iLoc-Hum or http://icpr.jci.edu.cn/bioinfo/iLoc-Hum.



iLoc-Hum: Predicting subcellular localization of human proteins with single and multiple sites

Your Input Sequences:

```
>P82932 (query protein 1; example of single subcellular location) length:125
MPRYELALILKAMQRPETAATLKRTIEALMDRGAIVRDLENLGERALPYRISAHSQQHNR
GGYFLVDFYAPTAAVESMVEHLSRDIDVIRGNIVKHPLTQELKECEGIVPVPLAEKLYST
KKRKK
>Q9Y3M2 (query protein 2; example of two subcellular locations) length:126
MPFFGNTFSPKKTPPRKSASLSNLHSLDRSTREVELGLEYGSPTMNLAGQSLKFENGQWI
AETGVSGGVDRREVQRLRRRNQQLEEENNLLRLKVDILLDMLSESTAESHLMEKELDELR
ISRKRK
```

Predicted Result:

1. Protein P82932 may locate in: **Mitochondrion**
2. Protein Q9Y3M2 may locate in: **Golgi apparatus; Nucleus**
3. Protein Q9Y2Y0 may locate in: **Centrosome; Cytoplasm; Mitochondrion**

Fig. 4 A semi-screenshot to show the output of iLoc-Hum. The input was taken from the three protein sequences listed in the Example window of the iLoc-Hum web-server (*cf.* Fig. 3).

under computation; once the results are available, you will be notified by e-mail".

Step 5: click on the Citation button to find the relevant papers that document the detailed development and algorithm of iLoc-Hum.

Step 6: click on the Data button to download the benchmark datasets used to train and test the iLoc-Hum predictor.

Caveat: to obtain the predicted result with the expected success rate, the entire sequence of the query protein rather than its fragment should be used as an input. A sequence with less than 50 amino acid residues is generally deemed as a fragment. Also, if the query human protein is known to be not in one of the 14 locations as shown in Fig. 1, stop the prediction because the result thus obtained will not make any sense.

IV. Results and discussion

1. Three different cross-validation tests

In statistical prediction, it would be meaningless to simply report a success rate of a predictor without specifying what method and benchmark dataset were used to test its accuracy. As is well known, the following three methods are often used to examine the quality of a predictor: independent dataset test, subsampling test, and jackknife test.[50] Since the subsampling test and the jackknife test can be performed with one benchmark dataset and that the independent dataset test can be treated as a special case of the subsampling test, one benchmark dataset would suffice to serve all the three kinds of cross-validations. However, it is instructive to point out that, of the three cross-validation test methods, the jackknife test is deemed as the least arbitrary or most objective.[41] The reasons are as follows. (1) For the independent dataset test, although all the proteins used to test a predictor are outside the training dataset used to train it so as to exclude the "memory" effect or bias, the way of how to select the independent proteins to test the predictor could be quite arbitrary. This kind of arbitrariness might result in completely different conclusions. For instance, a predictor achieving a higher success rate than the other predictor for a given independent testing dataset might fail to keep so when tested by another independent testing dataset.[50] (2) For the subsampling test, the concrete procedure usually used in the literature is the 5-fold, 7-fold or 10-fold cross-validation. The problem with this kind of subsampling test is that the number of possible selections in dividing a benchmark dataset is an astronomical figure even for a very simple dataset, as demonstrated by eqn (28)–(30) in ref. 32. Therefore, in any actual subsampling cross-validation test, only an extremely small fraction of the possible selections was taken into account. Since different selections will always lead to different results even for the same benchmark dataset and the same predictor, the subsampling test cannot avoid the arbitrariness either. A test method unable to yield a unique outcome cannot be deemed as an ideal one. (3) In the jackknife test, all the proteins in the benchmark dataset will be singled out one-by-one and tested by the predictor trained by the remaining protein samples. During the process of jackknifing, both the training dataset and the testing dataset are actually open, and each protein sample will be in turn moved between the two. The jackknife test can exclude the "memory" effect. Also, the arbitrariness problem as mentioned above for the independent dataset test and the subsampling test can be avoided because the outcome obtained by the jackknife cross-validation is always unique for a given benchmark dataset. Accordingly, the jackknife test has been widely recognized and increasingly used to examine the power of various predictors (see, *e.g.*, ref. 51–54). Accordingly, in this study, the jackknife test will be adopted as a key metric to evaluate the power of iLoc-Hum, while the independent dataset test will be used as a demonstration.

2. Comparison with Hum-mPLoc[28] *via* the jackknife test

It is instructive to point out that even if the jackknife test is used to examine the accuracy, the same predictor may still yield obviously different success rates when tested by different benchmark datasets. This is because the more stringent a benchmark dataset is in excluding homologous sequences, the more difficult it is for a predictor to achieve a high success rate. Also, the more number of subsets (subcellular locations) a benchmark dataset covers, the more difficult to achieve a high overall success rate as elucidated in a recent review.[32]

Among the existing benchmark datasets constructed for investigating human protein subcellular location prediction, the benchmark dataset $\mathbb{S}$ used in this study is the most stringent and harsh one as reflected by the facts that it covers 14 subcellular location sites with both singleplex and multiplex human proteins, and that none of the proteins included has $\geq 25\%$ pairwise sequence identity to any other in the same subset (*cf.* the Materials section).

Also, as mentioned in the Introduction section, there are many existing predictors for identifying protein subcellular locations, such as TargetP,[14] PSORT,[13,17,19] GNBSL,[55] WoLF PSORT,[56] and MultiLoc,[20] which have been widely used with each having its own advantage. However, these predictors were not developed for identifying the subcellular locations of human proteins, and most of them were designed for dealing with bacterial or plant proteins covering only five or less subcellular location sites. Particularly, none of these predictors has the capacity to deal with multiplex proteins that can simultaneously exist in two or more subcellular location sites.

Actually, for the current dataset $\mathbb{S}$ given in ESI† S1 that contains both single-location and multiple-location proteins distributed among 14 subcellular location sites in human cells, so far only two existing predictors, *i.e.*, Hum-mPLoc[28] and its improved version Hum-mPLoc 2.0,[29] have the capacity to deal with it. Besides, it has already been demonstrated[29] that Hum-mPLoc 2.0 is superior to Hum-mPLoc.[28] Therefore, to demonstrate the power of the current predictor, it would suffice to just compare iLoc-Hum with Hum-mPLoc 2.0.[29]

Listed in Table 1 are the results obtained with iLoc-Hum on the aforementioned benchmark dataset $\mathbb{S}$ by the jackknife test. As we can see from Table 1, for such a stringent and complicated benchmark dataset, the overall success rate achieved by iLoc-Hum is over 76%, which is about 13% higher than 62.7%, the overall jackknife success rate by Hum-mPLoc 2.0 on the same benchmark dataset $\mathbb{S}$ as reported in ref. 29. Besides, the corresponding overall sensitivity and specificity[57] obtained by iLoc-Hum were also quite high, with the values of 79.30% and 94.99%, respectively.

Note that during the course of the jackknife test with iLoc-Hum, the false positives (over-predictions) and false negatives (under-predictions) were also taken into account to reduce the scores for calculating the overall success rate. As for the detailed process of how to count the over-predictions and under-predictions for a system containing both single-location and multiple-location proteins, see eqn (43)–(48) and Fig. 4 in a comprehensive review.[24]

3. Absolute true success rate

To provide a more intuitive and easier-to-understand accuracy rate, let us introduce a new scale, the so-called "absolute true" success rate, to measure the accuracy of a predictor, as defined by

$$\Lambda = \frac{\sum_{i=1}^{N} \Delta(i)}{N} \tag{28}$$

where Λ represents the absolute true rate, N the number of total proteins investigated, and

$$\Delta(i) = \begin{cases} 1, & \text{if all the subcellular locations of the } i\text{th protein are} \\ & \text{correctly predicted without any overprediction} \\ 0, & \text{otherwise} \end{cases} \tag{29}$$

The absolute true success rate as defined by eqn (28) and (29) is particularly useful when dealing with a system consisting of both single-location proteins and multiple-location proteins. According to the above definition, for a protein belonging to, say, three subcellular locations, if only two of the three are correctly predicted, or the predicted result contains a location not belonging to the three, the prediction score will be counted as 0. In other words, when and only when all the subcellular locations of a query protein are exactly predicted without any underprediction or overprediction can the prediction be scored with 1. Therefore, the absolute true scale is much more strict and harsh than the scale used previously[24,29] for measuring the success rate for a system consisting of both singleplex and multiplex proteins. However, even if such a stringent criterion was used on the same benchmark dataset by the jackknife test, iLoc-Hum could still achieve the absolute true success rate as high as 2118/3106 = 68.19%.

Why can iLoc-Hum enhance the success rate so remarkably? One of the key reasons is that the GO formulation for protein samples in iLoc-Hum contains more information than that in Hum-mPLoc 2.0,[29] as can be illustrated as follows. For instance, let us consider a query protein **P** with the sequence given below:

$$\mathbf{P}=\left\{\begin{array}{l} \text{GAEDWPGQQLELDEDEASCCRWGAQHAGARELAALYSPGKRLQEWCSVIL} \\ \text{CFSLIAHNLVHLLLLARWEDTPLVILGVVAGALIADFLSGLVHWGADTWG} \\ \text{SVELPIVGKAFIRPFREHHIDPTAITRHDFIETNGDNCLVTLLPLLNMAY} \\ \text{KFRTHSPEALEQLYPWECFVFCLIIFGTFTNQIHKWSHTYFGLPRWVTLL} \\ \text{QDWHVILPRKHHRIHHVSPHETYFCITTGWLNYPLEKIGFWRRLEDLIQG} \\ \text{LTGEKPRADDMKWAQK} \end{array}\right\} \tag{30}$$

According to Steps 3 and 4 in the "Formulation *via* GO (gene ontology) Approach" section, we found four proteins in Swiss-Prot database that were homologous to **P** of eqn (30); *i.e.*, $\mathbb{N}_{\text{homo}}^{\mathbf{P}} = 4$. Their accession numbers are, respectively, A5PLL7, Q99LQ7, A6QLM0, and 8BVR0. Of the four proteins, A5PLL7 hits GO:0005783 (or GO_compressed:02508), and GO:0005789 (or GO_compressed:02514); Q99LQ7 hits GO:0005783 (or GO_compressed:02508), GO:0005789 (or GO_compressed:02514), GO:0016020 (or GO_compressed: 06305), and GO:0016021 (or GO_compressed:06306); A6QLM0 hits GO:0005783 (or GO_compressed:02508), GO:0005789 (or GO_compressed:02514), GO:0016020 (or GO_compressed: 06305), and GO:0016021 (or GO_compressed:6306); 8BVR0 hits GO:0003677 (or GO_compressed:01150), GO:0005634 (or GO_compressed:2411), and GO:0005737 (or GO_compressed:05737). In other words, GO_compress:01150 and GO_compress:02411 were each hit once; GO_compress:02469, GO_compress:02508, GO_compress:02514, GO_compress:06305, and GO_compress:06306 were each hit three times; all the other GO_compress numbers were not hit at all. Substituting these data into eqn (10) and (11) of Step 6, we obtain the uth (u = 1,2,...,11 118) component in eqn (9) for protein **P** of eqn (30) as

$$\psi_u^{\mathrm{G}}(\mathbf{P}) = \begin{cases} 1/4 = 0.25, & \text{if } u = 1150 \\ 1/4 = 0.25, & \text{if } u = 2411 \\ 3/4 = 0.75, & \text{if } u = 2469 \\ 3/4 = 0.75, & \text{if } u = 2508 \\ 3/4 = 0.75, & \text{if } u = 2514 \\ 3/4 = 0.75, & \text{if } u = 6305 \\ 3/4 = 0.75, & \text{if } u = 6306 \\ 0, & \text{otherwise} \end{cases} \tag{31}$$

In contrast, according to the formulation in Hum-mPLoc 2.0[29] the same protein **P** would be formulated as

$$\psi_u^{\mathrm{G}}(\mathbf{P}) = \begin{cases} 1, & \text{if } u = 1150 \\ 1, & \text{if } u = 2411 \\ 1, & \text{if } u = 2469 \\ 1, & \text{if } u = 2508 \\ 1, & \text{if } u = 2514 \\ 1, & \text{if } u = 6305 \\ 1, & \text{if } u = 6306 \\ 0, & \text{otherwise} \end{cases} \tag{32}$$

As can be clearly seen by comparing eqn (31) with eqn (32), although the elements in the 1150th, 2411th, 2469th, 2508th, 2514th, 6305th, and 6306th components are all not zero in both equations, in the one defined in Hum-mPLoc 2.0[29] all the elements are either 1 or zero, completely ignoring their weights. In other words, the GO formulation in the current iLoc-Hum contains more information than that in Hum-mPLoc 2.0[29] and hence leading to better prediction results.

The other reason is that in Hum-mPLoc 2.0[29] the number of subcellular location sites for a query protein was determined by a threshold factor θ^* (*cf.* eqn (48) in ref. 24) that actually functioned as a "black box" without providing any physicochemical rationale. In contrast, it is very much different in the current iLoc-Hum as reflected by the fact that the number of subcellular location sites for a query protein is determined according to the nearest neighbor (NN) principle (*cf.* eqn (26)), and that its concrete location sites are determined according to the CL scale (*cf.* eqn (22) and (27)). A combination of the above two approaches, *i.e.* the "Accumulation-label K-Nearest Neighbor" or "AC-KNN" algorithm, is particularly useful and more natural for investigating proteins with multiple subcellular location sites.

Owing to the aforementioned improved features of iLoc-Hum against Hum-mPLoc 2.0,[29] many overpredicted (false positive) results and underpredicted (false negative) results generated by Hum-mPLoc 2.0 can be corrected by iLoc-Hum.

For example, according to the experimental observations as annotated in Swiss-Prot database,[8] the protein with accession number P78329 can simultaneously occur in the following two location sites: "Endoplasmic reticulum" and "Microsome". However, when inputting the sequence of P78329 into the web-server of Hum-mPLoc 2.0,[29] the output was: "Endoplasmic reticulum; Extracell; Microsome", meaning that the location of "Extracell" was a false positive (overprediction). Now, upon inputting the same sequence into the web-server of iLoc-Hum, we obtained the output as "Endoplasmic reticulum; Microsome", meaning that its two subcellular locations were exactly predicted without any overprediction (false positive).

Also, according to the experimental annotation in Swiss-Prot database,[8] the protein A5PLL7 only occurs in the organelle of "Endoplasmic reticulum". However, if Hum-mPLoc 2.0[29] was used to predict its subcellular localization, the outcome was "Cytoplasm; Nucleus"; *i.e.*, both are false positive (overpredicted). Now, if the current iLoc-Hum is used to perform the prediction, the outcome is "Endoplasmic reticulum", exactly the same as the location observed by experiments.

Moreover, according to the experimental observations the protein with accession number O95866 can simultaneously occur in the following three location sites: "Endoplasmic reticulum", "Golgi apparatus" and "Plasma membrane". However, when inputting the sequence of O95866 into the web-server of Hum-mPLoc 2.0,[29] the output was: "Endoplasmic reticulum; Plasma membrane", meaning that one of its location sites was missing or underpredicted. But upon inputting the same sequence into the web-server of iLoc-Hum, we obtained the output as "Endoplasmic reticulum; Golgi apparatus; Plasma membrane", meaning all its three subcellular locations were perfectly correctly predicted without any false positive and false negative at all.

4. Comparison with MultiLoc[20] *via* the independent dataset test

Although Table 1 and the above analysis have provided very compelling evidence to indicate that iLoc-Hum is superior to Hum-mPLoc 2.0[29] as well as any of the existing predictors specialized for identifying the subcellular locations of human proteins, people might be still interested in knowing the outcome when comparing iLoc-Hum with the other existing predictors in identifying the subcellular locations of proteins regardless of what organisms they are from. To address this problem, let us consider MultiLoc, a powerful predictor for identifying the subcellular locations of animal, fungal, and plant proteins that had been demonstrated[20] to be superior to iPSORT[58] and TargetP.[14] The web-server of MultiLoc[20] is at http://abi.inf.uni-tuebingen.de/Services/MultiLoc/, and it covers the following 11 subcellular location sites: (1) chloroplast, (2) cytoplasm, (3) endoplasmic reticulum, (4) extracellular, (5) Golgi apparatus, (6) lysosome, (7) mitochondrion, (8) nucleus, (9) peroxisome, (10) plasma membrane, and (11) vacuole. Of the 11 organelles, chloroplast and vacuole are not present in human cells but in plant or some protist, bacterial, and fungal cells. To make iLoc-Hum and MultiLoc[20] comparable, let us consider the overlapping region of the 14 locations of iLoc-Hum with the 11 locations of MultiLoc.[20] Such an overlapping region consists of the following nine sites: (1) cytoplasm, (2) endoplasmic reticulum, (3) extracellular, (4) Golgi apparatus, (5) lysosome, (6) mitochondrion, (7) nucleus, (8) peroxisome, and (9) plasma membrane.

Thus, in order to make it possible to perform a head-to-head comparison between MultiLoc[20] and the current predictor iLoc-Hum, let us construct an independent testing dataset by randomly picking each of the constituent tested proteins from

Table 2 A head-to-head comparison between iLoc-Hum and MultiLoc[20] by the success rates in predicting the subcellular locations for the 270 proteins in the independent dataset as given in the ESI S2a

Subcellular location	iLoc-Hum[a]	MultiLoc2[b]
Cytoplasm	22/30 = 73.33%	13/30 = 43.33%
Endoplasmic reticulum	27/30 = 90.00%	11/30 = 36.67%
Extracellular	23/30 = 76.66%	20/30 = 66.67%
Golgi apparatus	15/30 = 50.00%	5/30 = 16.67%
Lysosome	25/30 = 83.33%	8/30 = 26.67%
Mitochondrion	26/30 = 86.67%	10/30 = 33.33%
Nuclear	20/30 = 66.67%	13/30 = 43.33%
Peroxisome	26/30 = 86.67%	18/30 = 60.00%
Plasma membrane	30/30 = 100.00%	15/30 = 50.00%
Overall	214/270 = 79.26%	113/270 = 41.85%

[a] Here the following absolute true scale (eqn (28)) was used to score the prediction point for the results identified by iLoc-Hum: when and only when the subcellular location (locations) of a query protein is (are) exactly predicted without any underprediction or overprediction, can the point be scored with 1 point; otherwise, scored with 0. [b] MultiLoc2 is an updated version of MultiLoc;[20] its web-server is at http://abi.inf.uni-tuebingen.de/Services/MultiLoc2.

Swiss-Prot database according to the following criteria: (1) it must belong to one of the aforementioned nine locations, as clearly annotated in Swiss-Prot database; (2) it must be a single-location protein because MultiLoc[20] does not have the capacity to deal with multiple-location proteins; (3) it must neither occur in the training dataset of MultiLoc nor occur in the training dataset of iLoc-Hum in order to avoid the memory bias. The degenerate independent testing dataset thus generated consists of 30 × 9 = 270 single-location proteins classified into the above nine location sites with each containing 30 proteins. The accession numbers and sequences of the 270 tested proteins are given in ESI† S2a.

The detailed predicted results by the current iLoc-Hum and MultiLoc[20] for each of the 270 independent tested proteins are listed in ESI† S2b, from which we can derive that the overall success rate by iLoc-Hum is 79.26% (Table 2), which is about 37% higher than 41.85%, the corresponding rate by MultiLoc.[20]

Furthermore, to demonstrate the power of iLoc-Hum in dealing with proteins with multiple location sites, let us consider the dataset given in ESI† S2c. It contains 30 protein sequences that were randomly picked from Swiss-Prot database under the following conditions: (1) none of the proteins occurs in the training dataset of iLoc-Hum; (2) each of them is known to belong to two or more subcellular locations distributed within the scope of the aforementioned nine sites covered by MultiLoc.[20] The outcomes generated by inputting the 30 sequences into the web-servers iLoc-Hum and MultiLoc[20] are, respectively, listed in ESI† S2d, from which we can see that among the 30 proteins, 26 were perfectly predicted by iLoc-Hum for their multiple-location sites without any false positive and false negative, and 4 were partially correctly predicted. This kind of power of iLoc-Hum in dealing with multiple-location proteins is beyond the reach of MultiLoc.

5. Comparison with Wolf PSORT[56] *via* the independent dataset test

As one more demonstration, let us also consider the comparison of iLoc-Hum with Wolf PSORT,[56] another well-known predictor with a web-server at http://wolfpsort.org/, for identifying the subcellular locations of proteins from animal, plant, and fungi organisms. Wolf PSORT covers the following 12 subcellular location sites: (1) "chol" (chloroplast), (2) "cyto" (cytosol or the soluble portion of the cytoplasm), (3) "cysk" (cytoskeleton), (4) E.R. (endoplasmic reticulum), (5) "extr" (extracellular), (6) "golg" (Golgi apparatus), (7) "lyso" (lysosome), (8) "mito" (mitochondria), (9) "nucl" (nuclear), (10) "pero" (peroxisome), (11) "plas" (plasma membrane), and (12) "vacu" (vacuolar membrane). Of the 12 sites, "chol" and "vacu" are not present in human cells but in plant or some other organisms cells. To make iLoc-Hum and Wolf PSORT[56] comparable, let us consider the overlapping region of the 14 locations of iLoc-Hum with the 12 locations of MultiLoc.[20] Such an overlapping region consists of the following ten sites: (1) cytoplasm, (2) cytoskeleton, (3) endoplasmic reticulum, (4) extracellular, (5) Golgi apparatus, (6) lysosome, (7) mitochondrion, (8) nucleus, (9) peroxisome, and (10) plasma membrane.

By following a similar procedure as described above in constructing the independent testing dataset for comparing iLoc-Hum with MultiLoc,[20] we obtained a degenerate independent testing dataset consisting of 30 × 10 = 300 single-location proteins classified into the above ten location sites with each containing 30 proteins. The accession numbers and sequences of the 300 tested proteins are given in ESI† S3a.

The detailed predicted results by the current iLoc-Hum and Wolf PSORT[56] for each of the 300 independent tested proteins are listed in ESI† S3b, from which we can derive that the overall success rate by iLoc-Hum is 88.33% (Table 3), which is 63% higher than 25.00%, the corresponding rate by Wolf PSORT.[56]

Moreover, to show the difference between iLoc-Hum and Wolf PSORT[56] in dealing with multiple-location proteins, let us consider the dataset given in ESI† S3c. It contains 30 protein sequences that were randomly picked from Swiss-Prot database under the following conditions: (1) none of the proteins occurs in the training dataset of iLoc-Hum; (2) each of them is known to belong to two or more subcellular locations distributed within the scope of the above ten sites covered by Wolf PSORT.[56] The outcomes generated by inputting the 30 sequences into the web-servers iLoc-Hum and Wolf PSORT[56] are, respectively, listed in ESI† S3d, from which we can see that among the 30 proteins,

Table 3 A head-to-head comparison between iLoc-Hum and Wolf PSORT[56] in predicting the subcellular locations for the 300 proteins in the independent dataset as given in the ESI S3a

Subcellular location	iLoc-Hum[a]	Wolf PSORT[b]
Cytoplasm	28/30 = 93.33%	13/30 = 43.33%
Cytoskeleton	24/30 = 80.00%	0/30 = 0.00%
Endoplasmic reticulum	26/30 = 86.67%	3/30 = 10.00%
Extracellular	24/30 = 80.00%	18/30 = 60.00%
Golgi apparatus	15/30 = 50.00%	1/30% = 3.33%
Lysosome	30/30 = 100.00%	0/30 = 0.00%
Mitochondrion	30/30 = 100.00%	12/30 = 40.00%
Nuclear	29/30 = 96.67%	17/30 = 56.67%
Peroxisome	30/30 = 100%	4/30 = 13.33%
Plasma membrane	29/30 = 96.67%	7/30 = 23.33%
Overall	265/300 = 88.33%	75/300 = 25.00%

[a] See footnote a of Table 2 for further explanation. [b] The web-server of Wolf PSORT[56] is at http://wolfpsort.org/.

26 were perfectly predicted by iLoc-Hum for their multiple-location sites without any false positive and false negative, and 4 were partially correctly predicted, once again demonstrating the power of iLoc-Hum in dealing with multiple-location proteins, which is beyond the reach of Wolf PSORT.

V. Conclusion

From the above comparisons of iLoc-Hum with Hum-mPLoc 2.0,[29] MultiLoc,[20] and Wolf PSORT,[56] we can now make the following points crystal clear.

The more stringent a benchmark dataset is in excluding homologous and high similarity sequences, or the more subcellular location sites it covers, the more difficult it is for a predictor to achieve a high overall success rate, as can be easily understood by considering the following cases. For a benchmark dataset only covering four subcellular locations each containing the same number of proteins, the overall success rate by random assignments would generally be 1/4 (25%); while for a benchmark dataset covering 14 subcellular locations, the overall success rate by random assignments would be only 1/14 (7.1%). This means that the former is more than three times the latter.

Also, for a predictor tested by jackknife cross-validation it is very difficult to yield a high success rate when performed on a stringent benchmark dataset in which none of the proteins included has $\geq 25\%$ pairwise sequence identity to any other in the same subset (subcellular location). That is why the overall success rate achieved by iLoc-Hum was only 76.3% (Table 1) when tested by the jackknife cross-validation on the benchmark dataset given in ESI† S1 but was over 79% (Table 2) and 88% (Table 3) when tested by the independent datasets given in ESI† S2a and S3a, respectively.

However, regardless of which test method or test dataset is used, one thing is crystal clear, *i.e.*, the overall success rates achieved by the current iLoc-Hum are significantly higher than those by its counterparts, as shown in Tables 1–3.

Meanwhile, it has also become understandable why the overall success rates reported by many existing predictors in this area were over-estimated. For example, the overall success rate reported in ref. 20 for MultiLoc was 75%, but its success rate by the independent dataset test here was only about 41% (Table 2). This is because the benchmark dataset used by MultiLoc to estimate its success rate contains many homologous sequences. Actually, in constructing the benchmark datasets for establishing MultiLoc,[20] the cutoff thresholds were set at 80%, meaning that only those sequences which have $\geq 80\%$ pairwise sequence identity to any other in the same subset were excluded. Compared with the current benchmark dataset (*cf.* ESI† S1) in which none of the proteins included has $\geq 25\%$ pairwise sequence identity to any other in the same subset, the benchmark dataset used in MultiLoc[20] is much less stringent and hence cannot avoid homologous bias and over estimation.

Particularly, as shown in ESI† S2d and S3d, many multiple-location proteins can be precisely identified by iLoc-Hum for their subcellular location sites without any false positive and false negative. This kind of function is far beyond the reach of the existing predictors, such as MultiLoc[20] and Wolf PSORT,[56] although they are widely used by biologists.

Finally, it should be pointed out that although iLoc-Hum is more powerful than the existing predictors in identifying the subcellular locations of human proteins, there is much room for further improvement in future studies. As shown in Table 1, the success rates by iLoc-Hum for proteins belonging to main cellular compartments (such as nuclear, extracellular, plasma membrane, mitochondria, and cytoplasm) are more accurate. However, for cellular compartments containing fewer proteins (such as endosome, peroxisome, and synapse), the success rates are mostly lower than 50%. This is because, compared with the main cellular compartments, the numbers of proteins in these non-main cellular compartments are not sufficiently large to train the prediction engine in a more effective way. As a consequence, owing to the presence of fewer proteins in these compartments in the training database, the correct neighbor could easily be competed out by a false positive. Nevertheless, it is anticipated that with more experimental data available for these non-main cellular compartments in the future, the situation will be improved and the anticipated success rates by iLoc-Hum will be further enhanced.

Acknowledgements

The authors are very much indebted to the two anonymous reviewers for their constructive comments, which were very helpful for strengthening the presentation of this paper. This work was supported by the grants from the National Natural Science Foundation of China (No. 60961003), the Key Project of Chinese Ministry of Education (No. 210116), the Province National Natural Science Foundation of JiangXi (2009GZS0064), the Department of Education of Jiang-Xi Province (No. GJJ09271), and the plan for training youth scientists (stars of Jing-Gang) of Jiangxi Province.

References

1 C. J. Tsai, B. Ma and R. Nussinov, *Trends Biochem. Sci.*, 2009, **34**, 594–600.
2 L. Chen, T. Huang, X. H. Shi, Y. D. Cai and K. C. Chou, *Molecules*, 2010, **15**, 8177–8192. (Openly accessible at www.mdpi.com/journal/molecules).
3 J. S. Ehrlich, M. D. Hansen and W. J. Nelson, *Dev. Cell*, 2002, **3**, 259–270.
4 E. Glory and R. F. Murphy, *Dev. Cell*, 2007, **12**, 7–16.
5 B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts and J. D. Watson, *Molecular Biology of the Cell*, Garland Publishing, New York & London, 3rd edn, 1994, ch. 1.
6 H. Lodish, D. Baltimore, A. Berk, S. L. Zipursky, P. Matsudaira and J. Darnell, *Molecular Cell Biology*, Scientific American Books, New York, 3rd edn,1995, ch. 3.
7 C. Smith, http://www.biocompare.com/Articles/TechnologySpotlight/976/Subcellular-Targeting-Of-Proteins-And-Drugs.html 2008.
8 A. Bairoch and R. Apweiler, *Nucleic Acids Res.*, 2000, **25**, 31–36.
9 H. Nakashima and K. Nishikawa, *J. Mol. Biol.*, 1994, **238**, 54–61.
10 J. Cedano, P. Aloy, J. A. P'erez-Pons and E. Querol, *J. Mol. Biol.*, 1997, **266**, 594–600.
11 A. Reinhardt and T. Hubbard, *Nucleic Acids Res.*, 1998, **26**, 2230–2236.
12 K. C. Chou and D. W. Elrod, *Protein Eng.*, 1999, **12**, 107–118.
13 K. Nakai and P. Horton, *Trends Biochem. Sci.*, 1999, **24**, 34–36.
14 O. Emanuelsson, H. Nielsen, S. Brunak and G. von Heijne, *J. Mol. Biol.*, 2000, **300**, 1005–1016.
15 K. C. Chou and Y. D. Cai, *J. Biol. Chem.*, 2002, **277**, 45765–45769.
16 K. J. Park and M. Kanehisa, *Bioinformatics*, 2003, **19**, 1656–1663.

17 J. L. Gardy, C. Spencer, K. Wang, M. Ester, G. E. Tusnady, I. Simon, S. Hua, K. deFays, C. Lambert, K. Nakai and F. S. Brinkman, *Nucleic Acids Res.*, 2003, **31**, 3613–3617.

18 S. Matsuda, J. P. Vert, H. Saigo, N. Ueda, H. Toh and T. Akutsu, *Protein Sci.*, 2005, **14**, 2804–2813.

19 J. L. Gardy, M. R. Laird, F. Chen, S. Rey, C. J. Walsh, M. Ester and F. S. Brinkman, *Bioinformatics*, 2005, **21**, 617–623.

20 A. Hoglund, P. Donnes, T. Blum, H. W. Adolph and O. Kohlbacher, *Bioinformatics*, 2006, **22**, 1158–1165.

21 P. Mundra, M. Kumar, K. K. Kumar, V. K. Jayaraman and B. D. Kulkarni, *Pattern Recognit. Lett.*, 2007, **28**, 1610–1615.

22 Y. D. Cai, J. He, X. Li, K. Feng, L. Lu, X. Kong and W. Lu, *Protein Pept. Lett.*, 2010, **17**, 464–472.

23 K. Nakai, *Adv. Protein Chem.*, 2000, **54**, 277–344.

24 K. C. Chou and H. B. Shen, *Anal. Biochem.*, 2007, **370**, 1–16.

25 A. Garg, M. Bhasin and G. P. Raghava, *J. Biol. Chem.*, 2005, **280**, 14427–14432.

26 K. C. Chou and H. B. Shen, *Biochem. Biophys. Res. Commun.*, 2006, **347**, 150–157.

27 A. H. Millar, C. Carrie, B. Pogson and J. Whelan, *Plant Cell*, 2009, **21**, 1625–1631.

28 H. B. Shen and K. C. Chou, *Biochem. Biophys. Res. Commun.*, 2007, **355**, 1006–1011.

29 H. B. Shen and K. C. Chou, *Anal. Biochem.*, 2009, **394**, 269–274.

30 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 and G. Sherlock, *Nat. Genet.*, 2000, **25**, 25–29.

31 E. Camon, M. Magrane, D. Barrell, V. Lee, E. Dimmer, J. Maslen, D. Binns, N. Harte, R. Lopez and R. Apweiler, *Nucleic Acids Res.*, 2004, **32**, D262–D266.

32 K. C. Chou, *J. Theor. Biol.*, 2011, **273**, 236–247.

33 S. F. Altschul, in *Theoretical and Computational Methods in Genome Research*, ed. S. Suhai, Plenum: New York, 1997, pp. 1–14.

34 J. C. Wootton and S. Federhen, *Comput. Chem.*, 1993, **17**, 149–163.

35 H. Nakashima, K. Nishikawa and T. Ooi, *J. Biochem. (Tokyo)*, 1986, **99**, 152–162.

36 K. C. Chou and C. T. Zhang, *J. Biol. Chem.*, 1994, **269**, 22014–22020.

37 G. P. Zhou and K. Doctor, *Proteins: Struct., Funct., Genet.*, 2003, **50**, 44–48.

38 K. C. Chou, *PROTEINS: Structure, Function, and Genetics*, 2001, **43**, 246–255; Erratum: ibid., 2001, **44**, 60.

39 E. Camon, M. Magrane, D. Barrell, D. Binns, W. Fleischmann, P. Kersey, N. Mulder, T. Oinn, J. Maslen, A. Cox and R. Apweiler, *Genome Res.*, 2003, **13**, 662–672.

40 D. Barrell, E. Dimmer, R. P. Huntley, D. Binns, C. O'Donovan and R. Apweiler, *Nucleic Acids Res.*, 2009, **37**, D396–D403.

41 K. C. Chou and H. B. Shen, *Nat. Protocols*, 2008, **3**, 153–162.

42 A. A. Schaffer, L. Aravind, T. L. Madden, S. Shavirin, J. L. Spouge, Y. I. Wolf, E. V. Koonin and S. F. Altschul, *Nucleic Acids Res.*, 2001, **29**, 2994–3005.

43 Y. Loewenstein, D. Raimondo, O. C. Redfern, J. Watson, D. Frishman, M. Linial, C. Orengo, J. Thornton and A. Tramontano, *Genome Biol.*, 2009, **10**, 207.

44 M. Gerstein and J. M. Thornton, *Curr. Opin. Struct. Biol.*, 2003, **13**, 341–343.

45 K. C. Chou, *Curr. Med. Chem.*, 2004, **11**, 2105–2134.

46 K. V. Mardia, J. T. Kent and J. M. Bibby, *Multivariate Analysis*, Chapter 11 Discriminant Analysis; Chapter 12 Multivariate analysis of variance; Chapter 13 cluster analysis, Academic Press, London, 1979, pp. 322–381.

47 P. C. Mahalanobis, *Proc. Natl. Inst. Sci. India*, 1936, **2**, 49–55.

48 K. C. S. Pillai, in *Encyclopedia of Statistical Sciences*, ed. S. Kotz and N. L. Johnson, John Wiley & Sons, This reference also presents a brief biography of Mahalanobis who was a man of great originality and who made considerable contributions to statistics: New York, 1985, vol. 5, pp. 176–181.

49 T. Huang, X. H. Shi, P. Wang, Z. He, K. Y. Feng, L. Hu, X. Kong, Y. X. Li, Y. D. Cai and K. C. Chou, *PLoS One*, 2010, **5**, e10972.

50 K. C. Chou and C. T. Zhang, *Crit. Rev. Biochem. Mol. Biol.*, 1995, **30**, 275–349.

51 M. Masso and I. I. Vaisman, *J. Theor. Biol.*, 2010, **266**, 560–568.

52 H. Mohabatkar, *Protein Pept. Lett.*, 2010, **17**, 1207–1214.

53 H. Ding, L. Liu, F. B. Guo, J. Huang and H. Lin, *Protein Pept. Lett.*, 2011, **18**, 58–63.

54 M. Hayat and A. Khan, *J. Theor. Biol.*, 2011, **271**, 10–17.

55 J. Guo, Y. Lin and X. Liu, *Proteomics*, 2006, **6**, 5099–5105.

56 P. Horton, K. J. Park, T. Obayashi, N. Fujita, H. Harada, C. J. Adams-Collier and K. Nakai, *Nucleic Acids Res.*, 2007, **35**, W585–W587.

57 J. Chen, H. Liu, J. Yang and K. C. Chou, *Amino Acids*, 2007, **33**, 423–428.

58 H. Bannai, Y. Tamada, O. Maruyama, K. Nakai and S. Miyano, *Bioinformatics*, 2002, **18**, 298–305.