Web supplement

"DiANNA 1.1: An extension of the DiANNA web server for ternary cysteine classification"

F. Ferrè[1] and P. Clote[2]

March 10, 2006

Abstract

DiANNA (1,2) is a recent state-of-the-art artificial neural network and web server, which determines the *cysteine oxidation state* and *disulfide connectivity* of a protein, given only its amino acid sequence. Version 1.1 of DiANNA (3) has extended functionality for cysteine oxidation state prediction. By using a support vector machine (SVM) with spectrum kernel, DiANNA 1.1 predicts whether a cysteine is reduced (free in sulfhydryl state), a half-cystine (involved in a disulfide bond) or bound to a metallic ligand. In the latter case, DiANNA predicts the ligand among iron, zinc, cadmium and carbon.

Here we describe the method used for the ternary cysteine state classifier in the web server DiANNA 1.1. For economy of space, this method was not described in (3). Additionally, we show the results of binary classification experiments that support the assessment of the method performance.

1. SVM prediction using string kernels

Support vector machines (SVM) were introduced by Vapnik within the context of a mathematically rigorous statistical learning theory – see (4) for a very clear exposition of this topic. Often demonstrating better prediction accuracy than neural networks, SVMs have become increasingly popular in bioinformatics, with applications ranging

[1] Department of Biology, Boston College, Chestnut Hill, MA 02467, ferref@bc.edu
[2] Corresponding author. Departments of Biology and Computer Science (courtesy appointment), Boston College, Chestnut Hill, MA 02467, clote@bc.edu

from translation initiation site determination (5), remote homology detection in proteins (6), viral protease cleavage site prediction (7), fast computation of Z-scores for minimum free energy of RNA (8), etc.

The theory of SVMs is quickly summarized as follows. Let $X = \{x_1, \ldots, x_n\} \subseteq \mathrm{R}^k$ be a finite set of real-valued vectors, with the property that each x_i is associated to a value $y_i \in \{-1, +1\}$. Positive instances x_i satisfy $y_i = +1$, while negative instances x_i satisfy $y_i = -1$. For example, k could be 100, and each x_i could be a sequence of five amino acids each encoded in *unary* (with alanine encoded by a 1 followed by 19 zeros, arginine encoded by 01 followed by 18 zeros, etc.). Positive vectors could be those 5-mers which occur in an α-helix, in which case, negative vectors are those 5-mers which do not occur in an α-helix. The goal of statistical learning theory is to determine a function $f : X \to \{-1, +1\}$ such that for each $1 \leq i \leq n$, $f(x_i) = y_i$. Generally, one randomly splits the set X into a training set and a test set, where accuracy is measured on the test set.

Following (9), a positive definite, real *kernel* is a function $k : X \times X \to \mathrm{R}$ which is *(i)* symmetric: $k(x, x') = k(x', x)$ and *(ii)* positive definite: $\sum_{i,j} c_i c_j k(x_i, x_j) \geq 0$. It is known that every real positive definite kernel satisfies the property that $k(x, x')$ is the inner product $\langle \Phi(x), \Phi(x') \rangle$, for some mapping $\Phi : X \to H$ from X into a *feature space* $H \subseteq \mathrm{R}^\ell$, which latter is generally of higher dimension with $\ell \gg k$. Intuitively, the kernel value $k(x, x')$ is a measure of the similarity between x and x'.

Finally, by solving an optimization problem, which unlike the case of neural networks, has a unique global optimum with rapid convergence, the desired function f can be defined by

$$f(x) = \mathrm{sng}\left(\sum_{i=1}^{n} y_i \alpha_i k(x, x_i) + b \right).$$

Here the *signum* function satisfies $\mathrm{sgn}(x) = +1$ if $x \geq 0$, and -1 otherwise. A *support vector* is a vector x_i, such that $\alpha_i \neq 0$; clearly the decision procedure $f(x)$ then depends only on the support vectors.

To apply support vector machines to the ternary cysteine classification problem, we must encode amino acid sequences (contents of size w windows) into vectors with

real coordinates. To that end, we use the *spectrum* representation[3] (10) which describes an amino acid sequence by specifying the vector of k-mers which occur; i.e. for peptide p, define $\Phi_k(p) = \langle \phi_a(x) : a \in A^k \rangle$, where $\phi_a(x)$ is the number of occurrences of the k-mer a in p, and A is the set of 1-letter codes of amino acids. In this paper, we obtained the best results when $k = 3$, so that the amino acid sequence p in each size w window is encoded by the vector $\Phi_3(p)$ of 8000 coordinates, giving the number of occurrences of each 3-mer in p. With the spectrum representation, we used the software `libSVM` (11) with a degree 2 polynomial kernel, such that the cost parameter $C = 1$ – see (11) for explanation of these parameters. The radial basis kernel and degree 3 polynomial yielded similar results, while the linear kernel (i.e. the original spectrum kernel of (10)) did not perform quite as well. We tried the same kernels (degree 2 and 3 polynomial kernel and radial basis kernel) when using the mismatch representation (12) and the profile-based spectrum representation (13), which latter makes use of PSIBLAST-derived profiles. Nevertheless, in our experiments, we found that the spectrum representation obtained the best results.

To train and test the SVMs we used 5-fold cross-validation, splitting positive and negative datasets into 5 random subsets of approximatively the same size. Using `libSVM`, the SVM multiclass classifier outputs, for each cysteine in the input sequence, the probability of being a free cysteine (FC), a half-cystine (HC) and ligand-bound (LC). To measure the performance of the algorithm we used the Q_3 score, which is the ratio between correctly predicted examples and the total number of examples. The results - Table 3 of (3) - show that the highest Q_3 score is obtained using for the spectrum representation with degree 2 polynomial kernel (0.76). This is somewhat unexpected since the papers (10) resp. (12,14) report that the mismatch kernel resp. profile-based kernel outperform the spectrum kernel in protein classification experiments.

---

[3]Leslie et al. use the term *spectrum kernel* resp. *mismatch kernel* in (11,13), and Busuttil et al. use the term *profile-based kernel* in (15). More rigorously speaking, these authors actually apply classical kernels (e.g. the linear kernel in (11,13)) for new *representations* of amino acid sequences – the spectrum representation, mismatch representation, profile-based spectrum representation. Note that it follows by the Altschul-Erikson algorithm that there are distinct peptides which have identical spectrum representations. In this paper, after experimentation with polynomial and radial based kernels for the spectrum, mismatch and profile-based spectrum representations, we obtained the best results for degree 2 polynomial kernel for the spectrum representation.

2. Binary Classification

To measure the performance of the algorithm we used the Q_3 score, which is the ratio between correctly predicted examples and the total number of examples. The Q_3 score was used previously for ternary classifier performance estimation, for example for three-states - helix, sheet, coil - secondary structure prediction (15). Nevertheless, since the three classes - LC, HC and FC- are differently represented (25%, 25% and 50%, respectively), it would be possible to obtain a Q_3 of 75% just predicting correctly all the instances of two classes (FC and alternatively LC or HC) and incorrectly all the instances of the remaining class. To check whether our classifier is correctly predicting instances of all three classes, we run the following experiments. First, we train the ternary classifier as described in (3), then we test the classifier performance for treating one class (i.e. LC) as *positive* examples, and the remaining two classes (i.e. HC + FC) as *negative* examples. Thus, we converted the ternary classifier into a binary classifier, for which we can compute the confusion matrix (true positives, true negatives, false positives, false negatives) and the following performance measures:

accuracy, or Q_2

$$\frac{TP+TN}{TP+TN+FP+FN} \text{ or } \frac{TP+TN}{P+N},$$

sensitivity, TP rate (tpr), or Q_c

$$\frac{TP}{P} \text{ or } \frac{TP}{TP+FN},$$

specificity, or Q_{nc}

$$\frac{TN}{N} \text{ or } \frac{TN}{TN+FP} = 1-\frac{TP}{TP+FP},$$

and *Matthew's correlation coefficient*, or *MCC*

$$\frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FN)\cdot(TP+FP)\cdot(TN+FP)\cdot(TN+FN)}}.$$

The FP rate (fpr) is $\frac{TP}{TP+FP} = 1-Q_{nc}$.

The results of these experiments, reported in Table 1, clearly show that the ternary classifier correct predictions are well distributed among the three classes.

Additionally, we trained and tested authentic binary classifiers for LC vs. HC, LC vs. FC, HC vs. FC. The classifiers are SVMs implementing a spectrum representation with degree 2 polynomial kernel. Results are shown in Table 2 and Figure 1. From

these results it appears to be easier to discriminate ligand-bound cysteines from half-cystines than from free cysteines. This observation can imply that ligand-bound cysteine molecular context is more similar to the context of reduced cysteines, and this is an important and somewhat surprising result.

3. References

1. Ferre, F. and Clote, P. (2005) DiANNA: a web server for disulfide connectivity prediction. *Nucleic Acids Res*, **33,** W230-232.
2. Ferre, F. and Clote, P. (2005) Disulfide connectivity prediction using secondary structure information and diresidue frequencies. *Bioinformatics*, **21,** 2336-2346.
3. Ferre, F. and Clote, P. (2006) DiANNA 1.1: An extension of the DiANNA web server for ternary cysteine classification. *under revision for Nucleic Acids Research*.
4. Vapnik, V. (1995) *The nature of statistical learning theory*. Springer, New York.
5. Zien, A., Ratsch, G., Mika, S., Scholkopf, B., Lengauer, T. and Muller, K.R. (2000) Engineering support vector machine kernels that recognize translation initiation sites. *Bioinformatics*, **16,** 799-807.
6. Jaakkola, T., Diekhans, M. and Haussler, D. (1999) Using the Fisher kernel method to detect remote protein homologies. *Proc Int Conf Intell Syst Mol Biol***,** 149-158.
7. Narayanan, A., Wu, X. and Yang, Z.R. (2002) Mining viral protease data to extract cleavage knowledge. *Bioinformatics*, **18 Suppl 1,** S5-13.
8. Washietl, S., Hofacker, I.L. and Stadler, P.F. (2005) Fast and reliable prediction of noncoding RNAs. *Proc Natl Acad Sci U S A*, **102,** 2454-2459.
9. Schölkopf, B. and Smola, A.J. (2002) *Learning with Kernels*. MIT Press, Cambridge, MA.
10. Leslie, C., Eskin, E. and Noble, W.S. (2002) The spectrum kernel: a string kernel for SVM protein classification. *Pac Symp Biocomput***,** 564-575.
11. Fan, R.-E., Chen, P.-H. and Lin, C.-J. (2005) Working set selection using the second order information for training SVM. *Journal of Machine Learning Research*, **6,** 1889-1918.
12. Leslie, C.S., Eskin, E., Cohen, A., Weston, J. and Noble, W.S. (2004) Mismatch string kernels for discriminative protein classification. *Bioinformatics*, **20,** 467-476.
13. Kuang, R., Ie, E., Wang, K., Siddiqi, M., Freund, Y. and Leslie, C. (2004) Profile-based string kernels for remote homology detection and motif extraction. *Proc IEEE Comput Syst Bioinform Conf***,** 152-160.
14. Busuttil, S., Abela, J. and Pace, G. (2004) Support vector machines with profile-based kernels for discriminative protein classification. *Genome Informatics*, **15,** 191-200.
15. Jones, D.T. (1999) Protein secondary structure prediction based on position-specific scoring matrices. *J Mol Biol*, **292,** 195-202.

Experiment	Measure	Score
LC vs. FC+HC	Acc	0.88
	Sen	0.64
	Spe	0.96
	MCC	0.66
HC vs. LC+FC	Acc	0.81
	Sen	0.36
	Spe	0.96
	MCC	0.43
FC vs. LC+HC	Acc	0.76
	Sen	0.89
	Spe	0.63
	MCC	0.55

Table 1. Performance measures for the binary cysteine state prediction using our new 3-state SVM classifier. To produce this data, we took our trained 3-state SVM classifier, which predicts cysteine state as either *free* (FC), *half-cystine* (HC) or *ligand-bound* (LC). Following the suggestion of an anonymous referee, we then combined two of the classes together. Thus in the first 4 lines, we computed the accuracy (Acc), sensitivity (Sen), specificity (Spe) and Matthew's correlation coefficient (MCC) for treating ligand-bound cysteines (LC) as *positive* examples, and non-ligand-bound cysteines (HC + FC) as *negative* examples. Similarly, we computed accuracy, sensitivity, specificity and Matthew's correlation coefficient for HC versus LC + FC, and for FC versus LC + HC.

Experiment	Measure	Score
LC vs. HC	Acc	0.88
	Sen	0.86
	Spe	0.9
	MCC	0.76
	AUC	0.94
LC vs. FC	Acc	0.83
	Sen	0.63
	Spe	0.94
	MCC	0.61
	AUC	0.84
HC vs. FC	Acc	0.83
	Sen	0.7
	Spe	0.92
	MCC	0.64
	AUC	0.88

Table 2. Performance measures for binary cysteine class prediction. Performance measure are accuracy (Acc), sensitivity (Sen), specificity (Spe), Matthew's correlation coefficient (MCC) and area under the ROC curve (AUC).

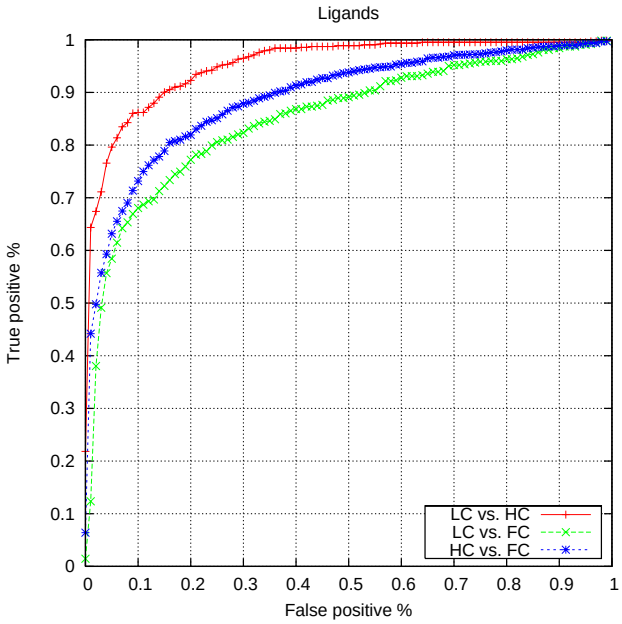

Figure 1. ROC curves for binary classification experiments obtained using the spectrum representation with polynomial kernel of degree 2.