A TWO-STAGE SVM ARCHITECTURE FOR PREDICTING THE DISULFIDE BONDING STATE OF CYSTEINES

Paolo Frasconi Andrea Passerini
Alessandro Vullo
Dipartimento di Sistemi e Informatica
Università di Firenze, Italy
Phone: +39 055 4796 362
Fax: +39 055 4796 363
Email: {paolo,passerini,vullo}@dsi.unifi.it
Web: http://www.dsi.unifi.it/

Abstract. Cysteines may form covalent bonds, known as disulfide bridges, that have an important role in stabilizing the native conformation of proteins. Several methods have been proposed for predicting the bonding state of cysteines, either using local context or using global protein descriptors. In this paper we introduce an SVM based predictor that operates in two stages. The first stage is a multi-class classifier that operates at the protein level. The second stage is a binary classifier that refines the prediction by exploiting local context enriched with evolutionary information in the form of multiple alignment profiles. The prediction accuracy of the system is 83.6% measured by 5-fold cross validation, on a set of 716 proteins from the September 2001 PDB Select dataset.

INTRODUCTION

Cysteines are one of the twenty amino acids that constitute proteins. The oxidized form of cysteines plays a fundamental role in the stabilization process of the native conformation of proteins. The covalent bonds formed by cysteines, known as disulfide bridges, may connect very distant portion of the sequence. The location of these bonds is a very informative constraint on the conformational space, and the associated information represents a significant step towards folding or understanding structural properties of the protein. Prediction of disulfide bridges from sequence is thus one of the important (and difficult) tasks in structural genomics. Recent works in this area suggest methodologies based on two steps. First, the disulfide-bonding state of each cysteine is predicted (a binary classification problem) [6, 7, 13]. Subsequently, once candidate cystines are known, other algorithms can be used

0-7803-76161/02/$17.00 (c) 2002 IEEE

to predict the actual location of disulfide bridges [5]. In this paper we are interested in the first step. Currently available predictors are all based on neural network approaches.

The program CYSPRED developed by Fariselli et al. [6] (accessible at `http://gpcr.biocomp.unibo.it/predictors/cyspred/`), uses a neural network with no hidden units, fed by a window of $2k + 1$ residues, centered around the target cysteine. Each element of the window is a vector of 20 components (one for each amino acid) obtained from multiple alignment profiles. This method achieved 79% accuracy (correct assignment of the bonding state) measured by 20-fold cross validation and using a non-redundant set of 640 high quality proteins from PDB Select [8] of October 1997. Accuracy was boosted to 81% using a jury of six networks.

The program CYSREDOX, later developed by Fiser & Simon [7] (accessible at `http://pipe.rockefeller.edu/cysredox/cysredox.html`) achieves state-of-the-art performance by exploiting the observation that cysteines and half cystines[1] rarely co-occur in the same protein. The important criterion in [7] is that if a larger fraction of cysteines are classified as belonging to one oxidation state, then all the remaining cysteines are predicted in the same state. The accuracy of this method is as high as 82%, measured by a jackknife procedure (leave-one-out applied at the level of proteins) on a set of 81 protein alignments.

More recently, Mucchielli-Giorgi et al. [13] have proposed a predictor that exploits both local context and global protein descriptors (normalized statistics based on amino acid frequencies, protein size, and number of cysteines). One interesting finding in [13] is that prediction of covalent state based on global descriptors is more accurate (77.7%) than prediction based on local descriptors alone (67.3%). This is not surprising in the light of the results presented in [7] because when using global descriptors all the cysteines in a given protein are deemed to be assigned to the same state. Thus a good method for classifying proteins in two classes is also a good method for predicting the bonding state of each cysteine. The effect of local context however is not negligible: results in [13] show that 79.3% accuracy can be achieved by using an input vector joining global and local descriptors (results in this case are measured by 5-fold cross-validation on a set of 559 proteins from Culled PDB). Although results are not directly comparable because different datasets are used, the performance levels attained in [6] and [7] suggest that multiple alignment profiles are more discriminative than frequency-based descriptors when prediction is based on a local window only.

Starting from the above observations, in this paper we propose a novel approach for exploiting the key fact that cysteines and half cystines rarely co-occur. Classification is achieved in two stages. The first classifier predicts the type of protein based on the whole sequence. Classes in this case are "all", "none", or "mix", depending whether all, none, or some of the cysteines in the protein are involved in disulfide bridges. The second binary classifier is then trained to selectively predict the state of cysteines for proteins assigned

[1] a cystine is the dimer formed by a pair of disulfide-bonded cysteines.

to class "mix", using as input a local window with multiple alignment profiles. The overall model is implemented as a probabilistic combination of support vector machines, as detailed in the remainder of the paper.

TWO-STAGE CLASSIFICATION OF CYSTEINES

Let $Y_{i,t}$ be a binary random variable associated with the bonding state of cysteine at position t in protein i. By W_t^k we denote the context of cysteine t (a window of size $2k+1$ centered around position t) enriched with evolutionary information in the form of multiple alignment profiles. Moreover, let D_i denote a global set of attributes (descriptors) for protein i. We are interested in building a model for $P(Y_{i,t} = 1|D_i, W_t^k)$.

For each protein, let C_i be a three-state variable that represents the propensity of the protein to form disulfide bridges. The possible states for C_i are "all", "none", and "mix", depending whether all, none, or some of the cysteines in the protein are involved in disulfide bridges. After introducing C_i, the model can be decomposed as follows:

$$P(Y_{i,t}|D_i, W_t^k) = \sum_{C_i} P(Y_{i,t}|D_i, W_t^k, C_i)P(C_i|D_i, W_t^k). \quad (1)$$

We can simplify the above model by introducing some conditional independence assumptions. First, we assume that the type of protein C_i depends only on its descriptor: $P(C_i|D_i, W_t^k) = P(C_i|D_i)$. Second, we simplify Equation

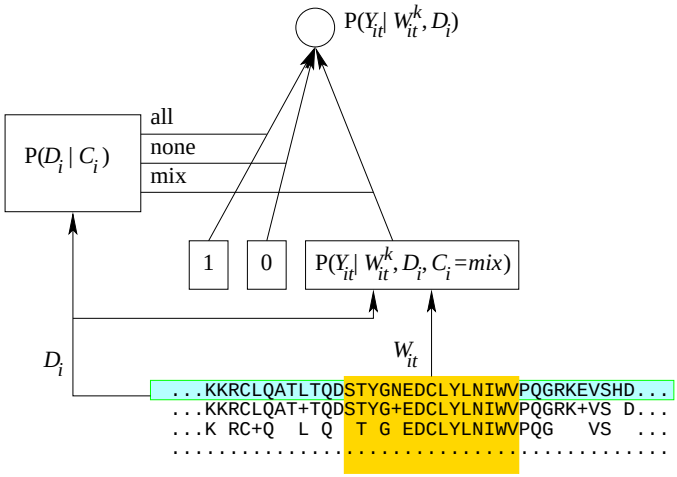

Figure 1: The two-stage system. The protein classifier on the left uses a global descriptor based on amino acid frequencies. The local context classifier is fed by profiles derived from multiple alignments.

1 by remembering the semantics of C_i:

$$\begin{array}{l} P(Y_{i,t} = 1|D_i, W_t^k, C_i = \text{all}) = 1 \\ P(Y_{i,t} = 1|D_i, W_t^k, C_i = \text{none}) = 0 \end{array} \quad (2)$$

(this can be seen as a particular form of context-specific independence [2]). As a result, the model in Equation 1 can be implemented by a cascade of two classifiers. Intuitively, we start with a a multi-class classifier for computing $P(C_i|D_i)$. If this classifier predicts one of the classes "all" or "none", then all the cysteines of the protein should be classified as disulfide-bonded or nondisulfide-bonded, respectively. If instead the protein is in class "mix", we refine the prediction using a second (binary) classifier for computing $P(Y_{i,t}|D_i, W_t^k, C_i = \text{mix})$. Thus the prediction is obtained as follows (see also Figure 1):

$$\begin{array}{rcl} P(Y_{i,t} = 1|D_i, W_t^k) & = & P(Y_{i,t} = 1|D_i, W_t^k, C_i = \text{mix})P(C_i = \text{mix}|D_i) \\ & & +P(C_i = \text{all}|D_i) \end{array} \quad (3)$$

By comparison, note that the method in [7] cannot assign different bonding states to cysteine residues in the same sequence.

IMPLEMENTATION USING PROBABILISTIC SVM

Kernel machines, and in particular support vector machines (SVM), are motivated by Vapnik's principle of structural risk minimization in statistical learning theory [17]. In the simplest case, the SVM training algorithm starts from a vector-based representation of data points and searches a separating hyperplane that has maximum distance from the dataset, a quantity that is know as the margin. More in general, when examples are not linearly separable vectors, the algorithm maps them into a high dimensional space, called *feature space* where they are almost linearly separable. This is typically achieved via a kernel function that computes the dot product of the images of two examples in the feature space. The popularity of SVM is due to the existence of theoretical results guaranteeing that the hypothesis obtained from training data minimizes a bound on the error associated with (future) test data.

The decision function associated with an SVM is based on the sign of the distance from the separating hyperplane:

$$f(\mathbf{x}) = \sum_{i=1}^{N} y_i \alpha_i K(\mathbf{x}, \mathbf{x}_i) \quad (4)$$

where $\mathbf{x}$ is the input vector, $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ is the set of support vectors, $K(\cdot, \cdot)$ is the kernel function, and y_i is the class of the i-th support vector (+1 or -1 for positive and negative examples, respectively).

Probabilistic outputs in SVM

In their standard formulation SVMs output hard decisions rather than conditional probabilities. However, margins can be converted into conditional probabilities in different ways both in the case of binary classification [11, 15] and in the case of multi-class classification [14]. The method used in this paper extends the algorithm presented in [15], where margins in Equation 4 are mapped into conditional probabilities using a logistic function, parameterized by an offset B and a slope A:

$$P(C_i = 1|\mathbf{x}) = \frac{1}{1 + \exp(-Af(\mathbf{x}) - B)} \tag{5}$$

In [15], parameters A and B are adjusted according to the maximum likelihood principle, assuming a Bernoulli model for the class variable. This is extended here to the multi-class case by assuming a multinomial model and replacing the logistic function by a softmax function [3]. More precisely, assuming Q classes, we train Q binary classifiers, according to the one-against-all output coding strategy. In this way, for each point $\mathbf{x}$, we obtain a vector $[f_1(\mathbf{x}), \cdots, f_Q(\mathbf{x})]$ of margins, that can be transformed into a vector of probabilities using the softmax function as follows:

$$g_q(\mathbf{x}) = P(C = q|\mathbf{x}) = \frac{e^{A_q f_q(\mathbf{x}) + B_q}}{\sum_{r=1}^{Q} e^{A_r f_r(\mathbf{x}) + B_r}} \tag{6}$$

The softmax parameters A_q, B_q are determined as follows. First, we introduce a new dataset $\{(f_1(\mathbf{x}_i), \ldots, f_Q(\mathbf{x}_i), \mathbf{z}_i), i = 1, \ldots, m\}$ of examples whose input portion is a vector of Q margins and output portion is a vector $\mathbf{z}$ of indicator variables encoding (in one hot) one of Q classes. As suggested in [15] for the two classes case, this dataset should be obtained either using a hold-out strategy, or a k-fold cross validation procedure. Second we derive the (log) likelihood function under a multinomial model, and search the parameters A_q and B_q that maximize

$$\ell = \sum_i \sum_{q=1}^{Q} z_{q,i} \log g_q(\mathbf{x}_i) \tag{7}$$

where $z_{q,i} = 1$ if the i-th training example belongs to class q and $z_{q,i} = 0$ otherwise.

A fully-observed mixture of SVM experts

While the above method yields multiclass conditional probabilities it does not yet implement the model specified by Equation 3. We now discuss the following general model, that can be seen as a variant of the mixture-of-

experts architecture [9]:

$$P(Y = 1|\mathbf{x}) = \sum_{q=1}^{Q} P(C = q|\mathbf{x})P(Y = 1|C = q, \mathbf{x}) \tag{8}$$

In the above equation, $P(C = q|\mathbf{x})$ is the probability that q is the expert for data point $\mathbf{x}$, and $P(Y = 1|C = q, \mathbf{x})$ is the probability that $\mathbf{x}$ is a positive instance, according to the q-th expert. Collobert *et al.* [4] have recently proposed a different SVM embodiment of the mixture-of-experts architecture, the main focus in their case being on the computational efficiency gained by problem decomposition. Our present proposal for cysteines is actually a simplified case since the discrete variable C associated with the gating network is not hidden[2]. Under this assumption there is no credit assignment problem and a simplified training procedure for the model in Equation 8 can be derived as follows.

Let $f'_q(\mathbf{x})$ denote the margin associated with the q-th expert. We may obtain estimates of $P(Y = 1|C = q, \mathbf{x})$ using a logistic function as follows:

$$p_q(\mathbf{x}) = P(Y = 1|C = q, \mathbf{x}) = \frac{1}{1 + \exp(A'_q f'_q(\mathbf{x}) + B'_q)}. \tag{9}$$

Plugging Equations 6 and 9 into Equation 8, we obtain the overall output probability as a function of $4Q$ parameters: A_q, B_q, A'_q, and B'_q. These parameters can be estimated by maximizing the following likelihood function

$$\ell = \sum_{i=1}^{m} \frac{1 - y_i}{2} \log \left(\sum_{q=1}^{Q} g_q(\mathbf{x}_i) p_q(\mathbf{x}_i) \right) \tag{10}$$

The margins to be used for maximum likelihood estimation are collected by partitioning the training set into k subsets. On each iteration all the $2Q$ SVMs are trained on $k - 1$ subsets and the margins computed on the held-out subset. Repeating k times we obtain as many margins vectors $(f_1(\mathbf{x}), \cdots, f_Q(\mathbf{x}), f'_1(\mathbf{x}), \cdots, f'_Q(\mathbf{x}))$ as training examples. These vectors are used to fit the parameters A_q, B_q, A'_q, and B'_q. Finally, the $2Q$ machines are re-trained on the whole training set.

DATA PREPARATION

All the experiments were carried out using a significant fraction of the current representative set of non homologous protein data bank chains (PDB Select [8]). We extracted the chains in the file 2001_Sep.25 listing 1641 chains with percentage of homology identity less than 25%. From this set we retained only high quality proteins on which the DSSP program [10] does not crash,

[2]Actually the architecture in Figure 1 for cysteines is even simpler since two of the experts output a constant prediction.

determined only by X-ray diffraction, without any physical chain breaks and resolution threshold less than 2.5 Å. The DSSP program was also used to identify disulfide bonds between cysteines. Proteins with interchain bonds were not included in the final dataset containing 716 proteins for a total of 4859 cysteines, 1820 of which in disulfide-bonded state and 3039 in nondisulfide-bonded state. In this dataset, 187 proteins are of type "all", 478 are of type "none", and 51 (i.e. only 7%) of type "mix".

Evolutionary information is expressed in the form of profiles derived from multiple sequence alignments. The actual profiles were extracted from the HSSP database [16].

Input encoding

The descriptor D_i used by the protein classifier is real vector with 24 components, similar to the one used in [13]. The first 20 features are $\log(N_i^j/N^j)$, where N_i^j is the number of occurrences of amino acid type j in protein i and N^j is the number of occurrences of amino acid type j in the whole training set.The 21st feature is $\log(N_i/N_{avg})$ where N_i is the length in residues of sequence i and N_{avg} is the average length of the proteins in the training set. The next two features are N_i^{cys}/N_{max}^{cys} and N_i^{cys}/N_i where N_i^{cys} and N_{max}^{cys} are respectively the number of cysteines in protein i and the maximum number of observed cysteines in the training set. The last feature is a flag indicating whether the cysteine count is odd.

The local input window W_t^k used by the second stage classifier is represented as the set of multiple sequence profile vectors of the residues flanking cysteine at position t. In the experiments, we used a symmetrical window centered at each cysteine varying the window size parameter k from 8 to 10. Note that although the central residue is always a cysteine, the corresponding feature is still taken into account since the profile in this case indicates the degree of conservation of the cysteine. For each of the $2k + 1$ positions we used a vector of 22 components, enriching the 20-components profile with relative entropy and conservation weight.

RESULTS

For each classifier we run a preliminary set of experiments to help the choice of kernel type. In these experiments we used roughly 66% of the proteins for training and the remaining as a validation set. We tried linear, polynomial, and radial basis function (RBF) kernel types. The RBF kernels yielded the best results for the multi-class protein classifier, while binary classification of cysteines was more accurate when using polynomial kernels.

Keeping fixed the type of kernel, we used a 5-fold cross-validation procedure to assess classification performance. The training procedure has been described in the implementation section, but on each fold we used the framework of algorithmic stability recently proposed in [1] as a tool for tuning ker-

Table 1: Summary of experimental results.

	$k = 8$			$k = 9$			$k = 10$		
Method	A	P	R	A	P	R	A	P	R
L	79.00	74.07	67.24	79.40	76.07	66.24	79.49	75.72	67.76
G	75.04	65.42	70.59	76.74	70.57	66.35	77.73	72.71	65.79
L+G	81.28	77.11	70.76	82.96	82.54	69.43	82.50	81.84	69.10
M	81.34	76.92	71.37	83.64	82.09	72.43	82.48	80.81	70.50

nel (hyper)parameters. In particular, we selected RBF radii or polynomial exponents that minimized the generalization error bound based on the leave-one-out error. Softmax parameters (see equations 6 and 9) were estimated by 3-fold cross validation (inside each fold of the outer 5-fold cross-validation), *after* kernel parameter estimation.

Table 1 reports four types of results obtained on the 716 proteins dataset. Each of the three major columns is relative to a different size k of the local window. Minor columns report classification accuracy A, precision P, and recall R. Accuracy (also denoted as Q_2 in other papers) is the fraction of correctly classified cysteines. Precision (or sensitivity) is the fraction of cysteines predicted in the disulfide-bonded state that are actually bonded. Recall (or specificity) is the fraction of disulfide-bonded cysteines that are correctly assigned to their state by the predictor.

Results are reported for four different methods. The first method (L) is a single SVM classifier (polynomial kernel) taking a local window of multiple alignments profiles as input. The second method (G) is a single SVM classifier (RBF kernel) taking as input 24 protein descriptors. It is interesting to note that the local window approach outperforms a predictor based on global descriptors. This result is in contrast to findings in [13], where amino acid frequencies were used as information flanking cysteine residues. It confirms that evolutionary profiles significantly contribute to classification accuracy. The third method (L+G) is a single SVM classifier (polynomial kernel) taking as input both local profiles and global protein descriptors. Like in [13], the combination of local and global features obtains better classification accuracy. The level of 82.96% obtained with a window of size 19 is higher than any previously published results based on neural networks. The fourth and last method (M) is the two-stage classifier of Figure 1 described in the methodology section. We can note that decomposing the task into two subtasks is actually beneficial and the best accuracy is further improved to 83.64%.

CONCLUSIONS

We have proposed a novel method for predicting the bonding state of cysteines, achieving state-of-the-art performance on the most recent set of non-redundant sequences from the Protein Data Bank. There are several obvious directions for further improving this method. First, we have seen that reliable detection of proteins that do not contain mixed types of cysteines is very

important for the overall performance. While the current method employs descriptors based on amino acid frequencies [13], keeping the whole sequence information by means of specialized kernels such as the spectrum kernel [12] may improve accuracy. Moreover, in [13] it was shown that higher prediction accuracy is obtained by training and testing within groups of homogeneous proteins. This result suggests that a mixture-of-experts approach, where the gating network is in charge of determining the protein group, is also likely to yield improved performance.

REFERENCES

[1] O. Bousquet and A. Elisseeff, "Stability and Generalization," **Journal of Machine Learning Research**, vol. 2, 2002.

[2] C. Boutilier, N. Friedman, M. Goldszmidt and D. Koller, "Context-specific independence in Bayesian networks," in **Prof. 12th Conf. on Uncertainty in Artificial Intelligence**, Morgan Kaufmann, 1996, pp. 115–123.

[3] J. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in F. Fogelman-Soulie and J. Hérault (eds.), **Neuro-computing: Algorithms, Architectures, and Applications**, Springer-Verlag, 1989.

[4] R. Collobert, S. Bengio and Y. Bengio, "A Parallel Mixture of SVMs for Very Large Scale Problems," **Neural Computation**, vol. 14, no. 5, 2002.

[5] P. Fariselli and R. Casadio, "Prediction of disulfide connectivity in proteins," **Bioinformatics**, vol. 17, pp. 957–964, 2001.

[6] P. Fariselli, P. Riccobelli and R. Casadio, "Role of Evolutionary Information in Predicting the Disulfide-Bonding State of Cysteine in Proteins," **Proteins**, vol. 36, 340–346 1999.

[7] A. Fiser and I. Simon, "Predicting the oxidation state of cysteines by multiple sequence alignment," **Bioinformatics**, vol. 16, no. 3, pp. 251–256, 2000.

[8] U. Hobohm and C. Sander, "Enlarged representative set of protein structures," **Protein Science**, vol. 3, pp. 522–524, 1994.

[9] R. Jacobs, M. Jordan, S. Nowlan and G. E. Hinton, "Adaptive mixtures of local experts," **Neural Computation**, vol. 3, no. 1, pp. 79–87, 1991.

[10] W. Kabsch and C. Sander, "Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features," **Biopolymers**, vol. 22, pp. 2577–2637, 1983.

[11] J. Kwok, "Moderating the outputs of support vector machine classifiers," **IEEE Transactions on Neural Networks**, vol. 10, no. 5, pp. 1018–1031, 1999.

[12] C. Leslie, E. Eskin and W. Noble, "The spectrum kernel: A string kernel for SVM protein classification," in **Proc. Pacific Symposium on Biocomputing**, 2002, pp. 564–575.

[13] M. Mucchielli-Giorgi, S. Hazout and P. Tuffèry, "Predicting the Disulfide Bonding State of Cysteines Using Protein Descriptors," **Proteins**, vol. 46, pp. 243–249, 2002.

[14] A. Passerini, M. Pontil and P. Frasconi, "From Margins to Probabilities in Multiclass Learning Problems," in F. van Harmelen (ed.), **Proc. 15th European Conf. on Artificial Intelligence**, 2002.

[15] J. Platt, "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods," in A. Smola, P. Bartlett, B. Scholkopf and D. Schuurmans (eds.), **Advances in Large Margin Classifiers**, MIT Press, 2000.

[16] R. Schneider, A. de Daruvar and C. Sander, "The HSSP database of protein structure-sequence alignments," **Nucleic Acids Res.**, vol. 25, pp. 226–230, 1997.

[17] V. Vapnik, **Statistical Learning Theory**, New York: John Wiley, 1998.