Molecular BioSystems

RSC Publishing

PAPER

View Article Online
View Journal

iLoc-Animal: a multi-label learning classifier for predicting subcellular localization of animal proteins†

Cite this: DOI: 10.1039/c3mb25466f

Wei-Zhong Lin,[ab] Jian-An Fang,[a] Xuan Xiao*[bc] and Kuo-Chen Chou*[c]

Predicting protein subcellular localization is a challenging problem, particularly when query proteins have multi-label features meaning that they 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-label proteins. Actually, multi-label proteins should not be ignored because they usually bear some special function worthy of in-depth studies. By introducing the ''multi-label learning'' approach, a new predictor, called iLoc-Animal, has been developed that can be used to deal with the systems containing both single- and multi-label animal (metazoan except human) proteins. Meanwhile, to measure the prediction quality of a multi-label system in a rigorous way, five indices were introduced; they are ''Absolute-True'', ''Absolute-False'' (or Hamming-Loss''), ''Accuracy'', ''Precision'', and ''Recall''. As a demonstration, the jackknife cross-validation was performed with iLoc-Animal on a benchmark dataset of animal proteins classified into the following 20 location sites: (1) acrosome, (2) cell membrane, (3) centriole, (4) centrosome, (5) cell cortex, (6) cytoplasm, (7) cytoskeleton, (8) endoplasmic reticulum, (9) endosome, (10) extracellular, (11) Golgi apparatus, (12) lysosome, (13) mitochondrion, (14) melanosome, (15) microsome, (16) nucleus, (17) peroxisome, (18) plasma membrane, (19) spindle, and (20) synapse, where many proteins belong to two or more locations. For such a complicated system, the outcomes achieved by iLoc-Animal for all the aforementioned five indices were quite encouraging, indicating that the predictor may become a useful tool in this area. It has not escaped our notice that the multi-label approach and the rigorous measurement metrics can also be used to investigate many other multi-label problems in molecular biology. As a user-friendly web-server, iLoc-Animal is freely accessible to the public at the web-site http://www.jci-bioinfo.cn/iLoc-Animal.

Received 22nd October 2012,
Accepted 14th January 2013

DOI: 10.1039/c3mb25466f

www.rsc.org/molecularbiosystems

Introduction

Knowledge of the subcellular locations of proteins can provide key hints and useful insight into revealing their functions, helping to understand the intricate pathways that regulate biological processes at the cellular level. It is also very useful for identifying and prioritizing drug targets during the process of drug development. With the explosion of protein sequences generated in the post-genomic era, it is urged to develop computational methods for timely and effectively identifying the subcellular location of uncharacterized proteins based on their sequence information alone.

Actually, in the last two decades or so, many efforts have been made in this regard (see, *e.g.*, ref. 1–19 as well as a long list of references cited in two comprehensive review articles[20,21]). However, relatively much fewer predictors were developed specialized for identifying the subcellular localization of animal proteins, particularly for those animal proteins that may simultaneously reside at, or move between, two or more different subcellular locations. Proteins with multiple location sites or dynamic feature of this kind, the so-called multiplex proteins,[13,16] should draw our special attention because they may have some unique biological functions worthy of in-depth investigations for both basic research and drug development.[22,23]

[a] *Information Science and technology School, Donghua University, Shanghai, 200261, China. E-mail: lin_weizhong@yahoo.com.cn, jafang@dhu.edu.cn*
[b] *Computer Department, Jing-De-Zhen Ceramic Institute, Jing-De-Zhen 333046, China. E-mail: xxiao@gordonlifescience.org, xiaoxuan0326@yahoo.com.cn*
[c] *Gordon Life Science Institute, San Diego, California 92130, USA. E-mail: kcchou@gordonlifescience.org*

† Electronic supplementary information (ESI) available: The benchmark dataset $\mathbb{S}$ used in this study contains 5048 different animal protein sequences, or 9552 locative protein sequences classified into 20 subcellular locations. Among the 5048 different proteins, 2284 belong to one location, 1740 to two locations, 510 to three locations, 368 to four locations, 111 to five locations, 20 to six locations, 9 to seven locations, and 6 to 8 locations. Both the accession numbers and sequences are given. Except for the subsets of ''acrosome'', ''centriole'', ''cell cortex'' and ''melanosome'', none of the proteins included has $\geq$25% sequence identity to any other in the same subset (subcellular location). See DOI: 10.1039/c3mb25466f

Besides, recent evidence has indicated an increasing number of proteins with multiple locations in the cell, as reported by Millar *et al.*[24] Although in some of the existing methods the multi-label learning technique was introduced to identify the subcellular location sites of multiplex proteins (see, *e.g.*, ref. 14–16, 19, 25, and 26), they have the following shortcomings.

First, the benchmark datasets used to train and test the aforementioned methods had a very low "multiplicity degree", and hence could not be effectively used to reflect the prediction quality for those proteins with multiple subcellular locations. For a given benchmark dataset $\mathbb{S}$, the multiplicity degree, or the "label cardinality" according to the term used in the multi-label learning classifier technique,[27] is defined by

$$\mathrm{MD}(\mathbb{S}) = \frac{\sum_{k=1}^{N} n^{\mathrm{L}}(k)}{N}, \quad n^{\mathrm{L}}(k) \geq 1 \qquad (1)$$

where MD is the abbreviation of multiplicity degree, N the total number of proteins in the benchmark dataset $\mathbb{S}$, and $n^{\mathrm{L}}(k)$ the number of different labels (or subcellular annotations) for the k-th protein in $\mathbb{S}$. As we can see from eqn (1), when all the proteins in $\mathbb{S}$ have only one subcellular location, we have $\mathrm{MD}(\mathbb{S}) = 1$. Therefore, the closer to 1 the multiplicity degree is, the fewer the number of proteins in the benchmark dataset that have multiple subcellular locations. For the benchmark datasets used in the aforementioned papers, their multiplicity degrees are all very close to 1. For example, the multiplicity degree of the benchmark dataset used for iLoc-Plant[14] is 1.0726, that for iLoc-Gneg[15] is 1.0460, and that for iLoc-Euk[13] is 1.1619.

Secondly, for the existing predictors that are able to deal with the multiplex proteins, the gene ontology (GO) approach was an important part or actually their cornerstone. With the development in gene ontology,[28] the terms of GO increased rapidly. For instance, the number of GO terms utilized in developing iLoc-Euk,[13] iLoc-Plant,[14] and iLoc-Hum[16] was about ten thousand, but now the number of GO terms that can be utilized is around forty thousand. Therefore, it is necessary to update the GO approach accordingly.

The present study was devoted to develop a subcellular location predictor specialized for animal proteins by improving the aforementioned shortcomings.

As summarized in ref. 29, to establish a really useful statistical predictor for identifying the subcellular localization of proteins according to their sequence information, we usually need to consider the following procedures: (1) construct or select a valid benchmark dataset to train and test the predictor; (2) formulate the protein sequence samples with a feature vector that can truly reflect the intrinsic correlation with the subcellular location sites; (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 describe how to deal with these procedures.

Materials

To construct a high quality and updated benchmark dataset for developing a predictor to identify the subcellular localization of animal proteins, the sequences were collected from the release 2012_07 of UniProtKB/Swiss-Prot at http://www.uniprot.org according to the following steps.

Step 1. Only those protein sequences were collected that had the annotation of "metazoa" as well as clear experimental subcellular location annotations.

Step 2. Remove those from step 1 that belong to humans.

Step 3. Remove those annotated with "fragment"; sequences with less than 50 amino acid residues were also removed because they might belong to fragments.

Step 4. To reduce the redundancy and homology bias, the program CD-HIT[30] was utilized to remove those proteins that had $\geq$25% pairwise sequence identity to any other in the same subset except for the subsets of "acrosome", "centriole", "cell cortex" and "melanosome" because the numbers of proteins in these four sites were quite few; otherwise, the data in the four subsets might be too few to be statistically significant.

Finally, we obtained 5048 animal proteins, of which 2284 occur in one subcellular location, 1740 in two locations, 510 in three locations, 368 in four locations, 111 in five locations, 20 in six locations, 9 in seven locations, 6 in eight locations, and none in nine or more locations. These proteins form the benchmark dataset $\mathbb{S}$ for the current study; it covers 20 different subcellular locations (Fig. 1), as can be formulated by

$$\mathbb{S} = \mathbb{S}_1 \cup \mathbb{S}_2 \cup \mathbb{S}_3 \cup \mathbb{S}_4 \cup \mathbb{S}_5 \cup \mathbb{S}_6 \cup \ldots \cup \mathbb{S}_{20} \qquad (2)$$

where $\mathbb{S}_1$ represents the subset for the subcellular location of "acrosome", $\mathbb{S}_2$ for "cell membrane", $\mathbb{S}_3$ for "centriole", and so

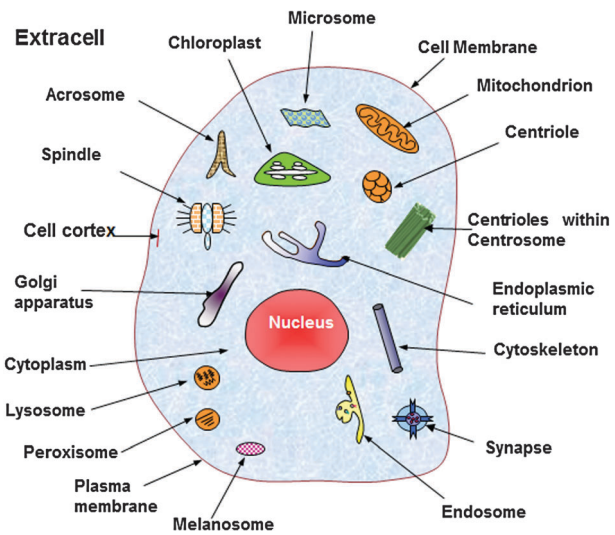

Fig. 1 A schematic drawing to show the 20 subcellular locations of animal proteins. The 20 locations are: (1) acrosome, (2) cell membrane, (3) centriole, (4) centrosome, (5) cell cortex, (6) cytoplasm, (7) cytoskeleton, (8) endoplasmic reticulum, (9) endosome, (10) extracell, (11) Golgi apparatus, (12) lysosome, (13) mitochondrion, (14) melanosome, (15) microsome, (16) nucleus, (17) peroxisome, (18) plasma membrane, (19) spindle, and (20) synapse.

Table 1 The benchmark dataset $\mathbb{S}$ contains 5048 animal proteins classified into 20 subcellular location sites (*cf.* eqn (1) and Fig. 1). Of the 5048 proteins, 2284 occur in one subcellular location, 1740 in two locations, 510 in three locations, 368 in four locations, 111 in five locations, 20 in six locations, 9 in seven locations, 6 in eight locations, and none in nine or more locations. See the text for further explanation

Subset	Subcellular location	Number of proteins
$\mathbb{S}_1$	Acrosome	87
$\mathbb{S}_2$	Cell membrane	1096
$\mathbb{S}_3$	Centriole	75
$\mathbb{S}_4$	Centrosome	243
$\mathbb{S}_5$	Cell cortex	108
$\mathbb{S}_6$	Cytoplasm	2170
$\mathbb{S}_7$	Cytoskeleton	729
$\mathbb{S}_8$	Endoplasmic reticulum	541
$\mathbb{S}_9$	Endosome	185
$\mathbb{S}_{10}$	Extracellular space	105
$\mathbb{S}_{11}$	Golgi apparatus	413
$\mathbb{S}_{12}$	Lysosome	136
$\mathbb{S}_{13}$	Mitochondrion	595
$\mathbb{S}_{14}$	Melanosome	49
$\mathbb{S}_{15}$	Microsome	71
$\mathbb{S}_{16}$	Nucleus	1458
$\mathbb{S}_{17}$	Peroxisome	81
$\mathbb{S}_{18}$	Plasma membrane	1096
$\mathbb{S}_{19}$	Spindle	159
$\mathbb{S}_{20}$	Synapse	155
Total different locative proteins		9552
Total different proteins		5048

forth (Table 1); while $\cup$ represents the symbol for "union" in the set theory. For convenience, hereafter let us just use the subscripts of eqn (1) as the codes of the 20 location sites; *i.e.*, "1" for "acrosome", "2" for "cell membrane", "3" for "centriole", and so forth.

Because some proteins may simultaneously occur in two or more locations, it is instructive to introduce the concept of "locative protein"[16] 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 sites, it will be counted as three locative proteins, and so forth. Thus, the number of total locative proteins can be expressed as

$$N(\mathrm{loc}) = N(\mathrm{seq}) + \sum_{m=1}^{M} (m-1)N(m) \tag{3}$$

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. Substituting the data of the last paragraph into eqn (3), we obtain

$$\begin{aligned} N(\mathrm{loc}) &= N(\mathrm{seq}) + (1-1)\times 2284 + (2-1)\times 1740 + (3-1)\times 510 \\ &\quad + (4-1)\times 368 + (5-1)\times 111 + (6-1)\times 20 \\ &\quad + (7-1)\times 9 + (8-1)\times 6 + \textstyle\sum_{m=9}^{20}(m-1)\times 0 \\ &= 5048 + 1740 + 1020 + 1104 + 444 + 100 + 54 + 42 = 9552 \end{aligned} \tag{4}$$

meaning that the total number of the locative proteins in the current benchmark dataset $\mathbb{S}$ is 9552, which is actually also the sum of the protein numbers for the 20 subsets $\mathbb{S}_i$ (i = 1, 2,..., 20) listed in Table 1. As we can see from eqn (3) and (4), 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.

For readers' convenience, the subcellular locations for each of these locative proteins as well as its sequence and accession number are given in the ESI.†

Methods

To develop a powerful predictor for statistically predicting protein subcellular localization based on the sequence information, one of the keys is to formulate the protein sequences with an effective mathematical expression that can truly reflect the intrinsic correlation with their subcellular localization.[31] However, it is not a trivial and easy job because this kind of correlation is usually deeply hidden in piles of complicated sequences.

The most straightforward method to formulate the sample of a query protein **P** of L amino acid residues is to use its entire amino acid sequence, as can be expressed by

$$\mathbf{P} = \mathrm{R}_1\mathrm{R}_2\mathrm{R}_3\mathrm{R}_4\mathrm{R}_5\mathrm{R}_6\mathrm{R}_7\ldots\mathrm{R}_L \tag{5}$$

where R_1 represents the 1st residue of the protein **P**, R_2 the 2nd residue, R_3 the 3rd residue, and so forth, and they each belong to one of the 20 native amino acids. To identify the subcellular location(s) of **P**, the tools for computing protein sequence similarity, such as BLAST,[32,33] were utilized to search protein database for those targets that have high sequence similarity to the query protein **P**. Subsequently, the subcellular location annotations of the targeted proteins thus found were used to infer the subcellular location(s) for the query protein **P**. Unfortunately, this kind of straightforward sequential model, although containing the entire sequence information of a protein sample, failed to work when the query protein **P** did not have any significant sequence similarity to location-known proteins.

To avoid the above difficulty, which is inherent to the sequential model, various non-sequential or discrete models to formulate protein samples were proposed in hopes to enhance the prediction power.

Among the discrete models, the simplest one is the amino acid (AA) composition or AAC.[34] According to the AAC-discrete model, the protein **P** of eqn (5) can be formulated by[35]

$$\mathbf{P} = [f_1 \quad f_2 \quad \cdots \quad f_{20}]^{\mathbf{T}} \tag{6}$$

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. 1–4, 7, and 36). Unfortunately, it can be obviously seen from eqn (6) that if the ACC model is used to represent the protein **P**, its sequence-order information would be totally lost, and hence might considerably limit the prediction quality.

To avoid completely losing the sequence-order information, the pseudo amino acid composition (PseAAC) was proposed to represent the sample of a protein.[37,38] The idea of PseAAC has been widely used in bioinformatics, proteomics, and system biology,[31] such as predicting protein structural class,[39] predicting metalloproteinase family,[40] predicting protein subcellular localization,[41] predicting protein submitochondrial localization,[42] predicting DNA-binding proteins,[43] identifying allergenic proteins,[44] identifying bacterial virulent proteins,[45] predicting the protein folding rate,[46] predicting GABA(A) receptor proteins,[47] predicting protein supersecondary structure,[48] predicting cyclin proteins,[49] classifying amino acids,[50] predicting enzyme family class,[51] identifying the risk type of human papillomaviruses,[52] identifying protein quaternary structural attributes,[53,54] identifying GPCRs and their types,[55] and discriminating outer membrane proteins,[56] among many others (see a long list of references cited in ref. 29). Recently, the concept of PseAAC was further extended to deal with the problems in the DNA area, such as identifying nucleosomes[57] and predicting the recombination spots.[58] Because of its wide and increasing usage, in 2012 a powerful software called ''PseAAC-Builder'' (http://www.pseb.sf.net)[59] was established for generating various special modes of PseAAC, in addition to the web-server PseAAC[60] built in 2008.

According to a recent review article,[29] the general form of PseAAC for a protein **P** is formulated by

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

where the subscript Ω is an integer, and its value as well as the components $\psi_1, \psi_2, \ldots$ will depend on how to extract the desired information from the amino acid sequence of **P** (*cf.* eqn (5)). Below, let us describe how to extract the core and essential features from a protein sequence to define the components in eqn (7).

1. GO approach

GO, or gene ontology, is a major bioinformatics initiative to unify the representation of gene and gene product attributes across all species.[28,61,62] The ontology covers three domains: cellular component, molecular function, and biological process.[63] Therefore, protein samples defined in a GO database space would be clustered in a manner much closely correlated with their subcellular locations.[64] However, the current GO approach is quite different from those of the previous studies[13,14,16,25,65–67] in the following two aspects. First, the number of GO terms has increased rapidly in recent years. The feature vector in ref. 13, 14, 16, and 25 was derived from the GO database in 2009 that only contained about 10 000 GO terms, but now its number is nearly 4 times as large. If all the GO terms are used to formulate the feature vector for a protein sample, we might face the high-dimension disaster problem[68] or machine breakdown problem. Secondly, in order to incorporate more information into the feature vector, instead of only using 0 and 1 as its elements as done in ref. 66, 69, and 70, a refined approach was adopted here to enrich the information of the feature vector. The detailed procedures to formulate the feature vector of a protein sample are given below.

Step 1. Compression and reorganization of the existing GO numbers that are marked with ''cellular component'' only. The GO database (version 2012-08-18) contains 36 294 GO numbers, of which only 3043 GO terms are annotated with ''cellular component''. These 3043 GO terms were singled out for compression and reorganization. For example, after such a treatment, the original GO number GO:0000015, GO:0000109, GO:0000110, GO:0000111, GO:0000112, GO:0000113, GO:0000118,..., GO:0098061 would become GO_compress:0001, GO_compress:0002, GO_compress:0003, GO_compress:0004, GO_compress:0005, GO_compress:0006, GO_compress:0007,....., GO_compress:3043, respectively. The GO database thus obtained from the original GO database is called GO_compress database, which contains 3043 GO numbers increasing successively from 1 to the last one.

Step 2. Using eqn (7) with Ω = 3043 to formulate the protein **P**; *i.e.*

$$\mathbf{P}_{\mathrm{GO}} = \left[\psi_1^{\mathrm{G}} \quad \psi_2^{\mathrm{G}} \quad \cdots \quad \psi_u^{\mathrm{G}} \quad \cdots \quad \psi_{3043}^{\mathrm{G}}\right]^{\mathbf{T}} \tag{8}$$

where ψ_u^{G} (u = 1, 2, ...3034) are defined according to the following steps.

Step 3. Use BLAST[71] program (version 2.2.25+) to search all the proteins in the Swiss-Prot database (version 2012-04-12) for those having homologous sequences to the protein **P** (during the search process the BLAST parameters for the expected value and the threshold were set at 10 and 10^{-9}, respectively).

Step 4. Those proteins found from step 3 were collected into a set, $\mathbb{S}_{\mathrm{P}}^{\mathrm{homo}}$, called the ''homology set'' of **P**. All the elements in $\mathbb{S}_{\mathrm{P}}^{\mathrm{homo}}$ 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[75] at http://www.geneontology.org/ (version 2012-08-18) to find the corresponding GO number.

Step 6. Based on the results obtained in step 5, the elements in eqn (8) can be written as

$$\psi_u^{\mathrm{G}} = \frac{\sum_{k=1}^{\mathbb{N}_{\mathrm{P}}^{\mathrm{homo}}} \delta(u,k)}{\mathbb{N}_{\mathrm{P}}^{\mathrm{homo}}} \quad (u = 1, 2, \cdots, 3043) \tag{9}$$

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

$$\delta(u,k) = \begin{cases} 1 & \text{if the } k\text{-th representative protein hits} \\ & \quad \text{the } u\text{-th GO\_compress number} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

As we can see from eqn (8), the GO formulation derived from the above steps consists of 3043 real numbers rather than the integers 0 and 1 only as in the GO formulation adopted in ref. 65 and 76.

Note that the GO formulation of eqn (8) may become meaningless or a naught vector 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.*, $\mathbb{S}_{\mathbf{P}}^{\text{homo}} = \varnothing$ meaning that the homology set $\mathbb{S}_{\mathbf{P}}^{\text{homo}}$ is an empty one; (2) its representative protein does 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 information to formulate a protein sample, as described below.

2. Grey-PSSM approach

Biology is a natural science with historic dimension. All biological species have developed starting out from a very limited number of ancestral species. It is true for the protein sequence as well. Their evolution involves changes of single residues, insertions and deletions of several residues,[77] 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[74] and residing in the same subcellular location. To extract the sequential evolution information and use it to define the components of eqn (7), let us consider the PSSM (Position Specific Scoring Matrix), as described below.

According to ref. 71, the sequence evolution information of protein **P** with L amino acid residues can be expressed by a $L \times 20$ matrix, as given by

$$\mathbf{P}_{\text{PSSM}}^{(0)} = \begin{bmatrix} m_{1,1}^{(0)} & m_{1,2}^{(0)} & \cdots & m_{1,20}^{(0)} \\ m_{2,1}^{(0)} & m_{2,2}^{(0)} & \cdots & m_{2,20}^{(0)} \\ \vdots & \vdots & \vdots & \vdots \\ m_{L,1}^{(0)} & m_{L,2}^{(0)} & \cdots & m_{L,20}^{(0)} \end{bmatrix} \tag{11}$$

where $m_{i,j}^{(0)}$ represents the original score of amino acid residues in the i-th ($i = 1, 2, \ldots, L$) sequential position of the protein that is being changed to amino acid type j ($j = 1, 2, \ldots, 20$) during the evolution 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.[16] The $L \times 20$ scores in eqn (11) were generated by using PSI-BLAST[71] to search the UniProtKB/Swiss-Prot database (Release 2011_05) through three iterations with 0.001 as the E-value cutoff for multiple sequence alignment against the sequence of the protein **P**. In order to make every element in eqn (11) within the range of 0–1, a conversion was performed through the standard sigmoid function to make it become

$$\mathbf{P}_{\text{PSSM}}^{(1)} = \begin{bmatrix} m_{1,1}^{(1)} & m_{1,2}^{(1)} & \cdots & m_{1,20}^{(1)} \\ m_{2,1}^{(1)} & m_{2,2}^{(1)} & \cdots & m_{2,20}^{(1)} \\ \vdots & \vdots & \vdots & \vdots \\ m_{L,1}^{(1)} & m_{L,2}^{(1)} & \cdots & m_{L,20}^{(1)} \end{bmatrix} \tag{12}$$

where

$$m_{i,j}^{(1)} = \frac{1}{1 + \mathrm{e}^{-m_{i,j}^{(0)}}} \quad (1 \le i \le L,\ 1 \le j \le 20) \tag{13}$$

Now, let us extract the useful information from eqn (12) to define the components of eqn (7). According to the grey system theory,[78] if the information of a system investigated is fully known, it is called a "white system"; if completely unknown, a "black system"; if partially known, a "grey system". The model developed on the basis of such a theory is called a "grey model", which is a kind of nonlinear and dynamic model formulated by a differential equation. The grey model is particularly useful for solving complicated problems that lack sufficient information, or need to process uncertain information and to reduce random effects of acquired data. Using the grey system theory, we can extract the following information from the j-th column of eqn (12)

$$\begin{bmatrix} a_1^j \\ a_2^j \\ b^j \end{bmatrix} = (\mathbf{B}_j^{\mathrm{T}}\mathbf{B}_j)^{-1}\mathbf{B}_j^{\mathrm{T}}\mathbf{U}_j \quad (j = 1, 2, \cdots, 20) \tag{14}$$

where

$$\mathbf{B}_j = \begin{bmatrix} -m_{2,j}^{(1)} & -m_{1,j}^{(1)} - 0.5m_{2,j}^{(1)} & 1 \\ -m_{3,j}^{(1)} & -\sum_{i=1}^{2} m_{i,j}^{(1)} - 0.5m_{3,j}^{(1)} & 1 \\ \vdots & \vdots & \vdots \\ -m_{k,j}^{(1)} & -\sum_{i=1}^{k-1} m_{i,j}^{(1)} - 0.5m_{k,j}^{(1)} & 1 \\ \vdots & \vdots & \vdots \\ -m_{L,j}^{(1)} & -\sum_{i=1}^{L-1} m_{i,j}^{(1)} - 0.5m_{L,j}^{(1)} & 1 \end{bmatrix} \tag{15}$$

and

$$\mathbf{U}_j = \begin{bmatrix} m_{2,j}^{(1)} - m_{1,j}^{(1)} \\ m_{3,j}^{(1)} - m_{2,j}^{(1)} \\ \vdots \\ m_{k,j}^{(1)} - m_{k-1,j}^{(1)} \\ \vdots \\ m_{L,j}^{(1)} - m_{L-1,j}^{(1)} \end{bmatrix} \tag{16}$$

Therefore, when using the grey-PSSM approach, we can extract a total of $\Omega = 3 \times 20 = 60$ quantities from eqn (14) for the sequential evolution information. Thus, eqn (7) can now be formulated as

$$\mathbf{P}_{\text{Evo}} = \begin{bmatrix} \psi_1^{\mathrm{E}} & \psi_2^{\mathrm{E}} & \cdots & \psi_{60}^{\mathrm{E}} \end{bmatrix}^{\mathrm{T}} \tag{17}$$

where

$$\begin{cases} \psi^{\mathrm{E}}_{3j-2} = a^j_1 f_j w_1 \\ \psi^{\mathrm{E}}_{3j-1} = a^j_2 f_j w_2 \\ \psi^{\mathrm{E}}_{3j} = b^j f_j w_3 \end{cases} \quad (j = 1, 2, \cdots, 20) \qquad (18)$$

where f_j (j = 1, 2,. . ., 20) are the same as those in eqn (6), and w_1, w_2, and w_3 are the weight factors, which were all set to 1 in the current study.

3. 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 $\mathbf{P}_{\mathrm{GO}}$ (see eqn (8)), then all the protein samples used to train the prediction engine should also be formulated in the GO form; if the query protein is defined in the form of $\mathbf{P}_{\mathrm{Evo}}$ (see eqn (17)), then all the proteins used to train the predictor should be defined with the same form as well.

4. ML-KNN or AL-KNN classifier

In this study, the "multi-label K-nearest neighbor" (ML-KNN) classifier[13] or "accumulation-label K-nearest neighbor" (AL-KNN) classifier[16] was used to perform the prediction. The detailed description of how the two classifier works is clearly described in eqn (18)–(25) of ref. 13 or eqn (20)–(27) of ref. 16, and hence there is no need to repeat here.

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

5. Web-server guide

For users' convenience, a web-server for iLoc-Animal was established. Furthermore, for the majority of experimental scientists who are only interested in getting the desired results without the need to follow the complicated mathematics, a step-by-step guide is given below.

Step 1. Open the web server at site http://www.jci-bioinfo.cn/iLoc-Animal and you will see the top page of the predictor on your computer screen, as shown in Fig. 3. Click on the *Read Me* button to see a brief introduction about the iLoc-Animal predictor and the caveat when using it.

Step 2. Either type or copy–paste the query protein sequences into the input box shown at the center of 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. The maximum number of query proteins for each submission is limited to 5.

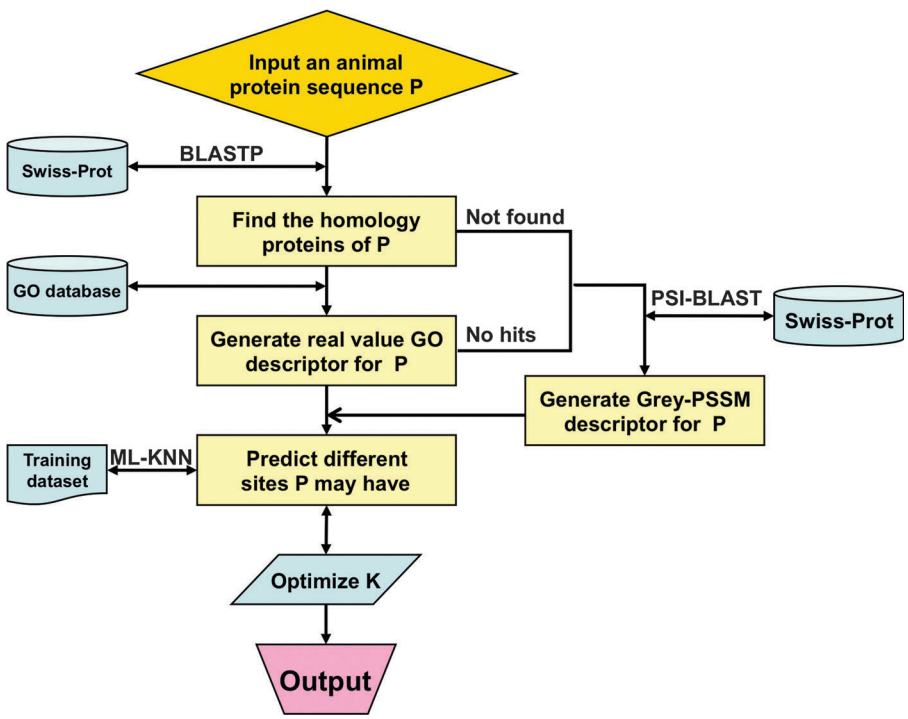

Fig. 2 A flowchart to show the prediction process of iLoc-Animal.

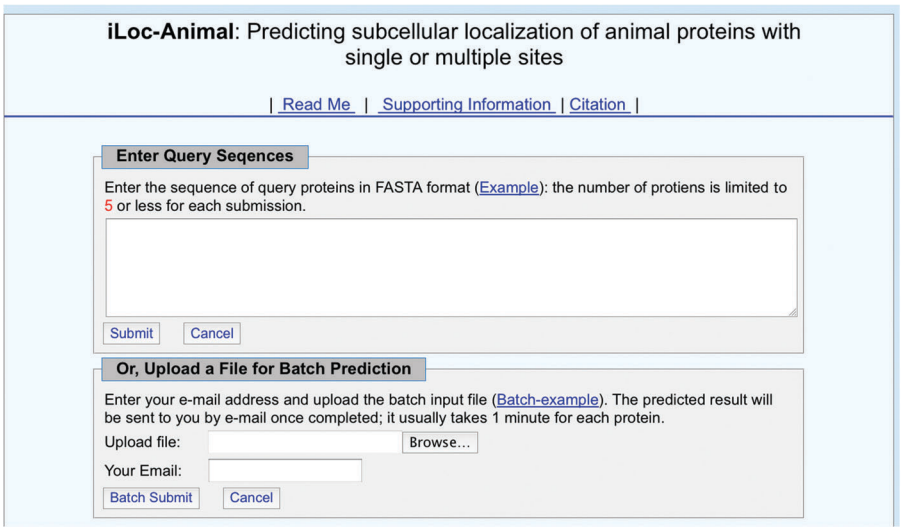

Fig. 3 A semi-screenshot to show the top page of the iLoc-Animal web-server. Its website address is http://www.jci-bioinfo.cn/iLoc-Animal.

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 on your screen that the predicted result for the 1st query protein is ''cell membrane'', ''cytoplasm'', ''cytoskeleton'', ''nucleus'', ''plasma membrane''; the predicted result for the 2nd query protein is ''cytoplasm''; the predicted result for the 3rd query protein is ''cytoplasm'', ''cytoskeleton'', ''spindle''. In other words, the 1st query protein (Q7TQJ1) is a multiplex protein that can simultaneously occur in five different subcellular location sites; the 2nd query protein (Q9CZV8) is a singleplex one residing at the site of ''cytoplasm'' only; the 3rd query protein (Q9D0P7) is again a multiplex one that can simultaneously occur in three different sites. All these results are fully consistent with the experimental observation as summarized in the ESI.† It takes about 20 seconds for the above computation before the predicted results appear on the computer screen; the more the number of query proteins and the longer each sequence, the more time 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 the FASTA format) *via* the ''Browse'' button. To see the sample of batch input file, click on the button *Batch-example*. 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.''

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

Step 6. Click on the ESI† button to download the benchmark datasets used to train and test the iLoc-Animal 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 animal protein is known outside of the 20 locations as covered by Fig. 1, stop the prediction because the result thus obtained will not make any sense.

Results and discussion

It is instructive to point out that the system investigated here is a multi-label one, *i.e.*, it contains proteins with both single and multiple location sites. Therefore, the existing methods or metrics used to evaluate the quality of a predictor on a single-label system would no longer be valid. The corresponding metrics for a multi-label system will be much more complicated.[16,21,27,79,80] Below, let us describe what metrics should be used to evaluate the prediction quality of a multi-label system.

1. Five different metrics for measuring the prediction quality of a multi-labeled system

Given a multi-label system consisting of N proteins, suppose M is the number of all possible subcellular locations, $\mathbb{L}$ the label set that contains the labels for all the possible subcellular locations concerned. Thus, the i-th protein $\mathbf{P}_i$ and its subcellular location(s) can be expressed by

$$\{\mathbf{P}_i, \mathbb{L}_i\} \quad (i = 1, 2, \cdots, N) \tag{19}$$

where $\mathbb{L}_i$ is the subset that contains all the location label(s) for the i-th protein. Obviously, we have

$$\mathbb{L}_1 \cup \mathbb{L}_2 \cup \cdots \cup \mathbb{L}_N \subseteq \mathbb{L} = \{\ell_1, \ell_2, \cdots \ell_M\} \tag{20}$$

where $\ell_i (i = 1, 2, \cdots, M)$ is the label for the i-th subcellular location. For the current study, $N = 5048$ and $M = 20$ (*cf.* Table 1). Suppose $\mathbb{L}_i^*$ represents the subset that contains all the predicted location label(s) for the i-th protein. Thus, we can have the

following five metrics to measure the prediction quality of the multi-label system.[16,27,79,81]

$$\begin{cases} \text{Absolute-False} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\left\|\mathbb{L}_i \cup \mathbb{L}_i^*\right\| - \left\|\mathbb{L}_i \cap \mathbb{L}_i^*\right\|}{M}\right) \\ \text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\left\|\mathbb{L}_i \cap \mathbb{L}_i^*\right\|}{\left\|\mathbb{L}_i \cup \mathbb{L}_i^*\right\|}\right) \\ \text{Precision} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\left\|\mathbb{L}_i \cap \mathbb{L}_i^*\right\|}{\left\|\mathbb{L}_i^*\right\|}\right) \\ \text{Recall} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\left\|\mathbb{L}_i \cap \mathbb{L}_i^*\right\|}{\left\|\mathbb{L}_i\right\|}\right) \\ \text{Absolute-True} = \frac{1}{N}\sum_{i=1}^{N}\Delta\left(\mathbb{L}_i, \mathbb{L}_i^*\right) \end{cases} \quad (21)$$

where $M = 20$ (*cf.* Table 1) is the total number of subcellular locations covered by the current benchmark dataset, $\cup$ the symbol of union in the set theory, $\cap$ the intersection symbol, $\|\|$ the operator acting on the set therein to count the number of its elements, and

$$\begin{cases} \Delta\left(\mathbb{L}_i, \mathbb{L}_i^*\right) = 1, & \text{if all the labels in } \mathbb{L}_i \text{ are identical to those in } \mathbb{L}_i^* \\ \Delta\left(\mathbb{L}_i, \mathbb{L}_i^*\right) = 0, & \text{otherwise} \end{cases} \quad (22)$$

Among the above five metrics, the rate of "Absolute-False" or "Hamming-Loss"[79] is opposite to those of the four others. As can be easily seen from eqn (21), when the multi-labels for all the proteins are correctly predicted, *i.e.*, $\mathbb{L}_i \equiv \mathbb{L}_i^*$ or $\left\|\mathbb{L}_i \cup \mathbb{L}_i^*\right\| = \left\|\mathbb{L}_i \cap \mathbb{L}_i^*\right\|$ ($i = 1, 2, \ldots, N$), the rate of Absolute-False is equal to 0. When each of $\mathbf{P}_i$ ($i = 1, 2, \ldots, N$) is predicted completely wrong, *i.e.*, belonging to all the possible locations except its own true location(s), *i.e.*, $\mathbb{L}_i \cup \mathbb{L}_i^* = \mathbb{L}$ and $\mathbb{L}_i \cap \mathbb{L}_i^* = \emptyset$, or $\left\|\mathbb{L}_i \cup \mathbb{L}_i^*\right\| = M$ and $\left\|\mathbb{L}_i \cap \mathbb{L}_i^*\right\| = 0$, the rate of Absolute-False is equal to 1. Therefore, the lower the Absolute-False is, the better the prediction quality will be. However, for the other four metrics, the meanings of their rates are just opposite; *i.e.*, the higher their rates are, the better the prediction quality will be.

Also, of the five metrics, the "Absolute-True" rate[16] or "Subset-Accuracy"[27] is the most intuitive and easier-to-understand one for a multi-label system. According to its definition, for a query protein, *e.g.*, the *i*-th query protein, when and only when all its subcellular location sites are exactly predicted without any underprediction or overprediction, *i.e.*, $\mathbb{L}_i \equiv \mathbb{L}_i^*$ (*cf.* eqn (21)), can the prediction event be scored with 1; otherwise, 0. For example, for a protein belonging to, say, four subcellular locations, if only three of the four are correctly predicted, or the predicted result contains a location not belonging to the four, the prediction score will be counted as 0.[13,16] Therefore, the absolute true rate is much more strict and harsh than the proportional success rate used previously.[21,69] For the former, if the locations of a query protein were partially correctly predicted, no score at all would be credited; but for the latter, a corresponding proportional score would be credited.

2. Cross-validation to evaluate success rates

In statistical prediction, the following three cross-validation methods are often used to examine a predictor for its effectiveness in practical application: independent dataset test, subsampling (K-fold cross-validation) test, and jackknife test. However, as elaborated by a recent review[29] and demonstrated by eqn (28)–(32) in that paper, among the three cross-validation methods, the jackknife test is deemed the least arbitrary and most objective because it can always yield a unique result for a given benchmark dataset, and hence has been increasingly used and widely recognized by investigators to examine the accuracy of various predictors (see, *e.g.*, ref. 45, 55, and 82–87). Accordingly, the jackknife test was also adopted in this study to examine the anticipated success rates of the current predictor.

The outcomes thus obtained on the benchmark dataset $\mathbb{S}$ for the five metrics as defined in eqn (21) are as follows

$$\begin{cases} \text{Absolute-False} = 0.0518 \\ \text{Accuracy} = 0.6288 \\ \text{Precision} = 0.7356 \\ \text{Recall} = 0.6949 \\ \text{Absolute-True} = 0.4562 \end{cases} \quad (23)$$

from which we can see that the overall absolute-false or Hamming-loss rate is very low ($\approx$5.2%), while the absolute-true rate is much higher ($\approx$46%), indicating that the iLoc-Animal is quite a promising multi-label predictor as will be further discussed later.

It is instructive to point out that, for a multi-label system like the current one, the absolute-true success rate for each of the individual subcellular locations is meaningless and misleading.[27,79] Therefore, rather than the absolute-true success rate for each of the individual subcellular locations, provided in Table 2 are the absolute true success rates for proteins with different numbers of labels (or subcellular location sites). Furthermore, for facilitating comparison, listed in Table 2 are also the corresponding rates by the completely random guess and weighted random guess, as defined below.

The completely random guess (CRG) rates were calculated according to the following equation

$$P(\text{CRG}) = \frac{1}{M} \cdot \frac{1}{C(M,m)} = \frac{1}{M\dfrac{M!}{(M-m)!m!}} \quad (m \le M) \quad (24)$$

where M is the total number of all the possible subcellular locations that is equal to 20 for the current benchmark dataset $\mathbb{S}$, m has the same meaning as in eqn (3), and the symbol $C(M, m)$ represents the number of combinations of M distinct things (or locations) taken m at a time.

The weighted random guess (WRG) rates were calculated according to the following equation

$$P(\text{WRG}) = \frac{N(m)}{N(\text{seq})} \cdot \frac{1}{C(M,m)} = \frac{N(m)}{\dfrac{N(\text{seq})M!}{(M-m)!m!}} \quad (m \le M) \quad (25)$$

where $N(\text{seq})$ and $N(m)$ have the same meaning as in eqn (3).

Table 2 The absolute true success rates for proteins with different numbers of subcellular location sites

		Absolute-true rate		
Number of location sites or labels	Number of proteins	iLoc-Animal	Completely random guess[a]	Weighted random guess[b]
1	2284	$\frac{1240}{2284} = 54\%$	$2.5 \times 10^{-1}\%$	2.2%
2	1740	$\frac{928}{1740} = 53\%$	$2.6 \times 10^{-2}\%$	$1.8 \times 10^{-1}\%$
3	510	$\frac{77}{510} = 15\%$	$4.4 \times 10^{-3}\%$	$8.9 \times 10^{-3}\%$
4	368	$\frac{53}{368} = 14\%$	$1.1 \times 10^{-3}\%$	$1.5 \times 10^{-4}\%$
5	111	$\frac{5}{111} = 4.5\%$	$3.2 \times 10^{-4}\%$	$1.4 \times 10^{-4}\%$
6	20	$\frac{0}{20} = 0.0\%$	$1.3 \times 10^{-4}\%$	$1.0 \times 10^{-5}\%$
7	9	$\frac{0}{9} = 0.0\%$	$6.5 \times 10^{-5}\%$	$2.2 \times 10^{-6}\%$
8	6	$\frac{0}{6} = 0.0\%$	$3.9 \times 10^{-5}\%$	$9.4 \times 10^{-7}\%$

[a] The completely random guess was calculated according to eqn (24). [b] The weighted random guess was calculated according to eqn (25).

From Table 2, we can see the following: (1) the more subcellular locations the protein have, the lower their absolute true rates are, meaning that the more difficult to predict their subcellular locations exactly without any over- or under-prediction; (2) although for the small numbers of proteins with 6 to 8 subcellular locations the absolute-true rates are zero, for most proteins with 1 to 5 subcellular locations the absolute-true rates achieved by iLoc-Animal are about 200 ~ 14 000 times high as those by the completely random guess, and about 25 ~ 93 000 times high as those by the weighted random guess.

Shown in Table 3 is a comparison of iLoc-Animal with IMMMLGP[19] and Hum-mPLoc2.0,[69] two powerful predictors developed recently that are also able to deal with a system with both single- and multiple-location proteins. As we can see from the table, the absolute-true success rate achieved by iLoc-Animal is significantly higher than those by IMMMLGP and Hum-mPLoc2.0. Particularly, it should be pointed out that, compared with the previous dataset used to test IMMMLGP and Hum-mPLoc2.0, the current benchmark dataset is much more difficult due to the following reasons. (1) It covers 20 subcellular location sites, while the dataset used to test IMMMLGP and Hum-mPLoc2.0 only covered 14 location sites. As is well known, the more classes a benchmark dataset covers, the more difficult to get a high success rate when using it to test a classifier.[29] (2) The multiplicity degree of the current benchmark dataset is 1.8922 (see Table 3), which is much higher than 1.1851, the multiplicity degree of the benchmark dataset used by IMMMLGP and Hum-mPLoc2.0. It is easy to imagine that the higher the multiplicity degree of a benchmark dataset is, the more multiple subcellular locations it contains, and hence the more difficult to achieve a high absolute-true success rate when using it to test a predictor (*cf.* Table 3). Even though, however, the overall absolute-true success rate achieved by iLoc-Animal is significantly higher than those by IMMMLGP and Hum-mPLoc2.0, indicating that iLoc-Animal holds a high potential to become a useful high throughput tool in this area.

Table 3 A comparison of the iLoc-Animal predictor with the other existing predictors

Predictor	Absolute-true rate[a]	Recall rate[a]	Multiplicity degree[b]	Number of locations covered
IMMMLGP[c]	0.2740	0.5950	1.1851	14
Hum-mPLoc2.0[d]	0.2940	0.5190	1.1851	14
iLoc-Animal	0.4562	0.6949	1.8922	20

[a] See eqn (18) for the definitions of "absolute-true" and "recall". [b] See eqn (1) for the definition of "multiplicity degree" for a benchmark dataset. [c] The predictor proposed by He *et al.*[19] [d] The predictor proposed by Shen and Chou.[69]

It is instructive to point out that of the 5048 protein samples in the benchmark dataset $\mathbb{S}$, 285 could not be formulated by the GO approach, and hence were formulated by the grey-PSSM approach. For such 285 samples, 83 were perfectly correctly predicted for their subcellular locations without any over- or under-prediction; *i.e.*, the absolute true rate was 83/25 = 29.12% which is lower than the overall absolute true rate as given in eqn (23). That is why the grey-PSSM approach was used to formulate a protein sample only when the GO-approach failed to do so.

3. Some remarks on the GO approach

The following questions might be prone to be asked regarding the GO approach. If a protein already has GO annotation, why does one need to predict its subcellular location? Is it merely a procedure of converting the annotation into another format? To address these questions, let us consider the following facts. In the literature almost all the existing benchmark datasets constructed by many investigators for predicting protein subcellular localization were taken from the Swiss-Prot database, in which all the proteins have subcellular location annotations determined by experiments. Can we say all these benchmark datasets are invalid, or any prediction based on these benchmark datasets is not prediction? Of course, we cannot. This is because all these predictors such as those proposed in ref. 4–11 would yield the desired subcellular locations of query proteins by using the input only containing their sequence information alone without needing any Swiss-Prot annotation information at all once these predictors had been established. This is exactly the same for the current predictor iLoc-Animal as well as those developed using the GO approach such as Euk-mPloc2.0,[66] iLoc-Euk,[13] iLoc-Hum,[16] and the predictors presented in ref. 19, 88 and 89. For these GO-approach predictors, once established, the only input for them to perform prediction is the sequences of query proteins alone without needing any of the GO annotation information whatsoever. Accordingly, as far as the requirement for the input is concerned, there is no difference at all between the non-GO-approach predictors and GO-approach predictors. So, why the former is a valid predictor, while the latter is not?

It is interesting to note that of the 5048 protein samples in the benchmark dataset $\mathbb{S}$, 3977 were found without any GO terms at all from the current GO database, and their feature vectors were defined *via* the homology set $\mathbb{S}_P^{homo}$ or the grey-PSSM approach. Among the 3977 proteins, 1744 were perfectly correctly predicted for their subcellular locations without any over- or under-prediction; *i.e.*, the absolute true rate for the 3977 "no GO term proteins" was 43.85%.

Actually, the essence of why using the GO approach can significantly improve the prediction quality is due to the fact that proteins mapped into the GO space (instead of Euclidean space or any other simple geometric space) would be clustered in a way much better reflecting their subcellular locations, as elaborated in ref. 64 and 90.

Acknowledgements

The authors wish to thank the two anonymous referees, whose constructive comments are 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, 31260273, 61261027), the Key Project of Chinese Ministry of Education (No. 210116), and the Department of Education of Jiang-Xi Province (No. GJJ11557, GJJ12490), and the Jiangxi Provincial Foundation for Leaders of Disciplines in Science (No. 20113BCB22008), and the Jiangxi Provincial Foreign Scientific and Technological Cooperation Project (No. 20121BDH80023). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

References

1 H. Nakashima and K. Nishikawa, *J. Mol. Biol.*, 1994, **238**, 54–61.
2 J. Cedano, P. Aloy, J. A. P'erez-Pons and E. Querol, *J. Mol. Biol.*, 1997, **266**, 594–600.
3 A. Reinhardt and T. Hubbard, *Nucleic Acids Res.*, 1998, **26**, 2230–2236.
4 K. C. Chou and D. W. Elrod, *Protein Eng.*, 1999, **12**, 107–118.
5 K. Nakai and P. Horton, *Trends Biochem. Sci.*, 1999, **24**, 34–36.
6 O. Emanuelsson, H. Nielsen, S. Brunak and G. von Heijne, *J. Mol. Biol.*, 2000, **300**, 1005–1016.
7 G. P. Zhou and K. Doctor, *Proteins: Struct., Funct., Genet.*, 2003, **50**, 44–48.
8 S. Matsuda, J. P. Vert, H. Saigo, N. Ueda, H. Toh and T. Akutsu, *Protein Sci.*, 2005, **14**, 2804–2813.
9 J. L. Gardy, M. R. Laird, F. Chen, S. Rey, C. J. Walsh, M. Ester and F. S. Brinkman, *Bioinformatics*, 2005, **21**, 617–623.
10 A. Hoglund, P. Donnes, T. Blum, H. W. Adolph and O. Kohlbacher, *Bioinformatics*, 2006, **22**, 1158–1165.
11 P. Mundra, M. Kumar, K. K. Kumar, V. K. Jayaraman and B. D. Kulkarni, *Pattern Recognit. Lett.*, 2007, **28**, 1610–1615.
12 Q. Xu, S. J. Pan, H. H. Xue and Q. Yang, *IEEE/ACM Trans. Comput. Biol. Bioinf.*, 2011, **8**, 748–759.
13 K. C. Chou, Z. C. Wu and X. Xiao, *PLoS One*, 2011, **6**, e18258.
14 Z. C. Wu, X. Xiao and K. C. Chou, *Mol. BioSyst.*, 2011, **7**, 3287–3297.
15 Z. C. Wu, X. Xiao and K. C. Chou, *Protein Pept. Lett.*, 2012, **19**, 4–14.
16 K. C. Chou, Z. C. Wu and X. Xiao, *Mol. BioSyst.*, 2012, **8**, 629–641.
17 J. Li, L. Xiong, J. Schneider and R. F. Murphy, *Bioinformatics*, 2012, **28**, i32–i39.
18 L. Li, Y. Zhang, L. Zou, C. Li, B. Yu, X. Zheng and Y. Zhou, *PLoS One*, 2012, **7**, e31057.
19 J. He, H. Gu and W. Liu, *PLoS One*, 2012, **7**, e37155.
20 K. Nakai, *Adv. Protein Chem.*, 2000, **54**, 277–344.
21 K. C. Chou and H. B. Shen, *Anal. Biochem.*, 2007, **370**, 1–16.
22 E. Glory and R. F. Murphy, *Dev. Cell*, 2007, **12**, 7–16.
23 C. Smith, http://www.biocompare.com/Articles/TechnologySpotlight/976/Subcellular-Targeting-Of-Proteins-And-Drugs.html, 2008.
24 A. H. Millar, C. Carrie, B. Pogson and J. Whelan, *Plant Cell*, 2009, **21**, 1625–1631.
25 X. Xiao, Z. C. Wu and K. C. Chou, *J. Theor. Biol.*, 2011, **284**, 42–51.
26 X. Xiao, Z. C. Wu and K. C. Chou, *PLoS One*, 2011, **6**, e20592.
27 G. Tsoumakas, I. Katakis and I. Vlahavas, in *Data Mining and Knowledge Discovery Handbook*, ed. O. Maimon and L. Rokach, Springer US, 2010, pp. 667–685.
28 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.
29 K. C. Chou, *J. Theor. Biol.*, 2011, **273**, 236–247.
30 Y. Huang, B. Niu, Y. Gao, L. Fu and W. Li, *Bioinformatics*, 2010, **26**, 680–682.
31 K. C. Chou, *Curr. Proteomics*, 2009, **6**, 262–274.
32 S. F. Altschul, in *Theoretical and Computational Methods in Genome Research*, ed. S. Suhai, Plenum, New York, 1997, pp. 1–14.
33 J. C. Wootton and S. Federhen, *Comput. Chem.*, 1993, **17**, 149–163.
34 H. Nakashima, K. Nishikawa and T. Ooi, *J. Biochem.*, 1986, **99**, 152–162.
35 K. C. Chou, *Proteins: Struct., Funct., Genet.*, 1995, **21**, 319–344.
36 K. C. Chou and D. W. Elrod, *Biochem. Biophys. Res. Commun.*, 1998, **252**, 63–68.
37 (*a*) K. C. Chou, *Proteins: Struct., Funct., Genet.*, 2001, **43**, 246–255; (*b*) K. C. Chou, *Proteins: Struct., Funct., Genet.*, 2001, **44**, 60.
38 K. C. Chou, *Bioinformatics*, 2005, **21**, 10–19.
39 S. S. Sahu and G. Panda, *Comput. Biol. Chem.*, 2010, **34**, 320–327.
40 M. Mohammad Beigi, M. Behjati and H. Mohabatkar, *J. Struct. Funct. Genomics*, 2011, **12**, 191–197.
41 S. W. Zhang, Y. L. Zhang, H. F. Yang, C. H. Zhao and Q. Pan, *Amino Acids*, 2008, **34**, 565–572.
42 L. Nanni and A. Lumini, *Amino Acids*, 2008, **34**, 653–660.

43 W. Z. Lin, J. A. Fang, X. Xiao and K. C. Chou, *PLoS One*, 2011, **6**, e24756.
44 H. Mohabatkar, M. M. Beigi, K. Abdolahi and S. Mohsenzadeh, *Med. Chem.*, 2013, **9**, 133–137.
45 L. Nanni, A. Lumini, D. Gupta and A. Garg, *IEEE/ACM Trans. Comput. Biol. Bioinf.*, 2012, **9**, 467–475.
46 J. Guo, N. Rao, G. Liu, Y. Yang and G. Wang, *J. Comput. Chem.*, 2011, **32**, 1612–1617.
47 H. Mohabatkar, M. Mohammad Beigi and A. Esmaeili, *J. Theor. Biol.*, 2011, **281**, 18–23.
48 D. Zou, Z. He, J. He and Y. Xia, *J. Comput. Chem.*, 2011, **32**, 271–278.
49 H. Mohabatkar, *Protein Pept. Lett.*, 2010, **17**, 1207–1214.
50 D. N. Georgiou, T. E. Karakasidis, J. J. Nieto and A. Torres, *J. Theor. Biol.*, 2009, **257**, 17–26.
51 X. B. Zhou, C. Chen, Z. C. Li and X. Y. Zou, *J. Theor. Biol.*, 2007, **248**, 546–551.
52 M. Esmaeili, H. Mohabatkar and S. Mohsenzadeh, *J. Theor. Biol.*, 2010, **263**, 203–209.
53 S. W. Zhang, W. Chen, F. Yang and Q. Pan, *Amino Acids*, 2008, **35**, 591–598.
54 X. Y. Sun, S. P. Shi, J. D. Qiu, S. B. Suo, S. Y. Huang and R. P. Liang, *Mol. BioSyst.*, 2012, **8**, 3178–3184.
55 R. Zia Ur and A. Khan, *Protein Pept. Lett.*, 2012, **19**, 890–903.
56 M. Hayat and A. Khan, *Protein Pept. Lett.*, 2012, **19**, 411–421.
57 W. Chen, H. Lin, P. M. Feng, C. Ding, Y. C. Zuo and K. C. Chou, *PLoS One*, 2012, **7**, e47843.
58 W. Chen, P. M. Feng, H. Lin and K. C. Chou, *Nucleic Acids Res.*, 2012, DOI: 10.1093/nar/gks1450.
59 P. Du, X. Wang, C. Xu and Y. Gao, *Anal. Biochem.*, 2012, **425**, 117–119.
60 H. B. Shen and K. C. Chou, *Anal. Biochem.*, 2008, **373**, 386–388.
61 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.
62 D. Barrell, E. Dimmer, R. P. Huntley, D. Binns, C. O'Donovan and R. Apweiler, *Nucleic Acids Res.*, 2009, **37**, D396–D403.
63 M. A. Harris, J. Clark, A. Ireland, J. Lomax, M. Ashburner, R. Foulger, K. Eilbeck, S. Lewis, B. Marshall, C. Mungall, J. Richter, G. M. Rubin, J. A. Blake, C. Bult, M. Dolan, H. Drabkin, J. T. Eppig, D. P. Hill, L. Ni, M. Ringwald, R. Balakrishnan, J. M. Cherry, K. R. Christie, M. C. Costanzo, S. S. Dwight, S. Engel, D. G. Fisk, J. E. Hirschman, E. L. Hong, R. S. Nash, A. Sethuraman, C. L. Theesfeld, D. Botstein, K. Dolinski, B. Feierbach, T. Berardini, S. Mundodi, S. Y. Rhee, R. Apweiler, D. Barrell, E. Camon, E. Dimmer, V. Lee, R. Chisholm, P. Gaudet, W. Kibbe, R. Kishore, E. M. Schwarz, P. Sternberg, M. Gwinn, L. Hannick, J. Wortman, M. Berriman, V. Wood, N. de la Cruz, P. Tonellato, P. Jaiswal, T. Seigfried and R. White, *Nucleic Acids Res.*, 2004, **32**, D258–D261.
64 K. C. Chou and H. B. Shen, *Nat. Protocols*, 2008, **3**, 153–162.
65 K. C. Chou and H. B. Shen, *PLoS One*, 2010, **5**, e11335.
66 K. C. Chou and H. B. Shen, *PLoS One*, 2010, **5**, e9931.
67 M. S. Scott, F. M. Boisvert, A. I. Lamond and G. J. Barton, *BMC Genomics*, 2011, **12**, 74.
68 T. Wang, J. Yang, H. B. Shen and K. C. Chou, *Protein Pept. Lett.*, 2008, **15**, 915–921.
69 H. B. Shen and K. C. Chou, *Anal. Biochem.*, 2009, **394**, 269–274.
70 K. C. Chou and H. B. Shen, *J. Proteome Res.*, 2007, **6**, 1728–1734.
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 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.
76 H. B. Shen and K. C. Chou, *J. Theor. Biol.*, 2010, **264**, 326–333.
77 K. C. Chou, *FEBS Lett.*, 1995, **363**, 123–126.
78 J. L. Deng, *J. Grey Syst.*, 1989, 1–24.
79 G. Tsoumakas and l. katakis, *Int. J. Data Warehousing Min.*, 2007, **3**, 13.
80 R. Cerri, R. da Silva and A. de Carvalho, in *Advances in Bioinformatics and Computational Biology*, ed. K. Guimarães, A. Panchenko and T. Przytycka, Springer, Berlin/Heidelberg, 2009, pp. 109–120.
81 R. Cerri, R. da Silva and A. de Carvalho, *Adv. Bioinf. Comput. Biol.*, 2009, **5676**, 109–120.
82 Y. K. Chen and K. B. Li, *J. Theor. Biol.*, 2013, **318**, 1–12.
83 M. Hayat and A. Khan, *J. Theor. Biol.*, 2012, **292**, 93–102.
84 S. Jahandideh, V. Srinivasasainagendra and D. Zhi, *J. Theor. Biol.*, 2012, **312**, 65–75.
85 L. Nanni, S. Brahnam and A. Lumini, *Amino Acids*, 2012, **43**, 657–665.
86 X. H. Niu, X. H. Hu, F. Shi and J. B. Xia, *Protein Pept. Lett.*, 2012, **19**, 940–948.
87 W. Z. Lin, J. A. Fang, X. Xiao and K. C. Chou, *PLoS One*, 2012, **7**, e49040.
88 S. Mei, *J. Theor. Biol.*, 2012, **293**, 121–130.
89 S. Mei, *J. Theor. Biol.*, 2012, **310**, 80–87.
90 K. C. Chou and H. B. Shen, *Nat. Sci.*, 2010, **2**, 1090–1103 (openly accessible at http://www.scirp.org/journal/NS/; DOI: 10.4236/ns.2010.210136).