Learning to discriminate between ligand bound and disulfide bound cysteines

Andrea Passerini [1] and Paolo Frasconi [1]

[1] Dipartimento di Sistemi e Informatica, Università di Firenze,
50139 Firenze, Italy
E-mail: {passerini,paolo}@dsi.unifi.it

Keywords. Genomics, Annotation, Knowledge representation

Introduction

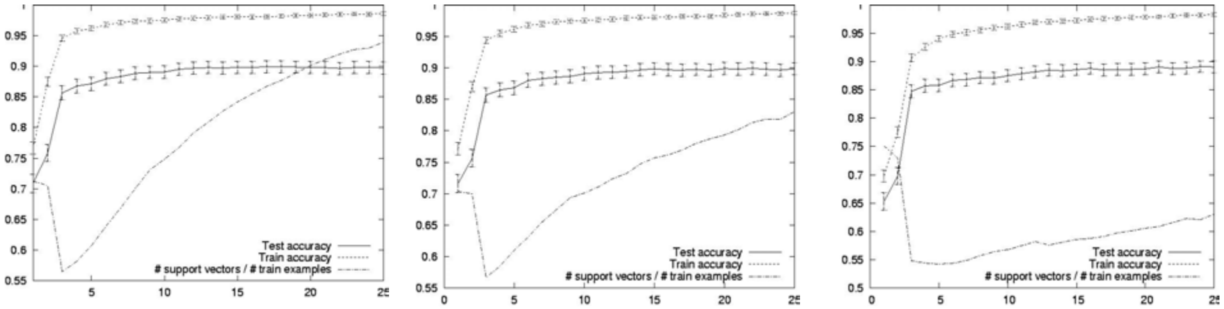
Non-free cysteines that are not involved in the formation of disulfide bridges are very often bound to prosthetic groups that include a metal ion and that play an important role in the function of a protein. The discrimination between the presence of a disulfide bridge (DB) or a metal binding site (MBS) in correspondence of a bound cysteine is often a necessary step during the NMR spectral assignment process of metalloproteins and its automation may significantly help towards speeding up the overall process. Several proteins are known where both situations are in principle plausible and it is not always possible to assign a precise function to each cysteine (see e.g. {2,1,5]).
We formulate the prediction task as a binary classification problem: given a non-free cysteine and information about flanking residues, predict whether the cysteine can bind to a prosthetic group containing a metal ion (positive class) or it is always bound to another cysteine forming a disulfide bridge (negative
class). Firstly, we suggest a nontrivial baseline predictor based on PROSITE pattern hits. Secondly, we introduce a classifier fed by multiple alignment profiles and based on support vector machines (SVM)[3]. We show that the latter classifier is capable of discovering the large majority of the relevant PROSITE patterns, but is also sensitive to signal in the profile sequence that cannot be detected by regular expressions and therefore outperforms the baseline predictor.

Materials and Methods

The data for cysteines involved in DB formation were extracted from PDB, while those for MBS were extracted from SWISS-PROT version 41.23, since PDB does not contain enough examples of metal ligands. In the latter case we included all entries containing at least one cysteine in a MBS, regardless of the annotation confidence. Intra-set redundancy due to sequence similarity was avoided by running the UniqueProt program [6] with hssp distance set to zero. Inter-set redundancy was kept in order to handle proteins with both DBs and MBS. It must be remarked that while inter-set redundancy can help the learning algorithm by providing additional data for training, it cannot favorably bias accuracy estimation since redundant cases should be assigned to opposite classes. We obtained in this way 2860 DB cysteines (in 529 chains) and 758 MBS (in 202 chains). Free cysteines were ignored.
For the data set described above the base accuracy, given by the frequency of the most common class, is 84.4%. In total absence of prior knowledge a predictor that performs better than the baseline is generally considered as successful. However, base accuracy does not account for precision/recall rates which are also needed in order to have a correct view of the classifier performance. In addition, for the task studied in this paper several well known consensus patterns exist that partially encode expert knowledge. For example the 4Fe-4S ferredoxin group is associated with the pattern `C-x(2)-C-x(2)-C-(x3)-C-[PEG]` [7]. It seems reasonable, when possible, to make use of them as a rudimentary prediction tool. Thus, in order to compare our prediction method with respect to a more interesting baseline than the mere base accuracy, we extracted features that consist of PROSITE [4] pattern hits. We found 199 patterns whose matches with the sequences in our data set contain the position of at least one bound cysteine. Many patterns are highly specific but false positives exist and some cysteines match several patterns. Thus a prediction rule based on pattern matches is difficult to craft by hand and we used the program C4.5 to create rules automatically from data. C4.5 induces a decision trees from labeled examples by recursively partitioning the instance space, using a greedy heuristic driven by information theoretic considerations [8].

Results and Discussion

Test performances were calculated by three fold *cross validation*: proteins were divided in three groups, mantaining in each group approximately the same distribution of disulfide bridges and different kinds of MBS. Table 1 reports our experimental results for PROSITE patterns and the polynomial kernel SVM. Results for the polynomial kernel are reported in figure 1(a). Train and test accuracies are plotted for growing size of the context window, with error bars for 95% confidence intervals, together to the fraction of support vectors (SV) over the train examples in the learned models, which is a rough indicator of the complexity of the learned models. The most evident improvement in test accuracy is obtained for a window of size *k*=3, and corresponds to the global minimum in the model complexity curve with about 56% of training examples as SV. Detailed results for such window are reported in table 1(b). A deeper analysis of individual predictions showed that the vast majority of predictions were driven by the presence of a well conserved CXXC pattern, taken as the indicator of a MBS. This explains the high rate of false negatives compared to the total number of negative examples, being most of them cysteines containing the pattern but involved in disulfide bridges, while most of the false positives are MBS missing it. The learned pattern is actually very common for most bindings involving iron-sulfur, iron-nickel and heme groups, and these kinds of MBS are actually predicted with the highest recall. The best accuracy is obtained for a window of size *k*=17, with a strong reduction of false negatives at the cost of a slight increase in the number of MBS predicted as DB. Figure 1(b) shows results for growing size of the context window, for a third degree polynomial kernel with McLachlan similarity matrix. While train and test accuracies are similar to those obtained without the similarity matrix (figure 1(a)), the corresponding models have less SVs, with reductions up to 11\% of the training set. This behaviour is even more evident for the Blosum62 substitution matrix (figure 1(c)) where a slight decrease in test accuracy, still within the confidence interval, corresponds to a reduction up to 30% of the training set. Note that the fraction of SVs over training examples is a loose upper bound on the leave one out error, which is an almost unbiased estimate of the true generalization error of the learning algorithm. These kernels are able to better exploit information on residue similarity, thus obtaining similar performances with simpler models.



(a) Polynomial Kernel

(b) Kernel with McLachlan Matrix

(c) Kernel with Blosum62 Matrix

Fig. 1 SVM results. Test and train accuracies with 95% confidence intervals are plotted, together to the fraction of SVs over the number of training examples, for growing sizes of the window of 2*k*+1 residues profiles around the target cysteine, with *k* going from 1 to 25. Results are averaged over a three fold cross validation procedure

Table 1 (a) decision rules learned by c4.5 from patterns extracted from PROSITE. (b) SVM with a window of size 3. (c) SVM with a window of size 17. P is precision, R is recall

	P	R	DB	MBS
DB	84%	99%	2845	15
MBS	93%	27%	556	202
Acc.	84.2%		Predicted	

	P	R	DB	MBS
DB	84%	99%	2845	15
MBS	93%	27%	556	202
Acc.	84.2%		Predicted	

	P	R	DB	MBS
DB	84%	99%	2845	15
MBS	93%	27%	556	202
Acc.	84.2%		Predicted	

References

[1] E. Balatri, L. Banci, I. Bertini, F. Cantini and S. Ciofi-Baffoni.. *Structure*, 11(11):1431--1443, 2003.

[2] Y.V. Chinenov. *J. Mol. Med.*, pages 239--242, 2000.

[3] C. Cortes and V. Vapnik. *Machine Learning*, 20:1--25, 1995.

[4] L. Falquet et al. *Nucleic Acids Res.*, 30(1):235--238, 2002.

[5] D.N. Heaton, G.N. George, G. Garrison, and D.R. Winge. *Biochemistry*, 40(3):743--751, 2001.

[6] S. Mika and B. Rost. *Nucleic Acids Res.*, 31(13):3789--3791, 2003.

[7] E. Otaka and T. Ooi. *J. Mol. Evol.*, 26(3):257--67, 1987.

[8] J.Ross. Quinlan. *Machine Learning*, 1:81--106, 1986.