Hybrid SVM-ANFIS for protein subcellular location prediction

Bo Jin*, Yuchun Tang and Yan-Qing Zhang

Department of Computer Science,
Georgia State University,
Atlanta, GA 30302-3994, USA
E-mail: jinbo@musc.edu
E-mail: yuchun_tang@securecomputing.com
E-mail: yzhang@cs.gsu.edu
*Corresponding author

Abstract: Predicting protein subcellular locations may help us understand protein functions and analyse protein interactions with other molecules. Many machine learning and computational techniques have been used to predict protein subcellular locations. In this paper, we propose a new hybrid classification system called SVM-ANFIS based on Support Vector Machines and Adaptive Neuro Fuzzy Inference System for protein subcellular location prediction. The experimental results show that the new system can not only achieve high total accuracies but also improve local accuracies in protein subcellular location prediction.

Keywords: SVMs; support vector machines; ANFIS; adaptive neuro fuzzy inference system; protein subcellular location; bioinformatics.

Reference to this paper should be made as follows: Jin, B., Tang, Y. and Zhang, Y-Q. (2009) 'Hybrid SVM-ANFIS for protein subcellular location prediction', *Int. J. Computational Intelligence in Bioinformatics and Systems Biology*, Vol. 1, No. 1, pp.59–73.

Biographical notes: Bo Jin is a research associate in the Department of Biostatistics, Bioinformatics and Epidemiology at Medical University of South Carolina. He received his PhD Degree in Computer Science at Georgia State University in 2007. His research interests involve data mining, text mining, machine learning, statistical learning, information theory, granular computing, evolutionary computation, fuzzy logic, bioinformatics and medical informatics. His current research focuses on developing statistical learning algorithms for literature-based gene annotation.

Yuchun Tang is currently a principal research scientist in McAfee Inc., Alpharetta, Georgia. He received the PhD Degree in Computer Science from Georgia State University, Atlanta, Georgia, USA, in 2006. His research interests include knowledge discovery and data mining, machine learning, statistical learning, computational intelligence, soft computing, granular computing, text mining, artificial intelligence, intelligent data analysis and decision support systems. He has been applying the above techniques in many domains including bioinformatics, medical informatics, computational biology, Web information retrieval and information extraction, spam Email filtering, Web security, and malware detection, business intelligence, etc.

Yan-Qing Zhang is currently an Associated Professor of the Computer Science Department at Georgia State University, Atlanta. His research interests include

computational intelligence, granular computing, data mining, bioinformatics, and computational Web intelligence. He has co-authored two books and co-edited two books. He published 12 book chapters, about 60 journal papers and 120 conference papers. He was a guest co-editor for Special Section on *Computational Intelligence Approaches in Computational Biology* and *Bioinformatics of IEEE/ACM Transactions on Computational Biology and Bioinformatics*, April–June, 2007. He was Program co-chair and Bioinformatics Track Chair of *IEEE 7th International Conference on Bioinformatics & Bioengineering* (IEEE BIBE 2007).

1 Introduction

Protein subcellular location plays an important role in the function of protein. The prediction of protein subcellular locations may help us not only understand protein functions but also protein interactions with other molecules. Many machine learning methods and computational techniques (Chou, 2000; Feng, 2002) were used to predict the protein subcellular locations. Nakai and Kanehisa (1992) presented an expert system to predict protein subcellular locations by using if-then rules and sequence motifs. The system can achieve prediction accuracies of 66% in training and 59% in testing. Reinhardt and Hubbard (1998) used neural networks to predict the subcellular locations of proteins and the accuracy can reach 81% in predicting three possible subcellular locations for prokaryotic proteins and 66% for four locations for eukaryotic proteins. Also based on the amino acid composition, Yuan (1999) presented Markov chain models for protein subcellular location prediction. For prokaryotic proteins, Yuan's method can achieve a prediction accuracy of 89.1% for three subcellular locations. For eukaryotic proteins, the prediction accuracies can reach 73.0% and 78.7% within four and three location categories respectively. Huang and Li (2004) introduced another method to predict protein subcellular locations from their dipeptide composition by using a fuzzy *k*-nearest neighbours (*k*-NN) algorithm, and its overall predictive accuracy is about 80% in the jack-knife test.

Besides the methods and techniques mentioned above, Support Vector Machines (SVMs) (Boser et al., 1992; Vapnik, 1998) were also used to predict protein subcellular locations. Hua and Sun (2001) first employed SVMs with the one-vs.-rest approach for protein subcellular location prediction. They built k binary SVM models for k protein subcellular location classes separately in the training. The ith SVM model is built with all samples in the ith class with positive labels and all other samples with negative labels. Once an unknown sample needs to be classified, it is first predicted by these SVM models, and then classified into the class corresponding to the SVM model with the highest output value. The data features were generated based on 20 single amino acid compositions only. In the jack-knife test, Hua and Sun's method achieved the overall prediction accuracy of 79.4% on the eukaryotic sequences and 91.4% on the prokaryotic sequences. Later, Cai et al. (2002) presented a SVMs based prediction method by incorporating the quasi-sequence-order effect and used both the amino acid composition and the sequence-order-coupling numbers to improve prediction. Their experimental data set includes 2191 protein sequences belonging to 12 groups, and the prediction accuracy was 75% by using the jack-knife test.

Park and Kanehisa (2003) also presented a SVMs based method with the one-vs.-rest voting approach. The main difference between this method and Hua and Sun's method is that besides the amino acid composition, the amino acid pair and gapped amino acid pair compositions (Jang et al., 1996) were used in the prediction. More than 7000 protein sequences were collected for the 12 subcellular locations groups. Park and Kanehisa tried to balance Total Accuracy (TA) and Local Accuracy (LA). Balance between TA and LA is an important issue in multi-class data classification, especially when some class groups are very small and others are very large. In general, TA is mainly affected by large class groups, while LA is calculated based on all groups equally. SVMs are Structural Risk Minimisation (SRM) based learning algorithms, which try to obtain the generalisation capabilities from the learning and some training errors are allowed within a limited range. In the case that SVMs are trained on the data sets heavily unbalanced among multi-class groups, the model's decision usually benefits the large class group. To reduce such negative effect, one way is to find out suitable weight parameters for each class, which could make SVMs' decision bias to the small training data set, but tuning SVM parameters is really time-consuming. The other way is to train several SVMs groups on the data sets with different kinds of features, and then use a scheme to fuse the outputs of SVMs, such as the method proposed in Park and Kanehisa (2003). In this paper, we propose a new hybrid classification system called SVM-ANFIS to improve LA while still keeping a good TA. In the system, Adaptive Neuro Fuzzy Inference System (ANFIS) is employed to learn the relationship between the data expected class and their related SVM outputs based on the amino acid and different gapped amino acid pair compositions. In the test phase, a voting scheme is preformed based on the ANFIS's outputs. The experimental results show that SVM-ANFIS can make a better balance between TA and LA.

The rest of this paper is organised as follows. Section 2 describes the basic SVM classification theories. Section 3 reviews fuzzy inference system and ANFIS. In Section 4, SVM-ANFIS is presented. Section 5 shows the experiments and results. Finally, Section 6 gives the conclusion.

2 Support Vector Machines

In this section, we introduce SVMs for data classification. For a binary classification problem, given a training data set $\{(\vec{x}_1, y_1), \ldots, (\vec{x}_l, y_l)\}$, where $\vec{x}_i \in R^n$, $i = 1, \ldots, l$ are training example vectors with *n* dimensions and $y_i \in \{-1, +1\}$, $i = 1, \ldots, l$ is the class label of $\vec{x}_i$. SVMs try to find an optimal hyperplane,

$$\langle \vec{w}, \vec{x} \rangle + b = 0 \tag{1}$$

where $\vec{w} \in R^n$ is constructed using some training example vectors and $b \in R$. The hyperplane separates the training examples with the maximum margin under the condition of

$$y_i(\langle \vec{w}, \vec{x}_i \rangle + b) \geq 1. \tag{2}$$

The decision function is

$$f(\vec{x}) = \text{sgn}(\langle \vec{w}, \vec{x} \rangle + b). \tag{3}$$

In practice, the optimal hyperplane is calculated by solving the following constrained optimisation problem,

Minimise

$$\frac{1}{2}\| \vec{w} \|^2 + C\sum_i \xi_i. \tag{4}$$

Subject to

$$y_i(\langle \vec{w}, \vec{x}_i \rangle + b) \geq 1 - \xi_i \tag{5}$$

where ξ_i are nonnegative slack variables and C is the regulation parameter. The problem described by equation (4) and Condition (5) is a QP problem, which can be transformed to minimise the following primal Lagrange,

$$L = \frac{1}{2}\| \vec{w} \|^2 - \sum_i \alpha_i (y_i(\langle \vec{w}, \vec{x}_i \rangle + b) - 1) \tag{6}$$

where α_i are Lagrange multipliers and $\alpha_i \geq 0$. According to primal-dual formulation, it is equivalent to solve the following problem

Maximise

$$\sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \langle \vec{x}_i, \vec{x}_j \rangle \tag{7}$$

Subject to

$$\sum_i \alpha_i y_i = 0 \tag{8}$$

where $0 \leq \alpha_i \leq C$, $i, j = 1, \ldots, l$ and those $\vec{x}_i$ with $\alpha_i \neq 0$ are called support vectors. The decision function becomes

$$f(\vec{x}) = \text{sgn}(\langle \vec{w}, \vec{x} \rangle + b) = \text{sgn}\left(\sum_i \alpha_i y_i \left\langle \vec{x}_i, \vec{x} \right\rangle + b \right) \tag{9}$$

where $\vec{w} = \sum \alpha_i y_i \vec{x}_i$. If using kernel functions $K(\cdot,\cdot)$ instead of the inner products $\langle \cdot, \cdot \rangle$, the problem becomes

Maximise

$$\sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j K(\vec{x}_i, \vec{x}_j) \tag{10}$$

and the new decision function is

$$f(\vec{x}) = \text{sgn}\left(\sum_j \alpha_i y_i K(\vec{x}, \vec{x}_i) + b \right). \tag{11}$$

The followings are two nonlinear kernel functions commonly used in SVMs,

Polynomial function

$$K(\vec{x},\vec{y}) = (\omega\vec{x} \bullet \vec{y} + 1)^d \tag{12}$$

RBF

$$K(\vec{x},\vec{y}) = \exp(-\gamma \| \vec{x} - \vec{y} \|^2). \tag{13}$$

3 Adaptive Neuro Fuzzy Inference System

In this section, we review Fuzzy Inference System (FIS) and Adaptive Neuro Fuzzy Inference System (ANFIS).

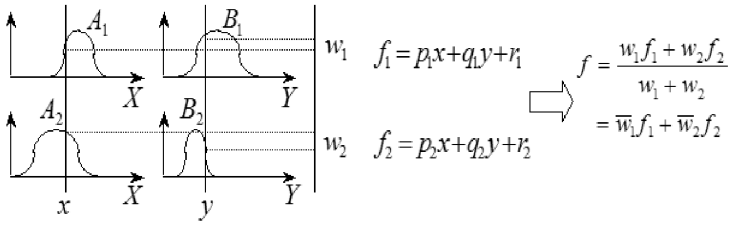
3.1 Fuzzy Inference System

A FIS (Jang et al., 1996) is also known as a fuzzy rule based system, which basically consists of three components: a rule base, which contains a set of fuzzy if-then rules; a database, which defines membership functions for fuzzy rules; and a reasoning mechanism, which performs the inference procedure and derive outputs based on the fuzzy rules. Figure 1(a) illustrates the fuzzy reasoning mechanism of the Surgeno type FIS with the following two if-then rules. Each rule's firing strength is usually obtained through product or min operations on the inputs' membership values in the premise part. The overall output can be chosen as the weighted average.

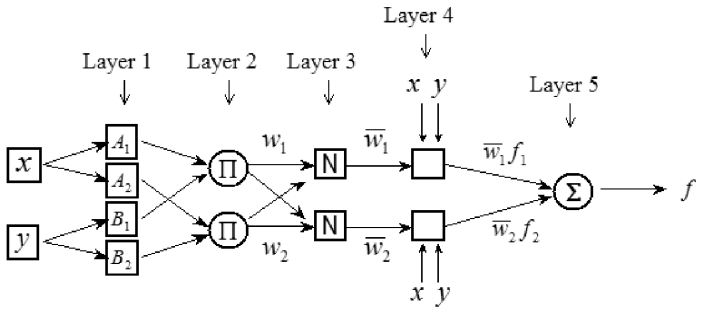
Rule 1: If x is A_1 and y is B_1, then $f_1 = p_1x + q_1y + r_1$.

Rule 2: If x is A_2 and y is B_2, then $f_2 = p_2x + q_2y + r_2$.

Figure 1 First-order Sugeno fuzzy model and the equivalent ANFIS architecture: (a) first-order sugeno fuzzy model and (b) equivalent ANFIS architecture



(a)



(b)

3.2 Adaptive Neuro Fuzzy Inference System

ANFIS (Jang, 1993; Lin and Lee, 1991; Wang and Mendel, 1992) is a kind of adaptive fuzzy inference system, which employs a hybrid-learning algorithm to identify fuzzy system parameters automatically. In ANFIS, the back-propagation gradient descent is employed to update the premise parameters for the membership functions and the least square method is used to identify consequent parameters for each rule's output. Figure 1(b) shows an ANFIS architecture example, which is equivalent to the fuzzy model in Figure 1(a). In Figure 1(b), nodes in Layer 1 are adaptive nodes and the membership functions are adjusted in this layer through the parameters, which are referred to as premise parameters. Nodes in Layer 2 are fixed and each output of them is the firing strength of a rule. In Layer 3, each node is also fixed and the ratio of each rule's firing strength to the sum of all rules' firing strengths is calculated by each related node. Layer 3's outputs are normalised weights. The product of each rule's output and the related weight are derived in Layer 4. Parameters in this layer are called consequent parameters. Finally the overall output is generated in Layer 5 as the sum of all incoming inputs.

4 SVM-ANFIS

In this section, we present the SVM-ANFIS classification system.

4.1 Binary SVM-ANFIS architecture

We use an example to explain SVM-ANFIS architecture. Figure 2 illustrates a binary SVM-ANFIS architecture with two SVMs and two fuzzy if-then rules.

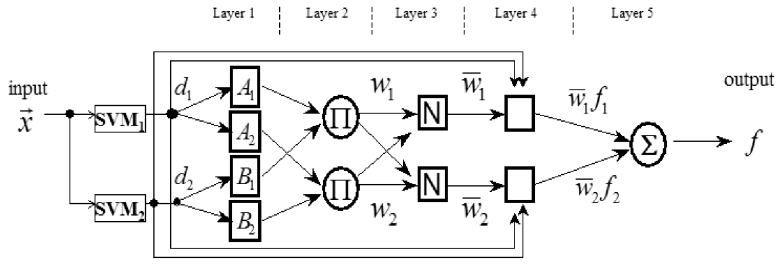
Rule 1: If d_1 is A_1 and d_2 is B_1, then $f_1 = p_1 d_1 + q_1 d_2 + r_1$.

Rule 2: If d_1 is A_2 and d_2 is B_2, then $f_2 = p_2 d_1 + q_2 d_2 + r_2$.

Where d_i is the input $\vec{x}$'s directional distance value from SVM$_j$'s decision hyperplane, which is calculated by using equation (14).

$$d_j = \sum_i \alpha_i y_i K(\vec{x}, \vec{x}_i) + b. \tag{14}$$

Figure 2 Binary SVM-ANFIS classification model



The followings are functions used in each layer.

$$O_{1,i} = \mu_{A_i}(d_1), \quad i = 1, 2 \tag{15}$$

$$O_{1,i} = \mu_{B_i}(d_2), \quad i = 1, 2 \tag{16}$$

where A_i (or B_i) is the linguistic label such as 'small' or 'large' and μ is the membership function for fuzzy set $A = \{A_i, B_i\}$. Here we use $O_{k,i}$ to represent the ith node's output in Layer k. In the experiment, we use the generalise bell function as the initial membership function.

$$\mu_A(d) = \frac{1}{1 + \left| \frac{d - c_i}{a_i} \right|^{2b_i}} \tag{17}$$

where a_i, b_i and c_i are parameters.

$$O_{2,i} = w_i = \mu_{A_i}(d_1)\mu_{B_i}(d_2), \quad i = 1, 2 \tag{18}$$

$$O_{3,i} = \overline{w}_i = \left(\frac{w_i}{w_1 + w_2} \right), \quad i = 1, 2 \tag{19}$$

$$O_{4,i} = \overline{w}_i f_i = \overline{w}_i (p_i d_1 + q_i d_2 + r_i), \quad i = 1, 2 \tag{20}$$

$$O_{5,1} = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}. \tag{21}$$

The final f function is

$$f = \frac{\sum_i w_i \left(p_i \left(\sum_j \alpha_j^1 y_j^1 K^1(\vec{x}, \vec{x}_j^1) + b^1 \right) + q_i \left(\sum_j \alpha_j^2 y_j^2 K^2(\vec{x}, \vec{x}_j^2) + b^2 \right) + r_i \right)}{\sum_i w_i} \tag{22}$$

where $\sum_j \alpha_j^l y_j^l \vec{x}_j^l (= w_j^l)$ and b^l are the hyperplane's weight and bias of SVM$_l$. For a binary classification problem, the final decision is made according to the sign of f.

In the training phase, the training data are split into two sub sets. One data set is used for SVMs training and the other is for SVMs evaluation. The evaluation results are used to train ANFIS model.

4.2 Voting scheme for multi-class classification

For multi-class classification problem, the final decision can be made by voting among binary SVM-ANFIS models (which were trained for each class separately). We define equation (23) and consider two cases in voting.

$$\delta = |f - 1|. \tag{23}$$

Case 1: Some fs have positive values. Voting is only made among the set of binary SVM-ANFIS models with positive fs values. One model's decision will be adopted if its related δ has the smallest value.

Case 2: Every binary SVM-ANFIS model's f has negative values. Voting is made among all binary SVM-ANFIS models. One model's decision will be adopted if its related δ has the smallest value.

5 Experiments and results

5.1 Data processing

The protein subcellular location data set (Park and Kanehisa, 2003) used in the experiment contains 7579 entries. Each protein sequence is transformed to five vectors respectively based on the amino acid composition, amino acid pair, 1 gapped amino acid pair, 2 gapped amino acid pair, and 3 gapped amino acid pair compositions respectively. For each sequence, its vector's features are all scaled into the range of 0~1 based on the sequence length. The vectors' feature number is 20 if the sequence is transformed based on the amino acid composition and 400 if it is based on the (gapped) amino acid pair compositions. For an amino acid pair composition, two amino χ and β are counted together as one unit $\chi\beta$ if χ and β happen continuously in one sequence. $\chi\beta$ and $\beta\chi$ are two different pairs if χ and β are different. The gapped amino acid pair composition means some number of intervening residues can exist in the pair. Table 1 lists the number of entries of each subcellular location in a decrease order.

Table 1 Subcellular location and number of entries

Subcelluar location	*No. of entries*
Nuclear	1932
Plasma membrane	1674
Cytoplasmic	1241
Extracellular	861
Mitochondrial	727
Chloroplast	671
Peroxisomal	125
Endoplasmic reticulum	114
Lysosomal	93
Vacuolar	54
Golgi apparatus	47
Cytoskeleton	40
Total	*7579*

5.2 5-fold cross-validation test

The system's prediction performance is tested using the 5-fold cross validation method. Each subcellular location's data set is divided into five subsets equally. We use S_i to represent the subcellular location data set i (listed in Table 1) and S_{ij} to represent the subset j of S_i, where $i = 1, \ldots, 1, 2$ and $j = 1, \ldots, 5$. The first test procedure of 5-fold cross validation is shown in Table 2. The other four tests' procedures are similar to the first one. Table 3 lists data sets assignment for 5-fold cross validation. Here sum (Σ) and plus (+) operations represent union operation on the data sets. We build 12 SVM-ANFIS

models, each of which has five SVMs and one ANFIS model. The prediction accuracies are evaluated using

$$\mathrm{TA} = \frac{\sum_{i=1}^{12} T_i}{N} \tag{24}$$

$$\mathrm{LA} = \frac{\sum_{i=1}^{12} \frac{T_i}{|S_i|}}{N} \tag{25}$$

where N is the total number of protein sequences (7579), T_i is the number of correctly predicted positive sequences in the subcellular location set S_i and $|S_i|$ is the number of sequences in S_i.

In the experiments, SVM part is built on Joachims' SVMlight 4.0 (Joachims, 1999) and most tests are preformed on SVM with RBF kernel, since generally SVM shows better performance with RBF kernel than with linear and polynomial kernels. ANFIS part is set up using the Fuzzy Toolbox of Matlab and the initial parameter is configured as the following. The initial Membership Function (MF) type is the generalise bell function type and two MFs for each input. The output type is set as linear type. Training epoch number is 10, training error goal is 0, the initial step size is 0.1, the step size decrease rate is 0.9 and the step size increase size is 1.1. The experiments are performed on two PCs with P4 2.0G and P4 2.8G processors each and each machine's memory size is 256M. The running time is about 12 hours for one 5-fold cross validation on all data sets.

Table 2 The first test procedure in 5-fold validation

For i = 1, …, 12

Label +1 on S_i's data

Label –1 on all S_k, where k = 1, …, 12 and $k \neq i$

Create the SVMs training set $T_{svm} = \sum_{l=1}^{12} \sum_{j=1}^{3} S_{lj}$

Create the SVMs evaluation set $E_{svm} = \sum_{l=1}^{12} \sum_{j=1}^{4} S_{lj}$

Create the SVM-ANFIS test set $SA = \sum_{l=1}^{12} S_{l5}$

For each of five kinds compositions

Train SVMs on T_{svm}

Evaluate SVMs on E_{svm}

For end

Train ANFIS with each entry's five evaluate results

Test SVM-ANFIS on SA

For end

Classify each SA entry according to the voting results among the 12 tests above.

Table 3 Data sets assignment

	Test 1	*Test 2*	*Test 3*	*Test 4*	*Test 5*
T_{svm}	$\sum_{l=1}^{12}\sum_{j=1}^{3} S_{lj}$	$\sum_{l=1}^{12}\sum_{j=2}^{4} S_{lj}$	$\sum_{l=1}^{12}\sum_{j=3}^{5} S_{lj}$	$\sum_{l=1}^{12}\sum_{j=4}^{5} S_{lj} + \sum_{l=1}^{12} S_{l1}$	$\sum_{l=1}^{12}\sum_{j=1}^{2} S_{lj} + \sum_{l=1}^{12} S_{l5}$
E_{svm}	$\sum_{l=1}^{12}\sum_{j=1}^{4} S_{lj}$	$\sum_{l=1}^{12}\sum_{j=2}^{5} S_{lj}$	$\sum_{l=1}^{12}\sum_{j=3}^{5} S_{lj} + \sum_{l=1}^{12} S_{l1}$	$\sum_{l=1}^{12}\sum_{j=4}^{5} S_{lj} + \sum_{l=1}^{12}\sum_{j=1}^{2} S_{lj}$	$\sum_{l=1}^{12}\sum_{j=1}^{3} S_{lj} + \sum_{l=1}^{12} S_{l5}$
SA	$\sum_{l=1}^{12} S_{l5}$	$\sum_{l=1}^{12} S_{l1}$	$\sum_{l=1}^{12} S_{l2}$	$\sum_{l=1}^{12} S_{l3}$	$\sum_{l=1}^{12} S_{l4}$

5.3 Experimental results

In the experiments we also implement the methods proposed in Hua and Sun (2001) and Park and Kanehisa (2003) and compared them to our method under the same conditions. We select RBF kernel and four groups of SVM parameters from the limited ranges (C = 10~30 and γ= 10~30). Park and Kanehisa gave two groups of SVM parameters (γ= 0.02 and γ= 0.03) and C is calculated by equation $C = N / \sum_{i=1}^{N} K(x_i, x_i)$, where N is the size of the training set). With these parameters, their method did not show good performance (*TA*≈0.533and *LA*≈0.21) in the experiments. The experimental results are listed in the following tables (Tables 4–10). From Tables 4–7, we can find all three methods can achieve better accuracy performances with parameters C = 30 and γ= 30 (see Table 6).

Table 4 Comparisons of prediction accuracies with RBF kernel (C = 10, γ= 10)

Location	*Amino acid (Hua's method)*	*Amino acid pair*	*One gapped amino acid pair*	*Two gapped amino acid pair*	*Three gapped amino acid pair*	*Park's method*	*SVM-ANFIS*
Nuclear	0.864	0.857	0.851	0.862	0.858	0.898	0.835
Plasma membrane	0.917	0.934	0.944	0.941	0.938	0.947	0.938
Cytoplasmic	0.635	0.654	0.657	0.658	0.675	0.72	0.646
Extracellular	0.653	0.634	0.623	0.617	0.622	0.686	0.689
Mitochondrial	0.172	0.311	0.31	0.267	0.202	0.238	0.487
Chloroplast	0.334	0.423	0.358	0.385	0.394	0.391	0.581
Peroxisomal	0.064	0.088	0.112	0.128	0.08	0.072	0.296
ER	0.035	0.307	0.272	0.263	0.202	0.193	0.421
Lysosomal	0.108	0.269	0.398	0.366	0.365	0.312	0.559
Vacuolar	0	0.074	0.111	0.13	0.074	0.019	0.259
Golgi apparatus	0.021	0.021	0.064	0.064	0.021	0.021	0.149
Cytoskeleton	0.3	0.225	0.1	0.35	0.325	0.25	0.35
TA	*0.651*	*0.682*	*0.678*	*0.679*	*0.672*	*0.701*	*0.725*
LA	*0.342*	*0.4*	*0.4*	*0.419*	*0.396*	*0.396*	*0.518*

Table 5 Comparisons of prediction accuracies with RBF kernel ($C = 20$, $\gamma = 20$)

Location	*Amino acid (Hua's method)*	*Amino acid pair*	*One gapped amino acid pair*	*Two gapped amino acid pair*	*Three gapped amino acid pair*	*Park's method*	*SVM-ANFIS*
Nuclear	0.855	0.854	0.845	0.857	0.852	0.896	0.843
Plasma membrane	0.913	0.93	0.935	0.932	0.931	0.943	0.94
Cytoplasmic	0.631	0.652	0.656	0.645	0.667	0.72	0.676
Extracellular	0.727	0.69	0.706	0.7	0.713	0.777	0.738
Mitochondrial	0.245	0.415	0.409	0.354	0.316	0.358	0.543
Chloroplast	0.498	0.556	0.514	0.526	0.537	0.557	0.63
Peroxisomal	0.048	0.16	0.2	0.2	0.152	0.152	0.32
ER	0.246	0.465	0.456	0.465	0.404	0.421	0.491
Lysosomal	0.28	0.538	0.602	0.516	0.613	0.548	0.602
Vacuolar	0.093	0.278	0.204	0.167	0.185	0.167	0.315
Golgi apparatus	0.043	0.064	0.149	0.106	0.043	0.064	0.213
Cytoskeleton	0.475	0.575	0.55	0.575	0.65	0.575	0.625
TA	*0.684*	*0.719*	*0.717*	*0.711*	*0.712*	*0.747*	*0.752*
LA	*0.421*	*0.515*	*0.519*	*0.504*	*0.505*	*0.515*	*0.578*

Table 6 Comparisons of prediction accuracies with RBF kernel ($C = 30$, $\gamma = 30$)

Location	*Amino acid (Hua's method)*	*Amino acid pair*	*One gapped amino acid pair*	*Two gapped amino acid pair*	*Three gapped amino acid pair*	*Park's method*	*SVM-ANFIS*
Nuclear	0.839	0.856	0.848	0.847	0.851	0.897	0.858
Plasma membrane	0.904	0.927	0.933	0.929	0.93	0.944	0.94
Cytoplasmic	0.645	0.666	0.657	0.654	0.658	0.729	0.707
Extracellular	0.717	0.722	0.734	0.722	0.733	0.801	0.769
Mitochondrial	0.286	0.487	0.466	0.418	0.387	0.443	0.554
Chloroplast	0.593	0.621	0.598	0.608	0.62	0.671	0.678
Peroxisomal	0.128	0.208	0.24	0.264	0.208	0.2	0.328
ER	0.404	0.526	0.5	0.482	0.465	0.509	0.553
Lysosomal	0.506	0.548	0.602	0.57	0.645	0.613	0.656
Vacuolar	0.13	0.426	0.296	0.204	0.297	0.278	0.333
Golgi apparatus	0.064	0.213	0.276	0.213	0.17	0.149	0.255
Cytoskeleton	0.5	0.675	0.675	0.65	0.675	0.65	0.65
TA	*0.699*	*0.742*	*0.737*	*0.729*	*0.731*	*0.774*	*0.772*
LA	*0.476*	*0.573*	*0.569*	*0.547*	*0.553*	*0.574*	*0.607*

Table 7 Comparisons of prediction accuracies with RBF kernel ($C = 20$, $\gamma = 30$)

Location	*Amino acid (Hua's method)*	*Amino acid pair*	*One gapped amino acid pair*	*Two gapped amino acid pair*	*Three gapped amino acid pair*	*Park's method*	*SVM-ANFIS*
Nuclear	0.84	0.855	0.847	0.85	0.847	0.897	0.844
Plasma membrane	0.909	0.929	0.935	0.934	0.93	0.944	0.943
Cytoplasmic	0.641	0.663	0.662	0.656	0.665	0.722	0.697
Extracellular	0.724	0.719	0.734	0.72	0.735	0.792	0.757
Mitochondrial	0.281	0.462	0.443	0.405	0.364	0.415	0.564
Chloroplast	0.574	0.614	0.578	0.59	0.589	0.648	0.654
Peroxisomal	0.104	0.224	0.232	0.264	0.184	0.2	0.36
ER	0.316	0.518	0.474	0.491	0.456	0.474	0.526
Lysosomal	0.473	0.516	0.602	0.559	0.624	0.581	0.624
Vacuolar	0.13	0.389	0.259	0.167	0.222	0.222	0.352
Golgi apparatus	0.043	0.149	0.234	0.17	0.106	0.128	0.213
Cytoskeleton	0.5	0.6	0.675	0.65	0.65	0.65	0.6
TA	*0.696*	*0.737*	*0.733*	*0.727*	*0.724*	*0.766*	*0.764*
LA	*0.461*	*0.553*	*0.556*	*0.538*	*0.531*	*0.556*	*0.594*

Table 8 Prediction accuracies using parameter-mixture RBF kernels ($C = 20$, $\gamma_1 = 20$, $\gamma_2 = 30$)

Location	*Amino acid (Hua's method)*	*Amino Acid pair*	*One gapped amino acid pair*	*Two gapped amino acid pair*	*Three gapped amino acid pair*	*Park's method*	*SVM-ANFIS*
Nuclear	0.851	0.859	0.85	0.854	0.854	0.899	0.844
Plasma membrane	0.915	0.93	0.933	0.935	0.93	0.944	0.943
Cytoplasmic	0.65	0.677	0.679	0.667	0.682	0.734	0.697
Extracellular	0.735	0.727	0.746	0.733	0.748	0.799	0.757
Mitochondrial	0.259	0.407	0.395	0.344	0.301	0.354	0.564
Chloroplast	0.498	0.541	0.513	0.53	0.54	0.562	0.654
Peroxisomal	0.04	0.152	0.192	0.208	0.152	0.16	0.36
ER	0.272	0.474	0.465	0.483	0.43	0.447	0.526
Lysosomal	0.333	0.495	0.581	0.538	0.624	0.538	0.624
Vacuolar	0.093	0.259	0.204	0.167	0.185	0.111	0.352
Golgi apparatus	0.021	0.043	0.128	0.149	0.064	0.043	0.213
Cytoskeleton	0.475	0.6	0.575	0.575	0.675	0.575	0.6
TA	*0.69*	*0.726*	*0.725*	*0.719*	*0.718*	*0.752*	*0.764*
LA	*0.428*	*0.514*	*0.522*	*0.515*	*0.515*	*0.514*	*0.594*

Table 9 Range of accuracy fluctuations with RBF kernel in the experiment

	Amino acid (Hua's method)	*Amino acid pair*	*One gapped Amino acid pair*	*Two gapped amino acid pair*	*Three gapped amino acid pair*	*Park's method*	*SVM-ANFIS*
TA	0.651~0.699	0.682~0.742	0.678~0.737	0.679~0.729	0.672~0.731	0.701~0.774	0.725~0.772
LA	0.342~0.476	0.4~0.573	0.4~0.569	0.419~0.547	0.396~0.553	0.396~0.574	0.518~0.607

Table 10 Best results summary for each method

	Cai et al. (2002)		*Hua's (2001)*		*Park's (2003)*		*SVM-ANFIS*	
Location	*No. of entries (total 2191)*	*Jack-knife*	*No. of entries (total 7579)*	*5-fold cross*	*No. of entries (total 7579)*	*5-fold cross*	*No. of entries (total 7579)*	*5-fold cross*
Nuclear	272	0.73	1932	0.839	1932	0.897	1932	0.858
Plasma membrane	699	0.91	1674	0.904	1674	0.944	1674	0.94
Cytoplasmic	571	0.88	1241	0.645	1241	0.729	1241	0.707
Extracellular	224	0.57	861	0.717	861	0.801	861	0.769
Mitochondrial	84	0.42	727	0.286	727	0.443	727	0.554
Chloroplast	145	0.57	671	0.593	671	0.671	671	0.678
Peroxisomal	27	0.04	125	0.128	125	0.2	125	0.328
ER	49	0.31	114	0.404	114	0.509	114	0.553
Lysosomal	37	0.54	93	0.506	93	0.613	93	0.656
Vacuolar	24	0.25	54	0.13	54	0.278	54	0.333
Golgi apparatus	25	0.12	47	0.064	47	0.149	47	0.255
Cytoskeleton	34	0.44	40	0.5	40	0.65	40	0.65
TA		*0.75*		*0.699*		*0.774*		*0.772*
LA		*0.48*		*0.476*		*0.574*		*0.607*

As shown in Table 6, both Park and Kanehisa's and our method can improve TA from Hua's 0.699 to 0.77 approximately. In LA comparison, we can see Park and Kanehisa's method can improve LA from 0.476 (Hua's) to 0.574. Our method can improve LA to 0.607, which is higher than Park and Kanehisa's LA by about 0.03. Furthermore, from each single location accuracy, we can find TA and LA of Park and Kanehisa's are much more affected by the first four location's prediction accuracies, all of which have the large data size. In contract, SVM-ANFIS can do a better balance between the prediction accuracies of locations with large data size and small data size. Using our method, although the first four location's prediction accuracies are lower than those of Park and Kanehisa's respectively, the total accuracy is as high as that of Park and Kanehisa's. In addition, from Table 6, we find the prediction based on (gapped) amino acid pair composition is also better than that based on simple amino acid composition.

From Tables 4–5 and 7, we see all methods with other three groups of parameters cannot achieve prediction accuracies as high as those with $C = 30$ and $\gamma = 30$. The interesting is that SVM-ANFIS still keeps relative high LA, while LA of Park's is reduced greatly. Using RBF kernel with $C = 10$ and $\gamma = 10$, LA of ours is 0.518, while LA of Park's is only 0.396, which is even worse than those based on the amino acid pair,

one gapped amino acid pair and two gapped amino acid pair compositions. It means Park and Kanehisa's method is more sensitive to the parameters of SVM. The prediction accuracies of Park's method with mixture RBF kernel are also tested and listed in Table 8. We use $\gamma = 30$ for the four large groups, $\gamma = 20$ for the rest and C is set as 20. The result of Park and Kanehisa's is not as expected. The range of accuracy fluctuations is listed in Table 9 for four groups of parameters. We can see the performance of SVM-ANFIS is much more stable than any others. The best results of each method are summarised in Table 10 and it is shown that our method is also better than Cai's by about 0.13.

6 Conclusion

In this paper, we proposed the hybrid SVM-ANFIS classification system and used it to predict protein subcellular location. Comparing to other SVM based systems, SVM-ANFIS is more stable and can make better predictions on the unbalanced data set. The experimental results demonstrate that our method can effectively improve the local accuracies without much affecting the total accuracies in protein subcellular location prediction.

Acknowledgement

This work was supported in part by NIH under Grant P20 GM065762.

References

Boser, B., Guyon, I. and Vapnik, V.N. (1992) 'A training algorithm for optimal margin classifiers', *Proc. 5th Annual Workshop on Computational Learning Theory*, ACM Press, New York, pp.144–152.

Cai, Y-D., Liu, X-J., Xu, X-B. and Chou, K-C. (2002) 'Support vector machines for prediction of protein subcellular location by incorporating quasi-sequence-order effect', *Journal of Cellular Biochemistry*, Vol. 84, No. 2, pp.343–348.

Chou, K.C. (2000) 'Prediction of protein structural classes and subcellular locations', *Current Protein and Peptide Science*, Vol. 1, No. 2, pp.171–208.

Feng, Z-P. (2002) 'An overview on predicting the subcellular location of a protein', *Silico Biology*, Vol. 2, No. 3, pp.291–303.

Hua, S. and Sun, Z. (2001) 'Support vector machine approach for protein subcellular localization prediction', *Bioinformatics*, Vol. 17, No. 8, pp.721–728.

Huang, Y. and Li, Y. (2004) 'Prediction of protein subcellular locations using fuzzy k-NN method', *Bioinformatics*, Vol. 20, No. 1, pp.21–28.

Jang, J-S.R. (1993) 'ANFIS: adaptive-network-based fuzzy inference systems', *IEEE Transactions on Systems, Man, and Cybernetics*, May, Vol. 23, No. 3, pp.665–685.

Jang, J-S.R., Sun, C-T. and Mizutani, E. (1996) *Neuro Fuzzy And Soft Computing A Computational Approach To Learning and Machine Intelligence*, Prentice-Hall, NJ.

Joachims, T. (1999) 'Making large-scale SVM learning practical', *Advances in Kernel Methods: Support Vector Learning*, MIT Press, Cambridge, MA, USA, pp.169–184.

Lin, C-T. and Lee, C.S.G. (1991) 'Neural-network-based fuzzy logic control and decision system', *IEEE Transactions on Computers*, Vol. 40, No. 12, December, pp.1320–1336.

Nakai, K. and Kanehisa, M. (1992) 'A knowledge base for predicting protein localization sites in eukaryotic cells', *Genomics*, Vol. 14, No. 4, pp.897–911.

Park, K-J. and Kanehisa, M. (2003) 'Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs', *Bioinformatics*, Vol. 19, No. 13, pp.1656–1663.

Reinhardt, A. and Hubbard, T. (1998) 'Using neural networks for prediction of the subcellular location of proteins', *Nucleic Acids Res.*, Vol. 26, No. 9, pp.2230–2236.

Vapnik, V.N. (1998) *Statistical Learning Theory*, John Wiley and Sons, New York.

Wang L.X. and Mendel, J.M. (1992) 'Back-propagation fuzzy systems as nonlinear dynamic system identifiers', *Proc. IEEE Int. Conf. Fuzzy Syst.*, San Diego, pp.1409–1418.

Yuan, Z. (1999) 'Prediction of protein subcellular locations using Markov chain models', *FEBS Lett.*, Vol. 451, No. 1, pp.23–26.