PROTEINS: Structure, Function, and Bioinformatics 54:557–562 (2004)

Prediction of Protein Relative Solvent Accessibility with Support Vector Machines and Long-Range Interaction 3D Local Descriptor

Hyunsoo Kim and Haesun Park*
Department of Computer Science and Engineering, University of Minnesota, Minneapolis, Minnesota

ABSTRACT **The prediction of protein relative solvent accessibility gives us helpful information for the prediction of tertiary structure of a protein. The SVMpsi method, which uses support vector machines (SVMs), and the position-specific scoring matrix (PSSM) generated from PSI-BLAST have been applied to achieve better prediction accuracy of the relative solvent accessibility. We have introduced a three-dimensional local descriptor that contains information about the expected remote contacts by both the long-range interaction matrix and neighbor sequences. Moreover, we applied feature weights to kernels in SVMs in order to consider the degree of significance that depends on the distance from the specific amino acid. Relative solvent accessibility based on a two state-model, for 25%, 16%, 5%, and 0% accessibility are predicted at 78.7%, 80.7%, 82.4%, and 87.4% accuracy, respectively. Three-state prediction results provide a 64.5% accuracy with 9%; 36% threshold. The support vector machine approach has successfully been applied for solvent accessibility prediction by considering long-range interaction and handling unbalanced data. Proteins 2004;54:557–562.** © 2003 Wiley-Liss, Inc.

Key words: protein structure prediction; solvent accessibility; support vector machines; PSSM; directed acyclic graph scheme; long range interaction

INTRODUCTION

The task of predicting protein structure from the sequence is important, since the function of a protein is closely related to its structure, which is difficult to determine experimentally. There are largely two types of methods in protein structure prediction. The first type includes threading and comparative modeling, which rely on prior knowledge of similarity among sequence and known structures. The second type, called de novo or ab initio methods, predicts the protein structure from the sequence alone, without relying on the similarity of known structure. Currently, it is difficult to predict high-resolution three-dimensional (3D) structure from ab initio methods for studying the docking of macromolecules, predicting protein partner, designing and improving ligands, and protein–protein interaction.[1]

For the knowledge-based methods, protein secondary structure prediction[2–8] has been studied as an intermediate step for predicting tertiary structure of proteins, especially in the case when the sequence similarity is lower than 30%, since the secondary structure is more conserved than the protein sequence. The protein solvent accessibility prediction has also been studied based on the neural network approach[9–13] with training by the conjugate gradient descent algorithm (i.e., back-propagation), Bayesian method,[14] or information theory.[15]

Though the prediction of solvent accessibility is less accurate than that of secondary structure from the homology approach, since it is less conserved than secondary structure,[10] there has been much effort to improve prediction accuracy to obtain important information regarding a buried or exposed residue for constructing tertiary structure from sequences. For example, the prediction of secondary structure and solvent accessibility can be aligned to known 3D structure to detect a putative remote homologue for threading. The predictions can also be used as additional constraints in ab initio methods.

In this article, we have introduced a long-range interaction 3D local descriptor and have used the SVMpsi[8] method, including feature weights, to improve prediction of protein relative solvent accessibility. We applied a directed acyclic graph (DAG) scheme[16] for the three-class classification problem in SVMpsi to avoid one-versus-rest classification, which has higher complexity than one-versus-one classification.

MATERIALS AND METHODS

Relative Solvent Accessibility

Amino acid solvent accessibility is the degree to which a residue in a protein is accessible to a solvent molecule. The relative solvent accessibility can be calculated from

$$RelAcc_i = 100^{*}Acc_i/MaxAcc_i, \quad (1)$$

where Acc_i for the ith residue is the solvent accessibility (given in Angstrom units) calculated from coordinates by

Grant sponsor: National Science Foundation; Grant number CCR-0204109 and ACI-0305543. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

*Correspondence to: Haesun Park, Department of Computer Science and Engineering, University of Minnesota, 200 Union Street St SE, 4-192 EE/CS Building, Minneapolis, MN 55455. E-mail: hpark@cs.umn.edu

Received 31 January 2003; Accepted 27 July 2003

the dictionary of protein secondary structure (DSSP) program.[17] The number of water molecules around a residue can be approximated by $Acc_i/10$, and $MaxAcc_i$ is the maximum accessibility for the ith residue, which is given for ambiguous (B, Z) or undetermined (X) residue, as well as 20 normal amino acids.[10]

We used two kinds of class definitions: (1) buried (B) and exposed (E); (2) buried (B), intermediate (I), and exposed (E). For the two-state definition, we chose various thresholds of the relative solvent accessibility such as 25%, 16%, 5%, and 0%. For the three-state (B, I, E) description of relative accessibility, one set of thresholds that we selected is the same as those in Rost and Sander[10]:

$$\text{Buried (B): } RelAcc < 9\%$$

$$\text{Intermediate (I): } 9\% \leq RelAcc < 36\%$$

$$\text{Exposed (E): } RelAcc \geq 36\%.$$

Various other thresholds for and three-state definition were chosen in addition to compare our results with the previously published results and to find the dependency of thresholds on the prediction accuracy of the relative solvent accessibility.

3D Local Descriptor Coding Scheme

A local descriptor that represents the local environment of sequences by sliding window coding scheme[2–18] can be enhanced by embedding the long-range interaction in order to reflect the 3D local environment. The 3D local descriptor represents the environment of a specific residue not only in the sequence but also in the 3D space.

There are essentially four significant driving forces that cause remote residues to contact. The first is a disulfide covalent bond, which makes the nearest neighbors contact. The most predominant linkage by disulfide bonds among the secondary structural elements is the coil–coil linkage.[19] The structure of coil region is relatively important, since functionally important residues which are involved in a key protein–protein interaction, usually lie in the coil regions. The second is a salt bridge. The oppositely charged residues between (Asp, Glu) and (Lys, Arg) tend to form a salt bridge. The third is hydrophobic interactions among (Phe, Ile, Leu, Val). Especially, homopairs between themselves give the most favorable hydrophobic interactions. The fourth is remote hydrogen bonds that frequently appear, since this is a major force that forms a beta sheet.

The most probable remote contact sequence block in an entire sequence with respect to the current local environment can be found by the long-range interaction matrix.[20] The matrix represents relative frequencies of long-range interaction for each amino acid pair. It was obtained from statistical analysis of the accumulation of long-range interactions, where 2 residues are separated by at least 10 residues in the sequence, and at least one of their atomic distances is less than the sum of the van der Waals radii of the two atoms plus 1.0 Å.[20] The remote contact expectation score between a given fragment o and the expected remote contact fragment e is

$$E_r = \sum_{i=1}^{w} P[o(i), e(i)], \tag{2}$$

where $o(i)$ is the ith amino acid in the window fragment, $e(i)$ is the ith amino acid in the candidate fragment, w is the window size, and $P(a, b)$ is the matrix component of the relative frequency of long-range interactions between two amino acids a and b. The fragments that stabilize proteins by building remote contacts tend to be more buried than the average accessibility of the rest of the sequences. However, rather high accessibility can also be found in the stabilization sequences, since the remote contact can be driven by a salt bridge between high polar residues. In a folded protein structure, hydrophilic side-chains tend to contact polar solvent, but the hydrophobic side-chains tend to minimize the contact with the polar solvent.[21] A weighted hydrophobicity for the current window can be expressed as

$$H_c = \sum_{i=1}^{w} h(i) \exp[-|i - (w+1)/2)|^2/100], \tag{3}$$

where $h(i)$ is the hydrophobicity of the ith amino acid in the current window. After identifying the most probable remote contact sequence block of w residues that has the highest remote contact expectation score E_r with the current window, we can also calculate a weighted hydrophobicity for remote contact H_r using a similar equation as for H_c, where $h(i)$ is the hydrophobicity of the ith amino acid in the remote contact sequence block.

The final position-specific scoring matrix (PSSM) from PSI-BLAST[22] against SWISS-PROT database[23] (after three iterations) is used as an input to support vector machines (SVMs). The matrix has $20 \times m$ elements, where m is the length of the target sequence and each element represents the log-likelihood of that particular residue substitution at that position in the template. The final PSSM from PSI-BLAST against the SWALL[23] nonredundant protein sequence database is used. We applied PFILT[24,25] to mask out regions of low complexity sequences, the coiled coil regions, and transmembrane spans. For PSI-BLAST, the E-value threshold for inclusion of 0.001 and three iterations were applied to search the nonredundant sequence database. The profile matrix elements in the range $[-7,7]$ are scaled to the [0,1] range.

Each residue is represented using 20 components in a vector, based on the PSSM. In order to allow a window to extend over the N-terminus and the C-terminus, an additional 21st unit (spacer) was attached to each residue. Then, each input vector has $21 \times w$ components, where w is a sliding window size. The values for H_c, E_r, and H_r are appended to the original feature vector to build a 3D local descriptor. Therefore, each input vector has $21 \times w + 3$ components. If the expected remote contact is not found [i.e., the expectation score is smaller than the threshold ($E_t = 1.2 \times w$)], E_r and H_r are filled with zeros. The window is shifted residue by residue through a protein chain.

TABLE I. Dependency of Testing Accuracy on the Window Length for Each Binary Classifier

Classifier	$l = 11$	$l = 13$	$l = 15$	$l = 17$	$l = 19$	l^*
E/B[1]	76.82	76.84	76.75	76.60	76.60	13
B/I[2]	69.11	69.06	69.08	69.16	69.12	17
E/B[2]	81.42	81.55	81.69	81.57	81.53	15
I/E[2]	67.94	68.24	68.08	68.02	68.03	13

Results for E/B[1] are obtained with threshold 25 in case of 2-state model. Results for B/I[2], E/B[2], and E/I[2] are obtained with thresholds 9;36 in case of the 3-state model. The results are on the RS126 with PSI-BLAST profile and an L_1 soft margin SVM with the RBF kernel function, using the corresponding optimized γ and C parameters. The l^* value is the optimal window length for each binary classifier. Combined results of 7-fold cross validation are shown.

PHDacc[10] consists of two different networks with window sizes of 9 and 13 consecutive residues for jury decision. In Jnet,[6] a neural network with a sliding window of 17 residues for the first input and 19 for the second input was designed. Both NETASA[26] and Naderi-Manesh et al's method[15] based on information theory used a window size of 17. We built input vectors considering 15 consecutive residues for predicting the central 8th residue after finding the optimal window length (see Table I).

Support Vector Machines

In many classification problems, different classes cannot be linearly separated in the original input space. An SVM finds a nonlinear decision function in the input space by implicitly mapping the data into a linear separable higher dimensional feature space and separating the data there by maximizing the geometric margin and minimizing the training error at the same time. The primal optimization problem is

$$\min_{\mathrm{w},\xi_\mathrm{i}} \quad \frac{1}{2}\mathrm{w}^t\mathrm{w} + C\sum_{i=1}^{n}\xi_i \tag{4}$$

$$s.t. \quad y_i[\mathrm{w}^t\mathrm{x}_i + b] \geq 1 - \xi_i,\ \xi_i \geq 0, \quad i = 1,\ldots,n,$$

where x_i represents an input vector, $\mathrm{y}_\mathrm{i} = \pm 1$ according to whether x_i is in the positive or negative class, n is the number of the training data, and C is a parameter that controls the trade-off between margin and classification error represented by slack variable ξ_is. The corresponding dual quadratic programming problem with an incorporation of a kernel function can be written as

$$\max_{\alpha_i} \quad \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j K(\mathrm{x}_i, \mathrm{x}_j), \tag{5}$$

$$s.t. \quad \sum_{i=1}^{n}\alpha_i y_i = 0,\ 0 \leq \alpha_i \leq C,\ i = 1,\ldots,n,$$

where α_i represents the influence of single ith training example limited by C, and $K(\mathrm{x}_\mathrm{i}, \mathrm{x}_\mathrm{j})$ is a kernel function to handle the nonlinear separable case.

SVMs find the unique minimum of a convex function for training a given data set.[27–30] The decision boundary is represented as a sparse linear combination of the training set examples.[31,32] Recently, SVMs have also been shown to perform well in multiple areas of biological analysis, including protein secondary structure prediction,[7,8] protein subcellular localization prediction,[33] multiclass protein fold recognition,[34–36] gene function prediction from microarray expression data,[37] cancer tissue classification from microarray expression data,[38,39] gene selection for cancer classification,[40] and protein–protein interaction problems.[41] Also, SVMs are well suited to solve pattern recognition problems, such as isolated handwritten digit recognition,[42] 3D object recognition,[43,44] speaker identification,[45] face detection,[46] and text categorization.[47–49]

Kernel Feature Weight Scheme

The solvent accessibility for a specific amino acid can be determined by its 3D local environment. We assumed that the contribution can be different, since the closer amino acids may have more influence on accessibility in the local environment. We scaled the feature values and derived a modified kernel function as

$$K_m(\mathrm{x}_i, \mathrm{x}_j) = K(W\mathrm{x}_i, W\mathrm{x}_j), \tag{6}$$

where W is a diagonal matrix which contains weight factors. However, we scaled all input vectors by multiplying them with W once, to avoid matrix vector multiplications whenever the kernel function is calculated. The 20 numerical values that are row elements of the PSSM for an amino acid were scaled by $\exp(-z^2/100) + 1.0$, where z is the sequential distance between the specific amino acid and the amino acid at the window center. The diagonal elements of W for the appended part for the remote contact residues were set to 1. The scale function was designed to cover the range of integers $z \in [-7, 7]$ for the optimal window length 15.

Parameter Optimization for SVMs

When using SVMs, we need to select a kernel function and the parameter C, and construct tertiary classifiers based on binary classifiers. After preliminary tests, it was found that the Gaussian RBF (radial basis function) kernel

$$K(\mathrm{x}_i, \mathrm{x}_j) = \exp(-\gamma\|\mathrm{x}_i - \mathrm{x}_j\|^2) \tag{7}$$

was appropriate for our classification problems.

The optimal separating hyperplane can be represented by support vectors of which α_i is nonzero. Each support vector contributes one local Gaussian function centered at that data point. The parameters γ and C can be selected from the optimization process, and were found to be $\gamma = 0.01$, $C = 1.0$ for our 2-state model, and $\gamma = 0.01$, $C = 1.5$ for our 3-state model. We also tested linear kernels and polynomial kernels

$$K(\mathrm{x}_i, \mathrm{x}_j) = \mathrm{x}_i \cdot \mathrm{x}_j \tag{8}$$

$$K(\mathrm{x}_i, \mathrm{x}_j) = [\alpha\mathrm{x}_i \cdot \mathrm{x}_j + b]^d \tag{9}$$

TABLE II. Prediction Accuracy With Different Types of Kernel Functions in SVMs

Kernel function	E/B[1]	B/I[2]	E/B[2]	I/E[2]
Linear	75.83	68.98	80.38	67.04
Polynomial ($d = 2$)	76.68	68.91	81.28	67.88
Polynomial ($d = 3$)	76.73	68.98	81.35	67.52
Polynomial ($d = 4$)	76.67	68.88	81.40	67.40
RBF	76.84	69.08	81.69	68.08

The sliding window size 13 was used for E/B, 15 for the others. Results for E/B[1] were obtained with the threshold of 25 in case of the 2-state model. Results for B/I[2], E/B[2], and E/I[2] are obtained with thresholds 9;36 in case of the 3-state model. The results are on the RS126 with PSI-BLAST profiles and an L_1 norm soft margin SVM. With the RBF kernel function, parameters γ and C are optimized based on the data set. Combined results of 7-fold cross validation are shown.

with various degrees d and $a = b = 1$. Various even and odd degrees for the polynomial kernels were tested, but no special difference in prediction accuracies was observed.

Table II shows that the RBF kernel produces the most accurate prediction results for the solvent accessibility.

Handling Unbalanced Data

In binary classification problems, if the number of samples of one class is much larger than that of the other class, the decision boundary tends to be determined to make a better decision for the larger class for the purpose of maximizing the total accuracy. For handling the unbalanced data, there are three kinds of approaches. The first method discards training points of the larger size class to balance the number of training points of both classes. Though this approach reduces the number of points to gain balance and lower complexity, it may eliminate points that contain critical information for classification. The second approach duplicates the training points of the smaller size to achieve balance. The third method uses different penalty parameters in the SVM formulation[30] such as

$$\max_{\alpha_i} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{10}$$

$$s.t. \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$0 \le \alpha_i \le C_+, \text{ if } y_i = 1$$

$$0 \le \alpha_i \le C_-, \text{ if } y_i = -1.$$

Using different penalty parameters (C_+ and C_-), we can resolve the situation that the recall value of the smaller class is too small to produce good prediction accuracy.

We used the third method to treat unbalanced data for most classifications. The duplicate method was used for fully buried residue classification, since it was difficult to choose a good pair of penalty parameters when the difference in the number of data points in two classes was too large. The first method was adopted for the KP480 data set, since there was not enough memory to store the entire KP480 sliding window data points.

Final Prediction

We obtained one-versus-one classifiers (E/B) for the 2-state (exposed/buried) relative solvent accessibility, and three one-versus-rest classifiers (B/~B, I/~I, E/~E) and three one-versus-one classifiers (B/I, E/B, and I/E) for the 3-state relative solvent accessibility from SVMs. We adopted a (DAG) scheme[16] for which prediction results were as good as the jury results that used all six binary classifiers. The jury decision scheme suffers from unbalanced data in its one-versus-rest classifiers.[8]

If a residue is predicted to be not buried (~B) from E/B one-versus-one classifier, I/E classifier is applied, whereas if the residue is not exposed (~E) from E/B classifier, B/I classifier is applied to check if it is buried or intermediate. Three different kinds of DAG schemes can be constructed; DAG1 (starts with B/I), DAG2 (starts with E/B), DAG3 (starts with I/E). We observed that the prediction results were almost the same in all cases.

RESULTS AND DISCUSSION

We used three different data sets for our computational experiments. The first data set (HMK24) consists of 19 training sequences and 5 test sequences. The training set contains the sequences of 1bp2, 1cpv, 1ctf, 1gcr, 1lz1, 1mbd, 1pcy, 1rn3, 1tpp, 2act, 2alp, 2apr, 2sga, 3dfr, 3tln, 4fxn, 451c, 5cpa, and 9pap. The test set contains the sequences of 1nxb, 1ubq, 2cpp, 2prk, and 2sns. The second data set (RS126) contains 126 proteins with less than 25% pairwise sequence identity. This data set has been used to study conservation and prediction of solvent accessibility in protein families.[10] The third data set (KP480) was designed based on CB513 by removing proteins that are shorter than 30 residues and those from the result of PSI-BLAST that contained only a few sequences in the first iteration.[8]

Three dimensional coordinates of proteins were obtained from the Protein Data Bank (PDB)[50] and the solvent accessibility was calculated with the DSSP program of Kalbsch and Sander.[17] The relative solvent accessibility was calculated by Eq. (1). The groups [buried (B), exposed (E); buried (B), intermediate (I), exposed (E)] were determined by the corresponding thresholds. The second and third data sets were divided into 7 folds that have similar number of proteins for cross-validation tests. The first data set was studied without any cross-validation test.

Holbrook et al[9] achieved 72.0% overall prediction accuracy for the test sequences of HMK24 in the binary model with a window size of 11, and 54.0% prediction accuracy of solvent accessibility in the ternary model with a window size of 7 using 10 hidden nodes. To compare our results with these previously published results[9] for the HMK24 data set, we used the same thresholds; that is, i.e. the buried residues were defined as those with less than 20% of relative solvent accessibility for the 2-state model, and buried (0–5%), intermediate (5–40%), or exposed (>40%)

TABLE III. Accuracy of the Relative Solvent Accessibility for Different Thresholds

	State threshold (%)				
Method	2-state (25%)	2-state (16%)	2-state (5%)	2-state (0%)	3-state (9%;36%)
PHDacc	—	74.8	—	86.0	57.5
SVMpsi*	76.8	77.8	79.8	86.2	59.6
Jnet*	75.0	—	79.0	86.6	—
Jnet†	76.2	—	79.8	86.5	—
BRNNs	77.2	—	81.2	86.5	—
NETASA	70.3	—	74.6	87.9	—
SVMpsi†	78.7	80.7	82.4	87.4	64.5

Jnet*, BRNNs, and SVMpsi methods are based on the PSI-BLAST profiles. Results for SVMpsi* are obtained for the same data set (i.e., RS126, with PHDacc). Results for SVMpsi† are obtained from KP480 data set. Combined results of 7-fold cross validation are shown, except that the results of BRNNs are obtained from 3-fold cross-validation. PHDacc results are from Rost and Sander[10]; Jnet results are from Cuff and Barton (Jnet*: PSI-BLAST profiles, Jnet†: combined PSI-BLAST and HMMER2 profiles)[6]; BRNNs results are from Pollastri et al.[13]; and NETASA results are from Ahmad and Gromiha.[26]

for the 3-state model. We obtained 78.7% accuracy for the 2-state model and 62.4% accuracy for the 3-state model using window size of 15. The improvement was 6.7% for the 2-state model and 8.4% for the 3-state model.

Manesh et al.[15] reported 70.0% prediction accuracy for the 2-state model with threshold 9%, and 58.1% for the 3-state model with thresholds 9%; 16% using information theory with a set of 215 protein sequences used by NETASA.[26] We discuss only these values for fair comparisons, although they also reported other results by calculating accessible surface area (ASA) instead of DSSP. PHDacc[10] reported 74.8% accuracy for the 2-state model with threshold 16%, and it showed that 86% of the completely buried sites were correctly predicted as having 0% relative accessibility for the RS126 data set. SVMpsi achieved 77.8% accuracy for the 2-state model, with threshold 16% for the same RS126 data set. Jnet[6] reported 75.0% prediction accuracy when the relative solvent accessibility threshold is 25% between buried and exposed, and 86.6% for fully buried residues using PSI-BLAST[22] profiles. We cannot directly compare Jnet and SVMpsi since Jnet used the CB480 data set, which is slightly different from the KP480 data set. Recently, Pollastri et al[13] achieved 77.2% for the 2-state model with a threshold of 25% by BRNNs (bidirectional recurrent neural networks), as well as PSI-BLAST profiles. They claimed that the improvement is due both to the larger training sets and the BRNN architectures, which can capture long-range interactions.

Table III shows that the methods using PSI-BLAST (i.e., Jnet,[6] BRNNs,[13] and SVMpsi[8]) were able to obtain much better prediction accuracies than other methods. The SVMpsi method with long-range interaction 3D local descriptor is comparable to or better than the other methods in predicting protein relative solvent accessibility. Though a direct comparison of our method with BRNNs is difficult, due to the fact that different training sets are used in the tests, both the BRNNs and SVMpsi, as methods that consider long-range interactions, produce relatively good prediction results.

We performed some additional experiments to test the influence of different factors on the prediction accuracy improvement. There are three factors (i.e., SVMs, long-range interaction 3D local descriptor, and kernel feature weight scheme). When we used only SVMs with $21 \times w$ components for each input vector without H_c, E_r, and H_r, the cross-validated prediction accuracies for KP480 data set with the 2-state models (25%, 16%, 5%, 0% thresholds) and the 3-state model (9%; 36% threshold) were 77.5%, 77.7%, 79.8%, 86.3%, and 61.9%, respectively. The results are lower than the SVMpsi† results in Table III that were achieved by taking advantage of all three factors. We also tested using 3D local descriptors and SVMs without a kernel feature weight scheme to estimate the contribution of the feature weight scheme. It was found that the contribution of the feature weight scheme was relatively small (less than about 0.2%) or sometimes not significant at all, since the results were almost the same as the SVMpsi† results. It shows that our prediction accuracy improvement was mainly due to SVMs and 3D local descriptor. We expect additional improvement with a more reliable long-range interaction matrix generated from a larger number of proteins and more accurate remote contact prediction methods.

The SVMpsi method has already been shown to achieve a good performance for protein secondary structure prediction in our previous work.[8] In this article, we present the first application of the SVM approach to predict protein relative solvent accessibility using a novel long-range interaction 3D local descriptor that contains hydrophobicity information for the current window, possibility of remote contact, and hydrophobicity for the expected remote contact window. While the protein secondary structure tends to be determined by local sequence environment, the solvent accessibility is much more related to the tertiary interactions between residues far apart in the sequence, but close in 3D space.

ACKNOWLEDGMENTS

We would like to thank the University of Minnesota Supercomputing Institute (MSI) for intensive numerical computing. We also thank Prof. Thorsten Joachims for making SVMlight software so widely available, James A. Cuff and Prof. Geoffrey J. Barton for providing the data set, and Prof. David T. Jones for PFILT software and his kind help.

REFERENCES

1. Baker D, Sali A. Protein structure prediction and structural genomics. Science 2001;294:93–96.
2. Rost B, Sander C. Prediction of secondary structure at better than 70% accuracy. J Mol Biol 1993;232:584–599.
3. King RD, Sternberg MJE. Identification and application of the concepts important for accurate and reliable protein secondary structure prediction. Protein Sci 1996;5:2298–2310.
4. Frishman D, Argos P. Seventy-five percent accuracy in protein secondary structure prediction. Proteins 1997;27:329–335.
5. Jones DT. Protein secondary structure prediction based on position-specific scoring matrices. J Mol Biol 1999;292:195–202.
6. Cuff JA, Barton GJ. Application of multiple sequence alignment

profiles to improve protein secondary structure prediction. Proteins 2000;40:502–511.
7. Hua SJ, Sun ZR. A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. J Mol Biol 2001;308:397–407.
8. Kim H, Park H. Protein secondary structure prediction based on an improved support vector machines approach. Protein Eng. 2003;16:553–560.
9. Holbrook SR, Muskal SM, Kim SH. Predicting surface exposure of amino acids from protein sequence. Protein Eng 1990;3(8):659–665.
10. Rost B, Sander C. Conservation and prediction of solvent accessibility in protein families. Proteins 1994;20:216–226.
11. Pascarella S, Persio RD, Bossa F, Argos P. Easy method to predict solvent accessibility from multiple protein sequence alignments. Proteins 1999;32:190–199.
12. Li X, Pan XM. New method for accurate prediction of solvent accessibility from protein sequence. Proteins 2001;42:1–5.
13. Pollastri G, Baldi P, Fariselli P, Casadio R. Prediction of coordination number and relative solvent accessibility in proteins. Proteins 2002;47:142–153.
14. Thompson MJ, Goldstein RA. Predicting solvent accessibility: higher accuracy using Bayesian statistics and optimized residue substitution classes. Proteins 1996;25:38–47.
15. Naderi-Manesh H, Sadeghi M, Arab S, Movahedi AAM. Predicting of protein surface accessibility with information theory. Proteins 2001;42:452–459.
16. Heiler M. Optimization criteria and learning algorithms for large margin classifiers. Thesis, University of Mannheim, Germany; 2002.
17. Kabsch W, Sander C. A dictionary of protein secondary structure. Biopolymers 1983;22:2577–2637.
18. Qian N, Sejnowski TJ. Predicting the secondary structure of globular proteins using neural network models. J Mol Biol 1988; 202:865–884.
19. Fiser A, Simon I. Predicting the oxidation state of cysteins by multiple sequence alignment. Bioinformatics 2000;3:251–256.
20. Dosztányi Z, Fiser A, Simon I. Stabilization centers in proteins: identification, characterization and predictions. J Mol Biol 1997; 272:597–612.
21. Rose GD, Geselowitz AR, Lesser GJ, Lee RH, Zehfus MH. Hydrophobicity of amino acid residues in globular proteins. Science 1985;229:834–838.
22. Altschul SF, Madden TL, Schaffer AA, Zhang JH, Zhang Z, Miller W, Lipman DJ. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 1997;25: 3389–3402.
23. Bairoch A, Apweiler R. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. Nucleic Acids Res 2000;28:45–48.
24. Jones DT, Taylor WR, Thornton JM. A model recognition approach to the prediction of all-helical membrane protein structure and topology. Biochemistry 1994;33:3038–3049.
25. Jones DT, Swindells MB. Getting the most from PSI-BLAST. Trends Biochem Sci 2002;27:161–164.
26. Ahmad S, Gromiha MM. NETASA: neural network based prediction of solvent accessibility. Bioinformatics 2002;18:819–824.
27. Burges CJC, Schölkopf B. Improving the accuracy and speed of support vector learning machines In: Mozer M, Jordan M, Petsche T, editors. Advances in neural information processing systems. Cambridge, MA: MIT Press; 1997. p 375–381.
28. Burges CJC. A tutorial on support vector machines for pattern recognition. Data Min Knowl Disc 1998;2:121–167.
29. Cristianini N, Shawe-Taylor J. Support vecor machines and other kernel-based learning methods. Cambridge University Press, Cambridge; 2000.
30. Osuna E, Freund R, Girosi F. Support vector machines: training and applications. Tech. Rep. AI Memo Now 1602. Cambridge, MA: MIT AI Laboratory; 1997.
31. Vapnik V. The nature of statistical learning theory. New York: Springer-Verlag; 1995.
32. Vapnik V. Statistical learning theory. New York: Wiley; 1998.
33. Hua SJ, Sun ZR. Support vector machine approach for protein subcellular localization preduction. Bioinformatics 2001;17:721–728.
34. Cai YD, Liu XJ, Xu XB, Zhou GP. Support vector machines for predicting protein structural class. BMC Bioinformatics 2001; 2(3).
35. Ding CHQ, Dubchak I. Multi-class protein fold recognition using support vector machines and neural networks. Bioinformatics 2001;17:349–358.
36. Leslie C, Eskin E, Noble WS. The spectrum kernel: a string kernel for SVM protein classification. In: Proceedings of the Pacific Symposium on Biocomputing, 2002. p 564–575.
37. Brown MPS, Grundy WN, Lin D, Cristianini N, Sugnet CW, Furey TS, Ares M, Haussler D. Knowledge-based analysis of microarray gene expression data by using support vector machines. Proc Natl Acad Sci USA 2000;97:262–267.
38. Mukherjee S, Tamayo P, Mesirov JP, Slonim D, Verri A, Poggio T. Support vector machine classification of microarray data. Tech. Rep. AI Memo No. 1677. Cambridge, MA: MIT; 1999.
39. Furey TS, Cristianini N, Duffy N, Bednarski DW, Schummer M, Hauessler D. Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics 2000;16:906–914.
40. Guyon I, Weston J, Barnhill S, Vapnik V. Gene selection for cancer classification using support vector machines. Mach Learn 2002;46: 389–422.
41. Bock JR, Gough DA. Predicting protein–protein interactions from primary structure. Bioinformatics 2001;17:455–460.
42. LeCun Y, Jackel LD, Bottou L, Brunot A, Cortes C, Denker JS, Drucker H, Guyon I, Muller UA, Sackinger E, Simard P, Vapnik V. Comparison of learning algorithms for handwritten digit recognition In: Fogelman F, Gallinari P, editors. International Conference on Artificial Neural Networks, 1995. p 53–60.
43. Blanz V, Scholkopf B, Bulthoff H, Burges C, Vapnik VN, Vetter T. Comparison of view-based object recognition algorithms using realistic 3D models In: von der Malsburg C, von Seelen W, Vorbruggen JC, Sendhoff B, editors. Proceedings of International Conference on Artificial Neural Networks (ICANN 96), 1996. p 251–256.
44. Pontil M, Verri A. Support vector machines for 3D object recognition. IEEE Trans Pattern Anal 1998;20:637–646.
45. Schmidt M. Identifying speakers with support vector networks. In: Proceedings of Interface 96, Sydney, Australia, 1996.
46. Osuna E, Freund R, Girosi F. Training support vector machines: an application to face detection. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997. p 130–136.
47. Joachims T. Text categorization with support vector machines: learning with many relevant features. In: Proceedings of the European Conference on Machine Learning. Berlin: Springer; 1998. p 137–142.
48. Lodhi H, Saunders C, Shawe-Taylor J, Cristianini N, Watkins C. Text classification using string kernels. J Mach Learn Res 2002;2: 419–444.
49. Kim H, Howland P, Park H. Dimension reduction in text classification using support vector machines. J of Mach Learn Res 2003. Forthcoming.
50. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE. The Protein Data Bank. Nucleic Acids Res 2000;28:235–242.