BIOINFORMATICS

Vol. 20 no. 5 2004, pages 653–659
DOI: 10.1093/bioinformatics/btg463

Disulfide connectivity prediction using recursive neural networks and evolutionary information

Alessandro Vullo* and Paolo Frasconi

Department of Systems and Computer Science, Università di Firenze Via di S. Marta 3, 50139-I Firenze, Italy

Received on August 7, 2003; revised on September 3, 2003; accepted on September 26, 2003
Advance Access publication January 22, 2004

ABSTRACT

Motivation: We focus on the prediction of disulfide bridges in proteins starting from their amino acid sequence and from the knowledge of the disulfide bonding state of each cysteine. The location of disulfide bridges is a structural feature that conveys important information about the protein main chain conformation and can therefore help towards the solution of the folding problem. Existing approaches based on weighted graph matching algorithms do not take advantage of evolutionary information. Recursive neural networks (RNN), on the other hand, can handle in a natural way complex data structures such as graphs whose vertices are labeled by real vectors, allowing us to incorporate multiple alignment profiles in the graphical representation of disulfide connectivity patterns.

Results: The core of the method is the use of machine learning tools to rank alternative disulfide connectivity patterns. We develop an *ad-hoc* RNN architecture for scoring labeled undirected graphs that represent connectivity patterns. In order to compare our algorithm with previous methods, we report experimental results on the SWISS-PROT 39 dataset. We find that using multiple alignment profiles allows us to obtain significant prediction accuracy improvements, clearly demonstrating the important role played by evolutionary information.

Availability: The Web interface of the predictor is available at http://neural.dsi.unifi.it/cysteines

Contact: vullo@dsi.unifi.it

1 INTRODUCTION

Proteins which contain cysteine residues are subject to post-translational covalent modifications and cysteines can occur either in oxidized or thiol form. Two oxidized cysteines uniquely pair to form a covalent bond, known as disulfide bridge. As reported by experiments in protein engineering (Matsumura *et al.*, 1989), disulfide bridges can increase the thermodynamic stability of the native state, because they contribute to a reduction of the number of unfolded conformations, thus of the entropic cost of folding a polypeptide chain into its native state (Harrison and Sternberg, 1994; Wedemeyer *et al.*, 2000). Depending on their number and location, these bonds may connect very distant portions of the sequence. Therefore, they add strong structural constraints that can be very helpful towards the *ab-initio* prediction of 3D structure.

In the absence of an experimentally determined structure, sequence archives do not report reliable information relating either the oxidized form of cysteines or disulfide bridge locations. The prediction task can thus be conveniently decomposed in two steps. First, the disulfide-bonding state of each cysteine is predicted from sequence, a binary classification problem that has been solved using several machine learning algorithms such as neural networks, (Fariselli *et al.*, 1999; Fiser and Simon, 2000), support vector machines (Frasconi *et al.*, 2002) and Hidden Markov models (Martelli *et al.*, 2002). Second, the location of disulfide bridges is predicted starting from knowledge of bonded cystines. This paper focuses on the second task which has received relatively scarce attention in the literature. To the best of our knowledge, the only published method (Fariselli and Casadio, 2001) is based on a weighted graph representation of disulfide bridges, where vertices are oxidized cysteines and undirected edges are labeled by the strength of interaction (contact potential) in the associated pair of cysteines. First, stochastic optimization is used to find an optimal set of weights. After a complete labeled graph is obtained, candidate bridges are located by finding the maximum weight perfect matching[1]. The problem can be solved in polynomial time using linear programming. Nevertheless, the computation of contact potentials is a time consuming process. In a subsequent improvement (Fariselli *et al.*, 2002), neural network predictions were used for labeling edges with cysteines pairwise interaction values, increasing the predictive power and concomitantly reducing the training time. This method achieves satisfactory results for the simplest cases (four and occasionally six oxidized cysteines).

*To whom correspondence should be addressed.

[1] A perfect matching of a graph (V, E) is a subset $E' \subseteq E$ such that each vertex $v \in V$ is met by only one vertex.

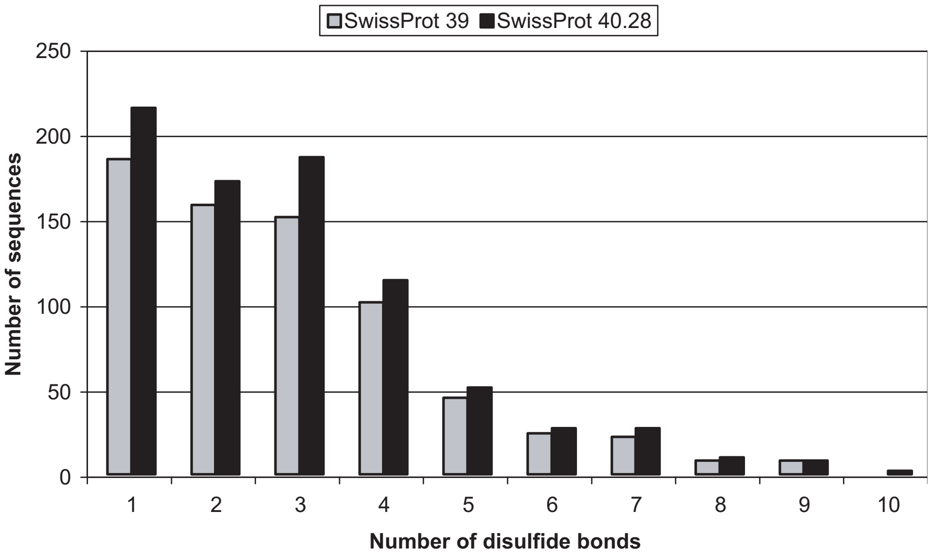

Fig. 1. Distribution of cysteine-rich sequences in two subsequent releases of the SWISS-PROT archive. Chains are grouped according to the number of disulfide bonds.

The method we propose in this paper is based on extended recursive neural networks (RNN) (Frasconi *et al.*, 1998), a connectionist model which allow us to formulate classification and regression tasks on structured data, like the graphs representing disulfide connectivity patterns. The network is trained to score candidate graphs according to a similarity metric with respect to the correct graph. During prediction, the score computed by the network is used to exhaustively explore the space of candidate graphs. We show how our method can easily incorporate and effectively exploit evolutionary information and how it can efficiently deal with a broad spectrum of sequences for the disulfide bridge prediction problem.

2 SYSTEM AND METHODS

2.1 The datasets of protein sequences

In order to compare our method to the alternative algorithm described in Fariselli and Casadio (2001), Fariselli *et al.* (2002), we replicated the same experimental setting. In particular, we extracted the same set of sequences from the SWISS-PROT database release no. 39 (SP39), October 2000 (Bairoch and Apweiler, 2000). We applied the same filtering procedure in order to include only high quality and experimentally verified intra-chain disulfide bridge annotations. Also, the experiments were carried out by excluding from the datasets all the chains having more than 10 oxidized cysteines. Less than 20% of SWISS-PROT sequences have more that five disulfide bridges (Fig. 1).

Table 1 reports the number of sequences used in our experiments, grouped by the number of disulfide bonds B and by structural classification taken from the SCOP database (Murzin *et al.*, 1995), latest release (1.63, May 2003). As can be seen from Table 1, most of the proteins in dataset have not yet been classified.

Table 1. Number of chains in the experimental dataset, grouped by number of disulfide bridges (B) and topology class

Fold type	$B = 2$	$B = 3$	$B = 4$	$B = 5$	$B = 2 \ldots 5$
α	6	6	4	1	17
β	13	9	4	2	28
α/β	5	3	1	1	10
$\alpha + \beta$	9	9	10	0	28
Small proteins	0	21	18	4	43
Peptides	8	2	1	0	11
Unclassified	115	96	61	37	309
All	156	146	99	45	446

2.2 Prediction of the location of disulfide bridges

A disulfide connectivity pattern has a simple representation in terms of an undirected graph $G = (V, E)$. The vertex set V represents the set of bonded cysteines and an edge $e \in E$ corresponds to a disulfide bridge between its adjacent cysteines. Admissible vertex and edge sets are constrained because an even number of intra-chain bonded cysteines is required and a cysteine can be bridged to only one different cysteine. Thus, we have $|V| = 2B$, $|E| = B$ and $degree(v) = 1$ for any $v \in V$ (perfect matching), where B denotes the number of disulfide bonds in a chain.

We introduce a simple formulation for the problem of predicting the correct connectivity pattern for a given disulfide bonded chain: find the best possible candidate as given by a suitable scoring function. This function maps undirected graphs to real numbers. Let $G^\star = (V, E^\star)$ denote the target connectivity pattern. Let $\mathcal{G}$ be the set of candidate solutions and let $s(E, V) : \mathcal{G} \longmapsto [0, 1]$ be a scoring function mapping $G \in \mathcal{G}$ into [0,1] and satisfying the following assumptions:

1. $s(E, V) = 1$ iff $E = E^\star$;
2. for every pair (E_1, E_2) of edge sets,
   $|E_1 \cap E^\star| \geq |E_2 \cap E^\star| \Rightarrow s(E_1, V) \geq s(E_2, V)$

The function $s(G)$ induces a partial order relation over the set of candidate pattern graphs sharing the same vertex set V. In other words, if $s(G) > s(G')$ then $G \succ G'$. Given $s(E, V)$ or an approximation of this function, a pattern can be predicted by a simple procedure enumerating all possible candidates and giving as output one graph with maximal score. The predicted pattern $\tilde{G} = (V, \tilde{E})$ is formally computed as:

$$\tilde{E} = \arg\max_{E \in \mathcal{E}} s(E, V) \qquad (1)$$

where $\mathcal{E}$ is the set of possible edge sets satisfying the given perfect matching constraints.

It can be easily shown that the function

$$s^\star(E) = \frac{|E \cap E^\star|}{|E|} \qquad (2)$$

satisfies the required assumptions and can be effectively used in the procedure of Equation (1). The given scoring function represents the fraction of correct pairs in the candidate solution. The definition of Equation (2) implies that $s\colon \mathcal{G} \longmapsto [0, 1]$ is neither injective nor onto $[0, 1]$. The codomain is the finite discrete set $\{s_0, \ldots, s_{B-2}, s_{B-1}\}$, with $s_i = i/B$, $i = 0 \ldots B - 2$ and $s_{B-1} = 1$. In this case, many different candidates are mapped into the same value, but still we have that $G = G^\star$ if $s_{B-1} = 1$. Therefore, given the true or a nearly perfect function, the search guarantees to find the target solution.

To analyze the computational complexity involved, first observe that we need to generate the whole set $\mathcal{E}$ of possible solutions. Given a chain with $|V| = 2B = n$ cysteines, the size of this set is

$$(n - 1)!! = \prod_{i \leq n/2} (2i - 1) = \frac{n!}{2^{n/2}(n/2)!}$$

By the last equality, it holds that

$$\left(\frac{n}{4}\right)^{n/2} < (n - 1)!! < \left(\frac{n}{2}\right)^{n/2},$$

thus $\Omega[(\sqrt{n}/2)^n]$ steps are necessary to compute $\mathcal{E}$. Each step requires the evaluation of the function s for the current candidate. Assuming the evaluation of s takes time linear in n, it follows that the algorithm takes time at least $\Theta[n(\sqrt{n}/2)^n]$. This computational complexity limits the application of the algorithm only to chains with few bridges (1–5 bridges). As previously noted by Fariselli and Casadio (2001), this is not a severe problem, since the constraint $B \leq 5$ is satisfied for most of the chains of interest (Fig. 1).

3 ALGORITHM

The apparent simplicity for finding $G^\star$ with the above procedure is due to the existence of an oracle that is able to compute $s(E, V)$ for every candidate graph: this requires the knowledge of $E^\star$, a kind of information which is obviously unknown at prediction time. In realistic situations, the scoring function for the algorithm of Equation (1) must rely only on V to assign scores to candidate solutions. In the following we propose a connectionist model capable of learning $s(E, V)$ from examples of known disulfide bond patterns.

3.1 Bi-recursive neural network architecture

A labeled graph on a set $\mathcal{X}$ is a pair $G = (V, E)$, with $E \subset V \times V$, and with a function $x\colon V \to \mathcal{X}$ that associates a label x_v with each vertex $v \in V$. The set of all finite size labeled graphs on $\mathcal{X}$ is denoted $\mathcal{X}^\#$. Classification or regression tasks for which the input portion consists of a labeled graph can be formulated as a mapping from $\mathcal{X}^\#$ to a set of categories (classification) or to real numbers (regression). Since graphs have variable size, regression in this case need to be represented as the composition of two functions, a mapping $F\colon \mathcal{X}^\# \to \Phi$ that transforms input graphs to an intermediate representation in a vector space Φ, and a mapping $g\colon \Phi \to I\!R$ from vectors to real numbers.

The solution used in this paper is based on a generalization of RNN. In this class of models, the mapping $F\colon \mathcal{X}^\# \to \Phi$ is adaptive and Φ is a low-dimensional space. This is in contrast to alternative approaches based for example on convolution kernels (Haussler, 1999), where Φ is a high-dimensional (or infinite dimensional) feature space and F is a fixed map.

The theory developed in Frasconi *et al.* (1998), briefly summarized below, holds in the case of directed ordered acyclic graphs (DOAG) with bounded connectivity and that possess a supersource. Ordered in this context refers to the existence of a total order defined on the set of children of each vertex. A supersource r is a vertex having the property that for every other vertex $v \in V$ there exists a directed path from r to v. Under these assumptions, F can be written recursively as follows. For each vertex v, we introduce a representation vector $\phi_v \in I\!R^n$ recursively computed as:

$$\phi_v = \begin{cases} 0 & \text{if } \mathrm{ch}_v = \emptyset \quad \text{(Base step)} \\ f(x_v, \boldsymbol{\phi}_{\mathbf{ch}_v}; \vartheta) & \text{otherwise} \quad \text{(Induction)} \end{cases} \qquad (3)$$

where $\boldsymbol{\phi}_{\mathbf{ch}_v} = (\phi_{\mathrm{ch}_v^1}, \ldots, \phi_{\mathrm{ch}_v^k})$ is the k-tuple of labels of v's children, ch_v^i denotes the i-th child of v and k is the maximum outdegree in the class of graphs being considered. In RNNs, functions $f(\cdot)$ and $g(\cdot)$ are implemented by adaptive neural networks with connection weights ϑ and θ, respectively. We assume x_v, the label of vertex v, to be encoded by a real vector in $I\!R^m$. Thus, the network of $f(\cdot)$ has $m + kn$ input units and n output units. The computation in Equation (3) proceeds in a bottom-up fashion, from 'leaf' vertices to the supersource. Since we assume that G is acyclic, propagation order can be obtained by sorting topologically the vertex set. The adaptive mapping from graphs to features is simply accomplished by taking the label at the supersource: $F(G) = \phi_r$. It turns out that for each vertex v, vector ϕ_v is the encoding of the subgraph induced by v and all its descendants.

Training is performed using a set of pairs $\{(G_i, y_i), i = 1, \ldots, N\}$ where $y_i \in [0, 1]$ is the desired output for graph G_i and $g[F(G_i); \theta] = g(\phi_r^i; \theta)$ is its network global output. Parameters θ and ϑ are adjusted by minimizing the error function

$$C(\theta, \vartheta) = \frac{1}{2} \sum_{i=1}^{N} \Big(y_i - g(\phi_r^i; \theta) \Big)^2 \qquad (4)$$

A gradient descent algorithm can be obtained by propagating errors backward in structure (i.e. in a top-down fashion, starting from the supersource and following a reverse topological sort of G) and taking into account the fact that weights are shared across different vertices v in Equation (3).

Graphs describing disulfide connectivity do not match the above framework since they are undirected, disconnected and unordered. However, we can introduce a suitable transformation that converts a disulfide graph $G = (V, E)$ into a DOAG G' having a supersource. The resulting model [bi-recursive neural networks (BiRnns)] was introduced in Vullo and Frasconi (2003) and proved to be effective for the prediction of protein coarse contact maps. First, note that the set of vertices V can be ordered reading the protein sequence from left to right. Edges can be thus oriented so that the source vertex precedes in sequence the target vertex. In this way, G is converted into a directed graph. Moreover, additional sequential edges can be added to connect vertices that are adjacent in sequence. After doing so G is connected and has a supersource. In practice, if we use a model like the one described by Equation (3), the role played by sequential links is to propagate information from left to right. More precisely, the feature vector ϕ_v associated with each half-cystine v would summarize information about all the upstream half-cystines and candidate bridges. Note, however, that dependencies in biological sequences are not unidirectional from left to right. To propagate information in both direction we can duplicate G and transpose the edge set of the copy, as shown in Figure 2. A similar approach already proved to be effective for the prediction of protein secondary structure (Baldi *et al.*, 1999). In our implementation, the recursive computation described by Equation (3) is actually piecewise stationary: $\vartheta_v = \vartheta_u$ if and only if v and u are both in the original graph or in the transposed copy. This means that we have three sets of adjustable weights for function F: one for vertices linked upstream to downstream, one for vertices linked downstream to upstream, and a distinguished set of weights for the supersource. It turns out that ϕ_v ($v \neq r$), is now represented by coupling the outputs of two functions $f(\cdot)$ and $b(\cdot)$ computed by two neural networks respectively with the up-to-downstream and down-to-upstream connection weights.

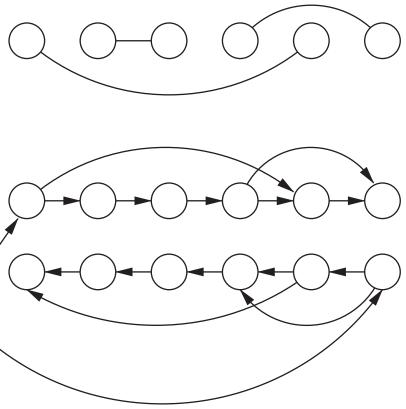

Fig. 2. Transforming a disulfide graph (top) into a supersource DOAG (bottom).

4 IMPLEMENTATION

4.1 Performance measures

Let $\mathcal{D}$ denote a set of proteins and let $G_i = (V_i, E_i)$ and $G_i^\star = (V_i, E_i^\star)$ denote the predicted and correct connectivity patterns for the i-th protein in the set $\mathcal{D}$, respectively. Prediction indices are defined as

$$Q_p = \frac{\sum_{i=1}^{|\mathcal{D}|} \delta(E_i, E_i^\star)}{|\mathcal{D}|}, \; Q_c = \frac{\sum_{i=1}^{|\mathcal{D}|} |E_i \cap E_i^\star|}{\sum_{i=1}^{|\mathcal{D}|} |E_i|} \qquad (5)$$

where $\delta(x, y) = 1$ if $x = y$ and 0 otherwise. The prediction index Q_p is pattern-based and measures the fraction of correctly assigned connectivity patterns. It estimates predictive performance at the protein level, namely the probability that a whole prediction is correct. Note that Q_p can be decreased by proteins having large values of B and yet the location of several bridges could have been predicted correctly. For this reason we also use the complementary couple-based prediction index Q_c, defined as the fraction of correctly predicted disulfide bridges.

4.2 Dataset generation and input data

For each chain in the dataset we generated connectivity graphs by including all the possible $B!!$ connectivity patterns. We obtained 55 683 undirected graphs for the SP39 dataset. Training is performed by assigning a target output score to each connectivity graph. More formally, if $G_i^\star = (V_i, E_i^\star)$ is the i-th instance for a dataset $\mathcal{D}$ with N chains and $\mathcal{E}_i = \{G_{i_1}, \ldots, G_{i_{|\mathcal{E}_i|}}\}$ is the set of candidate patterns for chain i, we used the set of pairs $\{(G_{i_j}, y_{i_j}), i = 1, \ldots, N \; j = 1, \ldots, |\mathcal{E}_i|\}$, where y_{i_j} represents the similarity between the candidate G_{i_j} and the true pattern $G_i^\star$, as computed by Equation (2).

Each vertex in a connectivity graph contains information describing the local environment of the corresponding bonded cysteine. More precisely, we used a window of size $2k + 1$ amino acids centered around the bonded cysteine and we encoded each amino acid position by a vector of 20 components. By comparison, Fariselli and Casadio (2001) used edge-labeled graphs where each edge was annotated by the contact potential between two adjacent vertices. As pointed out in the statistical analysis of Harrison and Sternberg (1994), sequence separation between bonded cysteines and sequence length correlate with specific connectivity patterns. To take advantage of this information we enriched label vectors with two additional features: the normalized sequence position t/L (where t is the cysteine position along the chain and L is the chain length in residues) and the relative sequence length $L/L_{\max}$ (where $L_{\max}$ is the maximum chain length observed in the database).

We consider two different encodings for the positions along the window: single-sequence and profile-based. In the single-sequence case, an all-zero-but-one binary vector identifies the residue type at a given window position. In this case the cysteine in the center of window is not taken into account, being always present and carrying no information. In profile-based encoding, each amino acid position is described by the vector of multiple sequence alignments profile. In this case the central position corresponding to the cysteine is retained. In all our experiments we used a window of size 5 ($k = 2$) resulting in label vectors of dimension 82 (single-sequence) or 102 (profile-based). The position-specific scoring matrix of each chain in the filtered SP39 dataset was created by running two iterations of the PSI-BLAST program (Altschul *et al.*, 1997) against the non-redundant SWISS-PROT + TrEmbl dataset of sequences.

4.3 Experimental protocol

Training and testing of the recursive neural model was performed according to the same 4-fold cross-validation procedure as used in Fariselli and Casadio (2001). In order to automatically stop the four training phases and to control overfitting, we adopted two strategies. In the first one, for each fold we used a validation set composed of 20% of randomly chosen sequences from the original training set. Compared to Fariselli and Casadio (2001), in this case we adopted less favorable training conditions, since test data remains the same. In order to exploit all available training data, in a second setting we used a weight-decay updating strategy.

Networks were trained by optimizing the cost function in Equation (4) using gradient descent—see Frasconi *et al.* (1998) for details on backpropagation in RNN. We used the online stochastic approximation, updating weights after the presentation of each graph.

Preliminary experiments were carried out testing different choices of the adjustable parameters. More precisely, model performances were evaluated varying the architecture of the neural networks implementing the forward and backward transition functions [$f(\cdot)$ and $b(\cdot)$, respectively] and the global output function $g(\cdot)$. All the results showed in the next section are relative to transition networks with 20 output units, no hidden layers and a single sigmoid unit representing the global output function (the score of a graph). During preliminary experiments we found that training separate networks for the sets of proteins having the same number of bridges helps improving generalization. In all the subsequent experiments we therefore used four separate networks for predicting the connectivity of proteins having $B = 2, 3, 4$ and 5, respectively. On average, a whole cross-validation procedure applied to the SP39 dataset took about 12 h on a single processor PIII 1GHz workstation.

Table 2. Comparison among different prediction algorithms

Method	$B = 2$		$B = 3$		$B = 4$		$B = 5$		$B = \{2 \ldots 5\}$	
	Q_p	Q_c	Q_p	Q_c	Q_p	Q_c	Q_p	Q_c	Q_p	Q_c
Frequency	0.58	0.58	0.29	0.37	0.01	0.10	0.00	0.23	0.29	0.32
MC graph-matching	0.56	0.56	0.21	0.36	0.17	0.37	0.02	0.21	0.29	0.38
NN graph-matching	0.68	0.68	0.22	0.37	0.20	0.37	0.02	0.26	0.34	0.42
BiRnn-1 sequence	0.59	0.59	0.17	0.30	0.10	0.22	0.04	0.18	0.28	0.32
BiRnn-1 profile	**0.65**	**0.65**	**0.46**	**0.56**	0.24	0.32	0.08	0.27	**0.42**	**0.46**
BiRnn-2 sequence	0.59	0.59	0.22	0.34	0.18	0.30	0.08	0.24	0.31	0.37
BiRnn-2 profile	**0.73**	**0.73**	**0.41**	**0.51**	0.24	0.37	0.13	0.30	**0.44**	**0.49**

Prediction indices Q_p and Q_c as in Equation (5). Methods as described in section 5. Results in bold indicate a statistically significant difference in performance between sequence-based and profile-based BiRNNs trained according to the same protocol (4-fold paired t-test, $p < 0.01$).

5 RESULTS AND DISCUSSION

Table 2 summarizes the results obtained by running the experiments with the B-specialized networks, as described in the previous section. We report the estimated pattern-based and couple-based prediction indices Q_p and Q_c for each group of chains having the same number of bonds. The rows report prediction results as obtained by the algorithms indicated under the column labeled Method. ‘Frequency’ is a trivial method consisting of always predicting the most frequent pattern observed in the training set. We use it to compare the algorithms against baseline performance in order to evaluate whether a network has learned more than simple first order statistics from the distribution of training instances. For the sake of comparison, the rows labeled MC and NN graph-matching report previous results published respectively in Fariselli and Casadio (2001) and Fariselli *et al.* (2002). The fourth and fifth rows report performances obtained by the BiRNN trained with validation sets and using as input respectively only the information encoded in the sequence (BiRnn-1 sequence) and multiple alignment profiles (BiRnn-1 profile). Similarly, the last two rows show results of the same model trained using weight-decay as stopping procedure.

Table 3. BiRnns pair-based performance

Fold type	$B = 2$			$B = 3$			$B = 4$			$B = 5$			$B = \{2 \dots 5\}$		
	Q_2	Q_{nc}	Q_c	Q_2	Q_{nc}	Q_c	Q_2	Q_{nc}	Q_c	Q_2	Q_{nc}	Q_c	Q_2	Q_{nc}	Q_c
α	0.83	0.88	0.75	0.68	0.80	0.20	0.90	0.94	0.67	1.0	1.0	1.0	0.81	0.88	0.56
β	0.76	0.82	0.64	0.73	0.83	0.32	0.86	0.92	0.50	0.80	0.89	0.10	0.78	0.85	0.47
α/β	0.92	0.94	0.88	0.60	0.75	0.00	0.79	0.88	0.25	0.78	0.88	0.00	0.77	0.86	0.27
$\alpha + \beta$	0.81	0.86	0.72	0.88	0.92	0.69	0.75	0.86	0.14	—	—	—	0.79	0.86	0.45
Small proteins	—	—	—	0.90	0.94	0.75	0.79	0.88	0.27	0.79	0.88	0.05	0.82	0.89	0.37
Peptides	0.83	0.88	0.75	1.0	1.0	1.0	0.71	0.83	0.00	—	—	—	0.85	0.89	0.69
Unclassified	0.83	0.87	0.74	0.80	0.87	0.49	0.82	0.89	0.37	0.85	0.92	0.33	0.82	0.88	0.50
All	0.82	0.87	0.73	0.80	0.88	0.51	0.81	0.89	0.37	0.84	0.91	0.30	0.82	0.88	0.49

Network model trained with multiple alignment profiles. Index Q_2 is the fraction of correctly assigned pairs and Q_{nc} is the fraction of correctly predicted pairs not involved in a disulfide bridge. Index Q_c as in Equation (5).

Surprisingly, the baseline method performs quite well in the case of 2 and 3 bonds. Apart for the case of 4 bonds, results of MC graph-matching and the sequence-based RNNs (BiRnn-1 sequence and BiRnn-2 sequence) are not significantly different from the trivial algorithm. This method is even better when the task is to predict connectivity on six cysteines. NN graph-matching performs better than single-sequence RNNs partly because it uses richer information compared to the input processed by the recursive model. Training RNNs with multiple alignment profiles allows us to obtain improvement in performance for all groups of chains. The difference is more significant for $B = 2, 3$ ($p < 0.025$) and less for $B = 4, 5$ ($p \leq 0.1$)[2]. In BiRnn-2 profile, we use the same amount of training data as in NN graph-matching and we are able to consistently outperform this method. Even though the BiRnn-2 profile method outperforms other algorithms in the case of 4 and 5 bonds, this performance is still substantially low compared to the case of 2 and 3 bonds. This highlights the increased difficulty of choosing the correct pattern among 105 ($B = 4$) and 945 ($B = 5$) possible candidates. On a global basis (see last column of Table 2), our methodology achieves correct prediction on slightly less than half of patterns and pairs. Overall, the combination of RNNs and profiles is not generally applicable, but considering the distribution of B (Fig. 1), it can be very informative in a real predictive context for a large subset of the sequences of interest (those with 2 and 3 bonds).

Since disulfide bridges can be seen as a special case of residue contacts, we analyzed in more detail the performance of our best method (BiRnn-2 profile, last row in Table 2) with the use of appropriate indices. Note that in this case the number of predicted contacts is constrained to be the same as the number of actual contacts. By this, precision and recall (considering respectively false positives and false negatives) have the same values and can be represented with a single index: Q_{nc} for pairs not in contact and Q_c for those in contact. The latter is obviously identical to the fraction of correctly predicted disulfide bridges in Equation (5). One additional index we use is Q_2, which estimates the probability of correct prediction at the level of individual pairs of bonded cysteines, either in contact or not. Table 3 shows the results of this kind of analysis. We report performance grouped according to the topology class and the number of disulfide bonds of each chain. From the last row and column we see that our approach yields 82% correct prediction at the level of individual pairs corresponding to about half of the contacts and 88% of non-contacts correctly classified. Not surprisingly, the values in the last row are nearly the same as those obtained for the scop-unclassified chains, since these belong to the over represented class in our dataset. The index Q_{nc} monotonically increases with B, as the proportion of non-contacts increases with the number of disulfide bonds. Overall, the fraction of correctly predicted pairs is consistently above 80%. By simple computation, we can compare the performance of our networks in each case against the baseline values for Q_2 and Q_{nc} which are respectively 33% and 50% for $B = 2$, 60% and 75% for $B = 3$, 71% and 83% for $B = 4$ and 78% and 88% for $B = 5$. As expected, the performance on the different fold types deviates from the global results, especially for chains having more than two bridges. This clearly depends on the unbalanced representation of the various classes (see Table 1). Despite this, the network operating on four bonded cysteines produces accurate and similar results for each of the represented topology classes. In this case the task is to predict the correct pattern among three possible alternatives and it is more likely that these classes share similar patterns.

6 CONCLUSIONS

We have proposed and tested a novel machine learning method for predicting disulfide connectivity patterns in cysteine-rich proteins. Performance is comparable or better than other algorithms in the literature. In addition, our model guarantees a significant decrease in training time and can easily incorporate and process evolutionary information in the form of multiple alignment profiles. Experimental studies

[2] In the latter case, note that the t-test is not maximally reliable, since the number of chains with 4 or 5 bonds in each fold is <30.

demonstrate the benefit of using this type of information. One obvious direction for further study is to combine cysteine bonding state predictors with a pairing algorithm like the one presented in this paper, in order to build a complete predictor of disulfide bridges. In this perspective, the use of neural networks for solving the pairing problem is potentially advantageous as it allows global optimization of the recursive network together with the parameters of the bonding state predictor. As previously stated, disulfide bridges can also be seen as a special (and important) case of residue contacts. Therefore it may be important to compare and combine predictors of disulfide bridges with predictors of contact maps whose performance is improving but still appears unsatisfactory for long ranged interactions.

ACKNOWLEDGEMENTS

This work is partially supported by a MIUR grant. We thank Piero Fariselli for suggestions and useful discussions on the topic. We also thank Aoife McLysaght for help in reviewing a preliminary draft of the paper.

REFERENCES

Altschul,S., Madden,T., Schaffer,A., Zhang,A., Miller,W. and Lipman,D. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleid Acids Res.*, **25**, 3389–3402.

Bairoch,A. and Apweiler,R. (2000) The SWISS-PROT protein sequence database and its supplement TrEMBL. *Nucleic Acids Res.*, **28**, 45–48.

Baldi,P., Brunak,S., Frasconi,P., Soda,G. and Pollastri,G. (1999) Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics*, **15**, 937–946.

Fariselli,P. and Casadio,R. (2001) Prediction of disulfide connectivity in proteins. *Bioinformatics*, **17**, 957–964.

Fariselli,P., Martelli,P.L. and Casadio,R. (2002) A neural network-based method for predicting the disulfide connectivity in proteins. In Damiani,E. *et al.* (eds), *Knowledge Based Intelligent Information Engineering Systems and Allied Technologies (KES 2002)*, Vol. 1, IOS Press, pp. 464–468.

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

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

Frasconi,P., Gori,M. and Sperduti,A. (1998) A general framework for adaptive processing of data structures. *IEEE Trans. on Neural Networks*, **9**, 768–786.

Frasconi,P., Passerini,A. and Vullo,A. (2002) A two stage SVM architecture for predicting the disulfide bonding state of cysteines. In *Proceedings of IEEE Neural Network for Signal Processing Conference* IEEE Press.

Harrison,P.M. and Sternberg,M. (1994) Analysis and classification of disulfide connectivity in proteins. *J. Mol. Biol.*, **244**, 448–463.

Haussler,D. (1999) Convolutional kernels on discrete structures. Tech. rep. UCSC-CRL-99-10, University of California at Santa Cruz.

Martelli,P., Fariselli,P., Malaguti,L. and Casadio,R. (2002) Prediction of the disulfide-bonding state of cysteines in proteins at 88% accuracy. *Protein Sci.*, **11**, 2735–2739.

Matsumura,M., Signor,G. and Mathews,B.W. (1989) Substantial increase in protein stability by multiple disulphide bonds. *Nature*, **342**, 291–293.

Murzin,A., Brenner,S., Hubbard,T. and Chothia,C. (1995) Scop: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, **247**, 536–540.

Vullo,A. and Frasconi,P. (2003) Prediction of protein coarse contact maps. *J. Bioinformatics Comput. Biol.*, **1**, 411–431.

Wedemeyer,W., Welkler,E., Narayan,M. and Scheraga,H. (2000) Disulfide bonds and protein-folding. *Biochemistry*, **39**, 4207–4216.