Available online at www.sciencedirect.com
ScienceDirect
Journal of Theoretical Biology 247 (2007) 259–265
Journal of Theoretical Biology
www.elsevier.com/locate/yjtbi

Prediction of membrane protein types from sequences and position-specific scoring matrices

Xian Pu[a], Jian Guo[b,*], Howard Leung[a], Yuanlie Lin[b]

[a]*Department of Computer Sciences, The City University of Hong Kong, Hong Kong*
[b]*Laboratory of Statistical Computation, Department of Mathematical Sciences, Tsinghua University, China*

Received 27 July 2006; received in revised form 22 December 2006; accepted 18 January 2007
Available online 30 January 2007

Abstract

Membrane protein plays an important role in some biochemical process such as signal transduction, transmembrane transport, etc. Membrane proteins are usually classified into five types [Chou, K.C., Elrod, D.W., 1999. Prediction of membrane protein types and subcellular locations. Proteins: Struct. Funct. Genet. 34, 137–153] or six types [Chou, K.C., Cai, Y.D., 2005. J. Chem. Inf. Modelling 45, 407–413]. Designing *in silico* methods to identify and classify membrane protein can help us understand the structure and function of unknown proteins. This paper introduces an integrative approach, IAMPC, to classify membrane proteins based on protein sequences and protein profiles. These modules extract the amino acid composition of the whole profiles, the amino acid composition of N-terminal and C-terminal profiles, the amino acid composition of profile segments and the dipeptide composition of the whole profiles. In the computational experiment, the overall accuracy of the proposed approach is comparable with the functional-domain-based method. In addition, the performance of the proposed approach is complementary to the functional-domain-based method for different membrane protein types.

Keywords: Membrane proteins type; Position-specific scoring matrix; Support vector machine

1. Introduction

Membrane proteins are those proteins which locate around cell membranes and execute crucial biological functions related to the cell membrane (such as signal transduction, ion transmembrane transport, etc.). According to Chou and Elrod (1999), membrane proteins can be generally classified as five main types: (1) type-I membrane protein; (2) type-II membrane protein; (3) multipass transmembrane proteins; (4) lipid chain-anchored membrane proteins; and (5) GPI-anchored membrane proteins. The above five types are defined by their modes associated with the lipid bilayer and by their biological functions.

Since the analysis of membrane protein by molecular biology experiment is time-consuming and labor-intensive, an automatic, efficient and effective method to identify (whether is the unknown protein a membrane protein) and classify (which type does the identified membrane protein belong to) membrane protein is desired. This paper focuses on the second problem.

The pioneering work about classification of membrane protein from the protein sequence was explored by Chou and Elrod (1999). After that, a number of methods were introduced to improve the classification performance. For example, the set of methods based on pseudo amino acid composition (PseAA) was originally proposed by Chou (2001) to extract information through a set of discrete correlation factors and various biochemical properties (Cai et al., 2004; Chou, 2000; Chou and Cai, 2003b, 2005b; Feng, 2001, 2002; Feng and Zhang, 2000; Gao et al., 2005; Pan et al., 2003; Shen and Chou, 2005; Shen et al., 2006; Sun and Huang, 2006; Wang et al., 2004, 2005; Wen et al., 2006; Xiao et al., 2005, 2006a; Zhou and Doctor, 2003). The set of methods (Cai and Chou, 2004, 2006; Cai et al., 2003; Chou and Cai, 2002, 2003a, 2004, 2005a, b; Zhang et al., 2006) based on protein functional annotation used the protein sample representation derived from a

*Corresponding author. Tel.: +86 10 62786651.
E-mail address: guojian1999@gmail.com (J. Guo).

0022-5193/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jtbi.2007.01.016

higher-level database, such as functional domain (FunD) database, Interpro database, gene ontology (GO) database, or their combinations. Recently, Liu et al. (2005) introduced a Fourier spectrum representation for membrane protein classification (Guo et al., 2006; Liu et al., 2005).

This paper introduced an integrated approach, IAMPC (Integrated Approach for Membrane Protein Classification), to predict membrane protein types from their sequences and profiles. IAMPC is composed of five modules, each of which extracts a particular feature from the protein sequences or profiles to train and test an individual support vector machine (SVM) classifier. The outputs from the five modules are combined by another SVM classifier for the final decision. On the same data set, the overall accuracy of IAMPC is comparable with the functional-domain-based method proposed by Cai and Chou (2006) and the performance of IAMPC is complementary to the functional-domain-based method for different membrane types.

2. Materials and methods

2.1. Data sets

The data set used to test the performance of IAMPC was created by Cai and Chou (2006). The data set was extracted from UniProt/Swiss-Prot Release 44 database by the following procedures (Cai and Chou, 2006; Chou and Elrod, 1999). (1) Selected those proteins with explicitly clear description of type I, type II, multipass, lipid-chain anchored, and GPI-anchored, i.e., excluded those with ambiguous annotations, such as "probable", "potential", and "by similarity". (2) Retained only one of the proteins with same name but different species. (3) Removed those proteins with the description of more than one type. (4) Removed those proteins with less than 50 residues. (5) Removed redundant proteins to promise that any pair of proteins in the data set has an identity less than 25%. In fact, this data set was a more strict version of the data set proposed by Chou and Cai (2005b). The final data set included 2763 sequences which were composed of 219 type-1 membrane proteins, 140 type-2 membrane proteins, 2137 multi-pass transmembrane proteins, 195 lipid chain-anchored membrane proteins, and 72 GPI-anchored membrane proteins.

2.2. Support vector machine

The support vector machine (SVM) is a widely used classification method based on the statistical learning theory. Here we briefly introduce its basic idea and interested readers can refer to Vapnik's (1995) book for more details.

Suppose we have a number of samples, each of which can be represented by a feature vector: $\boldsymbol{x}_i \in \Re^d$ with labels $y_i \in \{+1,-1\}$, $(i=1,\ldots,N)$. Here $+1$ and -1 indicate the two classes. Our goal is to best predict y_i according to the feature vector $\boldsymbol{x}_i$, i.e., to find an appropriate map from $\Re^d$ to $\{+1,-1\}$ so as to minimize the classification error. The SVM first maps the input vector $\boldsymbol{x} \in \Re^d$ into a high dimensional *Hilbert* space $\Phi(\boldsymbol{x}) \in \mathscr{H}$ to construct an optimized separating hyperplane. The hyperplane maximizes the margin, which is the largest distance between the hyperplane and the nearest data points of each class in the *Hilbert* space.

The decision function of SVM can be denoted as follows:

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} y_i \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b, \tag{1}$$

where $K(\boldsymbol{x}, \boldsymbol{x}_i)$ is called kernel function, representing the inner product in the *Hilbert* space, and the coefficient α_i is computed by solving the following convex quadratic programming problem:

$$\text{Maximize:} \quad \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

$$\text{Subject to:} \quad \sum_{i=1}^{N} \alpha_i y_i = 0, \quad i = 1,\ldots,N, \quad 0 \leqslant \alpha_i \leqslant C. \tag{2}$$

In Eq. (2), C is called regularization parameter which controls the trade off between: margin maximization and the classification errors and those x_i $(i=1,\ldots,N)$ corresponding to $\alpha_i > 0$ are called support vectors.

Some commonly used kernel functions are listed as follows:

$$K(\boldsymbol{x}, \boldsymbol{y}) = \langle \boldsymbol{x}, \boldsymbol{y} \rangle, \tag{3}$$

$$K(\boldsymbol{x}, \boldsymbol{y}) = (1 + \langle \boldsymbol{x}, \boldsymbol{y} \rangle)^d, \tag{4}$$

$$K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\gamma \|\boldsymbol{x} - \boldsymbol{y}\|^2), \tag{5}$$

where $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ represents the inner product of $\boldsymbol{x}$ and $\boldsymbol{y}$. Eq. (3) is called linear kernel, Eq. (4) is called polynomial kernel, and Eq. (5) is called radial basis kernel (RBF kernel).

In this paper, only the RBF kernel was used to train to test the SVM classifiers. The kernel parameter γ and the regularization parameter C are optimized individually and listed in Table 1.

2.3. Position-specific scoring matrix

Each protein sequence (called query sequence) in the proposed data set was used as a seed to search the homogenous sequences from the SWISSPROT 46.0 (Boeckmann et al., 2003) protein database using the PSI-BLAST program (Altschul et al., 1997) with parameters h and j being 0.001 and 3, respectively. These aligned sequences share some homogenous segments and belong to the same protein family. The aligned sequences were further converted into position-specific scoring matrices (PSSMs) to express their homogenous information. PSSM is a matrix with 20 rows and L columns, where L is the

Table 1
The optimized RBF kernel parameter γ and regularization parameter C

Parameter	Module 1	Module 2	Module 3	Module 4	Module 5	Fusion
γ	0.4	0.05	0.07	40	140	0.4
C	10	13	9	10	10	0.1

total number of the amino acids in the query sequence. Each column of a PSSM represents the log-likelihood of the residue substitutions at the corresponding position in the query sequence (Altschul et al., 1997). The (i,j)th entry of the matrix represents the chance of the amino acid in the jth position of the query sequence being mutated to amino acid type i during the evolution process. In this paper, the columns of each PSSM was normalized such that the sum of the squares of elements in each column added up to one. These PSSMs were used by module 1–module 4 of IAMPC.

For convenience, let us denote

$$\mathbf{P}^{(i)} = [\mathbf{p}_1^{(i)}, \mathbf{p}_2^{(i)}, \ldots, \mathbf{p}_{l^{(i)}}^{(i)}]$$

as the PSSM of the ith sequence, where

$$\mathbf{p}_j^{(i)} = [p_{j,1}^{(i)}, p_{j,2}^{(i)}, \ldots, p_{j,20}^{(i)}]^{\mathrm{T}}, \quad 1 \leqslant j \leqslant l^{(i)},$$

and $l^{(i)}$ is the total number of amino acids of the ith sequence.

2.4. Modules of IAMPC

IAMPC is composed of five modules, each of which extract a particular feature from PSSM or from protein sequence. The details of these modules are introduced as follows.

2.4.1. Module 1

This module extracts the amino acid composition from the entire PSSM. Denote

$$\mathbf{x}^{(i)} = [x_1^{(i)}, x_2^{(i)}, \ldots, x_{20}^{(i)}],$$

as a 20-dimensional feature vector extracted from the ith protein by this module. $x_k^{(i)}$ $(1 \leqslant k \leqslant 20)$ is the composition of the kth amino acid in the ith PSSM. It is calculated as follows:

$$x_k^{(i)} = \frac{1}{l^{(i)}} \sum_{j=1}^{l^{(i)}} p_{j,k}^{(i)}, \tag{6}$$

where $l^{(i)}$ is the number of amino acids in the ith protein.

2.4.2. Module 2

Module 2 uses the similar extraction approach as module 1 but it only computes the amino acid composition in N-terminus and C-terminus of the PSSM. Specifically, denote the amino acid composition in the N-terminus and the C-terminus of the PSSM of the ith protein as

$$\mathbf{y}^{(i)} = [y_1^{(i)}, y_2^{(i)}, \ldots, y_{20}^{(i)}],$$

and

$$\mathbf{z}^{(i)} = [z_1^{(i)}, z_2^{(i)}, \ldots, z_{20}^{(i)}],$$

respectively. Then $y_k^{(i)}$ $(1 \leqslant k \leqslant 20)$ is calculated as follows:

$$y_k^{(i)} = \frac{1}{L_N} \sum_{j=1}^{L_N} p_{j,k}^{(i)}, \tag{7}$$

and $z_k^{(i)}$ $(1 \leqslant k \leqslant 20)$ is calculated as follows:

$$z_k^{(i)} = \frac{1}{L_C} \sum_{j=l^{(i)}-L_C+1}^{l^{(i)}} p_{j,k}^{(i)}, \tag{8}$$

where L_N and L_C are the numbers of amino acids in the N-terminus and C-terminus of the ith protein. Then the feature vector extracted by this module is defined as

$$\mathbf{x} \oplus \mathbf{y} \oplus \mathbf{z} = [x_1^{(i)}, \ldots, x_{20}^{(i)}, y_1^{(i)}, \ldots, y_{20}^{(i)}, z_1^{(i)}, \ldots, z_{20}^{(i)}], \tag{9}$$

where $\oplus$ is the operator of the concatenation. In this paper, the length of the N-terminus L_N equals to 50 and the length of the C-terminus L_C equals to 40.

2.4.3. Module 3

This module assumes that different segments of PSSM have different amino acid distributions and provide complementary information for the prediction. The segment of a PSSM is defined as a matrix comprising of a number of consecutive columns of the PSSM. In this paper, the PSSM is divided into n_s segments with almost equal number of columns. For the ith protein with length (number of amino acid) $l^{(i)}$, denote l_s as the integral value of $l^{(i)}$ divided by n_s. Then the amino acid composition in the gth segment $(1 \leqslant g \leqslant n_s)$ is represented as

$$\mathbf{v}_g^{(i)} = [v_{g,1}^{(i)}, v_{g,2}^{(i)}, \ldots, v_{g,20}^{(i)}].$$

If $1 \leqslant g \leqslant n_s - 1$, the amino acid composition in the gth segment of the ith protein is calculated as follows:

$$v_{g,k}^{(i)} = \frac{1}{l_s} \sum_{j=(g-1)*l_s+1}^{g*l_s} p_{j,k}, \tag{10}$$

where $1 \leqslant k \leqslant 20$. If $g = n_s$, the amino acid composition in the gth segment of the ith protein is calculated as follows:

$$v_{g,k}^{(i)} = \frac{1}{l^{(i)} - (n_s - 1) * l_s} \sum_{j=(n_s-1)*l_s+1}^{l^{(i)}} p_{j,k}. \tag{11}$$

The last segment has more columns than the first $n_s - 1$ segments when $l^{(i)}$ is not divisible by n_s. Then feature vector of the ith protein $\mathbf{v}^{(i)}$ is the concatenation of the amino acid composition of all segments.

$$\mathbf{v}^{(i)} = \mathbf{v}_1^{(i)} \oplus \mathbf{v}_2^{(i)} \oplus \cdots \oplus \mathbf{v}_{n_s}^{(i)}, \tag{12}$$

where $\oplus$ is the operator of the concatenation. In this paper, the parameter n_s equals to 3.

2.4.4. Module 4

This module extracts the dipeptide composition in the PSSMs. Since there are 20×20 combinations of the dipeptides, the feature vector of this module is defined as a 400-dimensional vector (Chou and Elrod, 1999; Liu and Chou, 1999):

$$\mathbf{x}^{(i)} = [x_{1,1}^{(i)}, \ldots, x_{1,20}^{(i)}, x_{2,1}^{(i)}, \ldots, x_{2,20}^{(i)}, \ldots, x_{20,1}^{(i)}, \ldots, x_{20,20}^{(i)}],$$

then $x_{u,v}^{(i)}$ $(1 \leqslant u, v \leqslant 20)$ is calculated as follows:

$$x_{u,v}^{(i)} = \frac{1}{l^{(i)} - 1} \sum_{j=1}^{l^{(i)}-1} p_{j,u}^{(i)} \cdot \times p_{j,v}^{(i)}. \tag{13}$$

2.4.5. Module 5

This module extract features directly from the protein sequences rather than from the PSSMs like the previous four modules. The residue-couple composition of the protein sequence is extracted (Guo et al., 2005). The residue-couple is defined as a pair of amino acids in the proteins sequence consecutively or segregated by arbitrary amino acids. The consecutive amino acid pair is called rank-0 residue-couple, the amino acid pair segregated by 1 arbitrary amino acid is called rank-1 residue-couple, and by analogy, the amino acid pair segregated by m arbitrary amino acid is called rank-m residue-couple, where m is any non-negative integer. The rank-m residue-couple composition is a 400-dimensional vector with each entry representing the frequency of the occurrence of a particular rank-m residue-couple.[1]

Denote the rank-m residue-couple composition $(0 \leqslant m \leqslant n_{rc})$ of the ith protein as follows:

$$\mathbf{x}_m^{(i)} = [x_{m,1,1}^{(i)}, \ldots, x_{m,s,t}^{(i)}, \ldots, x_{m,20,20}^{(i)}],$$

where $1 \leqslant s, t \leqslant 20$. The entry $x_{m,s,t}^{(i)}$ represents the frequency of occurrence of the rank-m residue-couple with the front amino acid being type s and the back amino acid being type t. Then $x_{m,s,t}^{(i)}$ is calculated as follows:

$$x_{m,s,t}^{(i)} = \frac{1}{l^{(i)} - m - 1} \sum_{j=1}^{l^{(i)}-m-1} I_{s,t}(j, j+m+1), \tag{14}$$

where $l^{(i)}$ is the length of the ith protein and $I_{s,t}(j, j+m+1) = 1$ if the type of the jth amino acid (j is counted from the N-terminus of the sequence) is s and the type of the $(j+m+1)$th amino acid is t and $I_{s,t}(j, j+m+1) = 0$ otherwise.

The feature vector extracted by this module is denoted as follows:

$$\mathbf{x}_m^{(i)} = \mathbf{x}_0^{(i)} \oplus \mathbf{x}_1^{(i)} \oplus \cdots \oplus \mathbf{x}_{n_{rc}}^{(i)}, \tag{15}$$

where $\oplus$ is the operator of the concatenation. In this paper, $n_{rc} = 5$.

[1]There are 20 types of amino acid, so the number of possible types of residue-couple is $20 \times 20 = 400$.

2.5. SVM fusion

The fusion step attempts to integrate different information extracted by each module. There are different ways to implement the fusion process. For a multi-classification problem, the simplest way is voting, which classifies the sample to the class with the majority vote. In this paper, the SVM fusion method was used to implement the fusion process.

The SVM fusion uses the SVM as a classifier to reclassify the outputs from all modules. Specifically, the ith protein is predicted by module 1–module 5 and the output of the mth module $(1 \leqslant m \leqslant 5)$ is represented as a h-dimensional binary vector (h is the number of class).

$$\mathbf{w}_m^{(i)} = [w_{1,m}^{(i)}, w_{2,m}^{(i)}, \ldots, w_{h,m}^{(i)}].$$

If the ith protein is predicted to class q $(1 \leqslant q \leqslant h)$ by the mth module, the entry $w_{q,m}^{(i)}$ is 1 and other $h-1$ entries are 0. Then the feature vector is defined as

$$\mathbf{w}^{(i)} = \mathbf{w}_1^{(i)} \oplus \mathbf{w}_2^{(i)} \oplus \cdots \oplus \mathbf{w}_5^{(i)}, \tag{16}$$

where $\oplus$ represents the simple concatenation of two vectors. The feature vectors are used to train the SVM classifier for the fusion. The structure of IAMPC is illustrated in Fig. 1.

2.6. Assessment of performance

The leave-one-out cross validation (jackknife test) and k-fold cross validation are widely used to evaluate the performance of a method on a data set. The former is more rigorous and objective as elucidated in a comprehensive review (Chou and Zhang, 1995) and a series of follow-up papers (Feng, 2001, 2002; Guo et al., 2006; Liu et al., 2005; Luo et al., 2002; Shen and Chou, 2005; Shen et al., 2005; Sun and Huang, 2006; Wang et al., 2005; Xiao et al., 2005, 2006b; Zhou, 1998; Zhou and Assa-Munt, 2001; Zhou and Cai, 2006; Zhou and Doctor, 2003). However, the jackknife test is time-consuming so we used the 5-fold cross validation instead. In a 5-fold cross validation trial, the original data set was randomly divided into 5 subsets. Each

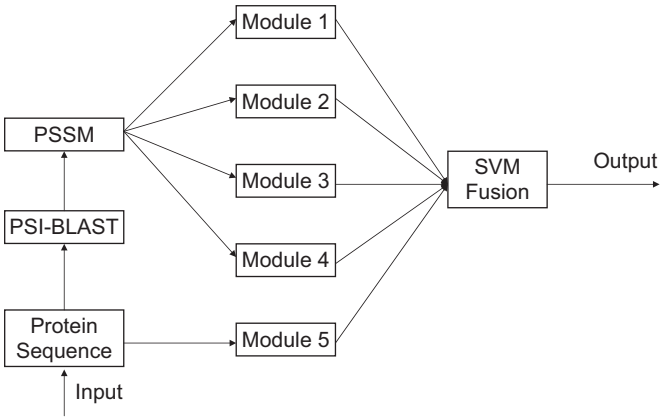

Fig. 1. The structure of the proposed prediction system.

subset was singled out in turn as a testing set, and the remaining ones were merged to train the classifier. The process was iterated five times until every subset has been used for testing and the prediction results from all iterations were averaged. The overall accuracy (OA), the accuracy for each class (Acc), and the Matthew's correlation coefficient (MCC) (Matthews, 1975) were used to assess the prediction result. MCC allows us to overcome the shortcoming of accuracy (Acc) on unbalanced data. For example, if the number of the positive samples are much larger than that of the negative samples, a classifier is easy to predict all samples as positive. Significantly it is not a good classifier because it predicts all negative samples incorrectly. In this case, the accuracy and MCC of the positive class are 100% and 0, respectively. Therefore, MCC is a better measure for unbalanced data classification.

Denote $\boldsymbol{M} \in \Re^{C\times C}$ as the confusion matrix of the prediction result, where h is the number of classes. Then $\boldsymbol{M}_{i,j}$ $(1\leqslant i,j\leqslant h)$ represents the number of proteins that actually belong to class i but are predicted as class j. We further denote

$$p_c = \boldsymbol{M}_{c,c}, \quad q_c = \sum_{i=1,i\neq c}^{h}\sum_{j=1,j\neq c}^{h}\boldsymbol{M}_{i,j},$$

$$r_c = \sum_{i=1,i\neq c}^{h}\boldsymbol{M}_{i,c}, \quad s_c = \sum_{j=1,j\neq c}^{h}\boldsymbol{M}_{c,j}, \tag{17}$$

where c $(1\leqslant c\leqslant h)$ is the index of a particular class. For class c, p_c is the number of true positive samples, q_c is the number of true negative samples, r_c is the number of false positive samples, and s_c is the number of false negative samples. Based on the notations above, the OA, the accuracy of class c (Acc_c), and the Matthew's correlation coefficient of class c (MCC_c) are

$$\mathrm{OA} = \frac{\sum_{c=1}^{h}\boldsymbol{M}_{c,c}}{\sum_{i=1}^{h}\sum_{j=1}^{h}\boldsymbol{M}_{i,j}}, \tag{18}$$

$$\mathrm{Acc}_c = \frac{\boldsymbol{M}_{c,c}}{\sum_{j=1}^{h}\boldsymbol{M}_{c,j}}, \tag{19}$$

$$\mathrm{MCC}_c = \frac{p_c q_c - r_c s_c}{\sqrt{(p_c+s_c)(p_c+r_c)(q_c+s_c)(q_c+r_c)}}. \tag{20}$$

3. Result and discussion

3.1. Result of every module

The prediction result of the five individual modules and the result of the SVM fusion were listed in Table 2. Module 1 only uses the amino acid composition of the profiles (the least information among modules 1–module 4) so it should be regarded as the baseline of the prediction. Module 2 used the amino acid composition of the N-terminus of the profiles, the C-terminus of the profiles and the whole profiles. The overall accuracy of module 2 reached 91.3%, which was 3.1% higher than that of module 1. The most significant improvement came from the GPI-anchored membrane proteins, for which the accuracy improved by 26.4%. The improvement was consistent with the biological knowledge. For example, because the C-terminus of the GPI-anchor membrane protein binds to the cell membrane by GPI-anchor, the amino acid composition of the C-terminus of the protein is very important for the classification. The improvement for the multipass transmembrane protein was minor because multipass transmembrane structure occurs in the middle part of the protein. The improvement implies that N-terminus and C-terminus of the protein can provide complementary information to improve the prediction. Module 3 used the amino acid composition of all segments of the profiles. The overall accuracy of module 3 reached 90.5%, which is 2.3% higher than that of module 1. Module 4 extracted the dipeptide composition of the profiles, which can be regarded as an extension of module 1. The overall accuracy of module 4 only improved 1.2% compared with module 1, while the accuracies of different membrane protein types of module 1 were similar to those of module 2. Unlike the

Table 2

Membrane type	Module 1		Module 2		Module 3		Module 4		Module 5		Fusion	
	Acc(%)	MCC	Acc(%)	MCC	Acc(%)	MCC	Acc(%)	MCC	Acc(%)	MCC	Acc(%)	MCC
Type-I	70.7	0.68	78.2	0.78	77.3	0.75	73.8	0.70	65.3	0.65	83.1	0.82
Type-II	47.0	0.50	57.0	0.63	58.9	0.60	54.3	0.58	23.2	0.37	55.6	0.63
Multipass	96.3	0.76	97.0	0.80	96.7	0.81	97.3	0.77	98.6	0.66	98.0	0.83
Lipid	68.6	0.68	78.9	0.79	75.8	0.78	68.2	0.72	50.7	0.63	83.9	0.83
GPI	27.8	0.38	54.2	0.61	37.5	0.45	20.8	0.35	11.1	0.29	40.3	0.58
Overall	88.2	–	91.3	–	90.5	–	89.4	–	86.6	–	92.3	–

Type-I: Type-I membrane protein; Type-II: Type-II membrane protein; multipass: multipass transmembrane protein; lipid: lipid chain-anchored membrane protein; GPI: GPI-anchored membrane protein.

former four modules, module 5 extracted the features from the protein sequences rather than from the protein profiles. It was not surprising that the overall accuracy of module 5 (86.6%) was lower than other profile-based modules. Since the protein profile represents the amino acid distribution of a family of aligned homologous proteins, it includes more worthy information for membrane protein prediction.

3.2. Result of SVM fusion

We fuse all modules because we assume that the feature vectors extracted by different modules provide complementary information that has potential to further improve the prediction performance. From Table 2 we noticed that these modules have different advantages for different membrane protein types. For example, module 2 performed best for type-I transmembrane protein, lipid-chain anchored membrane protein, and GPI-anchored membrane protein, while module 3 prefered on type-II transmembrane protein.

The result of the SVM fusion were listed in the last column of Tables 2 and 3. The overall accuracy of the SVM fusion reached 92.3%, which was slightly higher than that of module 2 (the module with the highest overall accuracy). The result of the SVM fusion was also compared with that of the method introduced by Cai and Chou (2006), in which the features are extracted from functional domain composition and pseudo-amino acid composition (Table 3). The result of Cai and Chou's method was obtained by leave-one-out cross validation test (jackknife) and the result of IAMPC was obtained by five-fold cross validation test(5FCV) to save computational time. The overall accuracy of IAMPC reached 92.3% and that of Cai and Chou's method reached 91.3%. Because of the different ways of cross validation (5FCV vs. jackknife), the two results can be regarded as comparable. In addition, the prediction accuracies of IAMPC and those of Cai and Chou's method were complemental for different types of membrane proteins. From Table 3, IAMPC achieved higher accuracy for lipid chain-anchored membrane proteins and Cai and Chou's method achieved higher accuracy for type-II membrane proteins and GPI-anchored membrane proteins. The two methods achieved comparable prediction accuracy for type-II membrane proteins and multipass transmembrane membrane proteins. Therefore, our future study is attempting to fuse IAMPC with Cai and Chou's method to further improve the prediction accuracy.

Table 3

Membrane type	Functional-domain-based method Acc(%)	IAMPC (SVM fusion)	
		Acc(%)	MCC
Type-I	83.6	83.1	0.82
Type-II	71.4	55.6	0.63
Multipass	97.6	98.0	0.83
Lipid	61.0	83.9	0.83
GPI	50.0	40.3	0.58
Overall	91.3	92.3	–

Type-I: Type-I membrane protein; Type-II: Type-II membrane protein; multipass: multipass transmembrane protein; lipid: lipid chain-anchored membrane protein; GPI: GPI-anchored membrane protein.

4. Conclusion

This paper introduces an integrative method (IAMPC) to predict the types of membrane proteins from their sequence and position-specific scoring matrix. IAMPC is composed of five individual modules, each of which can implement the prediction independently. Module 1–module 4 extract different features from the position-specific scoring matrices and the last module extracts residue-couple composition from the primary sequences of proteins. The SVM fusion is used to fuse the output of every module for the final prediction. The overall accuracy of IAMPC reaches 92.3%, which is comparable with that of Cai and Chou's method. In addition, IAMPC and Cai and Chou's method have different advantages for various types of membrane proteins. Therefore, IAMPC plays a complementary role to Cai and Chou's method.

Acknowledgements

This work is supported by the funds of Human Liver Proteome Project (2004BA711A21), The National Nature Science Foundation of China (10371063), and a grant from City University of Hong Kong (9360092).

References

Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J., 1997. Gapped blast and psi-blast: a new generation of protein database search programs. Nucleic Acids Res. 25, 3389–3402.

Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M.C., Estreicher, A., Gasteiger, E., Martin, M.J., Michoud, K., O'Donovan, C., Phan, I., Pilbout, S., Schneider, M., 2003. The swiss-prot protein knowledgebase and its supplement TrEMBL in 2003. Nucleic Acids Res. 31, 365–367.

Cai, Y.D., Chou, K.C., 2004. Predicting subcellular localization of proteins in a hybridization space. Bioinformatics 20, 1151–1156.

Cai, Y.D., Chou, K.C., 2006. Predicting membrane protein type by functional domain composition and pseudo-amino acid composition. J. Theor. Biol. 238, 395–400.

Cai, Y.D., Ricardo, P.W., Jen, C.H., Chou, K.C., 2004. Application of SVM to predict membrane protein types. J. Theor. Biol. 226, 373–376.

Cai, Y.D., Zhou, G.P., Chou, K.C., 2003. Support vector machines for predicting membrane protein types by using functional domain composition. Biophys. J. 84, 3257–3263.

Chou, K.C., 2000. Prediction of protein subcellular locations by incorporating quasi-sequence-order effect. Biochem. Biophys. Res. Commun. 278, 477–483.

Chou, K.C., 2001. Prediction of protein cellular attributes using pseudo amino acid composition. Proteins: Struct. Funct. Genet. 43, 246–255.

Chou, K.C., Cai, Y.D., 2002. Using functional domain composition and support vector machines for prediction of protein subcellular location. J. Biol. Chem. 277, 45765–45769.

Chou, K.C., Cai, Y.D., 2003a. A new hybrid approach to predict subcellular localization of proteins by incorporating gene ontology. Biochem. Biophys. Res. Commun. 311, 743–747.

Chou, K.C., Cai, Y.D., 2003b. Prediction and classification of protein subcellular location: sequence-order effect and pseudo amino acid composition. J. Cell. Biochem. 90, 1250–1260.

Chou, K.C., Cai, Y.D., 2004. Prediction of protein subcellular locations by GO-FunD-PseAA predicor. Biochem. Biophys. Res. Commun. 320, 1236–1239.

Chou, K.C., Cai, Y.D., 2005a. Predicting protein localization in budding yeast. Bioinformatics 21, 944–950.

Chou, K.C., Cai, Y.D., 2005b. Using GO-PseAA predictor to identify membrane proteins and their types. Biochem. Biophys. Res. Commun. 327, 845–847.

Chou, K.C., Elrod, D.W., 1999. Prediction of membrane protein types and subcellular locations. Proteins: Struct. Funct. Genet. 34, 137–153.

Chou, K.C., Zhang, C.T., 1995. Review: prediction of protein structural classes. Crit. Rev. Biochem. Mol. Biol. 30, 275–349.

Feng, Z., 2001. Prediction of the subcellular location of prokaryotic proteins based on a new representation of the amino acid composition. Biopolymers 58, 491–499.

Feng, Z., 2002. An overview on predicting the subcellular location of a protein. In Silico Biol. 2, 291–303.

Feng, Z.P., Zhang, C.T., 2000. Prediction of membrane protein types based on the hydrophobic index of amino acids. J. Protein Chem. 19, 269–275.

Gao, Y., Shao, S.H., Xiao, X., Ding, Y.S., Huang, Y.S., Huang, Z.D., Chou, K.C., 2005. Using pseudo amino acid composition to predict protein subcellular location: approached with Lyapunov index, Bessel function, and Chebyshev filter. Amino Acids 28, 373–376.

Guo, J., Lin, Y.L., Sun, Z.R., 2005. A novel method for protein subcellular localization: combining residue-couple model and SVM. In: Proceedings of APBC 2005, pp. 117–129.

Guo, Y.Z., Li, M., Lu, M., Wen, Z., Wang, K., Li, G., Wu, J., 2006. Classifying G protein-coupled receptors and nuclear receptors based on protein power spectrum from fast fourier transform, Amino Acids (doi: 10.1007/S00726-006-0332-z).

Liu, H., Wang, M., Chou, K.C., 2005. Low-frequency fourier spectrum for predicting membrane protein types. Biochem. Biophys. Res. Commun. 336, 737–739.

Liu, W., Chou, K.C., 1999. Protein secondary structural content prediction. Protein Eng. 12, 1041–1050.

Luo, R.Y., Feng, Z.P., Liu, J.K., 2002. Prediction of protein structural class by amino acid and polypeptide composition. Eur. J. Biochem. 269, 4219–4225.

Matthews, B.W., 1975. Comparison of predicted and observed secondary structure of T4 phage lysozyme. Biochim. Biophys. Acta 405, 442–451.

Pan, Y.X., Zhang, Z.Z., Guo, Z.M., Feng, G.Y., Huang, Z.D., He, L., 2003. Application of pseudo amino acid composition for predicting protein subcellular location: stochastic signal processing approach. J. Protein Chem. 22, 395–402.

Shen, H.B., Chou, K.C., 2005. Predicting protein subnuclear location with optimized evidence-theoretic k-nearest classifier and pseudo amino acid composition. Biochem. Biophys. Res. Comm. 337, 752–756.

Shen, H.B., Yang, J., Liu, X.J., Chou, K.C., 2005. Using supervised fuzzy clustering to predict protein structural classes. Biochem. Biophys. Res. Commun. 334, 577–581.

Shen, H.B., Yang, J., Chou, K.C., 2006. Fuzzy KNN for predicting membrane protein types from pseudo-amino acid composition. J. Theor. Biol. 240, 9–13.

Sun, X.D., Huang, R.B., 2006. Prediction of protein structural classes using support vector machines. Amino Acids (doi: 10.1007/S00726-005-0239-0).

Vapnik, V., 1995. The Nature of Statistical Learning Theory. Springer, New York.

Wang, M., Yang, J., Liu, G.P., Xu, Z.J., Chou, K.C., 2004. Weighted support vector machines for predicting membrane protein types based on pseudo amino acid composition. Protein Eng. Des. Sele. 17, 509–516.

Wang, M., Yang, J., Xu, Z.J., Chou, K.C., 2005. SLLE for predicting membrane protein types. J. Theor. Biol. 232, 7–15.

Wen, Z., Li, M., Li, Y., Guo, Y., Wang, K., 2006. Delaunay triangulation with partial least squares projection to latent structures: a model for g-protein coupled receptors classification and fast structure recognition. Amino Acids, 117–129.

Xiao, X., Shao, S., Ding, Y., Huang, Z., Huang, Y., Chou, K.C., 2005. Using complexity measure factor to predict protein subcellular location. Amino Acids 28, 57–61.

Xiao, X., Shao, S.H., Ding, Y.S., Huang, Z.D., Chou, K.C., 2006a. Using cellular automata images and pseudo amino acid composition to predict protein sub-cellular location. Amino Acids 30, 49–54.

Xiao, X., Shao, S.H., Huang, Z.D., Chou, K.C., 2006b. Using pseudo amino acid composition to predict protein structural classes: approached with complexity measure factor. J. Comput. Chem. 27, 478–482.

Zhang, S.W., Pan, Q., Zhang, H.C., Shao, Z.C., Shi, J.Y., 2006. Prediction of protein homo-oligomer types by pseudo amino acid composition: approached with an improved feature extraction and Naive Bayes Feature Fusion. Amino Acids 30, 461–468.

Zhou, G.P., 1998. An intriguing controversy over protein structural class prediction. J. Protein Chem. 17, 729–738.

Zhou, G.P., Assa-Munt, N., 2001. Some insights into protein structural class prediction. Proteins: Struct. Func. Genet. 44, 57–59.

Zhou, G.P., Cai, Y.D., 2006. Predicting protease types by hybridizing gene ontology and pseudo amino acid composition. Proteins: Struct. Funct. Bioinformatics 63, 681–684.

Zhou, G.P., Doctor, K., 2003. Subcellular location prediction of apoptosis proteins. Proteins: Struct. Funct. Genet. 50, 44–48.