Available online at www.sciencedirect.com

SCIENCE DIRECT®

Biochimica et Biophysica Acta 1648 (2003) 127–133

BIOCHIMICA ET BIOPHYSICA ACTA

BBA

www.bba-direct.com

Support vector machines for predicting rRNA-, RNA-, and DNA-binding proteins from amino acid sequence

Yu-dong Cai[a,*], Shuo Liang Lin[b]

[a] Shanghai Research Center of Biotechnology, Chinese Academy of Sciences, Shanghai, 200233, China
[b] Wyeth Research, Pearl River, NY 10965, USA

Received 14 October 2002; received in revised form 10 February 2003; accepted 11 February 2003

Abstract

Classification of gene function remains one of the most important and demanding tasks in the post-genome era. Most of the current predictive computer methods rely on comparing features that are essentially linear to the protein sequence. However, features of a protein nonlinear to the sequence may also be predictive to its function. Machine learning methods, for instance the Support Vector Machines (SVMs), are particularly suitable for exploiting such features. In this work we introduce SVM and the pseudo-amino acid composition, a collection of nonlinear features extractable from protein sequence, to the field of protein function prediction. We have developed prototype SVMs for binary classification of rRNA-, RNA-, and DNA-binding proteins. Using a protein's amino acid composition and limited range correlation of hydrophobicity and solvent accessible surface area as input, each of the SVMs predicts whether the protein belongs to one of the three classes. In self-consistency and cross-validation tests, which measures the success of learning and prediction, respectively, the rRNA-binding SVM has consistently achieved >95% accuracy. The RNA- and DNA-binding SVMs demonstrate more diverse accuracy, ranging from ~ 76% to ~ 97%. Analysis of the test results suggests the directions of improving the SVMs.

Keywords: Classification; Feature vector; Function; Functional genomic; Machine learning; Prediction; Pseudo-amino acid composition; Support vector machine, SVM

1. Introduction

Large-scale genome sequencing projects are producing gene sequences at unprecedented rates. Within merely a few years about 60 cellular genomes have been completely or nearly completely sequenced. An archean or bacterial genome may contain thousands of genes, while mammalian and plant genomes have tens of thousands. The enormous flux of genomic data exerts great pressure to the task of gene function determination. At present, only a number of predictive computer methods can keep up with the pace. Most of these methods adopt fast algorithms to search annotated databases for similarity in sequence, motif, profile, or hidden Markov models. Once sufficient similarity is found between the query sequence and one in the databases whose function is known, the query is predicted to possess a similar function. By this procedure, for example, researchers are able to assign functions to 69% of the 4524 putative proteins coded in the recently sequenced genome of an archean, *M. acetivorans* strain C2A [1]. While the coverage is remarkable for the latter, for this relatively small genome there are still ~ 1500 proteins to classify functionally. It will not be surprised if eventually only a minority of these proteins proves to possess a novel function.

The above example argues for the need to further advance protein function prediction methods. The question is, besides refining existing methods, in what directions can we explore? We notice that the current methods mostly operate on contiguous sequences or sequence segments, evaluating summations of per-position properties. However, protein sequence and function may not always associate with each other in this linear fashion. For instance, evolution may conserve among proteins of a common binding function the correlation between segments constituting the binding site, which are often short and discon-

* Corresponding author. Current address: Biomolecular Sciences Department, UMIST, P.O. Box 88, Manchester M60 1QD, UK. Tel.: +44-161-200-8936; fax: +44-161-236-0409.

E-mail address: y.cai@umist.ac.uk (Y. Cai).

1570-9639/03/$ - see front matter © 2003 Elsevier Science B.V. All rights reserved.
doi:10.1016/S1570-9639(03)00112-2

tinuous, rather than conserve the congruous sequences encompassing each of these segments. The quantification of correlation between positions across a distance amounts to a nonlinear mapping of the sequence. This observation prompts us to consider features nonlinear to protein sequence, and methods for nonlinear pattern comparison. These dual elements are rather new to protein function prediction, but have been used in other predictive applications. For instance, we have implemented Support Vector Machines (SVMs) to predict protein structural classes, subcellular locations, and proteolytic sites [2–4]. These SVMs each works with a nonlinear set of protein features, collectively referred to as pseudo-amino acid composition. They have achieved performance comparable or superior to other contemporary methods. Encouraged by the experiences, we are tempted to extend their application to protein function prediction.

SVM can be roughly described as follows. An SVM takes as input a set of features, called a feature vector. It outputs a classification. The SVM learns how to classify from a training set of feature vectors, whose expected outputs are already known. Figuratively, we may consider that the input vectors are mapped into a feature space. The training enables a binary classifying SVM to define a plane in the feature space, which optimally separates the training vectors of two classes. When a new feature vector is inputted, its class is predicted on the basis of which side of the plane it maps. Conceived recently as a pattern classification engine [5], SVM is based on rigorous statistical learning theories, and has been used in a wide range of problems, including image recognition and text classification [6] and drug design [7]. SVM has found increasing applications to protein classification problems, including fold recognition [8], gene expression data [9], etc.

Pseudo-amino acid composition of protein was proposed by Chou [10,11]. It is constructed by augmenting the traditional amino acid composition with a series of functions that couple the physicochemical properties between amino acids, which are separated by finite distances along the sequence. For these physicochemical properties, the coupling functions account for the effects of local sequence order, while they do not depend on the size, contiguity, and global order of the sequence. Pseudo-amino acid composition provides SVM with a high-dimensional feature vector. By selecting relevant physicochemical properties, a specific pseudo-amino acid composition can be formulated for a specific SVM classification application.

As a preliminary exploration of a machine learning method combining these two elements for protein function prediction, we have developed three prototypic SVMs. These SVMs predict whether or not a query sequence belongs to one of the rRNA-, RNA- and DNA-binding protein classes, respectively. We have obtained results that illustrate the feasibility of this approach. By fine tuning the SVMs and experimenting with various feature vector schemes [8,12], we can expect greater performance of the method in the future.

2. Materials and methods

2.1. Data

By searching the *SWISS-PROT* database (year 2000 release) with each of the keywords (KW) rRNA-binding, RNA-binding, and DNA-binding, 1056, 1496, and 7739 proteins were retrieved, respectively. These 3 collections are designated “positive” subsets, corresponding to the functions the keywords imply. A “negative” subset was composed by the following procedure:

(1) An “contrast” set of 10907 proteins was retrieved from *SWISS-PROT* by searching with a list of keywords suspicious of implying RNA/DNA-binding functionality, using the “or” logic. Namely, KW = Activator|ADP-ribosylation|Chroma-Chromatin regulator|Chromosomal protein|Chromosome partition|Core protein|DNA damage|DNA excision|DNA integration|DNA packaging|DNA priming|DNA recombination|DNA repair|DNA replication|DNA replication inhibitor|DNA synthesis|DNA-directed DNA polymerase|DNA-directed RNA polymerase|Endonuclease|Excision nuclease|Exonuclease|Helicase|Intron homing| Isomerase|mRNA processing|mRNA splicing|mRNA transport|Nuclear protein|Nuclease|Nucleocapsid|Nucleoprotein|Ribonucleoprotein|Ribosomal protein|Ribosome biogenesis|RNA repair|RNA replication|RNA-directed DNA polymerase|RNA-directed RNA polymerase|rRNA processing|Spli-Spliceosome|T-DNA|Topoisomerase|Trans-acting factor|Transcription|Transcription regulation|Transcription termination|Translation regulation|tRNA processing|tRNA-binding|DNA-binding|RNA-binding|rRNA-binding.

(2) The *SWISS-PROT* database depleted with the three positive subsets and the contrast set was randomly picked. The 4768 proteins resulted were entered to the negative subset. By combining the negative subset with each of the positive subsets, 3 function-specific datasets were obtained, i.e., the rRNA-binding dataset of 5824 proteins, the RNA-binding dataset of 6264 proteins, and the DNA-binding dataset of 12 507 proteins.

(3) The *SWISS-PROT* database depleted with the 3 positive subsets and the contrast set was reduced to a collection of 26100 proteins by removing homologous sequences using the CD-HIT program [13], with 40% cut-off. This collection was used as an alternative negative subset for the rRNA-binding SVM.

2.2. SVMs

The public domain software *SVMlight* [14] was used to build three SVMs for the prediction of rRNA-binding, RNA-binding, and DNA-binding proteins, respectively. The SVMs were binary classifying, meaning that each of

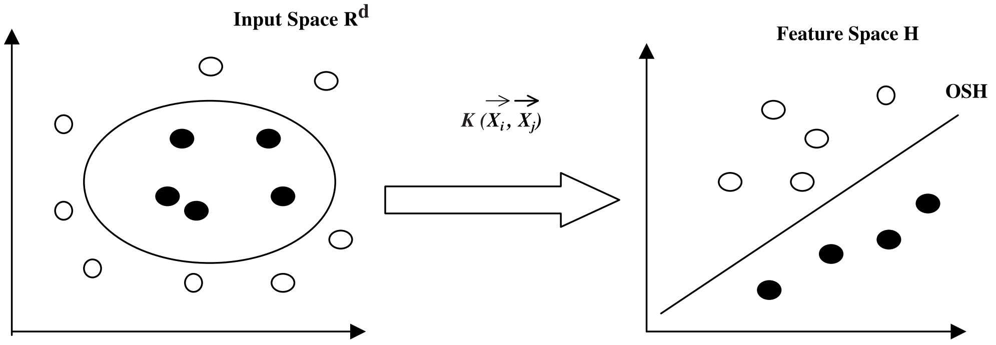

Fig. 1. The basic idea of SVM is to employ a mapping function ($K(X_i, X_j)$) to transform data from the input space (R^d), where the border between two classes may be nonlinear, to a feature space (H) where the border can be represented by a linear Optimal Separation Hyperplane (OSH).

them predicts whether or not an input protein possesses a targeted function. In the following, we explain the principles of SVM.

SVM is a learning machine based on statistical learning theory. The basic idea can be described briefly as follows. First, the inputs are formulated as feature vectors, of which each is associated with one of two classes. In training, the class of an input vector is known in advance. In prediction, the class is the output of SVM. Secondly, the feature vectors are mapped into a feature space (possibly with high dimensionality) by a kernel function, either linearly or nonlinearly. Thirdly, a division is computed in the feature space to optimally separate the two classes of training vectors. SVM training always automatically seeks global optimum and avoids over-fitting. These characteristics make it particularly suitable to deal with large numbers of features. For the application of SVM in pattern classification, the complete theory can be found in Vapnik's [5,15] monographs. Fig. 1 schematically illustrates the most basic idea of SVM.

In this work, SVM parameters were all set to the SVMlight default, except for that the width of the chosen kernel functions (Gaussian RBFs) was selected to minimize an estimate of the VC-dimension, and that the parameter C that controls the error-margin tradeoff was set to 1000. Readers interested in the terminology and other details should consult SVMlight specification [14] and Vapnik's monographs [5,15].

2.3. Pseudo-amino acid composition as feature vector

Pseudo-amino acid composition of a protein was used as a 40-dimensional input feature vector for SVM. The pseudo-amino acid composition involved the protein's amino acid composition and coupling functions applied to the charge, hydrophobicity, and accessible surface area of residues (Table 1), formulated according to a procedure prescribed by Chou [11] as follows.

First, for a protein chain of L amino acid residues

$R_1R_2R_3R_4R_5R_6R_7\ldots R_L$, a series of sequence-order-coupling numbers is calculated:

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

where τ_1 is called the first-rank sequence-order-coupling number that reflects the coupling mode between all the most

Table 1
Amino acid properties used to compose feature vector

Amino acid	Charge (C^0)	Hydrophobicity[a] (H^0)	Accessible surface area[b] (A^0)
Ala	0	1.8	44.1
Arg	1	− 4.5	152.9
Asn	0	− 3.5	80.8
Asp	− 1	− 3.5	76.3
Cys	0	2.5	56.4
Gln	0	− 3.5	100.6
Glu	− 1	− 3.5	99.2
Gly	0	− 0.4	0
His	1	− 3.2	98.2
Ile	0	4.5	90.9
Leu	0	3.8	92.8
Lys	1	− 3.9	139.1
Met	0	1.9	95.3
Phe	0	2.8	107.4
Pro	0	− 1.6	79.5
Ser	0	− 0.8	57.5
Thr	0	− 0.7	73.4
Trp	0	− 0.9	143.4
Tyr	0	− 1.3	119.1
Val	0	4.2	73.0

[a] The Kyte and Doolittle hydrophobicity scale (Kyte and Doolittle, 1982).
[b] Adapted from Creamer et al.'s (1995) Table 4, the N=25 column.

contiguous residues along a protein sequence, τ_2 is the second-rank sequence-order-coupling number that reflects the coupling mode between all the second most contiguous residues, and so forth. In Eq. (1), the coupling factor $J_{i,j}$ is a function of amino acids R_i and R_j, given by

$$J_{i,j} = D^2(R_i, R_j), \tag{2}$$

where $D(R_i,R_j)$ is the differential value of a physicochemical property between residues R_i to amino acid R_j. In this work,

$$D^2(R_i, R_j) = \frac{1}{3}\left\{[C(R_j) - C(Ri)]^2 + [H(R_j) - H(R_i)]^2 + [A(R_j) - A(R_i)]^2\right\} \tag{3}$$

where C, H and A are residual charge, hydrophobicity and accessible surface area, respectively, normalized from their original values C^0, H^0, A^0 (Table 1) by a standard conversion described by the following formulae [11]:

$$\begin{cases} C(i) = \dfrac{C^0(i) - \sum\limits_{i=1}^{20} \dfrac{C^0(i)}{20}}{\sqrt{\dfrac{\sum\limits_{i=1}^{20}\left[C^0(i) - \sum\limits_{i=1}^{20} \dfrac{C^0(i)}{20}\right]^2}{20}}} \\ H(i) = \dfrac{H^0(i) - \sum\limits_{i=1}^{20} \dfrac{H^0(i)}{20}}{\sqrt{\dfrac{\sum\limits_{i=1}^{20}\left[H^0(i) - \sum\limits_{i=1}^{20} \dfrac{H^0(i)}{20}\right]^2}{20}}} \\ A(i) = \dfrac{A^0(i) - \sum\limits_{i=1}^{20} \dfrac{A^0(i)}{20}}{\sqrt{\dfrac{\sum\limits_{i=1}^{20}\left[A^0(i) - \sum\limits_{i=1}^{20} \dfrac{A^0(i)}{20}\right]^2}{20}}} \end{cases} \tag{4}$$

where average is over the 20 amino acid types.

Secondly, suppose we have a set of N proteins, which is the union of m subsets, i.e.,

$$S = S_1 \cup S_2 \cup S_3 \cup S_4 \cup \cdots \cup S_m$$

Each subset has $n_\xi(\xi = 1,\ 2,\ 3\ldots,\ m)$ proteins of the same type. For the k-th protein in subset S_ξ, the traditional amino acid composition, i.e., the 20 normalized frequencies of amino acid occurrence $f_{k,j}^{\xi}(j = 1\ldots,\ 20)$, is augmented by φ sequence-order-coupling numbers to compute $20+\varphi$ features

$$X_k^{\xi} = \begin{bmatrix} X_{k,1}^{\xi} \\ X_{k,2}^{\xi} \\ \vdots \\ X_{k,20+\varphi}^{\xi} \end{bmatrix}, \quad (k = 1, 2, \ldots, n_\xi; \xi = 1, 2, \ldots, m), \tag{5}$$

in which

$$X_{k,u}^{\xi} = \begin{cases} \dfrac{f_{k,u}^{\xi}}{\sum\limits_{j=1}^{20} f_{k,j}^{\xi} + w\sum\limits_{q=1}^{\varphi} \tau_{k,q}^{\xi}}, & (1 \le u \le 20) \\ \dfrac{w\tau_{k,u-20}^{\xi}}{\sum\limits_{j=1}^{20} f_{k,j}^{\xi} + w\sum\limits_{q=1}^{\varphi} \tau_{k,q}^{\xi}}, & (20 + 1 \le u \le 20 + \varphi) \end{cases} \tag{6}$$

where $\tau_{k,q}^{\xi}$ is the qth-rank sequence-order-coupling number computed according to Eqs. (1) and (2) for the k-th protein in subset S_ξ, and w is the weight factor for the sequence-order effect. As we can see from Eqs. (3)–(6), the first 20 components reflect the effect of the amino-acid composition, while the components from $20+1$ to $20+\varphi$ reflect the effect of sequence order. We usually choose the parameters from the ranges $10<\varphi<40$ and $0.0<w<0.1$, depending on the coupling distance and strength suitable for the particular application. For this work, we chose $\varphi=20$ and $w=0.05$.

2.4. Computation

All computations were carried out on a Silicon Graphics IRIS Indigo Elan 4000 workstation with a 270-MHz IP27 processor.

3. Results

Each of the rRNA-binding, RNA-binding, and DNA-binding SVMs was trained with the corresponding dataset. A dataset included a positive subset and a negative subset. The proteins of the positive subset were known to possess the function the SVM was trained to recognize. The negative subset was known not to possess the function. The keyword annotation in the *SWISS-PROT* database was taken as prior knowledge of the protein function, regardless the function had been determined experimentally or predicted. Test predictions of these SMVs are reported here.

Two tests were conducted for each of the SVMs: self-consistency test and cross-validation test. In self-consistency test, an SVM trained with the full dataset was used to predict the function of every protein in the same dataset, to make comparison with known function. Test of the rRNA-binding SVM resulted in near-perfect correct prediction rates: 100% for the positive subset, 99.98% for the negative subset, and 99.98% for the full set. For the RNA-binding SVM, results were unbalanced: ~ 76% for the positive subset, ~ 97% for the negative subset, and ~ 92% overall. Results of the DNA-binding SVM were also unbalanced but tipping to the opposite direction: 93% for the positive subset, 77% for the negative subset, and 87% overall (Table 2).

In the cross-validation test, 90% of a dataset was used to train an SVM. Then the partially trained SVM was used to make function prediction and comparison for the remaining 10% data. Because the latter was unknown to the SVM during training, the prediction was realistic. In this test, the correct prediction rate of the rRNA-binding SVM ranged from ~ 95% to ~ 99%, and ~ 98% if combining all the 10 sets of prediction together. When using the alternative negative subset, the corresponding percentages were ~ 84%, ~ 95%, and 89%. For the RNA-binding SVM, the rates were in the ~ 82–91% range and ~ 86% overall. For the DNA-binding SVM, they were ~ 78–86% and ~ 81% (Table 2).

If the jackknife test were adopted, we expect that the correct prediction rates could be even higher and more consistent than demonstrated in the cross-validation test. Jackknife test is more objective and effective for appraising the prediction capability of the fully trained SVM, leaving away only a single data point from the training set for prediction [3,16]. However, permuting every sequence out from thousands would have taken too much CPU time than practical for us. Therefore, we had settled for the less rigorous, while still illustrating left-10%-out cross-validation scheme.

On the computer we used (Silicon Graphics IRIS Indigo Elan 4000 workstation with a 270 MHz IP27 processor), the CPU times consumed for training the rRNA, RNA, and DNA-binding SVMs were 5 min, 28 min, and 18 h, respectively (Table 3). For prediction, the CPU time was 2 min per query, averaged over 10 907 attempts. The disparate training times reflect how difficult the individual SVMs converge to a satisfactory separating hyperplane. In general, the greater the size of the training set, the more difficult the convergence, giving rise to a cost nonlinear to the size of dataset.

Table 2
Self-consistency test results

SVM	Correct prediction rate		
	Positive subset	Negative subset	Overall
rRNA-binding	1056/1056 = 100.0%	4767/4768 = 99.98%	5823/5824 = 99.98%
RNA-binding	1144/1496 = 76.47%	4634/4768 = 97.19%	5778/6264 = 92.24%
DNA-binding	7183/7739 = 92.82%	3675/4768 = 77.08%	10858/12507 = 86.82%

Table 3
CPU time of SVM training, testing, and prediction[a]

SVM	Training	Tests		Prediction
		Self-consistency	Cross-validation	
rRNA-binding	5 min	7 min	1 h	2 min
RNA-binding	28 min	30 min	4 h	2 min
DNA-binding	18 h	18 h	150 h	2 min

[a] For each SVM, training and testing used the corresponding rRNA_binding, RNA_binding and DNA_binding datasets. Prediction time was per_query average over the contrast set of 10907 proteins.

4. Discussion

Our work is among the early efforts of employing SVM, a machine learning method, to build classifiers for protein function. We have implemented three SVMs for binary classification of rRNA-, RNA- and DNA-binding proteins, respectively. Each of them predicts whether or not a query protein belongs to one of the three classes. These SVMs are exploratory prototypes, serving for an appraisal of the potential of our approach. The three classes of proteins present different levels of challenge by their different degrees of diversity in sequence and functionality. By assembling the training data from *SWISS-PROT* annotation, we have introduced a fair amount of noise virtually compatible to that in human knowledge. We have appraised the performance of the three SVMs by self-consistency and cross-validation tests. We consider their performance to be fair to stellar. From analyzing the test results, we have gained insights to how improvements can be made.

The self-consistency test demonstrates how successful SVM has turned training into internal knowledge. Revisiting the training data, the test measures the ability of SVM to reproduce known classification. In this test, the rRNA-binding SVM recognized the 1056 rRNA-binding proteins with 100% accuracy, and made only one mistake out of the 4768 proteins in the negative training set (Table 2). This nearly perfect performance indicates the following. (1) The pseudo-amino acid composition possesses an intrinsic correlation to the classification being pursued, even if the correlation cannot be explicitly expressed. (2) The SVM is able to abstract the correlation from training data, to construct correct class distinction rules.

To our best knowledge, rRNA-binding proteins have never been distinguished as a single class by algorithms based on linear sequential similarity. According to the *PROSITE* database, the 1056 rRNA-binding proteins in the positive training subset belong to dozens of different ribosomal protein families, each bearing its own signature motif(s). Between these families, sequence homology is generally low. For example, amino acid identity is mostly < 10% between members of ribosomal families 30S S4 and

50S L20, both abundantly presented in the training set. Therefore, the high performance of the rRNA-binding SVM in self-consistency test is not due to being trained with a unique, homologous positive subset. Rather, it showcases that SVM can find a factor common to a diverse set of positive training data, which the negative set lacks, and use it to achieve optimal classification. For the rRNA-binding proteins, it is tempting to uncover exactly what this common factor is. It would advance our understanding of the ribosomal proteins, and help derive more rational prediction methods. Unfortunately, like for other nonlinear learning machines, currently it is still difficult to translate the intricate SVM internal structure to biological terms.

Cross-validation test uses SVMs trained by part of the full training dataset to predict for the rest. The goal is twofold. First, it conducts a realistic prediction whose accuracy is measurable. Second, by alternating the part left for prediction, it examines the consistency in the potential of prediction. In this test, the rRNA-binding SVM achieved 95–99% accuracy (Table 4). The accuracy is both high and consistent. The less than perfect results agree with the aforementioned sequence diversity in the training set. Consistent with this interpretation, using a larger negative subset that is ∼ 26 times the size of the positive subset instead of the original ∼ 5 times to train the SVM, the accuracy is declined to 85–93%. The declination is mostly due to a lower recognition rate of the positives (data not shown). It is likely that the randomly picked, oversized alternative negative subset contains a greater number of false negatives that have confused the SVM. Therefore, with the original, more balanced dataset, the fully trained SVM is expected to possess higher prediction power than the partially trained versions. Nonetheless, however, slight as it is, the variation in accuracy still reminds us a character of SVM that probably applies to machine learning in general. Namely, its power of prediction is limited by the statistical rules it abstracts from the training set. By contrast, an idealistic, "mechanistic" rule that has grasped the essence of the data may have greater capacity for extrapolation. These two kinds of rules can converge when the training dataset has sufficient coverage and negligible noise, and the SVM is constructed perfectly. It is therefore desirable to reassemble the training data and re-train the SVM frequently.

Table 4
Cross-validation test results

SVM	Correct prediction rate		
	Low	High	Overall
rRNA-binding[a]	552/581 = 95.01%	574/581 = 98.80%	5640/5824 = 96.84%
	2662/2715 = 84.95%	2671/2715 = 92.51%	26726/27150 = 89.00%
RNA-binding	511/625 = 81.76%	567/625 = 90.72%	5371/6264 = 85.74%
DNA-binding	969/1249 = 77.58%	1071/1249 = 85.75%	10131/12507 = 81.00%

[a] The second row of rRNA-binding data was obtained using the alternative negative subset described in Materials and methods.

Both the RNA-binding and DNA-binding proteins are broad categories, known to be highly diverse in sequence and in the mode of nucleic acid binding. In each of these categories, the pairwise sequence is mostly below 20%, and the average homology is 8%. The RNA- and DNA-binding proteins are highly diverse in function. For instance, there are nucleic acid-processing enzymes recognizing specific RNA/DNA sites, and there are nonspecific genome-coating proteins. They present greater challenge than the more specialized class of rRNA-binding proteins. In the cross-validation test, the RNA-binding and DNA-binding SVMs resulted in 82–91% and 78–86% accuracy, respectively (Table 4). At the current state with these correction rates, we believe that these prototypic SVMs can be used for a first pass of functional prediction, complementary to existing methods. In the self-consistency test, both the SVMs showed unbalanced accuracy in separate results for the positive and negative subsets: 93% versus 77%, and 76% versus 97%, respectively (Table 2). For the higher scoring subset, sensitivity of the SVMs is high while selectivity falls behind. For the lower scoring subset, sensitivity and selectivity exchange positions. Armed with this knowledge, we can use these SVMs practically within appropriate context. Furthermore, we can find clues for improvement from the unbalance. In the language of feature space, the phenomenon implies that the optimal separating hyperplane is essentially able to isolate the vectors of one of the subsets, but unable to prevent the penetration of those of the other subset. Tuning the coupling functions incorporated in the features may result in a more desirable distribution of the vectors. On the other hand, if the penetration is not randomly diffusive, implementing the SVM with a different kernel function may "bend" the space toward a better separation.

Our method is intended for complementing those frequently used in functional genomics. Most of them opt for database search of similarity in sequence, motif, profile, or hidden Markov models, in order to transfer the functional annotation from hit(s) to the query. By BLAST and hidden Markov model searches lately, for example, 3571 of 5420 open reading frames of the completed genome of bacterium *P. putida* KT2440 were assigned with a putative function [17]. The ∼ 60% assignment rate is typical among more than a hundred completed genomes of the three major domains of life, and is expected for another hundred genomes being sequenced around the world—if assignment is limited to similar methods. Many of the thousands of genes not being assigned by this approach may actually perform known functions, without being homologous to any annotated protein. In this regard, it is interesting that a recently developed artificial neural network (ANN) method assigned enzymes to a much larger fraction of human genome than predicted in the original publications of the human genome draft [18]. Even though the ANN authors found that the differences in method made it difficult to interpret the differences in prediction, the work does exemplify possible benefits of adopting complementary methods. Comparing the performance of the ANN method with ours is also difficult, because different protein categories are studied. Nevertheless, we find it interesting that the ANN method also demonstrates tradeoff in the accuracy between predicting positives and negatives. For instance, a 90% accuracy in predicting the positives is accompanied by ~ 20–90% accuracy predicting the negatives for classifying 12 *SWISS-PROT* families. The worst of them are associated with the families of inherently heterogeneous, an observation consistent with our own experience. Comparing the accuracy levels, our method has shown its competitiveness even at an infantile stage. We believe that by joining force with other methods, it can contribute to the enhancement of coverage and accuracy of protein function prediction.

5. Conclusion

We have demonstrated the feasibility of combining SVM and pseudo-amino acid composition for a new protein function prediction method. Even as prototype, the three SVMs we implemented have shown practical performance. With appraisal tests, we have found clues to improve SVM. Selections of feature vector and kernel function, and an adequate and low-noise training set, are critical to the success of SVM. Apparently, the more specific a function is to predict, thus the more definite a training set can be assembled, and the higher predicting power the corresponding SVM can acquire. However, being specific in function does not require sequential similarity, as we evidently have shown with our SVMs. This segregation of sequential and functional similarity will be one of the most attractive attributes of using SVM for protein function prediction. Training, testing, and tuning SVM are computer-intensive. Predicting is very fast. In the future, we envisage an array of SVMs being trained to predict specific functions, and to parse genomic sequence data in parallel, complementing current methods to achieve more reliable, high-throughput gene function prediction.

References

[1] J.E. Galagan, et al., Genome Res. 12 (2002) 532–542.
[2] Y.D. Cai, X.J. Liu, X.B. Xu, K.C. Chou, Molec. Cell Biol. Res. Commun. 4 (2000) 230–233.
[3] Y.D. Cai, X.J. Liu, X.B. Xu, G.P. Zhou, BMC Bioinformatics 2 (2001) 1–3.
[4] Y.D. Cai, X.J. Liu, X.B. Xu, K.C. Chou, J. Comput. Chem. 23 (2002) 267–274.
[5] V. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.
[6] T. Joachims, Proceedings of the European Conference on Machine Learning. Springer, Berlin, 1998.
[7] R. Burbidge, T. Matthew, H. Sean, B. Bernard, Proceedings of the AISB'00 Symposium on Artificial Intelligence in Bioinformatics, Birmingham, England, 2000, pp. 1–4.
[8] C.H. Ding, I. Dubchak, Bioinformatics 17 (2001) 349–358.
[9] M.P. Brown, et al., Proc. Natl. Acad. Sci. U. S. A. 97 (2000) 262–267.
[10] K.C. Chou, J. Protein Chem. 18 (1999) 473–480.
[11] K.C. Chou, Proteins Struct. Funct. Genet. 43 (2001) 246–255.
[12] T. Jaakkola, M. Diekhans, D. Haussler, Proc. Int. Conf. Intell. Syst. Mol. Biol., AAAI Press, Menlo Park, CA, 1999, pp. 149–158.
[13] W. Li, L. Jaroszewski, A. Godzik, Bioinformatics 17 (2001) 282–283.
[14] T. Joachims, in: B. Schölkopf, C. Burges, A. Smola (Eds.), Advances in Kernel Methods-Support Vector Learning, MIT Press, Cambridge, MA, 1999, pp. 11–15.
[15] V. Vapnik, Statistical Learning Theory, Wiley-Interscience, New York, 1998.
[16] K.C. Chou, Biochem. Biophys. Res. Commun. 278 (2000) 477–483.
[17] K.E. Nelson, et al., Environ. Microbiol. 4 (2002) 799–808.
[18] L.J. Jensen, et al., J. Mol. Biol. 319 (2002) 1257–1265.