Predicting the Disulfide Bonding State of Cysteines with Combinations of Kernel Machines

Alessio Ceroni 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: {aceroni,paolo,passerini,vullo}@dsi.unifi.it
Web: http://www.dsi.unifi.it/

Abstract

Predicting the disulfide bonding state of each cysteine is a step towards location of disulphide bridges in proteins. The solution proposed here is a refinement of a recently introduced method based on a two-stage combination of multiple classifiers. In our approach, the first stage operates at the protein level and attempts to predict whether all, none, or some of the cysteines in the sequence are oxidized. The second stage is a binary classifier that refines the previous prediction by exploiting local context (a fixed-size window of residues flanking the target cysteine). The main contribution of this paper is the (combined) use of a spectrum kernel and the introduction of evolutionary information in the first stage of the system. We show that both implements allow to improve prediction accuracy. Our best system correctly classifies 85% cysteines as measured by 5-fold cross validation, on a set of 716 proteins from the September 2001 PDB Select dataset.

1 Introduction

Cysteines may form covalent bonds, known as disulfide bridges, that may connect very distant portion of a protein sequence. Thus, 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. Before disulfide bridges can be predicted, it is necessary to determine which cysteines in the sequence are actually oxidized, a binary classification problem that has received significant and increased attention during the last few years. Fariselli et al. [5] have proposed a jury of neural networks with no hidden units, fed by a window of $2k + 1$ residues (enriched by evolutionary information), centered around the target cysteine. This method achieved 81% 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 [9] of October 1997. Fiser & Simon [6] later proposed an improvement by exploiting the observation that cysteines and half cystines rarely co-occur in the same protein. Thus, 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 jack-knife procedure (leave-one-out applied at the level of proteins) on a set of 81 protein alignments. Mucchielli-Giorgi et al. [14] have proposed a predictor that exploits both local context and global protein descriptors. Interestingly, they found that prediction of covalent state based on global descriptors was 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 [6] 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 [14] 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). Starting from these observations, we have recently proposed a different approach for exploiting the key fact that cysteines and half cystines rarely co-occur [7]. Classification is achieved by using two cascaded classifier. 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 to class "mix", using as input a local window with multiple alignment profiles. The best accuracy reported in [7] was 83.6%, measured by 5-fold cross validation, on a set of 716 proteins from the September 2001 PDB Select dataset. Shortly after, Casadio et al. [3] have proposed yet another approach where disulfide-bonding state is predicted as in CYSPRED but predictions are then refined using a hidden Markov model trained to recognize the stochastic language that describes the alternate presence of bonding and non-bonding cysteines along the sequence. This improved method achieved the performance level of 88.% correct prediction on PDB select.

In this paper we study two extensions for improving the three-state classifier in the architecture presented in [7]. First, we use kernel machine based on the spectrum kernel [13] that exploits the whole protein sequence as input. Second, we introduce evolutionary information in the form of cystein conservation in multiple alignments. These modifications allow to improve prediction accuracy to 84.5% on the same test sequences studied in [7].

2 Two-stage classification of cysteines

Let us shortly review the method presented in [7]. We denote by $Y_{i,t}$ a binary random variable associated with the bonding state of cysteine at position t in protein i, by W_t^k the context of cysteine t, i.e. a window of size $2k+1$ centered around position t (enriched with evolutionary information in the form of multiple alignment profiles), and by D_i a global set of attributes (descriptors) for protein i.

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. We can now define a model for $P(Y_{i,t}=1|D_i,W_t^k)$ as follows:

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

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 1 by remembering the semantics of C_i:

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

(this can be seen as a particular form of context-specific independence [1]). 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):

$$P(Y_{i,t}=1|D_i,W_t^k) = P(Y_{i,t}=1|D_i,W_t^k,C_i=\text{mix})P(C_i=\text{mix}|D_i)+P(C_i=\text{all}|D_i) \tag{3}$$

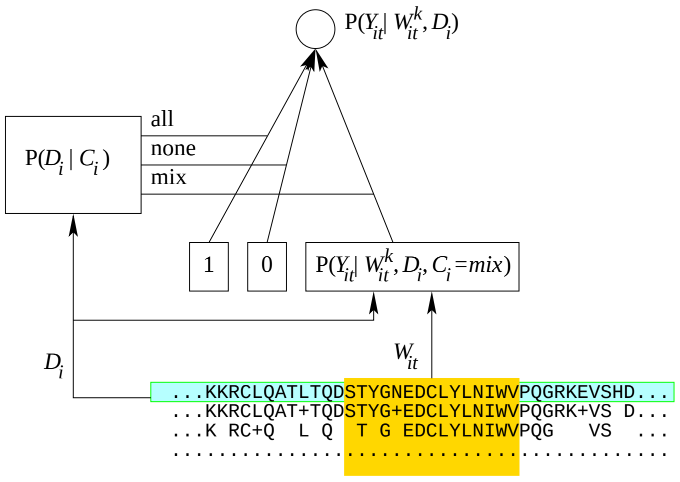
P(Y_it | W^k_it, D_i)
P(D_i | C_i)
all
none
mix
1
0
P(Y_it | W^k_it, D_i, C_i = mix)
D_i
W_it
...KKRCLQATLTQDSTYGNEDCLYLNIWVPQGRKEVSHD...
...KKRCLQAT+TQDSTYG+EDCLYLNIWVPQGRK+VS D...
...K RC+Q L Q T G EDCLYLNIWVPQG   VS ...

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.

By comparison, note that the method in [6] cannot assign different bonding states to cysteine residues in the same sequence.

3 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 [18]. 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:

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

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).

3.1 Spectrum kernel

The k-spectrum of a string s is the set of all the subsequences of s having length k. By $\Phi_k(s)$ we denote the vector associated with s in feature space. The components of $\Phi_k(s)$ count the number of occurrences of each k-subsequence in s (a "bag-of-subsequences" representation of s). Interestingly, descriptors based on amino acid frequencies as defined in [14], basically correspond to the use of a spectrum kernel with $k = 1$. Augmenting the feature space by incorporating short subsequences increases the expressive power of the model and may improve prediction accuracy if k is carefully chosen and enough training sequences are available. Leslie et al. [13] have recently introduced an algorithm based on suffix trees for efficiently computing the dot product in the feature space associated with spectrum kernels (its time complexity is proportional to k and to the sequence length). The experiments reported below have been carried out using our implementation of suffix trees [8].

3.2 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 [12, 16] and in the case of multi-class classification [15]. The method used in this paper extends the algorithm presented in [16], 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 [16], 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 [2]. 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 [16] 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.

3.3 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 [10]:

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

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[1]. 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.

4 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 [9]). 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 [11] does not crash, 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 derived from multiple sequence alignments, obtained in our case from the HSSP database [17].

4.1 Input encoding

The descriptor D_i described in [7] is real vector with 24 components, similar to the one used in [14]. The first 20 components 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 component 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 components 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 component 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

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

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.

4.2 Cysteine conservation

Cysteines tends to be conserved in multiple alignments when they form disulfide bridges. In the experiments reported below, we made available this information to the three-state classifier (none-all-mix) in two alternative ways. First, we defined an extended descriptor with H additional components related to the conservation of cysteines. For $h = 0, \dots, H-1$, the h-th extra component is the fraction of cysteines in the sequence whose multiple alignment conservation falls in the bin $[h/H, (h + 1)/H]$. Second, we defined a special sequential representation of proteins that incorporates evolutionary information. In this representation a protein is a string in an extended alphabet having $19+Z$ symbols, where occurrencies of C (cysteine) are replaced by a special symbol that indicates the degree of conservation of the cysteines in the correspondent positions of multiple alignments. For example if $Z = 2$, one symbol encodes highly (¿50%) conserved cysteines and another one encodes lowly conserved ones.

5 Results

The same settings as in [7] have been used in the present experiments. In particular, polynomial kernels have been choosen for the local experts. The subsequence length for the spectrum kernel was set to the best value of $k = 5$ determined using a validation set. Classification performance was assessed by a 5-fold cross-validation procedure. 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 major column corresponds to a different size k of the local window (from 7 to 10). Minor columns report classification accuracy, precision, and recall. 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 is the fraction of disulfide-bonded cysteines that are correctly assigned to their state by the predictor.

Results are reported for eight different classifiers. The first method (Desc24) is a global SVM classifier (RBF kernel) taking as input 24 protein descriptors, trained on a binary classification task as reported in [7]. In the next row (Spect20) we replaced descriptors by a spectrum kernel operating on strings of 20 symbols with $k = 5$. We can see that using frequencies of substrings of length up to five reduces two-state classification errors by about 7%. The third row (L) corresponds to a single classifier based on a support vector machine with polynomial kernel, that only takes a local window of multiple alignments profiles as input.

In the next five experiments, the local classifier is combined with different global classifiers using the architecture of Section 3.3. In the rows L+Spect20 and L+Desc24 we report results obtained by combining the local classifier with the two global classifiers mentioned above. In the remaining three experiments we incorporated evolutionary information in the input to the global classifier using the two representations explained in Section 4.2. In particular, in $L+Desc29$ we used a global descriptor with 5 extra inputs encoding cysteine conservation, while in L+Spect22 and L+Spect24 we used the spectrum kernel with extended alphabets ($Z = 3$ and $Z = 5$, respectively). The best result (85%) is obtained with the spectrum kernel, 5 symbols encoding cysteine conservation, and a local window of 21 residues.

6 Conclusions

We have shown that global features extracted through a spectrum kernel can improve prediction accuracy compared to global descriptors based on amino acid frequencies. As the baseline accuracy

Table 1: Summary of the experimental results.

	w15			w17			w19			w21		
Method	Acc	Pre	Rec	Acc	Pre	Rec	Acc	Pre	Rec	Acc	Pre	Rec
Desc24	75.04	65.42	70.59	75.04	65.42	70.59	75.04	65.42	70.59	75.04	65.42	70.59
Spect20	76.74	70.57	66.35	76.74	70.57	66.35	76.74	70.57	66.35	76.74	70.57	66.35
L	78.11	73.61	65.32	79.27	75.22	67.12	79.40	76.07	66.24	79.80	76.95	66.23
L+Spect20	83.14	82.16	70.58	83.29	82.36	70.80	82.96	82.54	69.43	82.82	81.95	69.75
L+Desc24	83.37	81.80	71.84	84.05	82.61	73.03	83.60	82.10	72.22	83.21	81.64	71.50
L+Spect22	83.72	78.96	77.66	84.45	80.76	77.28	83.94	79.96	76.80	83.86	79.83	76.74
L+Desc29	83.80	84.42	70.10	84.12	85.08	70.26	84.08	85.27	70.00	84.27	85.42	70.37
L+Spect24	84.37	81.66	75.53	84.82	82.70	75.54	84.92	83.75	74.50	85.05	83.60	75.09

obtained by our method is higher compared to neural network based classifiers, it is reasonable to ask whether HMM refinement of our system will yield overall accuracy over 88%.

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. n [14] 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.

Acknowledgments

We thank Rita Casadio, Piero Fariselli and Gigi Martelli for useful discussions about the methods described in this paper. Sauro Menchetti implemented part of the SVM code used in the experiments.

References

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

[2] J. Bridle. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In F. Fogelman-Soulie and J. Hérault, editors, *Neuro-computing: Algorithms, Architectures, and Applications*. Springer-Verlag, 1989.

[3] R. Casadio, P. Fariselli, and P.L. Martelli. Personal communication. 2002.

[4] R. Collobert, S. Bengio, and Y. Bengio. A parallel mixture of SVMs for very large scale problems. *Neural Computation*, 14(5), 2002.

[5] P. Fariselli, P. Riccobelli, and R. Casadio. Role of evolutionary information in predicting the disulfide-bonding state of cysteine in proteins. *Proteins*, 36, 340–346 1999.

[6] A. Fiser and I. Simon. Predicting the oxidation state of cysteines by multiple sequence alignment. *Bioinformatics*, 16(3):251–256, 2000.

[7] P. Frasconi, A. Passerini, and A. Vullo. A two-stage SVM architecture for predicting the disulfide bonding state of cysteines. In *Proc. of the IEEE Workshop on Neural Networks for Signal Processing*, 2002.

[8] Dan Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology.* Cambridge University Press, 1997.

[9] U. Hobohm and C. Sander. Enlarged representative set of protein structures. *Protein Science*, 3:522–524, 1994.

[10] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

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

[12] J.T. Kwok. Moderating the outputs of support vector machine classifiers. *IEEE Transactions on Neural Networks*, 10(5):1018–1031, 1999.

[13] C. Leslie, E. Eskin, and W.S. Noble. The spectrum kernel: A string kernel for SVM protein classification. In *Proc. Pacific Symposium on Biocomputing*, pages 564–575, 2002.

[14] M.H. Mucchielli-Giorgi, S. Hazout, and P. Tuffèry. Predicting the disulfide bonding state of cysteines using protein descriptors. *Proteins*, 46:243–249, 2002.

[15] A. Passerini, M. Pontil, and P. Frasconi. From margins to probabilities in multiclass learning problems. In F. van Harmelen, editor, *Proc. 15th European Conf. on Artificial Intelligence*, 2002.

[16] J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In A. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers.* MIT Press, 2000.

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

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