

# MemType-2L: A Web server for predicting membrane proteins and their types by incorporating evolution information through Pse-PSSM

Kuo-Chen Chou<sup>a,b,\*</sup>, Hong-Bin Shen<sup>b,1</sup>

<sup>a</sup> Gordon Life Science Institute, 13784 Torrey Del Mar Drive, San Diego, CA 92130, USA

<sup>b</sup> Institute of Image Processing & Pattern Recognition, Shanghai Jiaotong University, Shanghai 200030, China

Received 16 May 2007

Available online 15 June 2007

## Abstract

Given an uncharacterized protein sequence, how can we identify whether it is a membrane protein or not? If it is, which membrane protein type it belongs to? These questions are important because they are closely relevant to the biological function of the query protein and to its interaction process with other molecules in a biological system. Particularly, with the avalanche of protein sequences generated in the Post-Genomic Age and the relatively much slower progress in using biochemical experiments to determine their functions, it is highly desired to develop an automated method that can be used to help address these questions. In this study, a 2-layer predictor, called MemType-2L, has been developed: the 1st layer prediction engine is to identify a query protein as membrane or non-membrane; if it is a membrane protein, the process will be automatically continued with the 2nd-layer prediction engine to further identify its type among the following eight categories: (1) type I, (2) type II, (3) type III, (4) type IV, (5) multipass, (6) lipid-chain-anchored, (7) GPI-anchored, and (8) peripheral. MemType-2L is featured by incorporating the evolution information through representing the protein samples with the Pse-PSSM (Pseudo Position-Specific Score Matrix) vectors, and by containing an ensemble classifier formed by fusing many powerful individual OET-KNN (Optimized Evidence-Theoretic K-Nearest Neighbor) classifiers. The success rates obtained by MemType-2L on a new-constructed stringent dataset by both the jackknife test and the independent dataset test are quite high, indicating that MemType-2L may become a very useful high throughput tool. As a Web server, MemType-2L is freely accessible to the public at <http://chou.med.harvard.edu/bioinf/MemType>.

© 2007 Elsevier Inc. All rights reserved.

**Keywords:** Membrane protein type; Protein evolution; Pse-PSSM; OET-KNN; Ensemble classifier; Fusion; MemType-2L

## Introduction

As a “building block of life”, a cell is deemed the most basic structural and functional unit of all living organisms.

*Abbreviations:* PSSM, position-specific scoring matrix; Pse-PSSM, pseudo position-specific scoring matrix; OET-KNN, optimized evidence-theoretic K nearest neighbor.

\* Corresponding author. Address: Gordon Life Science Institute, 13784 Torrey Del Mar Drive, San Diego, CA 92130, USA.

*E-mail addresses:* [kcchou@gordonlifescience.org](mailto:kcchou@gordonlifescience.org) (K.-C. Chou), [hbshen@crystal.harvard.edu](mailto:hbshen@crystal.harvard.edu) (H.-B. Shen).

<sup>1</sup> Present address: BCMP, Harvard Medical School, Boston, MA 02115, USA.

It is highly organized with many functional units or organelles according to the cellular anatomy. Most of these units are “enveloped” by one or more membranes, which are the structural basis for many important biological functions. Although the lipid bilayer is the basic structure of membranes, most of the specific functions of the cell membrane are performed by the membrane proteins (see, e.g., [1,2]). For example, it is through membrane proteins that molecules can be transported into and out of cells by such methods as ion pumps, channel proteins and carrier proteins; that various chemical messages such as nerve impulses and hormone activity can be passed between cells; that parts of the cytoskeleton can be attached to the cell

membrane in order to provide shape; that cells can be attached to an extracellular matrix in grouping cells together to form tissues; and that the metabolism process and body’s defense mechanisms can be completed.

Membrane proteins possess different types. The function of a membrane protein is closely correlated with the type it belongs to. For instance, the transmembrane proteins can function on both sides of membrane or transport molecules across it, whereas proteins that function on only one side of the lipid bilayer are often associated exclusively with either the lipid monolayer or a protein domain on that side. Therefore, information about membrane protein type often offers important clues toward determining the function of an uncharacterized membrane protein. Furthermore, owing to the fluid nature of their infrastructure, membrane proteins can move around the cell membrane and hence they do not sit in one place but can move to where their function is required. Knowing the type of a membrane protein can provide insight into this kind of motion, which is indispensable for studying the biological process at the cellular level from the dynamic point of view. Therefore, it will certainly expedite the pace in determining the function of uncharacterized membrane proteins and in understanding their action process if the knowledge of their type can be timely acquired. Particularly, the number of sequences entering into databanks has been rapidly increasing. For instance, the number of total protein sequence entries in Swiss-Prot was only 3939 in 1986; recently, the number jumped to 265,950 according to the version 52.4 released on 01-May-2007 at <http://www.ebi.ac.uk/swissprot/>, meaning that the number of the entries now is more than 67 times the number of 1986! With the explosion of protein sequences entering into databanks and the fact that membrane proteins are encoded by 20–35% of genes but represent <1% of known protein structures to date [3], it is highly desirable to develop a sequence-based automated method for fast and effectively identifying a newly found protein according to the following two questions. (1) Is it a membrane protein? (2) If it is, which type does it belong to?

Actually, during the last eight years various prediction methods have been proposed in this area [4–14], yet all these methods have some of the following problems needed to be further addressed. (1) They were developed based on such a prerequisite that the query protein was already known belonging to membrane proteins without efforts made to identify whether the query protein was a membrane protein or non-membrane protein. To make the case logically more reasonable and practically more useful, such a procedure is indispensable. (2) The reported success rates were derived based on a benchmark dataset without being rigorously screened by a clear data-culling operation to avoid redundancy and homologous bias, and hence the reported success rates therein might be overestimated. (3) Only five membrane types were covered; with the development of protein databases, more types should be included to increase the scope of practical application. (4) None of these methods has provided a Web server for the public

usage, and hence their practical application value is quite limited. In this paper the aforementioned four problems will be explicitly addressed.

### Materials

Protein sequences were collected from the Swiss-Prot database at <http://www.ebi.ac.uk/swissprot/> (version 51.0 released on 6-October-2006). In order to collect as much desired information as possible and meanwhile ensure a high-quality for the benchmark dataset, the data were screened strictly according to the following criteria and order. (1) Sequences annotated with “fragment” were excluded; also, sequences with less than 50 amino acid residues were excluded because they might just be fragments. (2) Sequences annotated with ambiguous or uncertain terms, such as “potential”, “probable”, “probably”, “maybe”, or “by similarity”, were removed for further consideration. (3) For the sequences kept after the above screen procedures that all have clear experimental annotations, those annotated with “membrane protein” were stored in the membrane protein reservoir  $\mathbb{R}_{\text{mem}}$ ; while the rest stored in the non-membrane protein reservoir  $\mathbb{R}_{\text{non-mem}}$ . (4) Eight different membrane protein types (Fig. 1) were found in  $\mathbb{R}_{\text{mem}}$ ; to reduce the homology bias, a redundancy cutoff was operated by an in-house program to winnow those sequences which have  $\geq 80\%$  sequence identity to any other in a same membrane type. (5) A similar cutoff procedure was operated for the sequences in  $\mathbb{R}_{\text{non-mem}}$ ; from the data obtained after such a redundancy-reducing cutoff procedure, sequences were randomly picked to form the benchmark dataset for non-membrane proteins. Finally, we obtained a dataset  $\mathbb{S}$  containing 15,547 sequences of which 7582 belong to membrane proteins while 7965 to non-membrane proteins. According to their experimental annotations, the 7582 membrane proteins can be further classified into eight subsets. Thus, we have

$$\begin{cases} \mathbb{S} = \mathbb{S}^{\text{non-mem}} \cup \mathbb{S}^{\text{mem}} \\ \mathbb{S}^{\text{mem}} = \mathbb{S}_1^{\text{mem}} \cup \mathbb{S}_2^{\text{mem}} \cup \mathbb{S}_3^{\text{mem}} \cup \mathbb{S}_4^{\text{mem}} \cup \mathbb{S}_5^{\text{mem}} \cup \mathbb{S}_6^{\text{mem}} \cup \mathbb{S}_7^{\text{mem}} \cup \mathbb{S}_8^{\text{mem}} \end{cases} \quad (1)$$

where  $\mathbb{S}^{\text{non-mem}}$  is the set containing non-membrane proteins only,  $\cup$  is the symbol for union in the set theory,  $\mathbb{S}^{\text{mem}}$  is the set containing membrane proteins only,  $\mathbb{S}_1^{\text{mem}}$  is the subset containing the single-pass type I membrane proteins only,  $\mathbb{S}_2^{\text{mem}}$  is that for type II only, and so forth (Table 1).

On the basis of the membrane protein dataset  $\mathbb{S}^{\text{mem}}$ , two working datasets, i.e., a training dataset  $\mathbb{S}_A^{\text{mem}}$  and an independent testing dataset  $\mathbb{S}_B^{\text{mem}}$ , were constructed. In order to fully use the data in  $\mathbb{S}^{\text{mem}}$  and meanwhile guarantee that  $\mathbb{S}_A^{\text{mem}}$  and  $\mathbb{S}_B^{\text{mem}}$  be completely independent of each other, the following condition was imposed:

$$\mathbb{S}_A^{\text{mem}} \cup \mathbb{S}_B^{\text{mem}} = \mathbb{S}^{\text{mem}} \quad \text{and} \quad \mathbb{S}_A^{\text{mem}} \cap \mathbb{S}_B^{\text{mem}} = \emptyset \quad (2)$$

where  $\cup$ ,  $\cap$ , and  $\emptyset$  represent the symbols for “union”, “intersection”, and “empty set” in the set theory, respectively. To avoid the situation that the numbers of proteins in some subsets of the training dataset  $\mathbb{S}_A^{\text{mem}}$  might overwhelm those of the others, the following “bracket percentage distribution” criterion was used to randomly assign the protein samples to the corresponding subsets of  $\mathbb{S}_A^{\text{mem}}$  and  $\mathbb{S}_B^{\text{mem}}$ :

$$\begin{cases} n_i^A = 500 + \text{INT}\{(n_i - 500) \times 0.2\} & \text{if } n_i \geq 500 \\ n_i^A = \text{INT}\{n_i \times 0.8\} & \text{if } n_i < 500 \quad (i = 1, 2, \dots, 8) \\ n_i^B = n_i - n_i^A \end{cases} \quad (3)$$

where  $n_i$  is the number of protein samples in the  $i$ th subset of the original membrane protein dataset  $\mathbb{S}^{\text{mem}}$  (Eq. (1)),  $n_i^A$  that of the training dataset  $\mathbb{S}_A^{\text{mem}}$  (Eq. (2)),  $n_i^B$  that of the testing dataset  $\mathbb{S}_B^{\text{mem}}$ , and the symbol INT is the “integer truncation operator” meaning to take the integer part for the number in the brackets right after it. The numbers of proteins thus obtained for the eight membrane protein types in the training dataset  $\mathbb{S}_A^{\text{mem}}$  and testing dataset  $\mathbb{S}_B^{\text{mem}}$  are given in Table 2. The accession numbers and sequences for the corresponding membrane proteins in the training and testing datasets are given in Online Supporting Information A and B, respectively. Also, the accession numbers and sequences for the non-membrane protein benchmark dataset  $\mathbb{S}^{\text{non-mem}}$  are given in Online Supporting Information C.

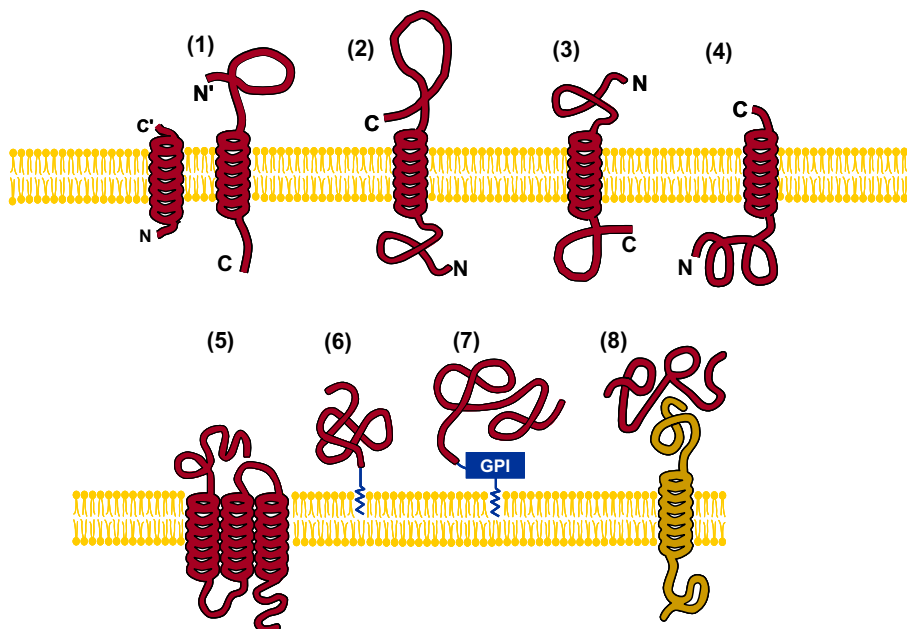


Fig. 1. Schematic illustration to show the eight types of membrane proteins: (1) type I transmembrane, (2) type II, (3) type III, (4) type IV, (5) multipass transmembrane, (6) lipid-chain-anchored membrane, (7) GPI-anchored membrane, and (8) peripheral membrane. As shown in the figure, types I, II, III, and IV are all of single-pass transmembrane proteins; see [56] for a detailed description about their difference.

Table 1  
Breakdown of the membrane and non-membrane protein sequences obtained by following steps (1)–(5) in Materials

Attribute	Type	Subset	Number of sequences
Membrane	Single-pass type I	$\mathbb{S}_1^{\text{mem}}$	1054
	Single-pass type II	$\mathbb{S}_2^{\text{mem}}$	390
	Single-pass type III	$\mathbb{S}_3^{\text{mem}}$	30
	Single-pass type IV	$\mathbb{S}_4^{\text{mem}}$	56
	Multipass	$\mathbb{S}_5^{\text{mem}}$	4581
	Lipid-chain-anchor	$\mathbb{S}_6^{\text{mem}}$	189
	GPI-anchor	$\mathbb{S}_7^{\text{mem}}$	228
	Peripheral	$\mathbb{S}_8^{\text{mem}}$	1054
Overall		$\mathbb{S}^{\text{mem}}$	7582
Non-membrane		$\mathbb{S}^{\text{non-mem}}$	7965

Table 2  
Number of membrane proteins in each of the eight types for the training and testing datasets randomly generated according to Eq. (3)

Subset	Type	Number of sequences in the training dataset $\mathbb{S}_A^{\text{mem}}$	Number of sequences in the testing dataset $\mathbb{S}_B^{\text{mem}}$
$\mathbb{S}_1^{\text{mem}}$	Single-pass type I	610	444
$\mathbb{S}_2^{\text{mem}}$	Single-pass type II	312	78
$\mathbb{S}_3^{\text{mem}}$	Single-pass type III	24	6
$\mathbb{S}_4^{\text{mem}}$	Single-pass type IV	44	12
$\mathbb{S}_5^{\text{mem}}$	Multipass	1316	3265
$\mathbb{S}_6^{\text{mem}}$	Lipid-chain-anchor	151	38
$\mathbb{S}_7^{\text{mem}}$	GPI-anchor	182	46
$\mathbb{S}_8^{\text{mem}}$	Peripheral	610	444
$\mathbb{S}^{\text{mem}}$	Overall	3249	4333

## Methods

Once the benchmark dataset is established, the subsequent problem is how to find an effective prediction engine and use what kind of profile to represent the protein samples for training the engine and conducting the prediction. Translated into a mathematical language, the problem can be formulated as

$$\mathbb{Q}@\mathbf{P} = C \in \begin{cases} \mathbb{S}^{\text{non-mem}} \cup \mathbb{S}^{\text{mem}} & \text{if among non-membrane and membrane} \\ \mathbb{S}^{\text{mem}} & \text{if among eight membrane protein types} \end{cases} \quad (4)$$

where  $\mathbb{Q}$  represents the prediction engine,  $\mathbf{P}$  the query protein,  $@$  is an action operator,  $C$  the predicted result,  $\in$  is a symbol in the set theory meaning “member of”, and  $\mathbb{S}$  is defined by Eq. (1). Before the prediction engine can be used, it must be trained by a training dataset where all the proteins must have the same profile as that of the query protein  $\mathbf{P}$ .

Various prediction engines have been introduced in this regard, such as BLAST [15], Covariant Discriminant algorithm [16], SVM [6,17], Weighted-SVM [7], SLLE [10], KNN [18,19], and Fuzzy-KNN [11]. In this

study, the OET-KNN (Optimized Evidence-Theoretic K Nearest Neighbor) classifier was utilized to identify the membrane proteins and their types. The OET-KNN classifier is a very powerful classification engine as demonstrated by its role in enhancing the success rates of protein sub-cellular localization [20], where a detailed formulation of OET-KNN classifier can be found.

As for the representation of protein samples, two different models were generally adopted: (1) sequential model; (2) discrete model. In the sequential model, the sample of a protein is represented by its amino acid sequence, and the sequence similarity search-based tools such as BLAST [21] are used to conduct prediction. However, this approach failed to work when a query protein did not have significant homology to character-known proteins. Thus, various discrete models were introduced by representing the sample of a protein with a set of discrete numbers. The simplest discrete model is to represent the sample of a protein with its amino acid composition (AAC) (see, e.g., [22–24]). However, in the AAC model, all the sequence-order effects are lost. To avoid completely lose the sequence-order information, the pseudo amino acid composition

(Pse-AAC) was introduced [25]. Using the Pse-AAC discrete model to represent protein samples can incorporate some sequence-order information through a set of correlation factors called “pseudo amino acid components”, and hence remarkably enhance the success rates in predicting various attributes of proteins as demonstrated by a series of recent publications (see, e.g., [26–37]). Because the Pse-AAC discrete model has been increasingly used, recently a Web server called PseAA was established at <http://chou.med.harvard.edu/bioinf/PseAA/>. Using the Web server, one can easily generate the pseudo amino acid components for any given protein sequence.

In this study, we are to introduce a new representation for the sample of a protein by incorporating its evolution information. To realize this, the PSSM (Position-Specific Scoring Matrix) [15] was used; i.e., the sample of a protein sequence **P** is represented by:

$$\mathbf{P}_{\text{PSSM}} = \begin{bmatrix} \mathbb{E}_{1 \rightarrow 1} & \mathbb{E}_{1 \rightarrow 2} & \cdots & \mathbb{E}_{1 \rightarrow 20} \\ \mathbb{E}_{2 \rightarrow 1} & \mathbb{E}_{2 \rightarrow 2} & \cdots & \mathbb{E}_{2 \rightarrow 20} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbb{E}_{i \rightarrow 1} & \mathbb{E}_{i \rightarrow 2} & \cdots & \mathbb{E}_{i \rightarrow 20} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbb{E}_{L \rightarrow 1} & \mathbb{E}_{L \rightarrow 2} & \cdots & \mathbb{E}_{L \rightarrow 20} \end{bmatrix} \quad (5)$$

where  $\mathbb{E}_{i \rightarrow j}$  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 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. The  $L \times 20$  scores in Eq. (5) were generated by using PSI-BLAST [15] to search the Swiss-Prot database (version 52.0 released on 6-March-2007) through three iterations with 0.001 as the E-value cutoff for multiple sequence alignment against the sequence of the protein **P**, followed by a standardization procedure given below:

$$\mathbb{E}_{i \rightarrow j} = \frac{\mathbb{E}_{i \rightarrow j}^0 - \frac{1}{20} \sum_{k=1}^{20} \mathbb{E}_{i \rightarrow k}^0}{\sqrt{\frac{1}{20} \sum_{u=1}^{20} (\mathbb{E}_{i \rightarrow u}^0 - \frac{1}{20} \sum_{k=1}^{20} \mathbb{E}_{i \rightarrow k}^0)^2}} \quad (i = 1, 2, \dots, L; j = 1, 2, \dots, 20) \quad (6)$$

where  $\mathbb{E}_{i \rightarrow j}^0$  represent the original scores directly created by PSI-BLAST that are generally shown as positive or negative integers. The standardized scores will have a zero mean value over the 20 amino acids and will remained unchanged if going through the same conversion procedure again. The positive score means that the corresponding mutation occurs more frequently in the alignment than expected by chance, while the negative one means just the opposite. However, according to the PSSM descriptor (Eq. (5)), proteins with different lengths will correspond to row-different matrices. To make the PSSM descriptor become a size-uniform matrix, one possible approach is to represent a protein sample **P** by

$$\mathbf{P}_{\text{PSSM}} = [\bar{\mathbb{E}}_1 \quad \bar{\mathbb{E}}_2 \quad \cdots \quad \bar{\mathbb{E}}_{20}]^T \quad (7)$$

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

$$\bar{\mathbb{E}}_j = \frac{1}{L} \sum_{i=1}^L \mathbb{E}_{i \rightarrow j} \quad (j = 1, 2, \dots, 20) \quad (8)$$

where  $\bar{\mathbb{E}}_j$  represents the average score of the amino acid residues in the protein **P** being changed to amino acid type  $j$  during the evolution process. However, if  $\mathbf{P}_{\text{PSSM}}$  of Eq. (7) was used to represent the protein **P**, all the sequence-order information during the evolution process would be lost. To avoid complete loss of the sequence-order information, the concept of the pseudo amino acid composition as originally proposed in [25,38] was adopted; i.e., instead of Eq. (7), let us use the pseudo position-specific scoring matrix (Pse-PSSM) as given by

$$\mathbf{P}_{\text{Pse-PSSM}}^{\xi} = [\bar{\mathbb{E}}_1 \quad \bar{\mathbb{E}}_2 \quad \cdots \quad \bar{\mathbb{E}}_{20} \quad G_1^{\xi} \quad G_2^{\xi} \quad \cdots \quad G_{20}^{\xi}]^T \quad (9)$$

to represent the protein **P**, where

$$G_j^{\xi} = \frac{1}{L - \xi} \sum_{i=1}^{L-\xi} [\mathbb{E}_{i \rightarrow j} - \mathbb{E}_{(i+\xi) \rightarrow j}]^2 \quad (j = 1, 2, \dots, 20; \xi < L) \quad (10)$$

meaning that  $G_j^1$  is the correlation factor by coupling the most contiguous PSSM scores along the protein chain for the amino acid type  $j$ ;  $G_j^2$  that by coupling the second-most contiguous PSSM scores; and so forth. Note that, as mentioned in the Material section, the length of the shortest protein sequence in the benchmark dataset is  $L = 50$ , and hence the value allowed for  $\xi$  in Eq. (10) must be smaller than 50. When  $\xi = 0$ ,  $G_j^{\xi}$  becomes a naught element and Eq. (6) is degenerated to Eq. (7).

Thus, according to the Pse-PSSM descriptor (Eqs. 9 and 10) a protein can be represented by one 20-D (dimensional) vector ( $\xi = 0$ ) and 49 different 40-D vectors each of which corresponds to a different  $\xi$  (1, ..., or 49). To avoid the over-fitting problem and reducing the cluster-tolerance capacity [39], instead of combining the 50 individual vectors ( $\xi = 0, 1, 2, \dots, 49$ ) into one  $(20 + 40 \times 49) = 1980$ -D vector, let us introduce an ensemble classifier by fusing the results obtained based on each of the individual vector descriptors through a voting system as formulated below.

As mentioned above, in this study the OET-KNN was used as the prediction engine and the query protein **P** represented by  $\mathbf{P}_{\text{Pse-PSSM}}^{\xi}$ . Thus, according to Eq. (4) we have

$$\begin{aligned} \text{OET-KNN} @ \mathbf{P}_{\text{Pse-PSSM}}^{\xi} &= C(K, \xi) \\ &\in \begin{cases} \mathbb{S}^{\text{non-mem}} \cup \mathbb{S}^{\text{mem}} & \text{if among non-membrane and membrane} \\ \mathbb{S}^{\text{mem}} & \text{if among eight membrane protein types} \end{cases} \quad (11) \\ &(K = 1, 2, \dots, 10; \xi = 0, 1, \dots, 49) \end{aligned}$$

where  $K$  is the number of the nearest proteins counted against the query protein during the prediction process, while  $C(1,0)$  is the result predicted with OET-KNN on  $\mathbf{P}_{\text{Pse-PSSM}}^0$  according to the 1-nearest-neighbor rule,  $C(2,1)$  is the result predicted with OET-KNN on  $\mathbf{P}_{\text{Pse-PSSM}}^1$  according to the 2-nearest-neighbor rule, and so forth. Generally speaking, for most training datasets, when  $K > 10$  the success rate drops down remarkably and hence we can narrow the scope of  $K$  from 1 to 10. The  $10 \times 50 = 500$  results of  $C(K, \xi)$  in Eq. (11) were fused into one through the following voting mechanism:

Suppose the voting score for the query protein **P** belonging to the  $i$ -th group  $\mathbb{G}_i$  is given by

$$Q_i = \sum_{K=1}^{10} \sum_{\xi=0}^{49} w_{K,\xi} \Delta\{C(K, \xi), \mathbb{G}_i\}, \quad (i = 1, 2, \dots, m) \quad (12)$$

where  $w_{K,\xi}$  is the weight and was set at 1 for simplicity, the delta function in Eq. (12) is given by

$$\Delta\{C(K, \xi), \mathbb{G}_i\} = \begin{cases} 1 & \text{if } C(K, \xi) \in \mathbb{G}_i \\ 0, & \text{otherwise} \end{cases} \quad (i = 1, 2, \dots, m) \quad (13)$$

thus the query protein **P** is predicted belonging to the group or subset for which the score of Eq. (12) is the highest; i.e.,

$$\mu = \arg \max_i \{Q_i\}, \quad (i = 1, 2, \dots, m) \quad (14)$$

where  $\mu$  is the argument of  $i$  that maximize  $Q_i$ . If there is a tie among two or more subsets, then the final outcome will be randomly assigned to one of their corresponding subsets although this kind of tie case rarely happens and actually was not observed in the current study.

The above algorithm is called Pse-PSSM OET-KNN ensemble classifier. When it is used to identify a query protein as membrane or non-membrane ( $m = 2$ ), just substitute  $\mathbb{G}_1$  and  $\mathbb{G}_2$  of Eqs. 12 and 13 with  $\mathbb{S}^{\text{mem}}$  and  $\mathbb{S}^{\text{non-mem}}$ , respectively; however, when used for identifying the membrane protein type ( $m = 8$ ),  $\mathbb{S}_i^{\text{mem}}$  ( $i = 1, 2, \dots, 8$ ) should be used to substitute  $\mathbb{G}_i$ . The overall predictor is called MemType-2L.

To provide an intuitive picture, a flowchart to show the process of how the Pse-PSSM OET-KNN ensemble classifier works is given in Fig. 2A, and the corresponding flowchart for the MemType-2L given in Fig. 2B.

## Results and discussion

In statistical prediction the independent dataset test, sub-sampling test, and jackknife test are often used in

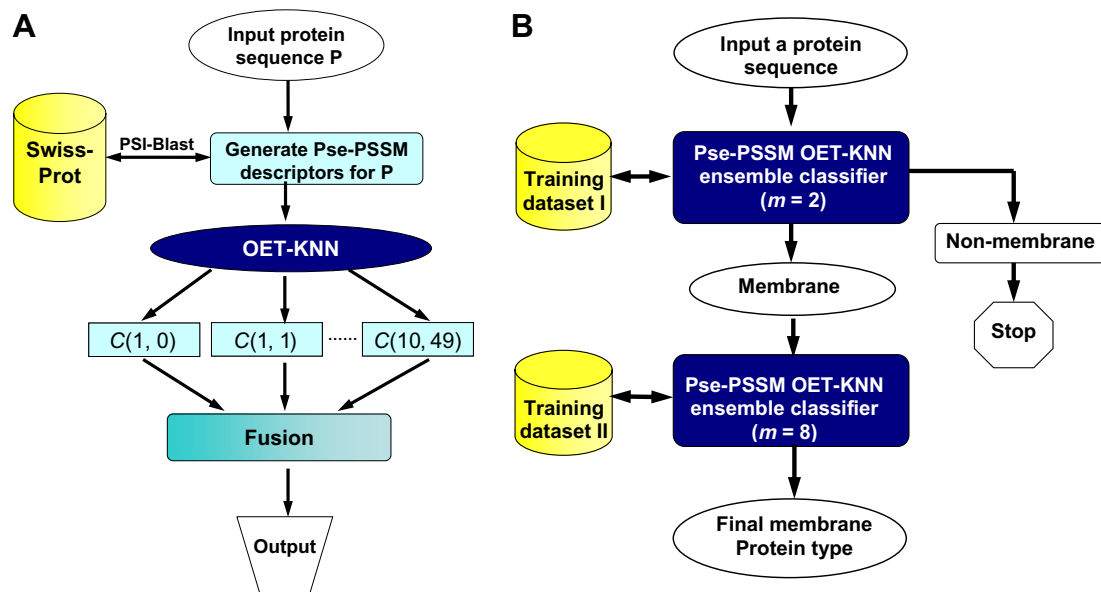


Fig. 2. A flowchart to show (A) the Pse-PSSM OET-KNN ensemble classifier, and (B) the MemType-2L.

Table 3

Success rates in discriminating membrane and non-membrane proteins by the jackknife test with different methods

Protein	Least Euclidean distance [54] (%)	ProtLoc [55] (%)	MemType-2L (%)
Membrane <sup>a</sup>	$\frac{5320}{7582} = 70.2$	$\frac{5512}{7582} = 72.7$	$\frac{6897}{7582} = 91.0$
Non-membrane <sup>b</sup>	$\frac{6688}{7965} = 84.0$	$\frac{6754}{7965} = 84.8$	$\frac{7520}{7965} = 94.4$
Overall	$\frac{12008}{15547} = 77.2$	$\frac{12266}{15547} = 78.9$	$\frac{14417}{15547} = 92.7$

<sup>a</sup> The data for the membrane proteins were taken from  $S_A^{\text{mem}}$  (Table 1), and their sequences are given in Online Supporting Information A and B.

<sup>b</sup> The data for the non-membrane proteins were taken from  $S_B^{\text{non-mem}}$  (Table 1), and their sequences are given in Online Supporting Information C.

literatures for examining the accuracy of a predictor. Among the three cross-validation examinations, the jackknife test is deemed the most rigorous and objective [40], and hence has been increasingly adopted by investigators [7–9,11–13,19,24,26–29,31–35,37,41–53] in examining the quality of various prediction methods. Therefore, the accuracy of a predictor should be mainly evaluated by the success rate of the jackknife test. However, as a demonstration to show the practical application, predictions were also made on the independent dataset  $S_B^{\text{mem}}$  as done in [4].

The predicted results obtained by MemType-2L are given in Tables 3 and 4, where, for facilitating comparison, the corresponding results by the other methods are also

Table 4

Success rates in identifying membrane protein types by the jackknife test and independent dataset test with different methods

Type	Jackknife test			Independent dataset test		
	Least Euclidean distance [54] (%)	ProtLoc [55] (%)	MemType-2L (%)	Least Euclidean distance [54] (%)	ProtLoc [55] (%)	MemType-2L (%)
Single-pass type I	$\frac{329}{610} = 53.9$	$\frac{329}{610} = 53.9$	$\frac{532}{610} = 87.2$	$\frac{229}{444} = 51.6$	$\frac{241}{444} = 54.3$	$\frac{386}{444} = 86.9$
Single-pass type II	$\frac{227}{312} = 29.2$	$\frac{125}{312} = 40.1$	$\frac{227}{312} = 72.8$	$\frac{27}{78} = 34.6$	$\frac{35}{78} = 44.9$	$\frac{55}{78} = 70.5$
Single-pass type III	$\frac{9}{24} = 37.5$	$\frac{8}{24} = 33.3$	$\frac{10}{24} = 41.7$	$\frac{1}{6} = 16.7$	$\frac{2}{6} = 33.3$	$\frac{2}{6} = 33.3$
Single-pass type IV	$\frac{32}{44} = 72.7$	$\frac{29}{44} = 65.9$	$\frac{33}{44} = 75.0$	$\frac{7}{12} = 58.3$	$\frac{8}{12} = 66.7$	$\frac{8}{12} = 66.7$
Multipass	$\frac{948}{1316} = 72.0$	$\frac{834}{1316} = 63.4$	$\frac{1260}{1316} = 95.7$	$\frac{2282}{3265} = 69.9$	$\frac{1148}{3265} = 35.2$	$\frac{3103}{3265} = 95.0$
Lipid-chain-anchor	$\frac{67}{151} = 44.4$	$\frac{69}{151} = 45.7$	$\frac{85}{151} = 56.3$	$\frac{14}{38} = 36.8$	$\frac{15}{38} = 39.5$	$\frac{16}{38} = 42.1$
GPI-anchor	$\frac{125}{182} = 42.3$	$\frac{78}{182} = 42.9$	$\frac{125}{182} = 68.7$	$\frac{18}{46} = 39.1$	$\frac{22}{46} = 47.8$	$\frac{35}{46} = 76.1$
Peripheral	$\frac{126}{610} = 20.7$	$\frac{78}{610} = 12.8$	$\frac{491}{610} = 80.5$	$\frac{81}{444} = 18.2$	$\frac{139}{444} = 31.3$	$\frac{365}{444} = 82.2$
Overall	$\frac{1679}{3249} = 51.7$	$\frac{1688}{3249} = 52.0$	$\frac{2763}{3249} = 85.0$	$\frac{2659}{4333} = 61.4$	$\frac{1610}{4333} = 37.2$	$\frac{3970}{4333} = 91.6$

<sup>a</sup>The jackknife test was performed on the 3,249 membrane proteins in the dataset  $S_A^{\text{mem}}$  (Table 2). See Online Supporting Information A for the sequences of the proteins in  $S_A^{\text{mem}}$ .

<sup>b</sup>Predictions were made by the ensemble classifier trained with the data in  $S_A^{\text{mem}}$  on the 4333 membrane proteins in  $S_B^{\text{mem}}$ . See Online Supporting Information B for the sequences of the proteins in  $S_B^{\text{mem}}$ .

listed. From the two tables, the following outcomes have been observed. (1) The overall jackknife success rate by the current MemType-2L in discriminating membrane and non-membrane proteins is 92.7%, which is about 13–16% higher than those by the least Euclidean algorithm [54] and ProtLoc [55] based on the conventional amino acid composition. (2) The overall jackknife success rate by MemType-2L in identifying the membrane protein type is 85.0%, which is about 33% higher than those by the other methods. (3) The overall independent dataset test success rate is 91.6%, which is about 30–54% higher than those by the other methods. All these indicate that MemType-2L is indeed very useful in identifying membrane proteins and their types.

## Conclusion

MemType-2L is a 2-layer predictor: the 1st layer prediction engine is to identify whether a query protein is membrane or not; the 2nd layer is to identify its type if the outcome from the 1st layer turns out to be positive. Compared with the existing predictors covering only 5–6 membrane protein types, MemType-2L can cover eight types. The high success rates obtained by MemType-2L is because (1) it takes into account the evolution information by representing the protein samples with the Pse-PSSM vectors derived from the results generated by PSI-BLAST, and (2) it operates by fusing many powerful individual classifiers so as to minimize both the information-missing problem and the over-fitting problem.

To support the people working in the relevant area, a Web server called **MemType-2L** is provided at <http://chou.med.harvard.edu/bioinf/MemType> that is freely accessible to the public and very user-friendly as well.

## Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at [doi:10.1016/j.bbrc.2007.06.027](https://doi.org/10.1016/j.bbrc.2007.06.027).

## References

- [1] H. Lodish, D. Baltimore, A. Berk, S.L. Zipursky, P. Matsudaira, J. Darnell, *Molecular Cell Biology*, Scientific American Books, New York, 1995, Chapter 3.
- [2] B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, J.D. Watson, *Molecular Biology of the Cell*, Garland Publishing, New York & London, 1994, chapter 1.
- [3] S.M. Douglas, J.J. Chou, W.M. Shih, DNA-nanotube-induced alignment of membrane proteins for NMR structure determination, *Proc. Natl. Acad. Sci. USA* 104 (2007) 6644–6648.
- [4] K.C. Chou, D.W. Elrod, Prediction of membrane protein types and subcellular locations, *Proteins: Struct., Funct., Genet.* 34 (1999) 137–153.
- [5] Z. M. Guo, Prediction of Membrane protein types by using pattern recognition method based on pseudo amino acid composition, Master Thesis, Bio-X Life Science Research Center, Shanghai Jiaotong University, 2002.
- [6] Y.D. Cai, G.P. Zhou, K.C. Chou, Support vector machines for predicting membrane protein types by using functional domain composition, *Biophys. J.* 84 (2003) 3257–3263.
- [7] M. Wang, J. Yang, G.P. Liu, Z.J. Xu, K.C. Chou, Weighted-support vector machines for predicting membrane protein types based on pseudo amino acid composition, *Protein Eng., Des., Sel.* 17 (2004) 509–516.
- [8] H. Liu, M. Wang, K.C. Chou, Low-frequency Fourier spectrum for predicting membrane protein types, *Biochem. Biophys. Res. Commun.* 336 (2005) 737–739.
- [9] H.B. Shen, K.C. Chou, Using optimized evidence-theoretic K-nearest neighbor classifier and pseudo amino acid composition to predict membrane protein types, *Biochem. Biophys. Res. Commun.* 334 (2005) 288–292.
- [10] M. Wang, J. Yang, Z.J. Xu, K.C. Chou, SLLE for predicting membrane protein types, *J. Theor. Biol.* 232 (2005) 7–15.
- [11] H.B. Shen, J. Yang, K.C. Chou, Fuzzy KNN for predicting membrane protein types from pseudo amino acid composition, *J. Theor. Biol.* 240 (2006) 9–13.
- [12] S.Q. Wang, J. Yang, K.C. Chou, Using stacked generalization to predict membrane protein types based on pseudo amino acid composition, *J. Theor. Biol.* 242 (2006) 941–946.
- [13] X.G. Yang, R.Y. Luo, Z.P. Feng, Using amino acid and peptide composition to predict membrane protein types, *Biochem. Biophys. Res. Commun.* 353 (2007) 164–169.
- [14] H.B. Shen, K.C. Chou, Using ensemble classifier to identify membrane protein types, *Amino Acids* 32 (2007) 483–488.
- [15] A.A. Schaffer, L. Aravind, T.L. Madden, S. Shavirin, J.L. Spouge, Y.I. Wolf, E.V. Koonin, S.F. Altschul, Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements, *Nucleic Acids Res.* 29 (2001) 2994–3005.
- [16] K.C. Chou, D.W. Elrod, Protein subcellular location prediction, *Protein Eng.* 12 (1999) 107–118.
- [17] K.C. Chou, Y.D. Cai, Using functional domain composition and support vector machines for prediction of protein subcellular location, *J. Biol. Chem.* 277 (2002) 45765–45769.
- [18] J.H. Friedman, F. Baskett, L.J. Shustek, An algorithm for finding nearest neighbors, *IEEE Trans. Inform. Theory* C-24 (1975) 1000–1006.
- [19] K.C. Chou, H.B. Shen, Hum-PLoc: A novel ensemble classifier for predicting human protein subcellular localization, *Biochem. Biophys. Res. Commun.* 347 (2006) 150–157.
- [20] K.C. Chou, H.B. Shen, Predicting eukaryotic protein subcellular location by fusing optimized evidence-theoretic K-nearest neighbor classifiers, *J. Proteome Res.* 5 (2006) 1888–1897.
- [21] S.F. Altschul, T.L. Madden, A.A. Schaffer, J. Zhang, Z. Zhang, W. Miller, D.J. Lipman, Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, *Nucleic Acids Res.* 25 (1997) 3389–3402.
- [22] K.C. Chou, C.T. Zhang, Predicting protein folding types by distance functions that make allowances for amino acid interactions, *J. Biol. Chem.* 269 (1994) 22014–22020.
- [23] K.C. Chou, A novel approach to predicting protein structural classes in a (20-1)-D amino acid composition space, *Proteins: Struct., Funct. Genet.* 21 (1995) 319–344.
- [24] G.P. Zhou, An intriguing controversy over protein structural class prediction, *J. Protein Chem.* 17 (1998) 729–738.
- [25] K. C. Chou, Prediction of protein cellular attributes using pseudo amino acid composition, *Proteins: Struct., Funct., Genet.* (Erratum: *ibid.*, 2001, vol.44, 60) 43 (2001) 246–255.
- [26] C. Chen, X. Zhou, Y. Tian, X. Zou, P. Cai, Predicting protein structural class with pseudo-amino acid composition and support vector machine fusion network, *Anal. Biochem.* 357 (2006) 116–121.
- [27] X. Xiao, S.H. Shao, Y.S. Ding, Z.D. Huang, K.C. Chou, Using cellular automata images and pseudo amino acid composition to predict protein subcellular location, *Amino Acids* 30 (2006) 49–54.

- [28] C. Chen, Y.X. Tian, X.Y. Zou, P.X. Cai, J.Y. Mo, Using pseudo-amino acid composition and support vector machine to predict protein structural class, *J. Theor. Biol.* 243 (2006) 444–448.
- [29] X. Xiao, S.H. Shao, Z.D. Huang, K.C. Chou, Using pseudo amino acid composition to predict protein structural classes: approached with complexity measure factor, *J. Comput. Chem.* 27 (2006) 478–482.
- [30] S.W. Zhang, Q. Pan, H.C. Zhang, Z.C. Shao, J.Y. Shi, Prediction protein homo-oligomer types by pseudo amino acid composition: Approached with an improved feature extraction and naive Bayes feature fusion, *Amino Acids* 30 (2006) 461–468.
- [31] P. Du, Y. Li, Prediction of protein submitochondria locations by hybridizing pseudo-amino acid composition with various physico-chemical features of segmented sequence, *BMC Bioinform.* 7 (2006) 518.
- [32] S. Mondal, R. Bhavna, R. Mohan Babu, S. Ramakumar, Pseudo amino acid composition and multi-class support vector machines approach for conotoxin superfamily classification, *J. Theor. Biol.* 243 (2006) 252–260.
- [33] H.B. Shen, K.C. Chou, Ensemble classifier for protein fold pattern recognition, *Bioinformatics* 22 (2006) 1717–1722.
- [34] H. Lin, Q.Z. Li, Predicting conotoxin superfamily and family by using pseudo amino acid composition and modified Mahalanobis discriminant, *Biochem. Biophys. Res. Commun.* 354 (2007) 548–551.
- [35] H. Lin, Q.Z. Li, Using pseudo amino acid composition to predict protein structural class: approached by incorporating 400 dipeptide components, *J. Comput. Chem.* 28 (2007) 1463–1466.
- [36] J.Y. Shi, S.W. Zhang, Q. Pan, Y.-M. Cheng, J. Xie, Prediction of protein subcellular localization by support vector machines using multi-scale energy and pseudo amino acid composition, *Amino Acids* (2007), doi:10.1007/s00726-00006-00475-y.
- [37] X. Pu, J. Guo, H. Leung, Y. Lin, Prediction of membrane protein types from sequences and position-specific scoring matrices, *J. Theor. Biol.* (2007), doi:10.1016/j.jtbi.2007.1001.1016.
- [38] K.C. Chou, Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes, *Bioinformatics* 21 (2005) 10–19.
- [39] K.C. Chou, A key driving force in determination of protein structural classes, *Biochem. Biophys. Res. Commun.* 264 (1999) 216–224.
- [40] K.C. Chou, C.T. Zhang, Review: prediction of protein structural classes, *Crit. Rev. Biochem. Mol. Biol.* 30 (1995) 275–349.
- [41] G.P. Zhou, K. Doctor, Subcellular location prediction of apoptosis proteins, *Proteins: Struct., Funct., Genet.* 50 (2003) 44–48.
- [42] K.C. Chou, H.B. Shen, Large-scale plant protein subcellular location prediction, *J. Cell. Biochem.* 100 (2007) 665–678.
- [43] Y. Cao, S. Liu, L. Zhang, J. Qin, J. Wang, K. Tang, Prediction of protein structural class with Rough Sets, *BMC Bioinform.* 7 (20) (2006).
- [44] Z.H. Zhang, Z.H. Wang, Z.R. Zhang, Y.X. Wang, A novel method for apoptosis protein subcellular localization prediction combining encoding based on grouped weight and support vector machine, *FEBS Lett.* 580 (2006) 6169–6174.
- [45] Q.B. Gao, Z.Z. Wang, C. Yan, Y.H. Du, Prediction of protein subcellular location using a combined feature of sequence, *FEBS Lett.* 579 (2005) 3444–3448.
- [46] K.C. Chou, H.B. Shen, Euk-mPLOC: a fusion classifier for large-scale eukaryotic protein subcellular location prediction by incorporating multiple sites, *J. Proteome Res.* 6 (2007) 1728–1734.
- [47] Q.B. Gao, Z.Z. Wang, Classification of G-protein coupled receptors at four levels, *Protein Eng. Des. Sel.* 19 (2006) 511–516.
- [48] J. Guo, Y. Lin, X. Liu, GNBSL: A new integrative system to predict the subcellular location for Gram-negative bacteria proteins, *Proteomics* 6 (2006) 5099–5105.
- [49] Q.S. Du, D.Q. Wei, K.C. Chou, Correlation of amino acids in proteins, *Peptides* 24 (2003) 1863–1869.
- [50] Q.S. Du, Z.Q. Jiang, W.Z. He, D.P. Li, K.C. Chou, Amino acid principal component analysis (AAPCA) and its applications in protein structural class prediction, *J. Biomol. Struct. Dyn.* 23 (2006) 635–640.
- [51] H.B. Shen, K.C. Chou, Predicting protein subnuclear location with optimized evidence-theoretic K-nearest classifier and pseudo amino acid composition, *Biochem. Biophys. Res. Commun.* 337 (2005) 752–756.
- [52] T. Zhang, Y. Ding, K.C. Chou, Prediction of protein subcellular location using hydrophobic patterns of amino acid sequence, *Computat. Biol. Chem.* 30 (2006) 367–371.
- [53] K.C. Chou, H.B. Shen, Signal-CF: a subsite-coupled and window-fusing approach for predicting signal peptides, *Biochem. Biophys. Res. Commun.* 357 (2007) 633–640.
- [54] H. Nakashima, K. Nishikawa, T. Ooi, The folding type of a protein is relevant to the amino acid composition, *J. Biochem.* 99 (1986) 152–162.
- [55] J. Cedano, P. Aloy, J.A. P'erez-Pons, E. Querol, Relation between amino acid composition and cellular location of proteins, *J. Mol. Biol.* 266 (1997) 594–600.
- [56] M. Spiess, Heads or tails - what determines the orientation of proteins in the membrane, *FEBS Lett.* 369 (1995) 76–79.