Protein Engineering, Design & Selection vol. 17 no. 6 pp. 509–516, 2004
Published online August 16, 2004 doi:10.1093/protein/gzh061

Weighted-support vector machines for predicting membrane protein types based on pseudo-amino acid composition

Meng Wang[1,2], Jie Yang[1], Guo-Ping Liu[1], Zhi-Jie Xu[1] and Kuo-Chen Chou[1,2,3,4,5,6]

[1]Institute of Image Processing and Pattern Recognition, Shanghai Jiaotong University, Shanghai 200030, [3]Bioinformatics Research Centre, Donghau University, Shanghai 200050, [4]Department of Biomedical Engineering, Shanghai Jiaotong University, Shanghai 200030, [5]Tianjin Institute of Bioinformatics and Drug Discovery, Tianjin, China and [6]Gordon Life Science Institute, San Diego, CA 92130, USA

[2]To whom correspondence should be addressed.
E-mail: mengking@sjtu.edu.cn; kchou@san.rr.com

Membrane proteins are generally classified into the following five types: (1) type I membrane proteins, (2) type II membrane proteins, (3) multipass transmembrane proteins, (4) lipid chain-anchored membrane proteins and (5) GPI-anchored membrane proteins. Prediction of membrane protein types has become one of the growing hot topics in bioinformatics. Currently, we are facing two critical challenges in this area: first, how to take into account the extremely complicated sequence-order effects, and second, how to deal with the highly uneven sizes of the subsets in a training dataset. In this paper, stimulated by the concept of using the pseudo-amino acid composition to incorporate the sequence-order effects, the spectral analysis technique is introduced to represent the statistical sample of a protein. Based on such a framework, the weighted support vector machine (SVM) algorithm is applied. The new approach has remarkable power in dealing with the bias caused by the situation when one subset in the training dataset contains many more samples than the other. The new method is particularly useful when our focus is aimed at proteins belonging to small subsets. The results obtained by the self-consistency test, jackknife test and independent dataset test are encouraging, indicating that the current approach may serve as a powerful complementary tool to other existing methods for predicting the types of membrane proteins.

Keywords: Chou's invariance theorem/covariant discriminant algorithm/pseudo-amino acid composition/spectral analysis/weighted υ-SVM

Introduction

Owing to the development of high-throughput sequencing technology, the data in various biology databases have been increasing at an unprecedented rate, which challenges the speed and ability of biologists and computational scientists to analyze these data. Because most of the specific functions of a cell are carried out by the membrane proteins (see e.g. Alberts *et al.*, 1994; Lodish *et al.*, 1995), prediction of membrane protein types has become a vitally important subject in molecular and cellular biology. Although the type of a membrane protein can be determined by various biochemical experiments, it is both time consuming and costly if the determination is based on an experimental approach alone. In view of this, it is highly desirable to develop an automated method to expedite the speed of determination.

Membrane proteins are generally classified into the following five types: (1) type I membrane proteins, (2) type II membrane proteins, (3) multipass transmembrane proteins, (4) lipid chain-anchored membrane proteins and (5) GPI-anchored membrane proteins (Figure 1). The function of a membrane protein is closely related to the type to which it belongs. Therefore, a fast and efficient method for predicting the type of membrane protein will significantly speed up the process of function determination for newly found membrane proteins. In a pioneering study, based on the amino acid composition, the covariant discriminant algorithm was introduced by Chou and Elrod (1999) to predict the types of membrane proteins. By definition, the conventional amino acid composition is a vector of 20 components, each representing the frequency of occurrence of one of the 20 native amino acids (Nakashima *et al.*, 1986; Chou, 1995; Zhou, 1998). Accordingly, using the amino acid composition to represent a sample of protein will miss all the sequence-order and sequence-length effects. In order to cope with this problem, a new concept, the so-called 'pseudo-amino acid composition', was proposed by Chou (2001). The pseudo-amino acid composition can bear the main features of amino acid composition, but meanwhile it can also incorporate some sequence order effects. Stimulated by its success in improving prediction quality, here we introduce a different approach to formulate the pseudo-amino acid composition.

Meanwhile, SVM (support vector machine) has recently been widely used in bioinformatics. However, its performance is greatly limited by the uneven sizes of the subsets in the training dataset. The classification results based on SVM are undesirably biased toward the class with more samples in the corresponding subset. In other words, the larger the size of a subset, the smaller is the classification error; whereas the smaller the size of a subset, the larger is the classification error. In the dataset constructed by Chou and Elrod (1999), the training subsets are uneven. To solve this problem, we use the weighted support vector machine (υ-SVM) to cope with this problem.

This paper is devoted to combining the concept of pseudo-amino acid composition and υ-SVM to develop a new predictor for predicting the membrane protein types.

Pseudo amino acid composition and discrete Fourier transform

A protein sequence can be represented as a series of amino acids by their single-character codes A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W and Y, formulated as

$$R_1R_2R_3R_4R_5R_6R_7R_8 \ldots R_L \qquad (1)$$

where the component R_1 is the first residue, R_2 the second residue and so forth.

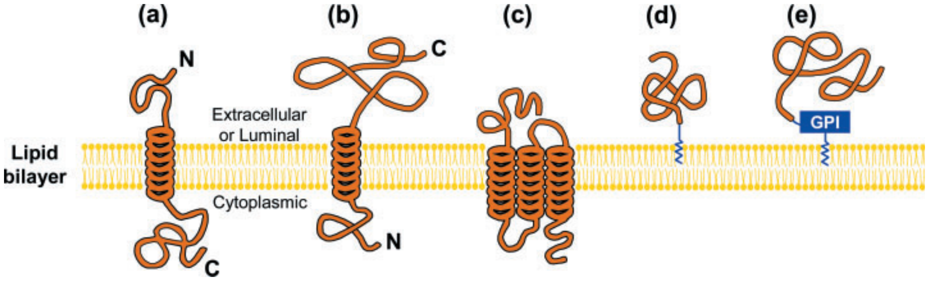

Fig. 1. Schematic drawing showing the following five types of membrane proteins: (**a**) type I transmembrane, (**b**) type II transmembrane, (**c**) multipass transmembrane, (**d**) lipid-chain anchored membrane and (**e**) GPI-anchored membrane. As shown, although both type I and type II membrane proteins are single-pass transmembrane, type I has a cytoplasmic C-terminus and an extracellular or luminal N-terminus for plasma membrane or organelle membrane, respectively, while the arrangement of N- and C-termini in type II membrane proteins is just the reverse. No such distinction was drawn between the extracellular (or luminal) and cytoplasmic sides for the other three types in the current classification scheme. Reproduced from Chou (2002), with permission.

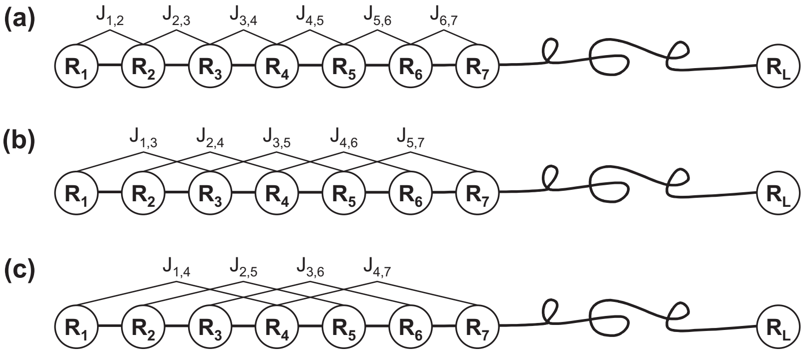

Fig. 2. A schematic drawing to show (**a**) the first-rank, (**b**) the second-rank and (**c**) the third-rank sequence-order correlation mode along a protein sequence. Panel (a) reflects the correlation mode between all the most contiguous residues, panel (b) that between all the second most contiguous residues and panel (c) that between all the third most contiguous residues. Adapted from Chou (2001), with permission.

The conventional amino acid composition is defined as 20 discrete numbers each representing the frequency of occurrence of one of the 20 native amino acids (Nakashima *et al.*, 1986; Chou and Zhang, 1994; Chou, 1995; Zhou, 1998). Compared with the conventional amino acid composition, the pseudo-amino acid composition is a vector with 20+λ discrete components (Chou, 2001) and hence may be viewed as a point in a (20+λ)-D space, as given by

$$\mathbf{P} = \begin{bmatrix} p_1 \\ \vdots \\ p_{20} \\ p_{20+1} \\ \vdots \\ p_{20+\lambda} \end{bmatrix}, \qquad (2)$$

where the first 20 components are the same as in the conventional amino acid composition, while the additional components $p_{20+1}, \ldots p_{20+\lambda}$ are related to λ different ranks (Figure 2) of sequence-order correlation factors as formulated by the following equation (Chou, 2001):

$$\begin{cases} \tau_1 = \frac{1}{L-1}\sum_{i=1}^{L-1} J_{i,i+1} \\ \tau_2 = \frac{1}{L-2}\sum_{i=1}^{L-2} J_{i,i+2} \\ \tau_3 = \frac{1}{L-3}\sum_{i=1}^{L-3} J_{i,i+3}, \quad (\lambda < l) \\ \cdots\cdots\cdots \\ \tau_\lambda = \frac{1}{L-\lambda}\sum_{i=1}^{L-\lambda} J_{i,i+\lambda} \end{cases} \qquad (3)$$

In the above equation, *L* denotes the length of the protein and τ_i is called the *i*th rank of coupling factor that harbours the *i*th sequence-order correlation factor. An illustration to show how these factors are associated with the sequence order effect is given in Figure 2. The coupling factor $J_{i,j}$ in Equation 3 is defined as a function of the amino acids R_i and R_j, such as the physicochemical distance (Schneider and Wrede, 1994; Chou, 2000) from R_i to R_j, or some combinations of several biochemical quantities related to R_i and R_j (Chou, 2001, 2002). Hence τ_i can be rewritten as:

$$\tau_m = \frac{1}{L-m}\sum_{j=1}^{L-m} f(R_i)f(R_j). \qquad (4)$$

As can be seen from Figure 2, the sequence-order effect of a protein can be, to some extent, reflected through a set of discrete numbers $\tau_1, \tau_2, \tau_3, \ldots, \tau_m$, as defined by Equation 4. Such information is very useful in the analysis of proteins with a set of discrete numbers. Accordingly, the first 20 components of Equation 1 reflect the effect of the amino acid composition, whereas the components from $20+1$ to $20+\lambda$ reflect some sequence-order effects. A set of $20+\lambda$ components as formulated by Equations (1) and (2) is called the pseudo-amino acid composition for protein P. Such a name is used because it still has the main features of amino acid composition, but on the other hand it contains the information beyond the conventional amino acid composition. The pseudo-amino acid composition thus defined has the following advantages: compared with the 210-D pair-coupled amino acid composition (Chou, 1999) and the 400-D first-order coupled amino acid composition (Liu and

Chou, 1999) that contain the sequence order effect only for a very short range (i.e. within two adjacent amino acid residues along a chain), the pseudo-amino acid composition incorporates much more sequence effects, i.e. those not only for the short range but also for the medium and long range, as indicated by a series of sequence-coupling factors with different tiers of correlation (see Figure 2 and Equations 2–4). Therefore, the prediction quality can be significantly improved by using the pseudo-amino acid composition to represent the sample of a protein. For detailed formulation and application of the pseudo-amino acid composition, readers are referred to two recent papers (Chou, 2001; Chou and Cai, 2003).

Below we shall use the technique of spectral analysis to formulate the pseudo-amino acid composition. As shown in Equation 1, the sequence of a protein is composed of a series of characters, which is hard for a computer to process because each element in the sequence is a linguistic symbol rather than a numerical value. To cope with this situation, each individual amino acid in the protein sequence has to be coded in a numerical way, i.e. expressed in terms of $f(R_i)$ of Equation 4. As is well known, the hydrophilic value of an amino acid is a very important physicochemical property that has crucial effects on the folding of a protein as well as its function, particularly for membrane proteins. In view of this, we choose the hydrophilic value of R_i for $f(R_i)$. Since a coded protein sequence can be treated as a stationary random process, many technologies in statistical signal processing can be used to characterize the sequence-order effects of a protein sequence.

In statistical signal processing, the correlation, covariance sequence and spectral density function are the three basic statistical quantities of discrete random signals. The true cross-correlation sequence is a statistical quantity defined as

$$R_{xy}(m) = \mathbf{E}\{(x_{n+m}y_n^*)\} = \mathbf{E}\{x_n y_{n-m}^*\} \tag{5}$$

where m is integer, and $\mathbf{E}\{\}$ is the expected value operator, and (*) denotes complex conjugate. The covariance sequence is the mean-removed cross-correlation sequence

$$C_{xy}(m) = \mathbf{E}\{(x_{n+m} - \mu_x)(y_n - \mu_y)^*\} \tag{6}$$

where μ_x and μ_y are the mean of stationary processes x_n and y_n, respectively.

The autocorrelation and autocovariance are their special cases as defined as follows

$$R_{xx}(m) = \mathbf{E}\{(x_{n+m}x_n^*)\} = \mathbf{E}\{x_n x_{n-m}^*\} \tag{7}$$

$$C_{xx}(m) = \mathbf{E}\{(x_{n+m} - \mu_x)(x_n - \mu_x)^*\} \tag{8}$$

In practice, one must estimate these sequences, because it is possible to access only a finite segment of the infinite-length random process. For example, the autocorrelation sequence is estimated as follows:

$$R_{xx}(m) = \frac{1}{L-|m|}\sum_{n=1}^{L-|m|} x(n)x(n+m), \tag{9}$$

where $x(n)$ are indexed from 1 to L, and $|m|$ is the absolute value of integer m.

Comparing Equation 9 and Equations 3 and 4, we can see that the sequence-order correlation factors τ_m defined by Chou (2001) are virtually an autocorrelation sequence of the coded protein sequence. Hence all the powerful tools in statistical signal analysis can be used to incorporate sequence-order effects. The goal of spectral analysis is to describe the distribution (over frequency) of the power contained in a signal, based on a finite set of data. The power spectrum of a stationary random process x_n is mathematically related to the correlation sequence by the discrete-time Fourier transform. In terms of physical frequency f (e.g. in hertz) is given by

$$S_{xx}(f) = \frac{1}{L}\sum_{m=1}^{L}\tau(m)\exp[-j2\pi fm/L], \quad f = 0, 1, 2, \ldots, L-1, \tag{10}$$

where f is the sampling frequency, $\tau(m)$ ($m = 1, \ldots, L$) is the autocorrelation sequence of x_n as defined in Equation 9, and j is the sign indicating the imaginary part. Hence the power spectral density (PSD) of the stationary signal x_n is defined as

$$P_{xx}(f) = \frac{S_{xx}(f)}{L}. \tag{11}$$

where L is the number of x_n's data point.

For real signals, the average power of a signal over a particular frequency and $[f_1,f_2], 0 \leqslant f_1 < f_2 \leqslant \frac{L}{2}$ can be found by integrating the PSD over that band:

$$\bar{P}_{[f_1,f_2]} = 2\int_{f_1}^{f_2} P_{xx}(f)df. \tag{12}$$

From the above expression, it can be seen that $P_{xx}(f)$ represents the power content of a signal in an infinitesimal frequency band, which is why we call it the power spectral density. The energy of $P_{xx}(f)$ is calculated by the following equation:

$$E(f) = |P_{xx}(f)|^2 = \mathrm{Re}^2[P_{xx}(f)] + \mathrm{Im}^2[P_{xx}(f)], \tag{13}$$

where $\mathrm{Re}[P_{xx}(f)]$ is the real part of $P_{xx}(f)$ and $\mathrm{Im}[P_{xx}(f)]$ is the imaginary part.

Since the low-frequency components of a given PSD better reflect the global information (Chou, 1988, 1989) of a given signal than the high-frequency components, we took the first 20 components of the energy spectrum to incorporate most of the sequence order information. Thus the pseudo-amino acid composition of a protein can be defined in a 40-D space as given by

$$\mathbf{P} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_i \\ \vdots \\ p_{40} \end{bmatrix}, \tag{14}$$

where

$$p_k = \begin{cases} \dfrac{f_k}{\sum_{i=1}^{20} f_i + w\sum_{j=20+1}^{40}\theta_j}, & (1 \leqslant k \leqslant 20) \\ \dfrac{w\theta_k}{\sum_{i=1}^{20} f_i + w\sum_{j=20+1}^{40}\theta_j}, & (20+1 \leqslant k \leqslant 40) \end{cases} \tag{15}$$

In the above equation, f_i ($i = 1, 2, \ldots, 20$) denote the frequencies of occurrence of the 20 native amino acids in a protein that actually reflect the amino acid composition (Nakashima *et al.*, 1986; Chou and Zhang, 1993; Chou, 1995) and θj ($j = 1, 2, \ldots, 20$) the first 20 low-frequency coefficients of the energy spectrum of a given protein sequence that reflect some sort of sequence-order effect. The parameter w is used to control how much sequence order information should be considered. In this paper, w was set as 0.15 (Pan *et al.*, 2003).

Compared with the original pseudo-amino acid composition introduced by Chou (2001), the current representation has the following features. (1) It can be seen from Equation 10 that the PSD is mathematically related to the correlation sequence by the discrete-time Fourier transform. Therefore, the sequence order information harboured by autocorrelation sequence is also preserved by the PSD since the transform is linear and will not lose any information. (2) The PSD is a more compact representation. For the coded protein sequences, most of the signal's power concentrates on the low-frequency components of the PSD. Hence the 20 components of the PSD are sufficient to represent the protein sequence as the high-frequency parts contain little information. (3) The low-frequency components represent the global information (Chou, 1988, 1989) of the coded sequence. The type of protein can be reflected by the curve of the hydrophobic values of the residues. The curve's global shape, represented by the low-frequency components of the PSD, is more important in determining the type of membrane protein. For example, the appearance of several peaks and valleys in the curve may indicate that the corresponding protein is likely to be a multi-pass trans-membrane protein (Figure 1).

Results and discussion

Using the dataset constructed by Chou and Elrod (1999), we test our approach to demonstrate its feasibility. The dataset contains 2059 membrane protein sequences. There are 435 type I transmembrane proteins, 152 type II transmembrane proteins, 1311 multi-pass transmembrane proteins, 51 lipid-chain anchored membrane proteins and 110 GPI anchored membrane proteins (Figure 1). Chou and Elrod classified the 2059 into five groups and the names of these proteins are given in Table I in Chou and Elrod (1999).

Instead of the covariant discriminant algorithm, which is a combination of the Mahalanobis distance (Mahalanobis, 1936; Pillai, 1985) and Chou's invariance theorem (Zhou and Assa-Munt, 2001; Pan *et al.*, 2003; Zhou and Doctor, 2003), here we are to use SVM, a kind of learning machine, to conduct prediction. However, owing to the highly uneven sizes of the sub-datasets investigated here, SVM is often undesirably biased toward the classes of membrane proteins with more samples. To deal with this problem, the weighted υ-SVM is proposed to improve the prediction accuracy of small subsets. For a detailed description of this algorithm, see Appendix A, where a full introduction to υ-SVM is given, followed by analysis of the reasons that lead to the undesirable bias toward the subset with more samples in the training set. Finally, by assigning the samples in different subsets with different weights, the unfavorable impact caused by the uneven class size is compensated.

During the operation, the width of the Gaussian RBFs was selected as 1 to minimize the estimation of the VC dimension. The parameter υ is assigned as 0.06. The weight s_i ($i = 1, 2, 3, 4, 5$) for each class is determined by the following procedure:

1. Set the largest set's $s_{\max}$ as 1.
2. Set other s_i ($i = 1, 2, 3, 4, 5$) by $S_i = \frac{\ell_{\max}}{\ell_i} S_{\max} = \frac{\ell_{\max}}{\ell_i}$, where $\ell_{\max}$ denotes the number of training samples of the largest data set and ℓ_i is the number of training samples of other data sets.
3. Train the weighted υ-SVM with the given sample's weight s_i ($i = 1, 2, 3, 4, 5$) for each class and classify the new entered data.

After being trained, the hyper-plane was built in the feature space and thus the output could be obtained. The prediction quality was examined by three methods (Chou and Zhang, 1995), the re-substitution test, the jackknife test and independent dataset test, as explained below.

Re-substitution test

The so-called re-substitution test is designed to examine the self-consistency of an identification method (Zhou, 1998; Cai, 2001; Zhou and Assa-Munt, 2001; Zhou and Doctor, 2003). When the re-substitution test is performed for the current classifier, the type of each membrane protein in a data set is in turn predicted using the rule parameters derived from the same training data set. In Table I, the success rate for the 2059 membrane proteins is listed; the overall success rate is

Table I. Overall rates of correct prediction for the five membrane protein types by different algorithms and test methods

Algorithm	Input form	Test method		
		Self-consistency[a]	Jackknife[a]	Independent dataset[b]
Least Hamming distance (Chou, 1980)	Amino acid composition	$\frac{1293}{2059} = 62.8\%$	$\frac{1279}{2059} = 62.1\%$	$\frac{1751}{2625} = 66.7\%$
Least Euclidean distance (Nakashima *et al.*, 1986)	Amino acid composition	$\frac{1307}{2059} = 63.5\%$	$\frac{1293}{2059} = 62.8\%$	$\frac{1816}{2625} = 69.2\%$
ProtLock (Cedano *et al.*, 1997)	Amino acid composition	$\frac{1372}{2059} = 66.6\%$	$\frac{1348}{2059} = 65.5\%$	$\frac{1674}{2625} = 63.8\%$
Covariant-discriminant (Chou and Elrod, 1999)	Amino acid composition	$\frac{1670}{2059} = 81.1\%$	$\frac{1573}{2059} = 76.4\%$	$\frac{2085}{2625} = 79.4\%$
Augmented covariant discriminant (Chou, 2001)	Pseudo-amino acid composition (Chou, 2001)	$\frac{1872}{2059} = 90.9\%$	$\frac{1665}{2059} = 80.0\%$	$\frac{2298}{2625} = 87.5\%$
Support vector machines	Functional-domain composition (Cai *et al.*, 2003b)	$\frac{1934}{2059} = 93.9\%$	$\frac{1776}{2059} = 86.3\%$	$\frac{1773}{2625} = 67.5\%$
υ-SVM	Pseudo-amino acid composition	$\frac{2030}{2059} = 98.59\%$	$\frac{1701}{2059} = 82.61\%$	$\frac{2376}{2659} = 90.51\%$
Weighted υ-SVM	Pseudo-amino acid composition	$\frac{2056}{2059} = 99.85\%$	$\frac{1696}{2059} = 82.37\%$	$\frac{2371}{2059} = 90.32\%$

[a] Conducted for the 2059 membrane proteins classified into five different types as described in the text and Figure 1.

[b] Conducted based on the rule parameters derived from the 2059 membrane proteins for the 2625 independent membrane proteins (see text).

99.85%, which shows that after being trained, the weighted υ-SVM has captured the complicated relationship between the pseudo-amino acid composition and the types of membrane proteins. Because the rule parameters derived from the training data set harbor the information of the query protein later plugged back into the test, the re-substitution test tends to underestimate the error and enhance the success rate. Therefore, the success rate thus obtained may give some sort of optimistic estimation (Chou and Zhang, 1994; Zhou, 1998; Cai, 2001; Zhou and Assa-Munt, 2001; Zhou and Doctor, 2003). However, the re-substitution test is definitely necessary because any algorithm whose self-consistency performance is poor cannot be deemed a good one. Namely, the re-substitution test is necessary but not sufficient for evaluating a classifier. Hence a cross-validation test for an independent testing data set is recommended because it can reflect the generalization of a classifier in practical applications. This is very useful when checking the validity of a training database for whether it contains sufficient information to reflect all the important features concerned so as to yield a high success rate in application. The results of the re-substitution test obtained for the 2059 membrane proteins are given in Tables I and II.

Jackknife test

The independent dataset test, sub-sampling test and jackknife test are the three most common methods for cross-validation in statistical prediction. Among these three, the jackknife test is regarded as the most objective and effective one; see, e.g., Chou and Zhang (1995) for a comprehensive discussion of this and Mardia *et al*. (1979) for the mathematical principles. During jackknifing, each membrane protein in the dataset is in turn taken out and all the rule parameters are calculated based on the remaining proteins. In other words, the type of each membrane is predicted using the rule parameter derived from all the other membrane proteins except that which is being identified. During the process of jackknifing, both the training data set and testing data set are actually open and a protein will move from one to the other in turn. The results of the jackknife test thus obtained for the 2059 membrane proteins are also given in Tables I and II.

Independent dataset test

Furthermore, predictions were also conducted for the 2625 independent membrane proteins based on the rule parameter derived from the 2059 proteins in the training dataset. The 2625 independent proteins were also taken from Chou and Elrod (1999). Of the 2625 proteins, 478 are type I transmembrane proteins, 180 type II transmembrane proteins, 1867 multi-pass transmembrane proteins, 14 lipid-chain anchored membrane proteins and 86 GPI anchored membrane proteins. The predicted results are also listed in Tables I and II.

From Table I, we may draw the following conclusions. (1) The success predictions obtained by the pseudo-amino acid composition approach are significantly higher than those obtained by the other approaches. (2) A comparison between the current approach and all the other approaches indicates that the success rates by the former are about 6% higher than those by the latter in the self-consistency test and 3% higher in independent dataset test. The only setback is that the jackknife result is about 3.7% lower than that of the functional-domain approach (Cai *et al*., 2003b), but higher than all other methods.

It should be pointed out that it is not sufficient only to compare the overall success rate. To make an in-depth comparison, one should look into the success rate for each type. In order to illustrate the weighted υ-SVM's ability to compensate for the bias caused by the imbalance of the dataset, a comparison with its original form was done as listed in Table 2, from which the following facts can be deduced. (1) For class 4, which is with the smallest size, the success rate by the weighted υ-SVM is about 31% higher than that by the υ-SVM by the self-consistency test and about 51% higher by the independent dataset test. For the jackknife test, the weighted υ-SVM also outperformed the υ-SVM by about 5%. This is fully consistent with what is expected because the effects caused by the dataset size have been taken into consideration during the process of algorithm formulation. (2) The success rates by the weighted υ-SVM are higher than or equal to those obtained by the original υ-SVM in the self-consistency test, jackknife test and independent dataset test, for classes 1, 2, 4 and 5. (3) For class 3, which has the largest size, the success rates by the weighted υ-SVM are lower than those by the υ-SVM in the jackknife test and independent dataset test. However, the overall success rates by the weighted υ-SVM are higher than those by υ-SVM. (4) The results are fully consistent with what we expected: the success rates for the classes with small size are improved at the cost of slightly reducing the success rates for large size classes. This is rational because the samples in small classes are treated as more important, i.e. assigning a larger coefficient for the data in the small class to improve its prediction accuracy. Meanwhile, the importance is that the overall success rates were enhanced.

Table II. Rates of correct prediction (%) for the five membrane protein types by the weighted υ-SVM and υ-SVM algorithms and different test methods

Test method	Algorithm	Membrane protein type				
		Class 1 (435)	Class 2 (152)	Class 3 (1311)	Class 4 (51)	Class 5 (110)
Self-consistency[a]	Weighted υ-SVM	100	100	99.92	100	100
	υ-SVM	99.77	95.39	99.85	68.63	97.27
Jackknife[a]	Weighted υ-SVM	81.38	40.79	91.53	54.90	47.27
	υ-SVM	79.08	38.81	93.29	49.02	45.45
Independent dataset[b]	Weighted υ-SVM	89.54	65.00	94.22	64.29	72.09
	υ-SVM	88.70	62.78	95.13	7.14	72.09

[a] Conducted for the 2059 membrane proteins classified into five different types as described in the text and Figure 1. The data for the 2059 proteins were taken from Table I of Chou and Elrod (1999).

[b] Conducted for the 2625 independent membrane proteins based on the rule parameters derived from the 2059 membrane proteins. The data for the 2625 independent proteins were taken from Chou and Elrod (1999).

Conclusion

The above results indicate that the types of membrane proteins are predictable with considerable accuracy. The development of the statistical prediction of protein attributes generally consists of two aspects: constructing a training dataset and formulating a prediction algorithm. The latter also consists of two aspects, i.e. how to define a protein and how to operate the prediction. The process in expressing a protein from the 20-D amino acid composition space (Nakashima *et al.*, 1986; Chou and Zhang, 1993; Chou, 1995; Zhou, 1998) to the (20+λ)-D pseudo-amino acid composition space (Chou, 2001) to the 2005-D functional-domain composition space (Chou and Cai, 2002; Cai *et al.*, 2003b) reflects the development in representing a protein sample. In this paper, the technique of signal spectrum analysis was introduced to represent a protein via the pseudo-amino acid composition (Chou, 2001) to incorporate the sequence-order information. The process from introducing the simple geometry distance algorithm (Nakashima and Nishikawa, 1994), to the Mahalanobis distance algorithm (Chou and Zhang, 1994; Chou, 1995), to the covariant discriminant algorithm (Chou and Elrod, 1999; Pan *et al.*, 2003; Zhou and Doctor, 2003) and to the current weighted υ-SVM algorithm reflects the development in operating algorithms. The weighted υ-SVM algorithm is particularly useful in solving the problem caused by uneven sizes of the subsets in the training dataset or dealing with the case where the classification accuracy is focused on a small subset.

Acknowledgements

The authors wish to thank the three anonymous referees whose comments were very helpful in strengthening the presentation of this work.

References

Bock,J.R. and Gough,D.A. (2001) *Bioinformatics*, **17**, 455–460.

Cai,Y.D. (2001) *Proteins*, **43**, 336–338.

Cai,Y.D., Liu,X.J., Xu,X.B. and Chou,K.C. (2000) *Mol. Cell Biol. Res. Commun.*, **4**, 230–233.

Cai,Y.D., Liu,X.J., Xu,X.B. and Chou,K.C. (2002a) *Comput. Chem.*, **26**, 293–296.

Cai,Y.D., Liu,X.J., Xu,X.B. and Chou,K.C. (2002b) *Internet Electron. J. Mol. Des.*, **1**, 219–226.

Cai,Y.D., Liu,X.J., Xu,X.B. and Chou,K.C. (2002c) *Peptides*, **23**, 205–208.

Cai,Y.D., Liu,X.J., Xu,X.B. and Chou,K.C. (2002d) *J. Cell. Biochem.*, **84**, 343–348.

Cai,Y.D., Lin,S. and Chou,K.C. (2003a) *Peptides*, **24**, 159–161.

Cai,Y.D., Zhou,G.P. and Chou,K.C. (2003b) *Biophys. J.*, **84**, 3257–3263.

Cedano,J., Aloy,P., Perez-Pons,J.A. and Querol,E. (1997) *J. Mol. Biol.*, **266**, 594–600.

Chew,H.G., Crisp,D.J. and Bogner,R.E. (2000) Target detecting in radar imagery using support vector machines with training size biasing. Singapore.

Chew,H.G., Bogner,R.E. and Lim,C.C. (2001) Dual nu-support vector machine with error rate and training size biasing. Salt Lake City, pp. 1269–1272.

Chou,J.J. and Zhang,C.T. (1993) *J. Theor. Biol.*, **161**, 251–262.

Chou,K.C. (1988) *Biophys. Chem.*, **30**, 3–48.

Chou,K.C. (1989) *Trends Biochem. Sci.*, **14**, 212.

Chou,K.C. (1995) *Proteins*, **21**, 319–344.

Chou,K.C. (1999) *J. Protein Chem.*, **18**, 473–480.

Chou,K.C. (2000) *Biochem. Biophys. Res. Commun.*, **278**, 477–483.

Chou,K.C. (2001) *Proteins*, **43**, 246–255; Erratum, 2001, **44**, 60.

Chou,K.C. (2002) In Weinrer,P.W. and Lu,Q. (eds), *Gene Cloning and Expression Technologies*. Westborough, MA, Eaton Publishing, pp. 57–70.

Chou,K.C. and Cai,Y.D. (2002) *J. Biol. Chem.*, **277**, 45765–45769.

Chou,K.C. and Cai,Y.D. (2003) *J. Cell. Biochem.*, **90**, 1250–1260; Addendum, 2004, **91**, 1085.

Chou,K.C. and Elrod,D.W. (1999) *Proteins*, **34**, 137–153.

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

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

Chou,P.Y. (1980) In *Abstracts of Papers, Part I, Second Chemical Congress of the North American Continent, Las Vegas*.

Cristianini,N. and Shawe-Taylor,J. (2000) *Support Vector Machines*. Cambridge: Cambridge University Press.

Ding,C.H. and Dubchak,I. (2001) *Bioinformatics*, **17**, 349–358.

Fan,X.W., Du,S.X. and Wu,T.J. (2003) *J. Image Graphics*, **8**, 1037–1042.

Guo,Z.M. (2002) Master's Thesis, Shanghai Jiaotong University.

Hua,S.J. and Sun,Z.R. (2001) *J. Mol. Biol.*, **308**, 397–407.

Karush,W. (1939) MSc Thesis, University of Chicago.

Knerr,S., Personnaz,L. and Dreyfus,G. (eds) (1990) *Single-layer Learning Revisited, a Stepwise Procedure for Building and Training Neural Networks*. Berlin: Springer.

Lee,Y.J. and Mangasarian,O.L. (2001) *RSVM, Reduced Support Vector Machines*.

Lin,C.F. and Wang,S.D. (2002) *IEEE Trans. Neural Networks*, **13**, 464–471.

Liu,W. and Chou,K.C. (1999) *Protein Eng.*, **12**, 1041–1050.

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

Mardia,K.V., Kent,J.T. and Bibby,J.M. (1979) *Multivariate Analysis*. London: Academic Press, pp. 322–381.

Nakashima,H. and Nishikawa,K. (1994) *J. Mol. Biol.*, **238**, 54–61.

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

Pan,Y.X., Zhang,Z.Z., Guo,Z.M., Feng,G.Y., Huang,Z.D. and He,L. (2003) *J. Protein Chem.*, **22**, 395–402.

Pillai,K.C.S. (1985) In Kotz,S. and Johnson,N.L. (eds), *Encyclopedia of Statistical Sciences*. New York: Wiley, pp. 176–181.

Schneider,G. and Wrede,P. (1994) *Biophys. J.*, **66**, 335–344.

Scholkopf,B., Smola,A. and Williamson,R.C. (2000) *Neural Comput.*, **12**, 1207–1245.

Suykens,J. and Vandewalle,J. (1999) *Neural Process. Lett.*, **9**, 293–300.

Vapnik,V.N. (1995) *The Nature of Statistical Learning Theory*. Berlin: Springer.

Vapnik,V.N. (1998) *Statistical Learning Theory*. New York: Wiley-Interscience.

Zhou,G.P. (1998) *J. Protein Chem.*, **17**, 729–738.

Zhou,G.P. and Assa-Munt,N. (2001) *Proteins*, **44**, 57–59.

Zhou,G.P. and Doctor,K. (2003) *Proteins*, **50**, 44–48.

Received May 10, 2004; revised July 23, 2004; accepted July 26, 2004

Edited by Alan Fersht

Appendix A: Weighted support vector machines

Owing to its characteristics of global optimization, sparseness of the solution and the use of kernel-induced feature spaces, SVMs are widely used in bioinformatics. SVMs have been applied to protein fold recognition (Ding and Dubchak, 2001), protein–protein interaction prediction (Bock and Gough, 2001), protein secondary structure prediction (Hua and Sun, 2001), protein structural class prediction (Cai *et al.*, 2002a), prediction of the specificity of GalNAc-transferase (Cai *et al.*, 2002c), subcellular location prediction (Cai *et al.*, 2000, 2002d; Chou and Cai, 2002), signal peptide prediction (Cai *et al.*, 2003a) and membrane protein type prediction (Cai *et al.*, 2002b, 2003b). Many variations have been proposed to improve its performance, such as C-SVM (Vapnik, 1995, 1998), one-class (Scholkopf *et al.*, 2000), RVSM (Lee and Mangasarian, 2001), υ-SVM (Scholkopf *et al.*, 2000), weighted-SVM (Chew *et al.*, 2001; Lin and Wang, 2002) and LS-SVM (Suykens and Vandewalle, 1999). When training datasets with uneven class sizes are used, the classification results based on support machines are undesirably biased toward the class with more samples in the subset. Namely, the larger the subset, the smaller is the classification error. Some efforts have been made in this regard (Chew *et al.*, 2000, 2001; Lin and Wang, 2002). The weighted-SVM (Fan *et al.*, 2003) compensates for the unfavorable impact caused by this kind of bias by assigning each subset a different penalty coefficient.

Based on the concept of weighted-SVM (Fan *et al.*, 2003), we adopt its specific form, the weighted υ-SVM, and apply it to the problem of prediction of membrane protein types.

υ-SVM

The basic idea of applying SVMs to pattern classification can be outlined as follows. First, map the input vectors into a feature space (possible with a higher dimension), either linearly or non-linearly, which is relevant to the selection of the kernel function. Then, within the feature space, seek an optimized linear division, i.e. construct a hyper-plane which can separate the entire samples to two classes (this can be extended to multi-classes) with the least errors and maximal margin. The SVMs training process always seeks a global optimized solution and avoids over-fitting, so it has the ability to deal with a large number of features. A complete description to the theory of SVMs for pattern recognition was given by Vapnik (1998).

Given a set of ℓ samples, i.e. a series of input vectors

$$x_i \in \Re^d (i = 1, \ldots, \ell), \tag{A1}$$

where x_i can be regarded as the ith protein or vector defined in the 40-D pseudo-amino acid space according to Equations 14 and 15 and $\Re^d$ is a Euclidean space with d dimensions. Since the multi-class identification problem can always be converted into a two-class identification problem, without loss of generality, the formulation below is given for the two-class case only. Suppose that the output derived from the learning machine is expressed by $y_i \in \{+1, -1\}$ $(i = 1, \ldots, N)$, where the indices -1 and $+1$ are used to stand for the two classes concerned, respectively, the goal here is to construct one binary classifier or derive one decision function from the available samples that has a small probability of misclassifying a future sample.

υ-SVM (Scholkopf *et al.*, 2000) uses the parameter υ to control the number of support vectors and errors. Its primal problem is

$$\begin{aligned} &\min_{\omega,b,\xi,\rho} \frac{1}{2}\omega^T\omega - \upsilon\rho + \frac{1}{l}\sum_{i=1}^{l}\xi_i \\ &\text{s.t.} \quad y_i\left(\omega^T\phi(x_i) + b\right) \geqslant \rho - \xi_i \\ &\qquad \xi_i \geqslant 0, i = 1, \ldots, l \\ &\qquad \rho \geqslant 0 \end{aligned} \tag{A2}$$

Its dual problem is

$$\begin{aligned} &\min_{\alpha} \frac{1}{2}\alpha^T Q\alpha \\ &\text{s.t.} \quad 0 \leqslant \alpha_i \leqslant \frac{1}{l}, \quad i = 1, \ldots, l \\ &\qquad y^T\alpha = 0 \\ &\qquad e^T\alpha \geqslant \upsilon \end{aligned} \tag{A3}$$

The decision function is

$$\tilde{f}(x) = \text{sign}\left(\sum_{i=1}^{l} y_i\alpha_i K(x_i, x) + b\right). \tag{A4}$$

To calculate b and ρ in the above equation, we need to select the same number of samples ($S > 0$ is the number of samples) from the two datasets. Suppose S_+ is the number of samples from the positive training dataset and S_- that from the negative training dataset. According to the Karush–Kuhn–Tucker (KKT) conditions (Karush, 1939; Cristianini and Shawe-Taylor, 2000), the condition in Equation A2

$$y_i\left(\omega^T\phi(x_i) + b\right) \geqslant \rho - \xi_i \tag{A5}$$

becomes

$$y_i\left(\omega^T\phi(x_i) + b\right) = \rho - \xi_i \tag{A6}$$

and

$$\xi_i = 0 \tag{A7}$$

Hence, with some deductions, we obtain the formulations to calculate b and ρ:

$$b = -\frac{1}{2s}\sum_{x\in S_+ \cup S_-}\sum_j \alpha_j y_j K\left(x, x_j\right) \tag{A8}$$

$$\rho = \frac{1}{2s}\left[\sum_{x\in S_+}\sum_j \alpha_j y_j K\left(x, x_j\right) - \sum_{x\in S_-}\sum_j \alpha_j y_j K\left(x, x_j\right)\right] \tag{A9}$$

In C-SVM, the only adjustable parameter is the constant C, which influences its performance greatly. However, because there is no natural interpretation of this parameter, it is hard to adjust it. In υ-SVM, the parameter C is replaced by υ. In this parameterization (Equation A3), υ places a lower bound on the sum of the α_i, which causes the linear term to be dropped from the objective function. Another connection between these two algorithms is that an increase in parameter C leads to a decrease in the number of support vectors in C-SVM, while a decrease in υ leads to a smaller number of support vectors.

Weighted υ-SVM

The performance of υ-SVM is also impaired when training sets with uneven class sizes are used. We propose the weighted υ-SVM to solve this problem. The primal problem of weighted υ-SVM is given by

$$\begin{aligned} &\mathbf{Min}_{\omega,b,\xi,\rho} \frac{1}{2}\omega^T\omega - \upsilon\rho + \frac{1}{l}\sum_{i=1}^{l} s_i\xi_i \\ &\text{s.t.} \quad y_i\left(\omega^T\phi(x_i) + b\right) \geqslant \rho - \xi_i \\ &\qquad \xi_i \geqslant 0, i = 1, \ldots, l, \rho \geqslant 0 \end{aligned} \tag{A10}$$

where s_i denotes the weight for each sample. We use the Lagrange multiplier method to solve the above optimization problem

$$\begin{aligned} &L(\omega, \xi_i, b, \rho, \alpha_i, \beta_i, \delta) \\ &\quad = \frac{1}{2}\omega^T\omega - \upsilon\rho + \frac{1}{l}\sum_{i=1}^{l} s_i\xi_i \\ &\qquad - \sum_{i=1}^{l}\left[\alpha_i\left(y_i\left(w^T\phi(x_i) + b\right) - \rho + \xi_i\right) + \beta_i\xi_i\right] - \rho\delta \end{aligned} \tag{A11}$$

where $\alpha_i \geqslant 0$, $\beta_i \geqslant 0$, $\delta \geqslant 0$ are all Lagrange multipliers. Differentiating and imposing a stationary condition, we obtain

$$\frac{\partial L}{\partial \omega} = \omega - \sum_{i=1}^{l}\alpha_i y_i \phi(x_i) = 0 \tag{A12}$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{l}\alpha_i y_i = 0 \tag{A13}$$

$$\frac{\partial L}{\partial \xi_i} = \frac{1}{l}s_i - \alpha_i - \beta_i = 0 \tag{A14}$$

$$\frac{\partial L}{\partial \rho} = -\upsilon + \sum_{i-1}^{l}\alpha_i - \delta = 0 \tag{A15}$$

Substituting Equations A12–A15 into Equation A11, we obtain its dual problem:

$$\begin{aligned} &\underset{\alpha}{\mathbf{Min}}\frac{1}{2}\alpha^T Q\alpha \\ \text{s.t.}\quad & 0 \leqslant \alpha_i \leqslant \frac{1}{l}s_i \\ & y^T\alpha = 0 \\ & e^T\alpha \geqslant \upsilon \end{aligned} \tag{A16}$$

In the weighted υ-SVM, by assigning training samples of different classes different weights, we can compensate for the unfavorable impact caused by the uneven class size. For simplicity, we analyze the case of two classes. We assign $s_i = s_+$ for positive class and $s_i = s_-$ for negative class. Hence the constraint in Equation A11 is reduced to

$$\begin{cases} 0 \leqslant \alpha_i \leqslant \frac{1}{l}s_+, & \forall y_i = +1 \quad i = 1, \ldots, \ell \\ 0 \leqslant \alpha_i \leqslant \frac{1}{l}s_-, & \forall y_i = -1 \quad i = 1, \ldots, \ell \end{cases} \tag{A17}$$

According to the constraint in Equation A9, we obtain

$$\sum_i \alpha_i y_i = \sum_{i:y_i=+1} \alpha_i - \sum_{i:y_i=-1} \alpha_i = 0 \tag{A18}$$

Because most of the gaps are not zero in the trained SVMs, according the constraint in Equation A1, ρ is greater than zero. By Kuhn–Tucher theory, we obtain $\delta\rho = 0$. If $\rho > 0$, then $\delta = 0$. Substituting this into Equation A6, we obtain

$$\sum_{i=1}^{\ell} \alpha_i = \upsilon \tag{A19}$$

Thus, from Equations A18 and A19, we deduce that

$$\sum_i \alpha_i = \sum_{i:y_i=+1} \alpha_i + \sum_{i:y_i=-1} \alpha_i = 2\sum_{i:y_i=+1} \alpha_i = \upsilon \tag{A20}$$

Before further analysis, we need to define several notations. The support vector (SV) is the training sample whose dual variables $\alpha_i \geqslant 0$. The normal support vector (NSV) is defined as the training sample whose dual variables $0 \leqslant \alpha_i \leqslant \frac{1}{l}$. The boundary support vector (BSV) is the training sample that both satisfy $\alpha_i = \frac{1}{l}$ and $\xi_i > 0$. The BSVs are misclassified training points.

Because the BSV has the property $\alpha_i = \frac{s_+}{\ell}$, the sum of N_{BSV_+} α_i (BSV's dual variable) is less than $\sum_{i:y_i=+1} \alpha_i N_{BSV_+}$, which is the number of BSVs that belong to the positive class:

$$2N_{BSV_+}\frac{S_+}{l} \leqslant 2\sum_{i:y_i=+1}^{l} \alpha_i = \upsilon \tag{A21}$$

i.e.

$$2N_{BSV_+}\frac{S_+}{\ell} \leqslant \upsilon \tag{A22}$$

In addition, the maximum of SV's dual variable α_i is $\frac{S_+}{\ell}$ and N_{sv_+} denotes the number of training samples belonging to the positive class. Thus, we have

$$\upsilon = 2\sum_{i:y_i=+1}^{\ell} \alpha_i \leqslant 2N_{SV_+}\frac{s_+}{\ell} \tag{A23}$$

i.e.

$$\upsilon \leqslant \frac{2N_{SV_+}s_+}{\ell} \tag{A24}$$

From Equations A15 and A17, we have

$$\frac{2N_{BSV_+}s_+}{\ell} \leqslant \upsilon \leqslant \frac{2N_{SV_+}s_+}{\ell} \tag{A25}$$

Likewise, we have a similar result for the case of negative class:

$$\frac{2N_{BSV_-}s_-}{\ell} \leqslant \upsilon \leqslant \frac{2N_{SV_-}s_-}{\ell} \tag{A26}$$

Equations A18 and A19 can be transformed into

$$\frac{N_{BSV_+}}{\ell_+} \leqslant \frac{\upsilon l}{2s_+\ell_+} \leqslant \frac{N_{SV_+}}{\ell_+} \tag{A27}$$

$$\frac{N_{BSV_-}}{\ell_-} \leqslant \frac{\upsilon l}{2s_-\ell_-} \leqslant \frac{N_{SV_-}}{\ell_-} \tag{A28}$$

where ℓ_+ is the number of positive training samples, ℓ_- the number of negative training samples and $\ell = \ell_+ + \ell_-$. The $\frac{N_{BSV_+}}{\ell_+}$ in Equation A18 and $\frac{N_{BSV_-}}{\ell_-}$ in Equation A19 may be interpreted as the rates of accuracy of positive and negative classes, respectively. From Equations A20 and A21, it can be shown that the rate of accuracy for positive class is upper bounded by $\frac{\upsilon\ell}{2s_+\ell_+}$ and the rate of accuracy for negative class is upper bounded by $\frac{\upsilon\ell}{2s_-\ell_-}$. Therefore, in order to balance two classes' rates of accuracy, we only need to force $\frac{\upsilon\ell}{2s_+\ell_+} = \frac{\upsilon\ell}{2s_-\ell_-}$. Then we have the following equation:

$$\frac{s_+}{s_-} = \frac{\ell_-}{\ell_+} \tag{A29}$$

In the weighted υ-SVM, by weighting the samples in the small class, the classification accuracy of the small class can be improved. Meanwhile, eliminating the bias toward the large class, the prediction accuracy of the large class is reduced slightly. Such a method can be applied directly to the prediction of membrane protein types where the training sets of five classes are highly uneven.

In this paper, we use the 'one-against-one' approach (Knerr *et al.*, 1990), in which $k(k-1)/2$ classifiers are constructed and each one trains data from two different classes; k is the number of classes and in this paper $k = 5$. During the training stage, we first calculate the ratio of different sample sizes according to Equation A29 and then assign different weights to the training samples of different classes as described in the next section. In classification we use a voting strategy: each binary classification is considered to be a voter where votes can be cast for all data points; the end point is designated to be in a class with maximum number of votes.