Journal of Theoretical Biology 271 (2011) 10–17

Contents lists available at ScienceDirect

Journal of Theoretical Biology

journal homepage: www.elsevier.com/locate/yjtbi

Predicting membrane protein types by fusing composite protein sequence features into pseudo amino acid composition

Maqsood Hayat, Asifullah Khan *

Department of Computer and Information Sciences, Pakistan Institute of Engineering and Applied Sciences, Nilore, Islamabad, Pakistan

ARTICLE INFO

Article history:
Received 29 July 2010
Received in revised form
10 November 2010
Accepted 10 November 2010
Available online 24 November 2010

Keywords:
Neural networks
SVM
Membrane protein
Composite protein sequence representation
Amino acid composition

ABSTRACT

Membrane proteins are vital type of proteins that serve as channels, receptors, and energy transducers in a cell. Prediction of membrane protein types is an important research area in bioinformatics. Knowledge of membrane protein types provides some valuable information for predicting novel example of the membrane protein types. However, classification of membrane protein types can be both time consuming and susceptible to errors due to the inherent similarity of membrane protein types. In this paper, neural networks based membrane protein type prediction system is proposed. Composite protein sequence representation (*CPSR*) is used to extract the features of a protein sequence, which includes seven feature sets; amino acid composition, sequence length, 2 gram exchange group frequency, hydrophobic group, electronic group, sum of hydrophobicity, and R-group. Principal component analysis is then employed to reduce the dimensionality of the feature vector. The probabilistic neural network (*PNN*), generalized regression neural network, and support vector machine (*SVM*) are used as classifiers. A high success rate of 86.01% is obtained using SVM for the jackknife test. In case of independent dataset test, *PNN* yields the highest accuracy of 95.73%. These classifiers exhibit improved performance using other performance measures such as sensitivity, specificity, Mathew's correlation coefficient, and *F*-measure. The experimental results show that the prediction performance of the proposed scheme for classifying membrane protein types is the best reported, so far. This performance improvement may largely be credited to the learning capabilities of neural networks and the composite feature extraction strategy, which exploits seven different properties of protein sequences. The proposed *Mem-Predictor* can be accessed at http://111.68.99.218/Mem-Predictor.

1. Introduction

Membrane proteins are important parts of proteins playing various roles in cells biology; some work as pumps or channel for transporting molecules into or out of the cells, while some provide the skeleton for the lipid bilayer membranes. About 30% of human genomes have been encoded from membrane protein. Knowledge of a given membrane protein type is helpful in determining its function. However, experimentally or manually detecting this information is difficult due to the intrinsic biochemical properties of membrane proteins or because of the need of growing huge body of the new proteins. Over the last decade, the technological improvements have rapidly increased the size of the biological data that contains gene sequences from various organisms. Currently the main challenges for the bioinformatics field are to store, analyze, and annotate this flood of unprocessed data. Additionally, the manual annotation of the membrane protein in some situations is very difficult or even impossible. In view of this, an effective membrane protein type predictor can provide immense help in understanding their functions by accurately predicting the types of the membrane proteins. Based on their functions, membrane proteins are classified into transmembrane proteins, which span across the membrane, and anchored proteins, which are attached to the membrane on one side. Five main sub-types of membrane proteins are; Type-I transmembrane, Type-II transmembrane, multipass transmembrane membrane, lipid chain-anchored membrane, and GPI-anchored membrane protein (Chou and Elrod, 1999). In the last few decades, several bioinformatics approaches are used for prediction of membrane protein types. A number of efforts have been carried out to predict membrane protein types from their sequence information. Chou and Elrod (1999) have first introduced the covariant discriminant algorithm (*CDA*) to identify the types of membrane proteins based on their amino acid (*AA*) composition. However, in case of *AA* composition, the sequence-order and sequence-length effects are lost. To avoid losing many important information hidden in protein sequences, the pseudo amino acid composition (PseAAC) was proposed (Chou, 2001; Chou, 2005) to replace the simple amino acid composition (AAC) for representing the sample of a protein. For a summary about its

* Corresponding author. Tel.: +92 51 2207381; fax: +92 51 2208070.
E-mail addresses: maqsood.hayat@pieas.edu.pk (M. Hayat), asif@pieas.edu.pk, khan.asifullah@gmail.com (A. Khan).

0022-5193/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jtbi.2010.11.017

development and applications, such as how to use the concept of Chou's PseAAC to develop 16 different forms of PseAAC, including those that are able to incorporate the functional domain information, gene ontology (GO) information, cellular automaton image information, sequential evolution information, among many others, see a recent comprehensive review (Chou, 2009). In this study, we are to introduce a different form of PseAAC for predicting membrane protein types. Chou (2001) has proposed the use of the *CDA* in conjunction with pseudo amino acid (*PseAA*) composition based feature extraction. Chou has carried out a series of works to improve the prediction accuracy of membrane protein types. Sonnhammer et al. (1998) have used the hidden markov model for predicting topology of membrane protein types. Cai et al. (2004) have also used *AA* composition and *SVM*. Similarly, Liu et al. (2005) have employed the Fourier spectrum and *SVM*, while Wang et al. (2004) have used weighted *SVM* and *PseAA* composition. Wang et al. (2006) have used *PseAA* and stacked generalization. Yang has used amino acid and peptide composition for prediction of membrane protein types. Chou and Shen (2007a) and Chou and Shen (2009) developed a web server for prediction of protein attributes and membrane protein types; they considered eight different membrane protein types in their dataset. The discrete wavelets transform (*DWT*) with cascaded neural network (Rezaei et al., 2008), and *SVM* in conjunction with *DWT* (Qiu et al., 2010) have also been employed for the prediction of membrane protein types. Wang, et al. (2010) have applied the dipeptide composition and *k*-nearest neighbor for predicting the membrane protein types. Neighborhood preserving embedding (*NPE*) technique has been used to reduce the dimensionality. Some research work has tried to explore the relationship between physiochemical properties and the types of membrane protein (Golmohammadi et al., 2007).

In this paper, composite protein sequence representation (*CPSR*) is used as feature extraction strategy. Principal component analysis (*PCA*) is used for the reduction of dimensionality. Probabilistic neural network (*PNN*), generalized regression neural network (*GRNN*), and support vector machine (*SVM*) are used as classifiers. Our main goal is to develop an efficient and high performance membrane protein type predictor by exploring the discrimination capability of *CPSR* and the learning capability of the neural network. First, each protein sequence is mapped into a novel feature-based vector. Next, the best performing classifier is selected to identify the type of membrane protein. Three statistical tests are performed using two large benchmark datasets for evaluating the performance of the proposed method.

The remaining part of the paper is organized as such: Section 2 describes Material and Methods. Section 3 describes the performance evaluation measures. Section 4 presents experimental results and discussion. Finally, Section 5 concludes the paper.

2. Material and methods

The dataset that we have used for the performance analysis and comparison has been developed by Chou and Elrod (1999). The dataset was passed from different phases. In the first phase, only those sequences were included in the dataset whose descriptions are clear. In the second phase, only one protein sequence was included from those who had the same name, but from different species. In the third phase, sequences whose type was described by two or more types were not included because of lack of uniqueness. After the above screening procedures, the dataset contain only 2059 protein sequences (Chou and Elrod, 1999). In training set where 435 are type-I, 152 type-II, 1311 multipass transmembrane, 51 lipid chain anchored membrane, and 110 are GPI anchored membrane protein sequences. The independent dataset contains 2625 membrane protein sequences. In which 487 are of type-I, 180 type-II, 1867 multipass, 14 lipid chain anchored, and 86 are GPI anchored membrane protein sequences.

2.1. Composite protein sequence representation

A protein sequence consists of 20 unique amino acids. All amino acids have a common basic chemical structure, but possess different chemical properties due to differences in their side chains. A protein can be represented by a chain of amino acids. Different proteins have different amino acid strings, in terms of the ordering and total number (length of the sequence). We have used seven distinct feature sets. These feature sets along with their corresponding number of features are shown in Table 1.

2.1.1. Amino acid composition

Amino acid composition of a protein is defined by 20 discrete numbers, each representing the occurrence frequency of the 20 native amino acids in the protein sequence. In amino acid composition, proteins can be expressed in 20D vector (Chou and Elrod, 1999)

$$\mathbf{p} = [p_1, p_2, \ldots, p_n]^T \quad (1)$$

where p_1, p_2, $p_3 \ldots p_{20}$ are the composition components of the 20 amino acids of a protein *P*, and *T* denotes transposition.

2.1.2. Sequence length

L is defined as the total number of amino acids in the given protein sequence.

2.1.3. 2-Gram exchange group composition

Thirty-six features are extracted by converting the sequence into its equivalent 6-letter exchange group representation (Wu et al., 1995; Golmohammadi et al., 2007) shown in Table 2, which has been derived from the PAM matrix (Dayhoff et al., 1978). The exchange groups are broader classes of amino acids that represent the effects of evolution. For example, all H, R, and K amino acids in the original sequence are replaced by e_1. After the amino acids are replaced, the resulting sequence consists of an alphabet of only six different characters. We compute the frequency of occurrence of each possible 2-gram pair of the consecutive exchange group of amino acids (Wu et al., 1992; Eghbal et al., 2009). 2-gram exchange group composition is the most important feature set and the reason for this discrimination is that it takes the sequence of amino acids, rather than just their composition.

2.1.4. Hydrophobic group

In Table 2, amino acids are categorized in two groups, i.e. hydrophobic and hydrophilic (Zumdahi, 2000; Waugh, 1954). The corresponding two features are based on counts of the hydrophobic and hydrophilic amino acids in the protein sequence.

Table 1
Feature based sequence representation.

Feature set	Number of features
Amino acid composition	20
Sequence length	1
2-Gram exchange group frequency	36
Hydrophobic group	2
Electronic group	6
Sum of hydrophobicity	1
R-group	5

2.1.5. Electronic group

The electronic group specifies whether a given amino acid is electrically neutral, donates electrons, or accepts electrons. We again compute the frequency of amino acids in each of the electronic groups as shown in Table 2.

2.1.6. Sum of hydrophobicity

Each amino acid has an associated hydrophobic affinity, which is often measured using a hydrophobic index. The Eisenberg hydrophobic index, given in Table 3, which provides information about the membrane-associated helices (Eisenberg et al., 1984), is applied in this feature set. This index is normalized and ranges between −2.53 for R (the least hydrophobic) and 1.38 for I (the most hydrophobic). Similarly (Kedarisetti et al., 2006), we have computed the sum of this hydrophobic index over all amino acids in the protein sequence, which gives a single feature.

2.1.7. R-group

Each amino acid has a different side chain. However, some of these side-chains have similar characteristics. They can thus be clustered into five sub-groups (Kedarisetti et al., 2006) as shown in Table 2. The composition of amino acids in each of these groups is computed. Overall, the resulting feature vector consists of 71 features, using seven feature sets.

2.2. Principal component analysis (PCA)

PCA is a statistical technique that is widely used in face recognition and image compression. It tries to find patterns in high dimensionality data. *PCA* is largely used for reducing the number of variables. It is useful when the number of variables is large and there is some redundancy in the data. The main advantage of the *PCA* is that it reduces the dimensionality but often does not lose much information. *PCA* is based on eigenvector-based multivariate analyses. If a multivariate dataset is visualized as a set of coordinates in a high-dimensional data space (1 axis per variable), *PCA* supplies the user with a lower-dimensional picture. In this paper, the original dimensionalities of the features vector are 71. Using *PCA*, we have reduced the dimensionalities from 71D to 48D. Fig. 1 shows the variance between the first principal component and the second principal component, where almost 99% of the variance is accounted for by the first two principal components.

2.3. Classification

Three different neural networks based approaches are used as classification algorithm; *PNN*, *GRNN*, and *SVM*. It is to be noted that *SVM* is often categorized as a neural network system, especially linear *SVM* is considered equivalent to the perceptron (Chen and Isaac, 2003).

2.3.1. Probabilistic neural network (PNN)

The probabilistic neural network was developed by Specht (1990). *PNN* is based on the Bayes theory. It estimates the likelihood of a sample being part of a learned category (Khan et al., 2010). *PNN* uses the radial basis function as kernel. *PNN* interprets the network structure in the form of probability density function and usually performs better as compared with the other state of the art neural networks. Practical advantage of the *PNN* is that, unlike many other neural networks, it operates completely in parallel without a need for feedback from the individual neurons to the inputs. Additionally,

Table 2
Property groups of amino acids used for deriving features (Eghbal et al., 2009; Zumdahi, 2000; Waugh, 1954; Kedarisetti et al., 2006).

Group	Sub-group	Amino acids
Exchange group	e_1	KHR
	e_2	DENQ
	e_3	C
	e_4	AGPST
	e_5	ILMV
	e_6	FYW
Hydrophobic group	Hydrophobic	ACFILMPVWY
	Hydrophilic	DEGHKNQRST
Electron group	Electron donor	DEPA
	Weak electron donor	VLI
	Electron acceptor	KNR
	Weak electron acceptor	FYMTQ
	Neutral	GHWS
	Special AA	C
R-group	Non-polar aliphatic	ALIV
	Glycine	G
	Non-polar	FMPW
	Polar uncharged	CNQSTV
	Charged	DEHKR

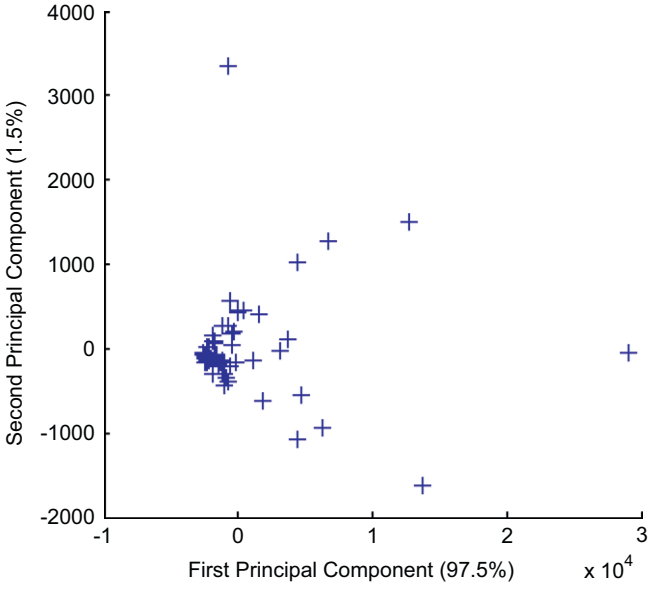

Fig. 1. First principal component versus second principal component.

Table 3
Eisenberg hydrophobicity index values of amino acids (Eisenberg et al., 1984).

Amino acid	Index value	Amino acid	Index value	Amino acid	Index value	Amino acid	Index value
A	0.62	E	−0.74	L	1.06	S	−0.18
R	−2.53	Q	−0.85	K	−1.5	T	−0.05
N	−0.78	G	0.48	M	0.64	W	0.81
D	−0.9	H	−0.4	F	1.19	Y	0.26
C	0.29	I	1.38	P	0.12	V	1.08

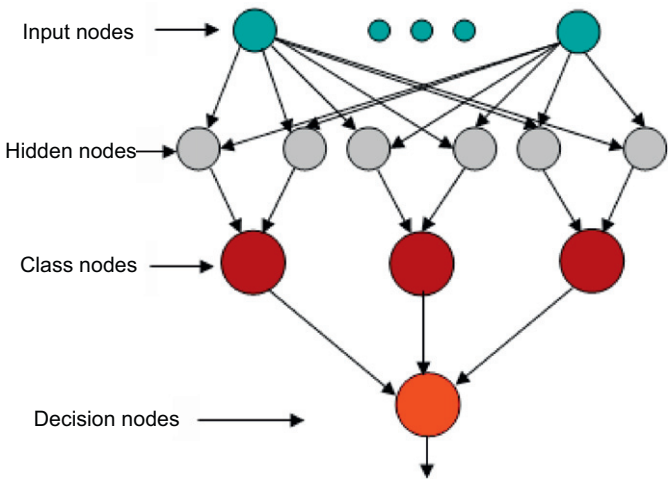

Fig. 2. Architecture of the probabilistic neural network.

PNN training is easy and instantaneous. *PNN* is used generally for classification, where the target variable is categorical and one has to compute the conditional probabilities for each of c classes. *PNN* takes n dimensional feature vector, $x=\{x_1, x_2, \ldots, x_n\}$ as input. The inputs are passed to the neurons in the first hidden layer (Fig. 2).

Then Gaussian probability distribution is calculated for each class $k(1 \le k \le c)$

$$p_j^k(x) = 1/(2\pi)^{n/2} \left| \sum_j^k \right|^{-1/2} \exp\left\{ -1/2(x-u_j^k)^T \left(\sum_j^k \right)^{-1} (x-u_j^k) \right\} \quad (2)$$

where n is dimension of the pattern vector x, j is the number of example in class k, u_j^k is the mean, and $\sum_j^k$ are covariance matrix. Therefore, m_k multivariate distributions exist for each class k. In the second layer, the class probability functions are calculated through a combination of these multivariate densities

$$o_k(x) = \sum_{j=1}^{m_k} \pi_j^k p_j^k(x) \quad (3)$$

where π_j^k is the mixing proportion within class and is nonnegative, and

$$\sum_{j=1}^{m_k} \pi_j^k = 1 \quad k = 1, \ldots, c \quad (4)$$

while

$$p_k(x) = \sum_{l=1}^{c} v_l^k a_l o_l(x) \quad (5)$$

where v_j^k is the risk function and a_l is the prior probability of class l. In the third layer, the decision risk is computed. When the *PNN* is used as a risk-based decision, class l with minimum risk p_l would be chosen

$$l = \text{argmin}_{1 \le k \le c}\{p_k(x)\} \quad (6)$$

2.3.2. Generalized regression neural network (GRNN)

GRNN are non-parametrical kernel regression estimators used in statistics. *GRNN* are based on Parzen–Rosenblatt density estimation. Main advantages of the *GRNN* as compared to the other types of neural networks are that it can accurately compute the approximation function from sparse data and also extract automatically the appropriate regression model (linear or nonlinear) from the data; it can train rapidly with very simple topology design. It is robust to noise and outliers due to its property of being instance-based techniques that works with weighted averages of the stored model (Devroye and Györfi, 1985; Goulermas et al., 2008). *GRNN* are used for classification problems where the target variable is continuous. It computes the conditional mean estimate for each class. Let $X=\{x_1, x_2, x_n\}$ is an n dimensional feature vector input of a function $y=f(x)$. Its corresponding output is real valued. The objective of the approximation function is to obtain an estimate $\hat{f}$ of f using X

$$\hat{f}(x) = E[y|x] = \int_{-\infty}^{\infty} yP(y,x)dy \Big/ \int_{-\infty}^{\infty} P(y,x)dy \quad (7)$$

$P(x, y)$ is joint probability density function and $E[y|x]$ is the conditional expectation. For example, the density $P(x)$ from a set of n training samples is give as

$$P(x) = 1/n\delta^d \sum_{i=1}^{n} W(x-x_i/\delta) \quad (8)$$

where W is the kernel function with the required condition that $\delta w(\xi) \ge 0$ and $\int w(\xi)d\xi = 1$.

If the parameter $\delta > 0$ then the kernel control the smoothness of the approximation. The drawback of the *GRNN* is high smoothness and dependency on the spatial density of the monitoring dataset.

2.3.3. Support vector machine (SVM)

SVM is a kind of learning machine based on statistical learning theory (Cao, 2003; Cai et al., 2002a, 2002b, 2002c, 2003a, 2003b, 2003c; Chou and Cai, 2002; Ding et al., 2007; Khan et al., 2008a, 2008b). It was developed by Cortes and Vapnik in 1995 (Zhang et al., 2009) and modified by Vapnik in 1999. When using *SVM* for classification, it performs operation into two steps; first, maps the sample data vector into a high dimensional space. The dimension of this space is significantly higher than the dimension of the original data space. In the second step, the algorithm finds a hyperplane in this space that has the largest margin separating the classes of the data. When the data is not linearly separable, *SVM* maps the data into a higher dimensional space (Duda et al., 2001) where maximal separating hyperplane is constructed and it then tries to separate the data. The maximum margin hyperplane has maximum distance from member to the non-member. *SVM* uses linear, polynomial, and radial basis kernel. The most remarkable characteristics of *SVM* are the absence of local minima. While ANNs can suffer from multiple local minima, *SVM* does not. *SVMs* have simple geometric interpretation and give a sparse solution. Nonlinear *SVM* based approaches use kernel functions to make a nonlinearly separable problem, a separable one (Fang et al., 2009). Unlike ANNs, the computational complexity of *SVMs* does not depend on the dimensionality of the input space. ANNs use empirical risk minimization, whilst *SVMs* use structural risk minimization.

3. Evaluation

The performance of the classifiers is accessed through the following standard parameters.

3.1. Accuracy

It is the percentage prediction of true examples namely, True prediction divided by the total number of examples. Let D is a dataset with N instances. Let Y_i and Z_i are the set of original and predicted labels, respectively, where $i \in D$, (Tsoumakas and Katakis, 2007). Then the overall accuracy is defined as

$$Accuracy = \frac{1}{N} \sum_{i=1}^{N} |Y_i \cap Z_i| / (Y_i \cup Z_i) \quad (9)$$

3.2. Sensitivity

It is the percentage of actual positives, which are predicted correctly

$$Sensitivity = \frac{TP}{TP+FN} \times 100 \quad (10)$$

3.3. Specificity

It is the percentage of actual negatives, which are predicted correctly

$$Specificity = \frac{TN}{FP+TN} \times 100 \quad (11)$$

3.4. Mathew's correlation coefficient (MCC)

It is considered to be one of the most rigorous performance parameters for any prediction methods. *MCC* takes values is the interval $[-1\ 1]$, whereby 1 means that the classifier predicts the entire correct one and -1 mean that the classifier predicts all incorrect one

$$MCC(i) = \frac{TP \times TN - FP \times FN}{\sqrt{[TP+FP][TP+FN][TN+FP][TN+FN]}} \quad (12)$$

The *MCC* recover the pitfall of accuracy on unbalance data. For example, if the number of positive examples is greater than the number of negative examples, then the classifier can easily predict all the examples as positive. Thus the performance of classifier will not be good because it predicts incorrectly all the negative examples. In this case, the accuracy and *MCC* of the positive examples is 100% and 0%, respectively. Therefore, the *MCC* is considered as the best performance parameter for the classification of unbalanced data.

3.5. F-measure

It is a statistical measure using for evaluating the performance of a classifier. It consists of precision *p* and recall *r*. *p* is the number of correct predictions divided by the number of all returned prediction and *r* is the number of correct predictions divided by the number of predictions. Best value of *F*-measure is one, while worst is 0

$$F\text{-}measure = 2 \times \frac{Precision \times Recall}{(Precision + Recall)} \quad (13)$$

$$Precision = \frac{TP}{TP+FP} \quad (14)$$

$$Recall = \frac{TP}{TP+FN} \quad (15)$$

where *TP*, *TN*, *FP*, and *FN* are true positive, true negative, false positive, and false negative, respectively.

The *F-measure* can be easily generalized for multiclass classification (Tsoumakas and Katakis, 2007). Then the Recall and Precision for label k are defined as

$$Recall_k = \sum_{\{i | i \in D \wedge k \in Y_i\}} |Y_i \cap Z_i| / |Y_i| \quad (16)$$

$$Precision_k = \sum_{\{i | i \in D \wedge k \in Z_i\}} |Y_i \cap Z_i| / |Z_i| \quad (17)$$

4. Result and discussion

In statistical prediction, the following three cross-validation methods are often used to examine a predictor for its effectiveness in practical application: independent dataset test, sub-sampling test, and jackknife test (Chou and Zhang, 1995). However, as elucidated in Chou and Shen (2010a) and demonstrated by Eq. (1) of Chou and Shen (2010a), among the three cross-validation methods, the jackknife test is deemed the most objective that can always yield a unique result for a given benchmark dataset, and hence has been increasingly used by investigators to examine the accuracy of various predictors (see, e.g., Chen et al., 2009; Chou and Shen, 2010b; Chou and Shen, 2010c; Ding et al., 2009; Lin, 2008; Mohabatkar, 2010; Vilar et al., 2009; Zeng et al., 2009; Zhou et al., 2007). Self-consistency is a first basic test, but the predictions of the classifier trained on self-consistency on the novel instances are not good. Jackknife test is also called *n* fold cross-validation, where *n* is the number of instances in the dataset. Each instance is in turn left out, and the learning method is trained on all the remaining instances. The results of all *n* judgments, one for each member of the dataset, are averaged and that average represents the final error estimate. When for training, one dataset is used and for testing another dataset then, it is called independent dataset test. Among these three tests, jackknife test is the most effective and objective one, because every instance is in turn considered as a novel sample, which increases the chance that the classifier will behave well on novel samples. Jackknife test is deterministic because no random sampling is involved.

4.1. Prediction performance using jackknife test

In Table 4, the predicted results of the classifiers using jackknife test are shown. Columns 2–6 show the accuracy, sensitivity, specificity, *MCC*, and the *F*-measure of all of the classifiers using jackknife test. The *SVM* yields the best performance using jackknife with an accuracy of 86.01%, as compared with the other state of the art neural networks. Its sensitivity, specificity, *MCC*, and *F*-measure are 85.1%, 85.8%, 0.63, and 0.71, respectively, which are higher than that of the other neural networks. *GRNN* yields relatively low accuracy of 78.14%. The main reason for the low performance using jackknife test is that it has low generalization capability compared to *SVM* and *PNN*. On the other hand, *PNN* achieves 82.51% accuracy which is better compared to *GRNN* but less than that of *SVM*. The accuracy, sensitivity, and specificity are shown in Fig. 3 while *MCC* and *F*-measure are shown in Fig. 4.

Table 4
Overall classification performance for jackknife and independent dataset tests.

Classifier	Jackknife test					Independent dataset test				
	Accuracy	Se	Sp	*MCC*	*F*-measure	Accuracy	Se	Sp	*MCC*	*F*-measure
SVM	**86.01**	85.15	85.82	0.63	0.71	95.23	93.28	95.51	0.85	0.88
PNN	82.51	82.46	82.09	0.56	0.65	**95.73**	96.22	95.41	0.87	0.89
GRNN	78.14	73.54	78.77	0.45	0.57	93.71	92.24	93.74	0.81	0.85

4.2. Prediction performance using independent dataset test

In Table 4, the prediction results in case of independent dataset test are shown. Columns 7–12 show the accuracy, sensitivity, specificity, *MCC*, and the *F*-measure for each of the classifiers using independent dataset test. *PNN* outperforms all the other classifiers in case of independent dataset test. It has obtained 95.73% accuracy, while it also performs well using the other parameters like sensitivity, specificity, *MCC*, and *F*-measure. Still the performance of GRNN is low as compared with that of *SVM* and *PNN*.

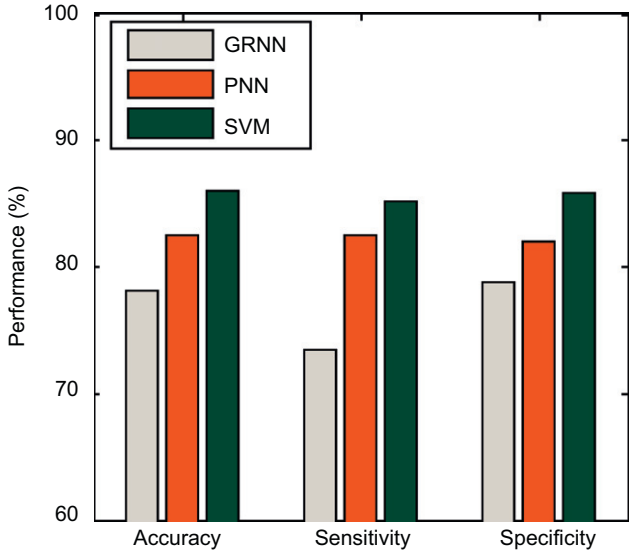

Fig. 3. Prediction performances of the different classifiers using Jackknife test.

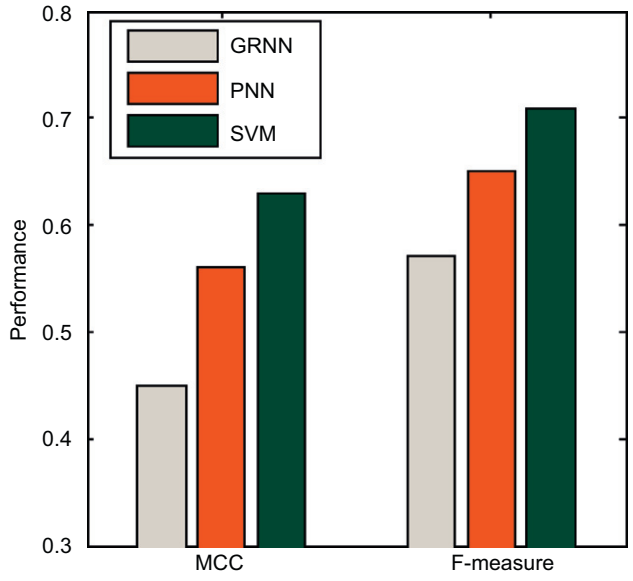

Fig. 4. *MCC* and *F*-measure of classifiers using jackknife test.

4.3. Prediction of each membrane protein types

In Table 5, columns 2–6 show the accuracy of each membrane protein type using jackknife test. *SVM* achieves highest prediction results in the case of type I and multipass transmembrane protein, which is 81.61% and 95.34%, respectively, due to large sample representations of these two classes in the data. The multipass transmembrane protein occupies maximum example (63.6%), therefore, it exhibits high prediction performance as compare to the other types. However, the performance of *SVM* is affected for type II, Lipid anchored membrane, and *GPI* anchored membrane protein because the number of examples of these three types is (15%) of the total examples. The obtained accuracy of each membrane protein type using jackknife test is shown in (Fig. 5). On the other hand, in Table 5, columns 7–12 show the same in case of independent dataset test. It is observed that in case of independent dataset test, the prediction performance of *PNN* on each membrane protein type is better as compared with *SVM* and *GRNN*. The prediction of each membrane protein type is useful for pharmaceutical industries, because the characteristics of these membrane protein types are investigated for drugs discovery in case of various diseases.

The proposed method is also compared to several state of the art existing methods as shown in Table 6. The advantage of the proposed method is that it uses seven feature vectors while the other method uses only one features vector. These feature vectors represents different properties of the amino acids. The high dimensionality, problem is however, tackled using kernel *PCA*. In the proposed method, the *SVM* obtains highest accuracy using self-consistency, and jackknife test, which is 99.9% and 86.01%, respectively. On the other hand, *PNN* yields the best performance, using self-consistency and independent dataset test, which is 99.9% and 95.73%, respectively. A similar improvement is observed in other performance measures also. We reckon that this effective performance

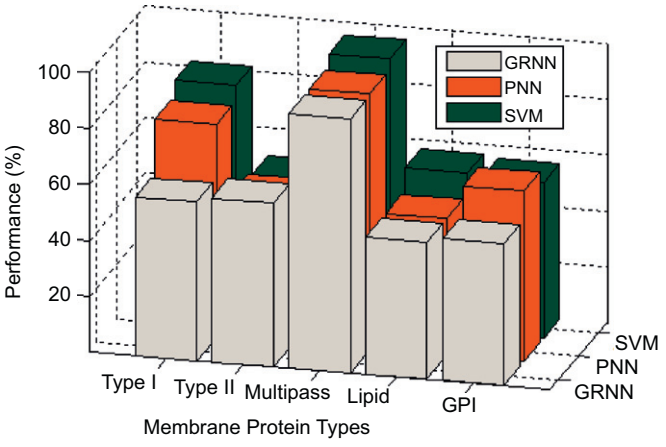

Fig. 5. Individual types of membrane protein using jackknife test.

Table 5
Classification performance for each type of membrane protein using jackknife and independent dataset test.

Classifier	Jackknife test					Independent dataset test				
	Type-I	Type-II	Multipass	Anchored	GPI	Type-I	Type-II	Multipass	Anchored	GPI
SVM	81.61	50.00	95.34	56.86	55.45	89.33	82.22	99.09	85.71	74.41
PNN	76.09	51.97	91.30	49.02	60.91	94.14	90.00	97.05	85.71	89.53
GRNN	56.78	58.55	90.92	49.02	50.91	85.77	89.44	96.83	71.42	82.55

Table 6
Performance comparison with existing approaches.

Methods	Self-consistency test (%)	Jackknife test (%)	Independent dataset test (%)
CDA (Chou and Elrod, 1999)	81.1	76.4	79.4
CDA and *PseAA* (Chou, 2001)	90.9	80.9	87.5
AA composition and *SVM* (Cai et al., 2004)	96.2	80.4	85.4
Low frequency Fourier spectrum (Liu et al., 2005)	99	78.0	87
Weighted u-*SVM* using *PseAA* (Wang et al., 2004)	99.8	82.3	90.3
PseAA and stacking (Wang et al., 2006)	98.7	85.4	94.3
Wavelet and cascade neural network (Rezaei et al., 2008)	96.8	81.3	91.4
Discrete wavelet and *SVM* (Qiu et al., 2010)	80.0	78.1	–
Dipeptide composition, *NPE* and *KNN* (Wang et al., 2010)	–	82.0	90.1
Proposed *CPSR* and *GRNN*	**99.9**	78.1	93.7
Proposed *CPSR* and *PNN*	**99.9**	82.5	**95.7**
Proposed *CPSR* and *SVM*	**99.9**	**86.0**	95.2

improvement is largely due to the good discrimination capabilities of the *CPSR* and the learning capabilities of the *SVM* and *PNN*.

5. Conclusions

In this paper, different neural networks based classification algorithms in conjunction with *CPSR* based feature extraction strategy are used for the prediction of membrane protein types. *PCA* is used to reduce the dimensionality of the obtained *CPSR* feature vector. Various statistical tests are used to evaluate the performance of these classifiers. Among the different classifiers, *SVM* performs the best using jackknife, while *PNN* does the same using independent dataset test. The *SVM* success rates obtained using self-consistency, jackknife, and independent dataset test are 99.9%, 86.01%, and 95.23%, respectively, while that of *PNN* are 99.9%, 82.51%, and 95.73%, respectively. These are the best prediction results reported so far and thus show the effectiveness of neural networks based classification strategies using *CPSR* based feature extraction for membrane protein type prediction.

Acknowledgment

This work is supported by the Higher Education Commission of Pakistan under the indigenous Ph.D. scholarship program 17-5-3 (Eg3-045)/HEC/Sch/2006).

Appendix A. Supplementary Material

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.jtbi.2010.11.017.

References

Cao, L., 2003. Support vector machines experts for time series forecasting. Neurocomputing. 51, 321–339.
Cai, Y.D., Liu, X.J., Xu, X.B., Chou, K.C., 2002a. Support vector machines for predicting the specificity of GalNAc-transferase. Peptides 23, 205–208.
Cai, Y.D., Liu, X.J., Xu, X.B., Chou, K.C., 2002b. Support vector machines for predicting HIV protease cleavage sites in protein. J. Comput. Chem. 23, 267–274.
Cai, Y.D., Liu, X.J., Xu, X.B., Chou, K.C., 2002c. Support vector machines for the classification and prediction of beta-turn types. J. Pept. Sci. 8, 297–301.
Cai, Y.D., Zhou, G.P., Chou, K.C., 2003a. Support vector machines for predicting membrane protein types by using functional domain composition. Biophys. J. 84, 3257–3263.
Cai, Y.D., Lin, S., Chou, K.C., 2003b. Support vector machines for prediction of protein signal sequences and their cleavage sites. Peptides 24, 159–161.
Cai, Y.D., Feng, K.Y., Li, Y.X., Chou, K.C., 2003c. Support vector machine for predicting alpha-turn types. Peptides 24, 629–630.
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.
Chen, C., Chen, L., Zou, X., Cai, P., 2009. Prediction of protein secondary structure content by using the concept of Chou's pseudo amino acid composition and support vector machine. Protein Pept. Lett. 16, 27–31.
Chen, Y., Isaac, G.C., 2003. An introduction to support vector machine overview. Al Magzine 24, 1–2.
Chou, K.C., Zhang, C.T., 1995. Review: prediction of protein structural classes. Crit. Rev. Biochem. Mol. Biol. 30, 275–349.
Chou, K.C., Elrod, D.W., 1999. Prediction of membrane protein types and subcellualar location. Proteins: Struct. Funct. Genet. 34, 137–153.
Chou, K.C., 2001. Prediction of protein subcellular attributes using pseudo-amino acid composition, proteins: structure, function. Genetics 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., 2005. Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes. Bioinformatics 21, 10–19.
Chou, K.C., Shen, H.B., 2007a. MemType-2 L: a web server for predicting membrane proteins and their types by incorporating evolution information through Pse-PSSM. Biochem. Biophys. Res. Commun. 360, 339–345.
Chou, K.C., Shen, H.B., 2009. Review: recent advances in developing web-servers for predicting protein attributes. Nat. Sci. 2, 63–92 openly accessible at 〈http://www.scirp.org/journal/NS/〉.
Chou, K.C., 2009. Pseudo amino acid composition and its applications in bioinformatics, proteomics and system biology. Curr. Proteomics 6, 262–274.
Chou, K.C., Shen, H.B., 2010a. Cell-PLoc 2.0: an improved package of web-servers for predicting subcellular localization of proteins in various organisms. Nat. Sci. 2, 1090–1103 openly accessible at 〈http://www.scirp.org/journal/NS/〉.
Chou, K.C., Shen, H.B., 2010b. A new method for predicting the subcellular localization of eukaryotic proteins with both single and multiple sites: Euk-mPLoc 2.0. PLoS ONE 5, e9931.
Chou, K.C., Shen, H.B., 2010c. Plant-mPLoc: a top-down strategy to augment the power for predicting plant protein subcellular localization. PLoS ONE 5, e11335.
Dayhoff, M.O., Schwartz, R.M., Orcutt, B.C., 1978. A model of evolutionay change in proteins. Atlas Protein Sequence Struct. 5, 345–352.
Devroye, L.P., Györfi, L., 1985. Nonparametric Density Estimation: The L1 View. Wiley, New York.
Ding, Y.S., Zhang, T.L., Chou, K.C., 2007. Prediction of protein structure classes with pseudo amino acid composition and fuzzy support vector machine network. Protein Pept. Lett. 14, 811–815.
Ding, H., Luo, L., Lin, H., 2009. Prediction of cell wall lytic enzymes using Chou's amphiphilic pseudo amino acid composition. Protein Pept. Lett. 16, 351–355.
Duda, R.O., Hart, P.E., Stock, D.G., 2001. Pattern Classification. John Wileys Sons, New York.
Eghbal, G.M., Mansoor, J.Z., Seraj, D.K., 2009. Protien superfamily classification using fuzzy rule-based classifier. IEEE Trans. Nanobiosci. 8, 1–8.
Eisenberg, D., Schwarz, E., Komaromy, M., Wall, R., 1984. Analysis of membrane and surface protein sequences with the hydrophobic moment plot. J. Mol. Biol. 179, 125–142.
Fang, G., Tao, G., Zang, S., 2009. A research on bioinformatics prediction of protein subcellular localization. Curr. Bioinformactics 4, 177–182.
Golmohammadi, S.K., Kurgan, L., Crowley, B., Reformat, M., 2007. Classification of cell membrane proteins. Front. Convergence Biosci. Information Technol., 153–158.
Goulermas, J.Y., Liatsis, P., Zeng, X.J., 2008. Kernel regression networks with local structural information andcovariance volume adaptation. Neurocomputing. 72, 257–261.
Kedarisetti, K., Kuragan, L., Dick, S., 2006. Classifier ensemble for protien structural class prediction with varying homolgoy. Biochem. Biophys. Res. Commun. 348, 981–988.

Khan, A., Tahir, S.F., Majid, A., Choi, Tae-Sun., 2008a. Machine learning based adaptive watermark decoding in view of an anticipated attack. Pattern Recognition 41, 2594–2610.
Khan, A., Tahir, S.F., Choi, Tae-Sun, 2008b. Intelligent extraction of a digital watermark from a distorted image. IEICE Transactions on Information Systems E91-D (7), 2072.
Khan, A., Majid, A., Choi, T.S., 2010. Predicting protein subcellular location: exploiting amino acid based sequence of feature spaces and fusion of diverse classifiers. Amino Acids 38, 347–350.
Liu, H., Wang, M., Chou, K.C., 2005. Low-frequency Fourier spectrum for predicting membrane protein types. Biochem. Biophys. Res. Commun. 336, 737–739.
Lin, H., 2008. The modified Mahalanobis discriminant for predicting outer membrane proteins by using Chou's pseudo amino acid composition. J. Theor. Biol. 252, 350–356.
Mohabatkar, H., 2010. Prediction of cyclin proteins using Chou's pseudo amino acid composition. Protein Pept. Lett. 17, 1207–1214.
Qiu, J.D., Sun, X.U., Huang, J.H., Liang, R.P., 2010. Prediction of the types of membrane proteins based on discrete wavelet transform and support vector machines. J. Protien 29, 114–119.
Rezaei, M.A., Maleki, P.A., Karami, Z., Asadabadi, E.B., Sherafat, M.A., Moghaddam, K.A., Fadaie, M., Forouzanfar, M., 2008. Prediction of membrane protein types by means of wavelet analysis and cascaded neural network. J. Theor. Biol. 255, 817–820.
Sonnhammer, E.L.L., Heijne, G.V., Krogh, A., 1998. A hidden Markov model for predicting transemembrane helices in protein sequences. In: Proceedings of Sixth International Conference on Intelligent Systems for Molecular Biology, AAAI/MIT Press, Menlo Park, CA, vol. 6, pp. 175–182.
Specht, D.F., 1990. Probabilisitc neural networks. Neural Network 3, 109–118.
Tsoumakas, G., Katakis, I., 2007. Multi-label classification: an overwiew. Int. J. Data Warehousing. Min. 3, 1–13.
Vilar, S., Gonzalez-Diaz, H., Santana, L., Uriarte, E., 2009. A network-QSAR model for prediction of genetic-component biomarkers in human colorectal cancer. J. Theor. Biol. 261, 449–458.
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. Sel. 17, 509–516.
Wang, S.Q., Yang, J., Chou, K.C., 2006. Using stacking generalization to preict membraen protien types based on pseudo-amino aid. J. Theor. Biol. 242, 941–946.
Wang, L., Yuan, Z., Chen, X., Zhou, Z., 2010. The prediction of membrane protein types with NPE. IEICE Electron. Express vol. 7 (No. 6), 397–402.
Waugh, D.F., 1954. Protien–Protein interactions. Adv. Protein Chem. 9, 325–437.
Wu, C.H., Berry, M., Fung, Y.S., Mclarty, J., 1995. Neural networks for full-scale protein sequence classification: sequence encoding with singular value decomposition. Mach. Learn. 21, 177–193.
Wu, C., Whitson, G., Mclarty, J., Ermongkonchai, A., Chang, T.C., 1992. Protein classification artificail neual system. Protein Sci. 1, 667–677.
Zhang, L.I., Lioa, B.O., Li, D., Zhu, W., 2009. A novel representation for apoptosis protein Subcellular localization prediction using support vector machine. J. Theor. Biol. 259, 361–365.
Zeng, Y.H., Guo, Y.Z., Xiao, R.Q., Yang, L., Yu, L.Z., Li, M.L., 2009. Using the augmented Chou's pseudo amino acid composition for predicting protein submitochondria locations based on auto covariance approach. J. Theor. Biol. 259, 366–372.
Zhou, X.B., Chen, C., Li, Z.C., Zou, X.Y., 2007. Using Chou's amphiphilic pseudo-amino acid composition and support vector machine for prediction of enzyme subfamily classes. J. Theor. Biol. 248, 546–551.
Zumdahi, S., 2000. Chemistry5th edition Houghton Mifflin Company.