Molecular BioSystems

Cite this: *Mol. BioSyst.*, 2011, **7**, 3287–3297

www.rsc.org/molecularbiosystems

PAPER

iLoc-Plant: a multi-label classifier for predicting the subcellular localization of plant proteins with both single and multiple sites†

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

Received 11th June 2011, Accepted 12th September 2011
DOI: 10.1039/c1mb05232b

Predicting protein subcellular localization is a challenging problem, particularly when query proteins may simultaneously exist at, or move between, two or more different subcellular location sites. Most of the existing methods can only be used to deal with the single-location proteins. Actually, multiple-location proteins should not be ignored because they usually bear some special functions worthy of our notice. By introducing the "multi-labeled learning" approach, a new predictor, called iLoc-Plant, has been developed that can be used to deal with the systems containing both single- and multiple-location plant proteins. As a demonstration, the jackknife cross-validation was performed with iLoc-Plant on a benchmark dataset of plant proteins classified into the following 12 location sites: (1) cell membrane, (2) cell wall, (3) chloroplast, (4) cytoplasm, (5) endoplasmic reticulum, (6) extracellular, (7) Golgi apparatus, (8) mitochondrion, (9) nucleus, (10) peroxisome, (11) plastid, and (12) vacuole, where some proteins belong to two or three locations but none has $\geq 25\%$ pairwise sequence identity to any other in a same subset. The overall success rate thus obtained by iLoc-Plant was 71%, which is remarkably higher than those achieved by any existing predictors that also have the capacity to deal with such a stringent and complicated plant protein system. As a user-friendly web-server, iLoc-Plant is freely accessible to the public at the web-site http://icpr.jci.edu.cn/bioinfo/iLoc-Plant or http://www.jci-bioinfo.cn/iLoc-Plant. Moreover, for the convenience of the vast majority of experimental scientists, a step-by-step guide is provided on how to use the web-server to get the desired results without the need to follow the complicated mathematic equations presented in this paper for its integrity. It is anticipated that iLoc-Plant may become a useful bioinformatics tool for Molecular Cell Biology, Proteomics, Systems Biology, and Drug Development.

I. Introduction

Knowledge of subcellular locations of proteins can provide key hints and useful insight for revealing their functions, helping to understand the intricate pathways that regulate biological processes at the cellular level.[1,2] It is also very useful for identifying and prioritizing drug targets[3] during the process of drug development.

The recent progresses of the plant genome sequencing projects have generated huge number of plant protein sequences. To timely use these new plant protein sequences for both basic research and drug discovery, we need to know of their subcellular locations. Actually, one of the fundamental goals in cell biology and proteomics is to identify the functions of proteins in the context of compartments that organize them in the cellular environment. Unfortunately, it is both time-consuming and expensive to determine the localization of an uncharacterized protein in a living cell purely based on experiments. Therefore, we have to resort to computational approaches to deal with this problem.

Actually, many computational methods have been developed for predicting the subcellular locations of proteins based on their sequence information alone (see, *e.g.*, ref. 4–56).

However, among the aforementioned methods, only the one called "TargetP"[13] and the one called "Predotar"[30] are specialized for plant proteins. Ever since the two predictors were proposed, they have been widely used for studying various plant protein systems and related areas. However, the two predictors only cover a very limited scope. For instance, TargetP[57] was established for identifying plant protein sequences among the following four sites: (i) mitochondria, (ii) chloroplast, (iii) secretory pathway, and (iv) other. And Predotar[30] was established for identifying plant protein sequences among: (i) endoplasmic reticulum, (ii) mitochondrion, (iii) plastid, and (iv) other. Since "other" is an ambiguous

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

† Electronic supplementary information (ESI) available. See DOI: 10.1039/c1mb05232b

location, the two predictors could actually be used to identify plant proteins among only three subcellular location sites. If a user tries to use them to identify a query plant protein located outside the aforementioned sites, such as cell wall, peroxisome, Golgi apparatus, or vacuole, the two predictors would either fail to work or the output thus generated would be meaningless.

To improve this kind of limitation in coverage, the predictor called "Plant-PLoc"[58] was developed to extend the coverage scope for plant proteins from the three locations covered by TargetP[57] or Predotar[30] to the following eleven: (i) cell (or plasma) membrane, (ii) cell wall, (iii) chloroplast, (iv) cytoplasm, (v) endoplasmic reticulum, (vi) extracellular, (vii) mitochondrion, (viii) nucleus, (ix) peroxisome, (x) plastid, and (xi) vacuole. However, Plant-PLoc[58] can only be used to deal with the single-location or "singleplex" plant proteins but not multiple-locations or "multiplex" plant proteins. The latter may simultaneously reside at, or move between, two or more different subcellular locations. Proteins with multiple location sites or a dynamic feature of this kind are particularly interesting because they may have some unique biological functions worthy of our special notice.[2,3] Particularly, as pointed out by Millar *et al.*,[59] recent evidence has indicated that an increasing number of proteins have multiple locations in the cell.

To make Plant-PLoc[58] be able to deal with multiplex plant proteins as well, a predictor called Plant-mPLoc[60] was developed recently, where the character "m" in front of "PLoc" stands for "multiple", meaning that it can be also used to deal with plant proteins with multiple locations. Meanwhile, the scope covered by Plant-mPLoc was further extended from 11 location sites to 12 by adding one more site: Golgi apparatus.

However, Plant-mPLoc 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.[61,62] Such an over-simplified formulation might cause some important information loss and hence limit the prediction quality. (2) In predicting the number of subcellular location sites for a query plant protein, an optimal threshold factor θ^* (see eqn (48) of ref. 63) was utilized without giving its statistical implication and learning process. It would be more instructive if we could find a different approach to determine this in a more natural and intuitive manner. (3) Although a web-server for Plant-mPLoc has been established at http://www.csbio.sjtu.edu.cn/bioinf/plant-multi/, only one query protein sequence at a time is allowed when using the web-server for prediction. For the convenience of users in handling many query protein sequences, such a rigid limit should be improved.

The present study was devoted to develop a new and more powerful predictor, called iLoc-Plant, for predicting plant protein subcellular localization by addressing the above three problems.

To establish a really useful statistical predictor for protein systems, one usually needs to consider the following procedures:[64] (i) select or construct a valid benchmark dataset to train and test the predictor; (ii) formulate the protein samples with an effective mathematical expression that can truly reflect their intrinsic correlation with the attribute to be predicted; (iii) introduce or develop a powerful algorithm (or engine) to operate the prediction; (iv) properly perform cross-validation tests to objectively evaluate the anticipated accuracy of the predictor; (v) establish a user-friendly web-server for the predictor that is accessible to the public. Below, we describe how to realize these steps.

II. Materials

Here, we use the same dataset $\mathbb{S}$ constructed by Chou and Shen[60] in establishing Plant-mPLoc as the benchmark dataset for the current study. The reasons to do so are as follows. (1) The dataset was constructed specially for plant proteins and it can cover 12 subcellular location sites (*cf.* ESI† S1); compared with the other datasets such as those in TargetP[13] and Predotar[30] that could only cover 3 or 4 subcellular locations, the coverage scope of the dataset $\mathbb{S}$ from ref. 60 is much wider. (2) None of the proteins included in $\mathbb{S}$ has $\geq 25\%$ pairwise sequence identity to any other in a same subcellular location; compared with most of the other benchmark datasets in this area, the dataset $\mathbb{S}$ is much more rigorous 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) Using the dataset $\mathbb{S}$ will also facilitate comparison because the results obtained by Plant-mPLoc on $\mathbb{S}$ have been well documented and reported.[60]

The dataset $\mathbb{S}$ contains 978 plant protein sequences, of which 904 occur in one subcellular location, 71 in two locations, 3 in three locations, and none in four or more locations. The dataset covers 12 subcellular locations (Fig. 1); *i.e.*,

$$\mathbb{S} = \mathbb{S}_1 \cup \mathbb{S}_2 \cup \mathbb{S}_3 \cup \mathbb{S}_4 \cup \mathbb{S}_5 \cup \mathbb{S}_6 \cup \ldots \cup \mathbb{S}_{12} \quad (1)$$

where $\mathbb{S}_1$ represents the subset for the subcellular location of "cell membrane", $\mathbb{S}_2$ for "cell wall", $\mathbb{S}_3$ for "chloroplast", and so forth (*cf.* Table 1); while $\cup$ represents the symbol for

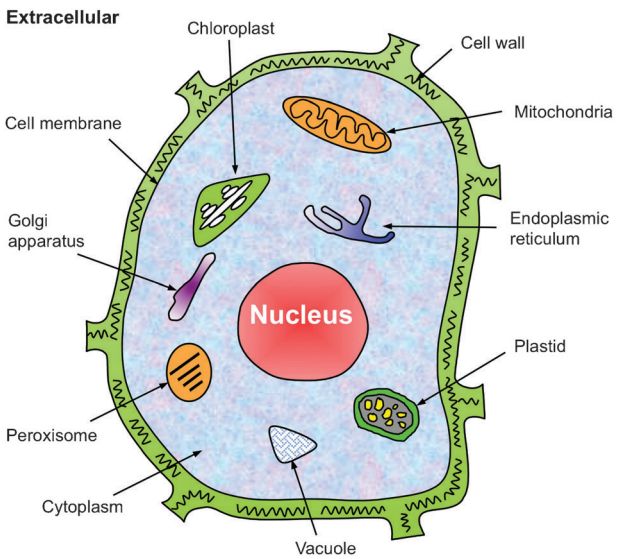

Fig. 1 Illustration to show the 12 subcellular locations of plant proteins. The 12 locations are: (1) cell membrane, (2) cell wall, (3) chloroplast, (4) cytoplasm, (5) endoplasmic reticulum, (6) extracellular, (7) Golgi apparatus, (8) mitochondrion, (9) nucleus, (10) peroxisome, (11) plastid, and (12) vacuole.

Table 1 Breakdown of the plant protein benchmark dataset $\mathbb{S}$ taken from ref. 60. None of the plant proteins included here has $\geq 25\%$ sequence identity to any other in a same subcellular location

Subset	Subcellular location	Number of proteins
$\mathbb{S}_1$	Cell membrane	56
$\mathbb{S}_2$	Cell wall	32
$\mathbb{S}_3$	Chloroplast	286
$\mathbb{S}_4$	Cytoplasm	182
$\mathbb{S}_5$	Endoplasmic reticulum	42
$\mathbb{S}_6$	Extracellular	22
$\mathbb{S}_7$	Golgi apparatus	21
$\mathbb{S}_8$	Mitochondrion	150
$\mathbb{S}_9$	Nucleus	152
$\mathbb{S}_{10}$	Peroxisome	21
$\mathbb{S}_{11}$	Plastid	39
$\mathbb{S}_{12}$	Vacuole	52
Total number of locative proteins N(loc)		1055[a]
Total number of different proteins N(seq)		978[b]

[a] See eqn (36)–(38) of ref. 63 for the definition of the number of locative proteins, and its relation with the number of different proteins. [b] Of the 978 different proteins, 904 have one subcellular location, 71 have two locations, 3 have three locations, and none have four or more locations.

"union" in the set theory. For convenience, hereafter we just use the subscripts of eqn (1) as the codes of the 12 location sites; *i.e.*, "1" for "cell membrane", "2" for "cell wall", "3" for "chloroplast", and so forth (Table 2).

For readers' convenience, the corresponding accession numbers and protein sequences in $\mathbb{S}$ are given in ESI† S1.

Note that because some proteins may occur in two or more locations, the 978 different plant proteins actually correspond to 1055 locative proteins. The concept of "locative proteins" was introduced for studying proteins with multiple subcellular location sites, as elaborated in ref. 63.

Table 2 A comparison of the jackknife success rates achieved by Plant-mPLoc[60] and the current iLoc-Plant on the benchmark dataset $\mathbb{S}$ (*cf.* ESI S1) that covers 12 location sites of plant proteins in which none of the proteins included has $\geq 25\%$ pairwise sequence identity to any other in a same location

Code	Subcellular location site	Success rate by jackknife test	
		Plant-mPLoc[a]	iLoc-Plant[b]
1	Cell membrane	24/56 = 42.9%	39/56 = 69.6%
2	Cell wall	8/32 = 25.0%	19/32 = 59.4%
3	Chloroplast	248/286 = 86.7%	252/286 = 88.1%
4	Cytoplasm	72/182 = 39.6%	114/182 = 62.6%
5	Endoplasmic reticulum	17/42 = 40.5%	21/42 = 50.0%
6	Extracellular	3/22 = 13.6%	2/22 = 9.1%
7	Golgi apparatus	6/21 = 28.6%	16/21 = 76.2%
8	Mitochondrion	114/150 = 76.0%	112/150 = 74.7%
9	Nucleus	136/152 = 89.5%	140/152 = 92.1%
10	Peroxisome	14/21 = 66.7%	6/21 = 28.6%
11	Plastid	4/39 = 10.3%	7/39 = 17.9%
12	Vacuole	26/52 = 50.0%	28/52 = 53.8%
Overall		672/1055 = **63.7%**[c]	756/1055 = **71.7%**[c]

[a] The predictor from ref. 60. [b] The parameter K for the KNN classifier in the current iLoc-Plant was 10, which was derived by optimizing the overall jackknife success rate obtained by iLoc-Plant on the benchmark dataset $\mathbb{S}$. [c] Note that instead of 978 (the number of total different proteins), here we use 1055 (the number of total different virtual proteins) for the denominator. This is because some proteins may have two or more location sites. See footnotes a and b of Table 1 for further explanation.

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.[64] 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 complicate 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 written by

$$\mathbf{P} = \mathrm{R}_1\ \mathrm{R}_2\ \mathrm{R}_3\ \mathrm{R}_4\ \mathrm{R}_5\ \mathrm{R}_6\ \mathrm{R}_7 \ldots \mathrm{R}_L \qquad (2)$$

where R_1 represents the 1st residue of the protein **P**, R_2 the 2nd residue, ..., R_L the L-th residue, and they each belong to one of the 20 native amino acids. In order to identify its subcellular location(s), sequence-similarity-search-based tools, such as BLAST,[65,66] were utilized to search protein databases 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, although this kind of straightforward sequential model is quite intuitive and able to contain the entire information of a protein sequence, it failed to work when the query protein **P** did not have significant sequence similarity to any location-known proteins.

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

Among the discrete models for a protein sequence sample, the simplest one is its amino acid (AA) composition or AAC.[67] According to the AAC-discrete model, the protein **P** of eqn (2) can be formulated by[68]

$$\mathbf{P} = [f_1 \quad f_2 \quad \cdots \quad f_{20}]^{\mathbf{T}} \qquad (3)$$

where f_i ($i = 1, 2, \ldots, 20$) 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 are based on the AAC-discrete model (see, *e.g.*, ref. 5,6,8,9). However, as we can see from eqn (3), 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 limited.

To avoid completely losing the sequence-order information, the pseudo amino acid composition (PseAAC) was proposed to represent the sample of a protein.[14] For a brief description about PseAAC, see a Wikipedia article at http://en.wikipedia.org/wiki/Pseudo_amino_acid_composition. However, according to a recent review,[64] the PseAAC for a protein **P** can be generally formulated as

$$\mathbf{P} = [\psi_1 \quad \psi_2 \quad \ldots \quad \psi_u \quad \ldots \quad \psi_\Omega]^{\mathbf{T}} \qquad (4)$$

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 (2)).

In order to catch the essential and core features of protein samples that are closely correlated with their subcellular localization, let us consider the following approaches through the general form of PseAAC (eqn (4)).

1. GO (Gene Ontology) formulation

GO database[61] was established according to the molecular function, biological process, and cellular component. Accordingly, protein samples defined in a GO database space would be clustered in a way better reflecting their subcellular locations.[63,69] However, in order to incorporate more information, instead of only using 0 and 1 elements as done in ref. 70, here let us consider a different approach as described below.

Step 1. Compression and reorganization of the existing GO numbers. The GO database (version 74.0 released 30 July 2009) contains many GO numbers. However, these numbers do not increase successively and orderly. For easier handling, a reorganization and compression procedure was used 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: 11 118, 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 (4) with $\Omega = 11\,118$, the protein **P** can be formulated as

$$\mathrm{P}_{\mathrm{GO}} = [\psi_1^{\mathrm{G}} \quad \psi_2^{\mathrm{G}} \quad \ldots \quad \psi_u^{\mathrm{G}} \quad \ldots \quad \psi_{11\,118}^{\mathrm{G}}]^{\mathbf{T}} \qquad (5)$$

where ψ_u^{G} ($u = 1,2, \ldots, 11\,118$) are defined *via* the following steps.

Step 3. Use BLAST[71] to search the homologous proteins of the protein **P** from the Swiss-Prot database (version 55.3), with the expect value $E \geq 0.001$ for the BLAST parameter.

Step 4. Those proteins which have $\geq 60\%$ pairwise sequence identity with the protein **P** are collected into a set, $\mathbb{S}_{\mathbf{P}}^{\mathrm{homo}}$, called the "homology set" of **P**. All the elements in $\mathbb{S}_{\mathbf{P}}^{\mathrm{homo}}$ can be deemed as the "representative proteins" of **P**, sharing some similar attributes such as structural conformations and biological functions.[72–74] Because they were retrieved from the Swiss-Prot database, these representative proteins must each have their own accession numbers.

Step 5. Search each of these accession numbers collected in Step 4 against the GO database at http://www.ebi.ac.uk/GOA/ to find the corresponding GO numbers.[75]

Step 6. Based on the results obtained in Step 5, the elements in eqn (5) can be written as

$$\psi_u^{\mathrm{G}} = \frac{\sum_{k=1}^{\mathbb{N}_{\mathbf{P}}^{\mathrm{homo}}} g(u,k)}{\mathbb{N}_{\mathbf{P}}^{\mathrm{homo}}} \quad (u = 1,2,\ldots,11118) \qquad (6)$$

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

$$g(u,k) = \begin{cases} 1, & \text{if the } k\text{-th representative protein hits the } u\text{-th GO\_compress number} \\ 0, & \text{otherwise} \end{cases} \qquad (7)$$

As we can see from eqn (5), the GO formulation derived from the above steps consists of 11 118 real numbers rather than only the elements 0 and 1 as in the GO formulation adopted in ref. 70.

Note that the GO formulation of eqn (5) 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.*, $\mathbb{S}_{\mathbf{P}}^{\mathrm{homo}} = \varnothing$ meaning the homology set $\mathbb{S}_{\mathbf{P}}^{\mathrm{homo}}$ is an empty one; (2) its representative proteins do not contain any useful GO information for statistical prediction based on a given training dataset.

Under such a circumstance, let us consider using the sequential evolution formulation to represent the protein **P**, as described below.

2. SeqEvo (Sequential Evolution) formulation

Biology is a natural science with a historic dimension. All biological species have developed starting out from a very limited number of ancestral species. It is true for protein sequences as well.[74] Their evolution involves changes of single residues, insertions and deletions of several residues,[76] 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 having basically the same biological function and residing in the same subcellular location.

To incorporate the sequential evolution information into the PseAAC of eqn (4), here let us use the information of the PSSM (Position-Specific Scoring Matrix),[71] as described below.

Step 1. According to ref. 71, the sequential evolution information of protein **P** can be expressed by a $20 \times L$ matrix as given by

$$\mathbb{PSSM} = \begin{bmatrix} E_{1\to1}^0 & E_{2\to1}^0 & \cdots & E_{L\to1}^0 \\ E_{1\to2}^0 & E_{2\to2}^0 & \cdots & E_{L\to2}^0 \\ \vdots & \vdots & \vdots & \vdots \\ E_{1\to20}^0 & E_{2\to20}^0 & \cdots & E_{L\to20}^0 \end{bmatrix} \qquad (8)$$

where L is the length of **P** (counted in the total number of its constituent amino acids as shown in eqn (1)), $E_{i\to j}^0$ represents the score of the amino acid residue in the i-th position of the protein sequence 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 $20 \times L$ scores in eqn (8) were generated by using PSI-BLAST[71] 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**. However, according to the formulation of eqn (8), proteins with different lengths will correspond to column-different matrices causing difficulty in developing a predictor able to uniformly cover proteins of any length. To make the descriptor become a size-uniform matrix, let us consider the following steps.

Step 2. Use the elements in $\mathbb{PSSM}$ of eqn (8) to define a new matrix M as formulated by

$$\mathbf{M} = \begin{bmatrix} E_{1\to1} & E_{2\to1} & \dots & E_{L\to1} \\ E_{1\to2} & E_{2\to2} & \dots & E_{L\to2} \\ \vdots & \vdots & \ddots & \vdots \\ E_{1\to20} & E_{2\to20} & \dots & E_{L\to20} \end{bmatrix} \quad (9)$$

with

$$E_{i\to j} = \frac{E^0_{i\to j} - \bar{E}^0_j}{\mathrm{SD}(\bar{E}^0_j)} \quad (i = 1, 2, \dots, L;\ j = 1, 2, \dots, 20) \quad (10)$$

where

$$\bar{E}^0_j = \frac{1}{L}\sum_{i=1}^{L} E^0_{i\to j} \quad (j = 1, 2, \dots, 20) \quad (11)$$

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

$$\mathrm{SD}(\bar{E}^0_j) = \sqrt{\sum_{i=1}^{L} [E^0_{i\to j} - \bar{E}^0_j]^2 / L} \quad (12)$$

is the corresponding standard deviation.

Step 3. Introduce a new matrix generated by multiplying **M** with its own transpose matrix $\mathbf{M}^{\mathbf{T}}$; *i.e.*,

$$\mathbf{M}\mathbf{M}^{\mathbf{T}} = \begin{bmatrix} \sum_{i=1}^{L} E_{i\to1}E_{i\to1} & \sum_{i=1}^{L} E_{i\to1}E_{i\to2} & \dots & \sum_{i=1}^{L} E_{i\to1}E_{i\to20} \\ \sum_{i=1}^{L} E_{i\to2}E_{i\to1} & \sum_{i=1}^{L} E_{i\to2}E_{i\to2} & \dots & \sum_{i=1}^{L} E_{i\to2}E_{i\to20} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{L} E_{i\to20}E_{i\to1} & \sum_{i=1}^{L} E_{i\to20}E_{i\to2} & \dots & \sum_{i=1}^{L} E_{i\to20}E_{i\to20} \end{bmatrix} \quad (13)$$

which contains 20 × 20 = 400 elements. Since $\mathbf{M}\mathbf{M}^{\mathbf{T}}$ is a symmetric matrix, we only need the information of its 210 elements, of which 20 are the diagonal elements and (400 − 20)/2 = 190 are the lower triangular elements, to formulate the protein **P**; *i.e.*, the general PseAAC form of eqn (4) can now be formulated as

$$\mathrm{P}_{\mathrm{Evo}} = [\psi^{\mathrm{E}}_1 \quad \psi^{\mathrm{E}}_2 \quad \dots \quad \psi^{\mathrm{E}}_u \quad \dots \quad \psi^{\mathrm{E}}_{210}]^{\mathbf{T}} \quad (14)$$

where the components ψ^{E}_u (u = 1, 2, …, 210) are, respectively, taken from the 210 diagonal and lower triangular elements of eqn (13) by following a given order, say from left to right and from the 1st row to the last as illustrated by the following equation

$$\begin{bmatrix} (1) & & & & \\ (2) & (3) & & & \\ (4) & (5) & (6) & & \\ \vdots & \vdots & \vdots & \ddots & \\ (191) & (192) & (193) & \dots & (210) \end{bmatrix} \quad (15)$$

where the numbers in parentheses indicate the order of elements taken from eqn (13) for (14).

3. The self-consistency formulation principle

Regardless of which formulation is used to represent protein samples, the following self-consistency principle must be observed during the course of prediction: if the query protein **P** is defined in the form of P_{GO} (see eqn (5)), then all the protein samples used to train the prediction engine should also be expressed in the GO formulation; if the query protein is defined in the form of $\mathrm{P}_{\mathrm{Evo}}$ (see eqn (14)), then all the training data should be expressed in the SeqEvo formulation as well.

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

4. Multi-Label KNN (*K*-Nearest Neighbor) classifier

In this study, we introduce a novel classifier, called the multi-label KNN or abbreviated as ML-KNN classifier, to predict the subcellular localization for the systems that contain both single-location and multiple-location proteins.

Suppose the m-th subset $\mathbb{S}_m$ of $\mathbb{S}$ (eqn (1)) contains N_m plant proteins, and $\mathbf{P}(m,j)$ is the j-th one in that subset. Thus, we have

$$\mathbf{P}(m,j) = \begin{cases} \mathbf{P}_{\mathrm{GO}}(m,j), & \text{in GO space} \\ \mathbf{P}_{\mathrm{Evo}}(m,j), & \text{in SeqEvo space} \end{cases} \quad (m = 1, 2, \dots, 12;\ j = 1, 2, \dots, N_m) \quad (16)$$

where $\mathbf{P}_{\mathrm{GO}}(m,j)$ and $\mathbf{P}_{\mathrm{Evo}}$ (m,j) have the same forms as $\mathbf{P}_{\mathrm{GO}}$ (eqn (5)) and $\mathbf{P}_{\mathrm{Evo}}$ (eqn (14)), respectively; the only difference is that the corresponding constituent elements are derived from the amino acid sequence of $\mathbf{P}(m,j)$ instead of **P**.

In sequence analysis, there are many different scales to define the distance between two proteins, such as Euclidean distance, Hamming distance,[77] and Mahalanobis distance.[68,78,79] In ref. 60, the distance between $\mathbf{P}(m,j)$ and **P** was defined by $1 - \cos^{-1}[\mathbf{P}, \mathbf{P}(m,j)]$. 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 distance between **P** and $\mathbf{P}(m,j)$ is defined here by

$$D\{\mathbf{P},\mathbf{P}(m,j)\} = \|\mathbf{P} - \mathbf{P}(m,j)\| \quad (17)$$

where $\|\mathbf{P} - \mathbf{P}(m,j)\|$ represents the module of the vector difference between **P** and $\mathbf{P}(m,j)$ in the Euclidean space. According to eqn (17), 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 $\mathbf{P}^*_1, \mathbf{P}^*_2, \dots, \mathbf{P}^*_K$ are the K nearest neighbor proteins to the protein **P** that forms a set denoted by $\mathbb{S}^{\mathbf{P}}_K$, which is a subset of $\mathbb{S}$; *i.e.*, $\mathbb{S}^{\mathbf{P}}_K \subseteq \mathbb{S}$. Based on the K nearest neighbor proteins in $\mathbb{S}^{\mathbf{P}}_K$, let us define an accumulation-layer (AL) scale, given by

$$\mathbb{Q}(\mathbf{P}, K) = \{\rho^K_1 \quad \rho^K_2 \quad \dots \quad \rho^K_m \quad \dots \quad \rho^K_M\} \quad (18)$$

where

$$\rho_m = \frac{\sum_{i=1}^{K} \delta(\mathbf{P}^*_i, m)}{\mathbb{N}^*_K} \quad (m = 1, 2, \dots, M) \quad (19)$$

where M = 12 is the number of total subcellular locations considered in this study, while

$$\delta(\mathbf{P}^*_i, m) = \begin{cases} 1, & \text{if } \mathbf{P}^*_i \text{ belongs to the } m\text{-th location} \\ 0, & \text{otherwise} \end{cases} \quad (20)$$

and

$$\mathbb{N}_K^* = \sum_{m=1}^{M} \sum_{i=1}^{M} \delta(\mathbf{P}_i^*, m) \qquad (21)$$

Note that $\mathbb{N}_K^* \geq K$ because a protein may belong to one or more subcellular location sites 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 $\mathbb{S}$. For example, suppose **P*** is the nearest protein to **P** in $\mathbb{S}$. If **P*** has only one subcellular location, then **P** will also have only one location; if **P*** has two subcellular locations, then **P** will also have two locations; and so forth. In general, if **P*** belongs to $\mathfrak{M}$ different location sites, then **P** will be predicted to have the same number, $\mathfrak{M}$, of subcellular locations as well, as can be formulated by

$$\mathfrak{M} = \text{Num}\{\mathbf{P}^* \Rightarrow \mathbb{L}\} = \text{Num}\{\mathbf{P} \Rightarrow \mathbb{L}\} \qquad (22)$$

where $\mathfrak{M}$ is an integer ($\leq M = 12$), Num $\{\mathbf{P}^* \Rightarrow \mathbb{L}\}$ represents the number of different subcellular locations to which **P*** belongs, and Num $\{\mathbf{P} \Rightarrow \mathbb{L}\}$ the number of different subcellular locations to which **P** belongs.

Step 2. However, the concrete location site(s) to which **P** belongs will not be determined by the location site(s) of **P***, but by the element(s) in eqn (18) that has (have) the highest score(s), as can be expressed by $\{\ell\}$, the subscript(s) of eqn (1). For example, if **P** is found to belong to only one location ($\mathfrak{M} = 1$) in Step 1, and the highest score in eqn (18) is ρ_2^K, then **P** will be predicted as $\{\ell\} = \{2\}$ meaning that it belongs to $\mathbb{S}_2$ or resides at "cell wall" (*cf.* Table 1). If **P** is found to belong to three locations ($\mathfrak{M} = 3$), and the first three highest scores in eqn (18) are ρ_1^K, ρ_9^K, and ρ_{12}^K, then **P** will be predicted as $\{\ell\} = \{1, 9, 12\}$ meaning that it belongs to $\mathbb{S}_1$, $\mathbb{S}_9$, and $\mathbb{S}_{12}$ or resides simultaneously at "cell membrane", "nucleus", and "vacuole". And so forth. In other words, the concrete predicted subcellular location(s) can be formulated as

$$\{\ell\} = \text{Max} \rhd_{\text{Sub}}^{\mathfrak{M}} \{\rho_1^K \quad \rho_2^K \quad \ldots \quad \rho_m^K \quad \ldots \quad \rho_M^K\} \ (\mathfrak{M} \leq M) \qquad (23)$$

where the operator "Max $\rhd_{\text{Sub}}^{\mathfrak{M}}$" means identifying the $\mathfrak{M}$ highest scores for the elements in the brackets right after it, followed by taking their $\mathfrak{M}$ subscripts. The value for the parameter K in eqn (23) 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-Plant, which can be used to predict the subcellular localization of both singleplex and multiplex plant proteins. To provide an intuitive picture, a flowchart is provided in Fig. 2 to illustrate the prediction process of iLoc-Plant.

5. Protocol guide

For user's convenience, a web-server for iLoc-Plant has been established. Below, we give a step-by-step guide on how to use it to get the desired results.

Step 1. Open the web server at site http://icpr.jci.edu.cn/bioinfo/iLoc-Plant and you will see the top page of the

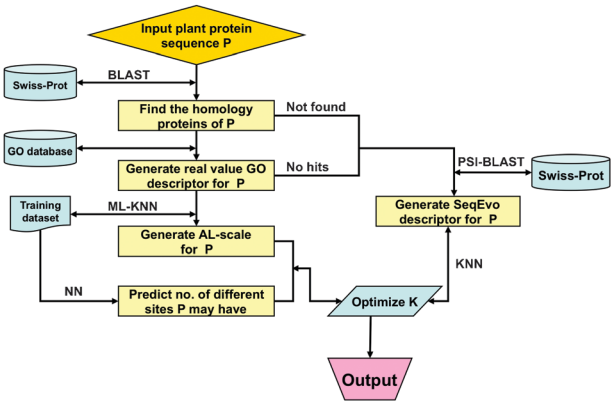

Fig. 2 A flowchart to show the prediction process of iLoc-Plant.

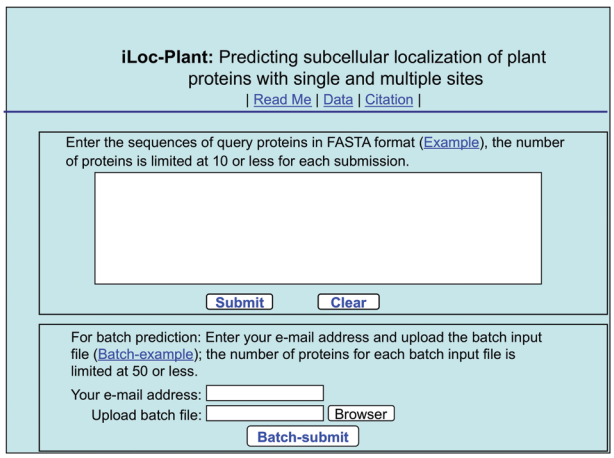

Fig. 3 A semi-screenshot to show the top page of the iLoc-Plant web-server. Its website address is http://icpr.jci.edu.cn/bioinfo/iLoc-Plant.

predictor on your computer screen, as shown in Fig. 3. Click on the Read Me button to see a brief introduction about iLoc-Plant predictor and the caveat when using it.

Step 2. Either type or copy and paste the query protein sequence into the input box shown at the center of Fig. 3. The input sequence should be in the FASTA format. A sequence in 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 FASTA format can be seen by clicking on the Example button right above the input box. For more information about FASTA format, visit http://en.wikipedia.org/wiki/Fasta_format. Different from Plant-mPLoc,[60] where only one query protein sequence at a time is allowed 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

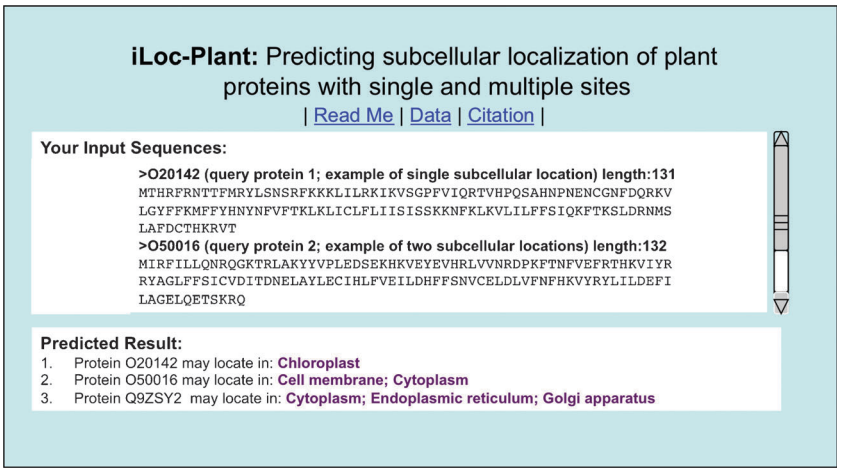

Fig. 4 A semi-screenshot to show the output of iLoc-Plant. The input was taken from the three protein sequences listed in the Example window of the iLoc-Plant web-server (*cf.* Fig. 3).

the Submit button, you will see Fig. 4 on your screen, indicating that the predicted result for the 1st query protein is "Chloroplast", that for the 2nd one is "Cell membrane; Cytoplasm", and that for the 3rd one is "Cytoplasm; Endoplasmic reticulum; Golgi apparatus". In other words, the 1st query protein (O20142) is a single-location one residing at "chloroplast" only, the 2nd one (O50016) can simultaneously occur in two different sites ("cell membrane" and "cytoplasm"), and the 3rd one (Q9ZSY2) can simultaneously occur in three different sites ("cytoplasm", "endoplasmic reticulum", and "Golgi apparatus"). 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 on 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 batch input file, click on the button Batch-example. The maximum number of the query proteins for each batch input file is 50. After clicking the button Batch-submit, you will see "Your batch job is under computation; once the results are available, you will be notified by e-mail." Note that if you submit a batch input file from an Apple computer, although it looks like to be in the FASTA format, your input might change to non-FASTA format in the server end and cause errors. Under such a circumstance, the safest way is to submit your input file in a pdf format.

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

Step 6. Click on the Data button to download the benchmark datasets used to train and test the iLoc-Plant 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 plant protein is known not to be in one of the 12 locations as shown in Fig. 1, stop the prediction because the result thus obtained will not make any sense.

IV. Results and discussion

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.[64] 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.[80] Because a subsampling test and a jackknife test can be performed with one benchmark dataset and an 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-validation. However, as illustrated by eqn (28)–(32) of ref. 64 and the relevant text therein, among the three cross-validation methods, the jackknife test is deemed the least arbitrary that can always yield a unique result for a given benchmark dataset and hence has been widely recognized and increasingly used to examine the power of various predictors (see, *e.g.*, ref. 81–86). Accordingly, in this study, the jackknife test will be adopted to evaluate the power of iLoc-Plant as well.

However, 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 elaborated in a recent review.[64]

As mentioned in the Materials section, the benchmark dataset used in this study is $\mathbb{S}$ (*cf.* ESI† S1), which is the same benchmark dataset constructed in ref. 60 for Plant-mPLoc.

Actually, for such a dataset containing both single-location and multiple-location plant proteins distributed among 12 subcellular location sites, so far only one existing predictor,

i.e., Plant-mPLoc,[60] has the capacity to deal with it. Therefore, to demonstrate the power of the current predictor, it would suffice to just compare iLoc-Plant with Plant-mPLoc.[60]

Listed in Table 2 are the results obtained with Plant-mPLoc[60] and iLoc-Plant on the aforementioned benchmark dataset $\mathbb{S}$ by the jackknife test. As we can see from Table 2, for such a stringent and complicated benchmark dataset, the overall success rate achieved by iLoc-Plant is over 71.7%, which is 8% higher than that by Plant-mPLoc.[60]

Note that during the course of the jackknife test by Plant-mPLoc and iLoc-Plant, the false positives (over-predictions) and false negatives (under-predictions) were also taken into account to reduce the scores in 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.[63]

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

$$\Lambda = \frac{\sum_{i=1}^{N} \Delta(i)}{N} \tag{24}$$

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

$$\Delta(i) = \begin{cases} 1, & \text{if all the subcellular locations of the } i\text{-th protein are correctly predicted without any overprediction} \\ 0, & \text{otherwise} \end{cases} \tag{25}$$

The absolute true success rate as defined by eqn (24) and (25) is particularly useful when dealing with a system consisting of both single-location proteins and multiple-location proteins. According to its definition, for a protein belonging to, say, two subcellular locations, if only one of the two is correctly predicted, or the predicted result contains a location not belonging to the two, 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[60,63] in measuring the success rate. However, even if such a stringent criterion was used on the same benchmark dataset by the jackknife test, the overall absolute true success rate achieved by iLoc-Plant was 666/978 = 68.1%.

Why can iLoc-Plant enhance the success rate so remarkably? One of the key reasons is that the GO formulation for protein samples in iLoc-Plant contains more information than that in Plant-mPLoc,[60] as can be illustrated as follows. Suppose a query protein **P**, according to Steps 3 and 4 in the section of "GO (Gene Ontology) Formulation", we found 12 proteins that were homologous to it; *i.e.*, $\mathbb{N}_{\mathbf{P}}^{\text{homo}} = 12$. Of the 12 homologous proteins, 12 hit GO_compress:01150, 12 hit GO_compress:01163, 12 hit GO_compress:02881, 12 hit GO_compress:02886, 8 hit GO_compress:04822, 9 hit GO_compress:04841, 1 hit GO_compress:05021, and 12 hit GO_compress:09878. Substituting these data into eqn (6) and (7), we have

$$\psi_u^{\text{G}}(\mathbf{P}) = \begin{cases} 12/12 = 1.0, & \text{if } u = 1150 \\ 12/12 = 1.0, & \text{if } u = 1163 \\ 12/12 = 1.0, & \text{if } u = 2881 \\ 12/12 = 1.0, & \text{if } u = 2886 \\ 8/12 = 0.667, & \text{if } u = 4822 \\ 9/12 = 0.75, & \text{if } u = 4841 \\ 1/12 = 0.083, & \text{if } u = 5021 \\ 12/12 = 1.0, & \text{if } u = 9878 \\ 0.0, & \text{otherwise} \end{cases} \quad (u = 1, 2, \ldots, 11\,118) \tag{26}$$

In contrast, if the same protein was represented according to the formulation in Plant-mPLoc,[60] it would be

$$\psi_u^{\text{G}}(\mathbf{P}) = \begin{cases} 1, & \text{if } u = 1150 \\ 1, & \text{if } u = 1163 \\ 1, & \text{if } u = 2881 \\ 1, & \text{if } u = 2886 \\ 1, & \text{if } u = 4822 \\ 1, & \text{if } u = 4841 \\ 1, & \text{if } u = 5021 \\ 1, & \text{if } u = 9878 \\ 0, & \text{otherwise} \end{cases} \quad (u = 1, 2, \ldots, 11\,118) \tag{27}$$

As can be clearly seen by comparing eqn (26) with (27), although the elements in the 1150th, 1163rd, 2881st, 2886th, 4822nd, 4841st, 5021st, and 9878th component are all not zero in both formulations, in the formulation (eqn (27)) as defined in Plant-mPLoc[60] all these eight elements are equal to 1, completely ignoring their weights. In other words, the GO formulation in the current iLoc-Plant contains more information than that in Plant-mPLoc[60] and hence leads to better prediction results.

The other reason is that in Plant-mPLoc[60] the number of the subcellular location sites for a query protein was determined by a threshold factor θ^* (*cf.* eqn (48) in ref. 63) that actually functioned as a "black box" without providing any physico-chemical rationale. In contrast, it is very much different in the current iLoc-Plant as reflected by the fact that the number of the subcellular location sites for a query protein is determined according to the nearest neighbor (NN) principle (*cf.* eqn (22)), and that its concrete location sites are determined according to the accumulation-layer scale (*cf.* eqn (18) and (23)).

Although Table 2 and the above analysis have provided very compelling evidence to indicate that iLoc-Plant is superior to Plant-mPLoc,[60] people might ask how about comparing iLoc-Plant with TargetP[13] and Predotar,[30] two popular predictors widely used for predicting the subcellular locations of plant proteins. As mentioned in Introduction, the two predictors only cover three or four location sites. Therefore, it can be easily conceived that they would yield even much

lower success rates when tested by the current benchmark dataset that covers twelve location sites.

Actually, even if tested by a benchmark dataset within the scope that can be covered by TargetP[13] or Predotar,[30] the success rate achieved by the current iLoc-Plant predictor is also much higher than those by the two predictors, as demonstrated below.

Let us compare iLoc-Plant with TargetP[13] first. The TargetP predictor also has a web-server at http://www.cbs.dtu.dk/services/TargetP/, with a built-in training dataset covering the following four items: "mitochondria", "chloroplast", "secretory pathway", and "other". Since the "secretory pathway" is not a final destination of subcellular location as annotated in Swiss-Prot databank, it was removed from the comparison. Also, the location of "other" is not a clear site for comparison, and should be removed as well. Thus, in order to compare TargetP with the new predictor iLoc-Plant, let us construct an independent testing dataset by randomly picking testing proteins according to the following criteria: (i) they must belong to plant proteins, as clearly annotated in Swiss-Prot database; (ii) they must neither occur in the training dataset of TargetP nor occur in the training dataset of iLoc-Plant in order to avoid the memory bias; (iii) their experimentally observed subcellular locations are known as clearly annotated in Swiss-Prot database, and also these locations must be within the scope covered by TargetP as a compromise for rationally utilizing its web-server. By following the above procedures, we obtained a degenerate independent testing dataset consisting of 100 plant proteins, of which 50 belong to chloroplast and 50 belong to mitochondrion. The accession numbers and sequences of these 100 plant proteins are given in ESI† S2a.

The results predicted by TargetP[13] and the current iLoc-Plant for each of the 100 independent testing proteins are listed in ESI† S2b, where for facilitating comparison, the corresponding experimental results are also given. By examining ESI† S2b, we can see the following. (1) Many proteins whose subcellular locations were misidentified by TargetP have been corrected by iLoc-Plant. (2) Many proteins, which were identified by TargetP as belonging to the location of "other", have been identified as "chloroplast" or "mitochondrion", fully consistent with experimental observations. (3) The overall success rate achieved by iLoc-Plant for the 100 independent proteins was 94%, which was 60% higher than that by TargetP[13] (see Table 3).

To demonstrate the power of iLoc-Plant in dealing with proteins with multiple location sites, let us consider the dataset in ESI† S2c. It contains 17 protein sequences that were randomly picked from Swiss-Prot database under the following conditions: (1) they all belong to plant proteins; (2) none of them occurs in the training dataset of iLoc-Plant; (3) each of them is known to belong to two or more subcellular locations according to their experimental annotations. The outcomes generated by inputting the 17 sequences into the web-servers iLoc-Plant and TargetP[13] are, respectively, listed in ESI† S2d, from which we can see that for all the 17 proteins their multiple-location sites were perfectly predicted by iLoc-Plant without any false positive and false negative. This kind of capacity of iLoc-Plant in dealing with multiple-location proteins is far beyond the reach of TargetP.[13]

Table 3 A head-to-head comparison between iLoc-Plant and TargetP[13] by the success rates in predicting the subcellular locations for the 100 proteins in the independent dataset as given in the ESI S2a

Subcellular location	iLoc-Plant[a]	TargetP[b]
Chloroplast	50/50 = 100.00%	12/50 = 24.00%
Mitochondrial	44/50 = 88%	22/50 = 44.00%
Overall	94/100 = **94.00%**	34/100 = **34.00%**

[a] Here the following absolute true scale (eqn (24)) was used to score the prediction point for the results identified by iLoc-Plant: 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] The web-server of TargetP is at http://www.cbs.dtu.dk/services/TargetP/.

Now, let us compare iLoc-Plant with Predotar.[30] The web-server of Predotar is at: http://urgi.versailles.inra.fr/predotar/predotar.html, with a built-in training dataset covering the following four items: "endoplasmic reticulum", "mitochondrion", "plastid", and "other". Since the term "other" is not a clear description for subcellular location, it was removed from comparison. Thus, by following the aforementioned similar criteria as in constructing the independent dataset for comparing TargetP with iLoc-Plant, we also constructed a degenerate independent dataset to compare Predotar[30] with iLoc-Plant. The dataset consists of 150 plant proteins, of which 50 belong to endoplasmic reticulum, 50 belong to mitochondrion, and 50 belong to plastid. The accession numbers and sequences of these 150 proteins are given in ESI† S3a. The results predicted by Predotar[30] and the current iLoc-Plant for the 150 independent testing proteins and their corresponding experimental results are listed in ESI† S3b, from which we can see the following. (1) Subcellular locations of many proteins correctly identified by iLoc-Plant were not identified by Predotar[30] although all these location sites are within its coverage scope. (2) Subcellular locations of many proteins misidentified by Predotar have been corrected by iLoc-Plant. (3) The overall success rate obtained by iLoc-Plant for the 150 independent proteins was 82.67%, which was about 50% higher than that by Predotar[30] (see Table 4).

Furthermore, to show the difference between iLoc-Plant and Predotar[30] in dealing with multiple-location proteins, let us consider the dataset in ESI† S3c. It contains 14 protein sequences that were randomly picked from Swiss-Prot

Table 4 A head-to-head comparison between iLoc-Plant and Predotar[30] by the success rates in predicting the subcellular locations for the 150 proteins in the independent dataset as given in the ESI S3a

Subcellular location	iLoc-Plant[a]	Predotar[b]
Endoplasmic reticulum	36/50 = 72.00%	22/50 = 44.00%
Mitochondrial	50/50 = 100.00%	20/50 = 40.00
Plastid	38/50 = 76.00%	4/50 = 8.00%
Overall	124/150 = **82.67%**	46/150 = **30.67%**

[a] Here the following absolute true scale (eqn (24)) was used to score the prediction point for the results identified by iLoc-Plant: 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] The web-server of Predotar is at http://urgi.versailles.inra.fr/predotar/predotar.html.

database under the following conditions: (1) they all belong to plant proteins; (2) none of them occurs in the training dataset of iLoc-Plant; (3) each of them is known to belong to two or more subcellular locations according to their experimental annotations. The outcomes generated by inputting the 14 sequences into the web-servers iLoc-Plant and Predotar[30] are, respectively, listed in ESI† S3d, from which we can see that of the 14 proteins, multiple-location sites for 13 were perfectly predicted by iLoc-Plant without any false positive and false negative, while 1 was partially correctly predicted. This kind of capacity of iLoc-Plant in dealing with multiple-location proteins is far beyond the reach of Predotar.[30]

From the above comparisons of iLoc-Plant with Plant-mPLoc,[60] TargetP,[13] and Predotar,[30] 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 covering only three subcellular locations each containing the same number of proteins, the overall success rate obtained by random assignments would generally be $1/3 \simeq 33.3\%$; while for a benchmark dataset covering 12 subcellular locations, the overall success rate obtained by random assignments would be only $1/12 \simeq 8.3\%$. This means that the former is more than four times the latter.

Also, a predictor tested by jackknife cross-validation 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 a same subset (subcellular location). That is why the overall success rate achieved by iLoc-Plant was only 71.7% when tested by the jackknife cross-validation on the benchmark dataset given in ESI† S1 but was 94% (Table 3) and over 82% (Table 4) when tested by the independent datasets given in ESI† S2a and ESI† 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-Plant are significantly higher than those by its counterparts, as shown in Tables 2–4.

Meanwhile, it has also become understandable why the overall success rates as originally reported for TargetP[13] and Predotar[30] were over-estimated. This is because the benchmark datasets adopted by TargetP[13] or Predotar[30] only cover less than one-third of the location sites that are covered by the current iLoc-Plant. Besides, the benchmark datasets used by TargetP[13] and Predotar[30] to estimate their success rates contain many homologous sequences. For the benchmark dataset used by Predotar,[30] the cutoff threshold was set as 80%, meaning that only those sequences which have $\geq 80\%$ pairwise sequence identity to any other in a same subset were excluded;[30] while for the benchmark dataset used in TargetP,[13] even such a cutoff percentage was not indicated. 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 a same subset, the benchmark datasets adopted in Predotar and TargetP are much less stringent and hence cannot avoid homologous bias and overestimation.

V. Conclusions

Prediction of protein subcellular localization is a challenging problem, particularly when the system concerned contains both singleplex and multiplex proteins. The reasons why iLoc-Plant can achieve higher success rates than Plant-mPLoc are as follows. (1) The GO formulation used to represent protein samples in iLoc-Plant is formed by the probabilities of hits (*cf.* eqn (6) and (7)) and hence contains more information than that in Plant-mPLoc[70] where only the number "0" or "1" was used regardless of how many hits were found for the corresponding component in the GO formulation. (2) The accumulation-layer scale has been introduced in iLoc-Plant which is more natural and effective in dealing with proteins having both single and multiple subcellular locations.

The protocol guide as presented in this paper is particularly helpful for the vast majority of experimental scientists, who wish to utilize iLoc-Plant to get the desired results without the need to understand the detailed mathematics.

Acknowledgements

The authors wish to thank the two anonymous referees, whose constructive comments 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 Natural Science Foundation of Jiangxi Province, China (2010GQS0127), 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. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

References

1 J. S. Ehrlich, M. D. Hansen and W. J. Nelson, *Dev. Cell*, 2002, **3**, 259–270.
2 E. Glory and R. F. Murphy, *Dev. Cell*, 2007, **12**, 7–16.
3 C. Smith, http://www.biocompare.com/Articles/TechnologySpotlight/976/Subcellular-Targeting-Of-Proteins-And-Drugs.html, 2008.
4 K. Nakai and M. Kanehisa, *Proteins: Struct., Funct., Genet.*, 1991, **11**, 95–110.
5 H. Nakashima and K. J. Nishikawa, *Mol. Biol.*, 1994, **238**, 54–61.
6 J. Cedano, P. Aloy, J. A. Pérez-Pons and E. J. Querol, *Mol. Biol.*, 1997, **266**, 594–600.
7 K. Nakai and P. Horton, *Trends Biochem. Sci.*, 1999, **24**, 34–36.
8 A. Reinhardt and T. Hubbard, *Nucleic Acids Res.*, 1998, **26**, 2230–2236.
9 K. C. Chou and D. W. Elrod, *Protein Eng.*, 1999, **12**, 107–118.
10 Z. Yuan, *FEBS Lett.*, 1999, **451**, 23–26.
11 K. Nakai, *Adv. Protein Chem.*, 2000, **54**, 277–344.
12 R. F. Murphy, M. V. Boland and M. Velliste, *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, 2000, **8**, 251–259.
13 O. Emanuelsson, H. Nielsen, S. Brunak and G. von Heijne, *J. Mol. Biol.*, 2000, **300**, 1005–1016.
14 K. C. Chou, *Proteins: Struct., Funct., Genet.*, 2001, **43**, 246–255 (Erratum: *ibid.*, 2001, **44**, 60).
15 Z. P. Feng, *Biopolymers*, 2001, **58**, 491–499.
16 S. Hua and Z. Sun, *Bioinformatics*, 2001, **17**, 721–728.
17 Z. P. Feng and C. T. Zhang, *Int. J. Biol. Macromol.*, 2001, **28**, 255–261.
18 Z. P. Feng, *In silico Biol.*, 2002, **2**, 291–303.

19 K. C. Chou and Y. D. Cai, *J. Biol. Chem.*, 2002, **277**, 45765–45769.
20 G. P. Zhou and K. Doctor, *Proteins: Struct., Funct., Genet.*, 2003, **50**, 44–48.
21 Y. X. Pan, Z. Z. Zhang, Z. M. Guo, G. Y. Feng, Z. D. Huang and L. He, *J. Protein Chem.*, 2003, **22**, 395–402.
22 K. J. Park and M. Kanehisa, *Bioinformatics*, 2003, **19**, 1656–1663.
23 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.
24 Y. Huang and Y. Li, *Bioinformatics*, 2004, **20**, 21–28.
25 X. Xiao, S. Shao, Y. Ding, Z. Huang, Y. Huang and K. C. Chou, *Amino Acids*, 2005, **28**, 57–61.
26 Y. Gao, S. H. Shao, X. Xiao, Y. S. Ding, Y. S. Huang, Z. D. Huang and K. C. Chou, *Amino Acids*, 2005, **28**, 373–376.
27 Z. Lei and Y. Dai, *BMC Bioinf.*, 2005, **6**, 291.
28 H. B. Shen and K. C. Chou, *Biochem. Biophys. Res. Commun.*, 2005, **337**, 752–756.
29 A. Garg, M. Bhasin and G. P. Raghava, *J. Biol. Chem.*, 2005, **280**, 14427–14432.
30 I. Small, N. Peeters, F. Legeai and C. Lurin, *Proteomics*, 2004, **4**, 1581–1590.
31 S. Matsuda, J. P. Vert, H. Saigo, N. Ueda, H. Toh and T. Akutsu, *Protein Sci.*, 2005, **14**, 2804–2813.
32 J. L. Gardy, M. R. Laird, F. Chen, S. Rey, C. J. Walsh, M. Ester and F. S. Brinkman, *Bioinformatics*, 2005, **21**, 617–623.
33 Q. B. Gao, Z. Z. Wang, C. Yan and Y. H. Du, *FEBS Lett.*, 2005, **579**, 3444–3448.
34 J. Guo, Y. Lin and X. Liu, *Proteomics*, 2006, **6**, 5099–5105.
35 X. Xiao, S. H. Shao, Y. S. Ding, Z. D. Huang and K. C. Chou, *Amino Acids*, 2006, **30**, 49–54.
36 A. Hoglund, P. Donnes, T. Blum, H. W. Adolph and O. Kohlbacher, *Bioinformatics*, 2006, **22**, 1158–1165.
37 K. Lee, D. W. Kim, D. Na, K. H. Lee and D. Lee, *Nucleic Acids Res.*, 2006, **34**, 4655–4666.
38 Z. H. Zhang, Z. H. Wang, Z. R. Zhang and Y. X. Wang, *FEBS Lett.*, 2006, **580**, 6169–6174.
39 J. Y. Shi, S. W. Zhang, Q. Pan, Y.-M. Cheng and J. Xie, *Amino Acids*, 2007, **33**, 69–74.
40 Y. L. Chen and Q. Z. Li, *J. Theor. Biol.*, 2007, **248**, 377–381.
41 Y. L. Chen and Q. Z. Li, *J. Theor. Biol.*, 2007, **245**, 775–783.
42 P. Mundra, M. Kumar, K. K. Kumar, V. K. Jayaraman and B. D. Kulkarni, *Pattern Recogn. Lett.*, 2007, **28**, 1610–1615.
43 H. Lin, H. Ding, F. B. Feng-Biao Guo, A. Y. Zhang and J. Huang, *Protein Pept. Lett.*, 2008, **15**, 739–744.
44 J. Y. Shi, S. W. Zhang, Q. Pan and G. P. Zhou, *Amino Acids*, 2008, **35**, 321–327.
45 F. M. Li and Q. Z. Li, *Protein Pept. Lett.*, 2008, **15**, 612–616.
46 E. Tantoso and X. B. Li, *Amino Acids*, 2008, **35**, 345–353.
47 X. Jiang, R. Wei, T. L. Zhang and Q. Gu, *Protein Pept. Lett.*, 2008, **15**, 392–396.
48 X. B. Zhou, C. Chen, Z. C. Li and X. Y. Zou, *Amino Acids*, 2008, **35**, 383–388.
49 Y. S. Ding and T. L. Zhang, *Pattern Recogn. Lett.*, 2008, **29**, 1887–1892.
50 S. W. Zhang, Y. L. Zhang, H. F. Yang, C. H. Zhao and Q. Pan, *Amino Acids*, 2008, **34**, 565–572.
51 Y. Jin, B. Niu, K. Y. Feng, W. C. Lu, Y. D. Cai and G. Z. Li, *Protein Pept. Lett.*, 2008, **15**, 286–289.
52 H. Lin, H. Wang, H. Ding, Y. L. Chen and Q. Z. Li, *Acta Biotheoretica*, 2009, **57**, 321–330.
53 L. Zhang, B. Liao, D. Li and W. Zhu, *J. Theor. Biol.*, 2009, **259**, 361–365.
54 Y. H. Zeng, Y. Z. Guo, R. Q. Xiao, L. Yang, L. Z. Yu and M. L. Li, *J. Theor. Biol.*, 2009, **259**, 366–372.
55 P. Du, S. Cao and Y. Li, *J. Theor. Biol.*, 2009, **261**, 330–335.
56 Y. D. Cai, J. He, X. Li, K. Feng, L. Lu, X. Kong and W. Lu, *Protein Pept. Lett.*, 2010, **17**, 464–472.
57 O. Emanuelsson, H. Nielsen and G. von Heijne, *Protein Sci.*, 1999, **8**, 978–984.
58 K. C. Chou and H. B. Shen, *J. Cell. Biochem.*, 2007, **100**, 665–678.
59 A. H. Millar, C. Carrie, B. Pogson and J. Whelan, *Plant Cell*, 2009, **21**, 1625–1631.
60 K. C. Chou and H. B. Shen, *PLoS One*, 2010, **5**, e11335.
61 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.
62 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.
63 K. C. Chou and H. B. Shen, *Anal. Biochem.*, 2007, **370**, 1–16.
64 K. C. Chou, *J. Theor. Biol.*, 2011, **273**, 236–247.
65 S. F. Altschul, in *Theoretical and Computational Methods in Genome Research*, ed. S. Suhai, Plenum, New York, 1997, pp. 1–14.
66 J. C. Wootton and S. Federhen, *Comput. Chem.*, 1993, **17**, 149–163.
67 H. Nakashima, K. Nishikawa and T. Ooi, *J. Biochem.*, 1986, **99**, 152–162.
68 K. C. Chou and C. T. Zhang, *J. Biol. Chem.*, 1994, **269**, 22014–22020.
69 K. C. Chou and H. B. Shen, *Nat. Sci.*, 2010, **2**, 1090–1103.
70 K. C. Chou and H. B. Shen, *PLoS One*, 2010, **5**, e9931.
71 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.
72 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.
73 M. Gerstein and J. M. Thornton, *Curr. Opin. Struct. Biol.*, 2003, **13**, 341–343.
74 K. C. Chou, *Curr. Med. Chem.*, 2004, **11**, 2105–2134.
75 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.
76 K. C. Chou, *FEBS Lett.*, 1995, **363**, 123–126.
77 K. V. Mardia, J. T. Kent and J. M. Bibby, *Multivariate Analysis*, ch. 11 Discriminant Analysis; ch. 12 Multivariate analysis of variance; ch. 13 cluster analysis, Academic Press, London, 1979, pp. 322–381.
78 P. C. Mahalanobis, *Proc. Natl. Inst. Sci. India*, 1936, **2**, 49–55.
79 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.
80 K. C. Chou and C. T. Zhang, *Crit. Rev. Biochem. Mol. Biol.*, 1995, **30**, 275–349.
81 S. Jahandideh, S. Hoseini, M. Jahandideh, A. Hoseini and F. M. Disfani, *J. Theor. Biol.*, 2009, **259**, 517–522.
82 S. Kannan, A. M. Hauth and G. Burger, *Protein Pept. Lett.*, 2008, **15**, 1107–1116.
83 M. Masso and I. I. Vaisman, *J. Theor. Biol.*, 2010, **266**, 560–568.
84 H. Mohabatkar, *Protein Pept. Lett.*, 2010, **17**, 1207–1214.
85 S. S. Sahu and G. Panda, *Comput. Biol. Chem.*, 2010, **34**, 320–327.
86 X. Xiao, P. Wang and K. C. Chou, *Mol. BioSyst.*, 2011, **7**, 911–919.