PROTEINS: Structure, Function, and Bioinformatics 62:617–629 (2006)

Large-Scale Prediction of Disulphide Bridges Using Kernel Methods, Two-Dimensional Recursive Neural Networks, and Weighted Graph Matching

Jianlin Cheng, Hiroto Saigo, and Pierre Baldi*

Institute for Genomics and Bioinformatics, School of Information and Computer Sciences, University of California, Irvine, Irvine, California

ABSTRACT **The formation of disulphide bridges between cysteines plays an important role in protein folding, structure, function, and evolution. Here, we develop new methods for predicting disulphide bridges in proteins. We first build a large curated data set of proteins containing disulphide bridges to extract relevant statistics. We then use kernel methods to predict whether a given protein chain contains intrachain disulphide bridges or not, and recursive neural networks to predict the bonding probabilities of each pair of cysteines in the chain. These probabilities in turn lead to an accurate estimation of the total number of disulphide bridges and to a weighted graph matching problem that can be addressed efficiently to infer the global disulphide bridge connectivity pattern. This approach can be applied both in situations where the bonded state of each cysteine is known, or in ab initio mode where the state is unknown. Furthermore, it can easily cope with chains containing an arbitrary number of disulphide bridges, overcoming one of the major limitations of previous approaches. It can classify individual cysteine residues as bonded or nonbonded with 87% specificity and 89% sensitivity. The estimate for the total number of bridges in each chain is correct 71% of the times, and within one from the true value over 94% of the times. The prediction of the overall disulphide connectivity pattern is exact in about 51% of the chains. In addition to using profiles in the input to leverage evolutionary information, including true (but not predicted) secondary structure and solvent accessibility information yields small but noticeable improvements. Finally, once the system is trained, predictions can be computed rapidly on a proteomic or protein-engineering scale. The disulphide bridge prediction server (DIpro), software, and datasets are available through www.igb.uci.edu/servers/pass.html. Proteins 2006;62:617–629.**

Key words: disulfide bridges; recursive neural networks; kernel methods

INTRODUCTION

Disulphide Connectivity

The formation of covalent links between cysteine (Cys) residues by disulphide bridges is an important and unique feature of protein folding and structure. Simulations,[1] experiments in protein engineering,[2–4] theoretical studies,[5–7] and even evolutionary models[8] stress the importance and selective advantage of disulphide bridges in stabilizing the native state of proteins. This stabilizing role of disulphide bridges derives from a reduction of the number of configurational states, thus of the entropic cost of folding a polypeptide chain into its native state.[21] Moreover, disulphide bridges not only contribute to the energetics of folding but, depending on their number and location, they can also contribute to catalytic activity.[4] Thus, knowledge or prediction of disulphide bridges in a protein is important: it can provide essential insights into its structure, function, and evolution, as well as valuable long-ranged structural constraints[9] that can be incorporated into a protein structure prediction pipeline. However, it is precisely because disulphide bridges link linearly distant portions of a protein that their prediction has remained a considerable challenge. To address this challenge, here we develop and test new methods that significantly improve the prediction of disulphide bridges.

Overview of Disulphide Connectivity Prediction

Only in recent years has the problem of predicting disulphide bridges in a systematic manner received sustained attention.[10–13] The prediction of disulphide bridges can in fact be subdivided into four related prediction subproblems (Fig. 1). First, only a minority of protein chains contain disulphide bridges. Thus, it is desirable to be able to classify protein chains into those containing disulphide bridges and those that are entirely devoid of disulphide bridges (chain classification). Second, even in a chain that contains disulphide bridges, not all the cysteines may be bonded. Thus, the second problem is the classifications of cysteine residues into bonded and non-

Grant sponsor: NIH Biomedical Informatics training grant; Grant number: LM-07443-01; Grant sponsor: NSF MRI grant; Grant number: EIA-0321390; Grant sponsor: University of California System wide Biotechnology Research and Education Program, Grant sponsor: Institute for Genomics and Bioinformatics at UCI.

*Correspondence to: Pierre Baldi, Institute for Genomics and Bioinformatics, School of Information and Computer Sciences, University of California, Irvine, Irvine, CA 92697. E-mail: pfbaldi@ics.uci.edu

Received 5 May 2005; Revised 11 August 2005; Accepted 6 September 2005

Published online 30 November 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/prot.20787

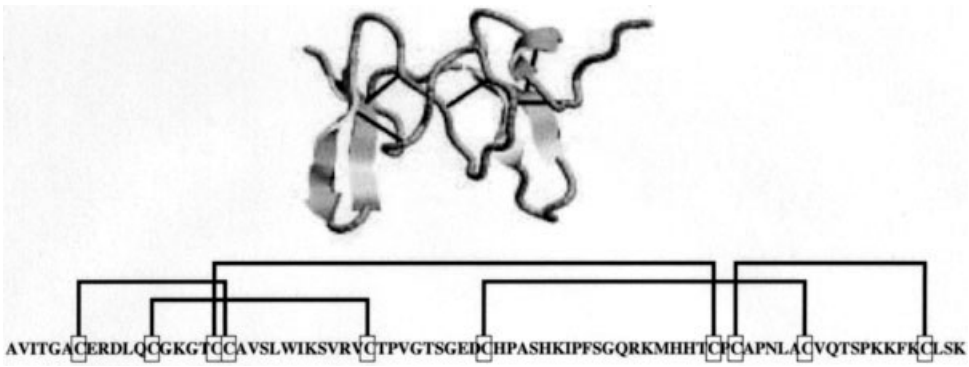

Fig. 1. Structure (top) and disulphide bridge connectivity pattern (bottom) of intestinal toxin 1, PDB code 1IMT. There are five disulphide bridges shown as thick lines.

bonded (residue classification). Third, given a pair of cysteines, one can ask whether they are linked or not by a disulphide bridge (bridge classification). And fourth, the most important and challenging problem is to determine all the pairs of cysteines that are bonded to each other by a disulphide bridge (connectivity prediction). Although these problems can be tackled separately, it is clear that they are not independent and that "mixed" solutions can also be considered. Furthermore, tackling them sequentially and independently of each other may not be always optimal. For example, deciding in isolation whether a given cysteine is bonded or not may fail to take into consideration information about the bonding state of other cysteines in the same sequence and the obvious *global* constraint that the total number of intrachain disulphide-bonded cysteines must be even. Finally, it is worth noting that interchain disulphide bridges associated with quaternary structure do occur also. However, they are considerably less frequent and very few such examples can be found in the Protein Data Bank (PDB).[14] Thus, in the current state of affairs, and consistently with all existing literature, it is not unreasonable to focus exclusively on intrachain disulphide bridges. Here, we address all four problems, with a particular emphasis on the most challenging problem of predicting intrachain disulphide connectivity, directly or in combination with the second and third problems.

None of the approaches published so far in the literature address all four problems. Published approaches to disulphide connectivity prediction use stochastic global optimization,[10] combinatorial optimization,[12] and machine learning techniques.[11,13] The early work in Fariselli and Casadio[10] provides a first, fairly comprehensive treatment of disulphide connectivity prediction by reducing it to a matching problem in a complete weighted graph, where the vertices represent oxidized cysteines. Edge weights correspond to interaction strengths or contact potentials between the corresponding pairs of cysteines. The weights are learned using a simulated annealing approach. A candidate set of bridges is then derived by finding the maximum weight perfect matching.* The prediction of which cysteines are oxidized (residue classification) is not addressed in this work. In a subsequent improvement,[11] neural network predictions are used for labeling edges with contact potentials, increasing the predictive power and reducing training time. This method achieves good results in the simplest cases of chains containing only two or three bridges.

The method in Vullo and Frasconi[13] attempts to solve the connectivity prediction problem using a different machine learning approach by modeling candidate connectivity patterns as undirected graphs (see Fig. 1, bottom). A recursive neural network architecture[15] is trained to score candidate graphs by their similarity with respect to the correct graph. The vertices of the graphs are labeled by fixed-size vectors corresponding to multiple alignment profiles in a local window around each cysteine. During prediction, the score computed by the network is used to exhaustively search the space of candidate graphs. This method yields slight improvements over Fariselli and colleagues[11] when tested on the same dataset. Unfortunately, for computational reasons, the applicability of this method remains limited because it too can only deal with sequences containing a small number of bridges, in practice up to five.

A different approach to predicting disulphide bridge connectivity is reported in Klepeis and Floudas,[12] where finding disulphide bridges is part of a more general protocol aimed at predicting the topology of β-sheets in proteins. The approach assumes hydrophobic rather than hydrogen interactions as the main driving force of β-sheet formation. Residue-to-residue contacts (including Cys–Cys bridges) are predicted by solving a series of integer linear programming problems in which customized hydrophobic contact energies must be maximized. Model constraints define allowable sheets and disulphide connectivity configurations. The most interesting aspect of this approach is its ability to predict cysteine–cysteine contacts, without assuming prior knowledge of the bonding state of the cysteines. This method, however, cannot be

* 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 edge in E'.

compared with the other approaches because the authors report validation results only two relatively short sequences with few bonds (2 and 3). In contrast, Fariselli and Casadio[10] and Vullo and Frasconi[13] assess their methods on a broad spectrum of sequences.

The simpler problem of predicting whether a given cysteine is bonded or not has also been addressed using a variety of machine learning methods including neural networks (NNs), hidden Markov models (HMMS), and support vector machines (SVMs).[16–20] For example, SVMs and kernels methods are used in Ceroni and colleague[20] to predict in two stages whether a given protein contains oxidized cysteines—in fact, whether all, none, or a mixture of its cysteines are oxidized—and subsequently to predict the oxidation state of each cysteine. The best accuracies reported in the literature are around 85%.

We present an integrated, modular approach to address all four problems. We leverage evolutionary information in the form of profiles and curated training sets in combination with kernel methods to address the chain classification problem. We use two-dimensional graphical models and recursive neural networks to predict the bonding probability of each pair of cysteines, leveraging in addition secondary structure and relative solvent accessibility information. These predictions can be derived for *all* the cysteines in a given chain, or only for the subset of disulphide-bonded cysteines, when pre-existing information about residue classification is available. Finally, we use graph matching methods to infer the disulphide bridge connectivity of each protein chain, which in turn yields a solution for both the bridge and residue classification problems, even in the case where the bonding state of individual cysteines is not known. Thus, the approach works for both situations where the bonded state of each cysteine is known or unknown and, after training, produces predictions that are rapid enough for genome-scale projects.

METHODS

Data Preparation

In order to assess our methods, we used two existing data sets (SP 39 and SP41, courtesy of Dr. A. Vullo) to compare our results with previously published results. We also curated a third, larger, data set (SPX), to take advantage of recent growth in the PDB.[14]

Previous data sets (SP39 and SP41)

SP39 is the set described and used in Fariselli and colleagues[11] and Vullo and Frasconi[13] compiled from the Swiss-Prot database[21] release no. 39 (October 2000). SP41 is the updated version, compiled with the same filtering procedures as SP39, using the Swiss-Prot version 41.19 (August 2003). Specifically, only chains whose structure is deposited in the PDB are retained. Protein chains with disulphide bonds assigned tentatively or inferred by similarity are filtered out yielding a data set comprising 966 chains, containing at least one, and up to 24, disulphide bridges. Because our method is not limited by the number of disulphide bonds, the entire set of chains is retained. This set contains a subset of 712 sequences containing at least two disulphide bridges ($K \geq 2$)—the case $K = 1$ being trivial when the cysteine bonding state is known. By comparison, SP39 contains 446 chains only, with no chain having more than five bridges. Thus, SP41 contains 266 additional sequences, and 112 of these have more than 10 oxidized cysteines.

In order to avoid biases during the assessment procedure and to perform k-fold cross validation, SP41 is partitioned into 10 different subsets, with the constraint that sequence similarity between two different subsets be less than or equal to 30%. This is comparable to the criteria adopted in Vullo and Frasconi[13] and Fariselli and Casadio,[10] where SP39 was split into four subsets. Sequence similarity is derived by a procedure analogous to the one adopted for building the PDB nonredundant selection of chains[22] by running an all-against-all rigorous Smith–Waterman local pairwise alignment,[23] using the BLOSUM65 scoring matrix with gap penalty 12 and gap extension 4. Pairs of chains with a negative distance,[24] or with and alignment length shorter than 30 residues, are considered unrelated. To address the chain classification problem, we augment the 966 positive sequences in SP41 with a set of 506 negative sequences, containing no disulphide bridged, taken form PDB Select.[25].

New dataset (SPX)

We downloaded all the proteins from the PDB on May 17, 2004. Some (26.8%) of these proteins contain at least one disulphide bridge (6827 out of 25,465). Among these 6827 proteins, 89% (6058) contain disulphide bridges that are exclusively intrachain and these are retained for further processing.[10,13,16] These proteins containing exclusively intrachain disulfide bonds yield a total of 10,793 chains, after removal of short sequences containing less than 12 amino acids. Among these 10,793 chains, 96% (10,378) have at least one intrachain disulfide bond. The disulfide bond information is extracted from the PDB files by analyzing SSBOND records.[26] To reduce overrepresentation of particular protein families, we use UniqueProt,[27] a protein redundancy reduction tool based on the HSSP[28] distance, to choose 1018 representative chains by setting the HSSP cut-off distance to 10. The HSSP distance is a similarity measure which takes into account sequence length. An HSSP distance of 10 between two sequences of length 250 is roughly equivalent to 30% sequence identity. To leverage and assess the role of secondary structure and solvent accessibility information during prediction, we use DSSP[29] to annotate secondary structure and solvent accessibility for all selected protein chains. These sequences contain 5983 cysteines in total, 85% (5082) of which are involved in disulphide bridges. These sequences are randomly split into 10 subsets of roughly the same size. During each 10-fold cross-validation experiment, 9 subsets are used for training and the remaining subset is used for validation. Final results are averaged across the 10 cross-validation experiments. To address the chain classification problem, we augment a subset of 897 positive sequences selected with an even lower HSSP cutoff distance of 5

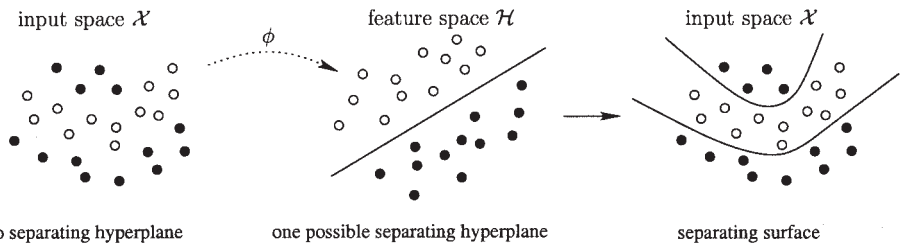

Fig. 2. Kernel methods for classification. (Left) Training patterns (white and black disks) which are not linearly separable in the original input space. (Middle) Linear separability is achieved in feature space via the mapping ϕ. (Right) The hyperplane in feature space defines a complex decision surface in input space.

(roughly below 20% similarity) with a set of 1650 negative sequences, containing no disulphide bridges, extracted from PDB and redundancy-reduced using UniqueProt with a stringent HSSP cutoff distance of 0 (no similarity).

Kernel Methods for Chain Classification

Kernels methods[30–32] are an important class of flexible machine learning methods that have proven useful for several problems in bioinformatics.[20, 33–37] The basic idea behind these methods is to try to retain the elegance and simplicity of linear methods when dealing with nonlinear data, by embedding the original data into a feature space, equipped with a dot product, where linear methods can be applied to perform classification, regression, and other computational tasks (Fig 2). The embedding is performed implicitly by defining the inner product between each pair of points in the embedding feature space through the kernel function. Thus, if ϕ denotes the embedding, the kernel function can be viewed as a measure of similarity between input points, or a metric in feature space, defined by

$$K(x, x') = \langle \phi(x), \phi(x') \rangle .$$

In a classification problem with training examples $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, where x denotes input points and y denotes binary classification variables ($\pm$ 1), it can be shown that given a point x_{n+1}, the linear decision surface is completely determined by the Gram matrix $K(x_i, x_j)$ of inner products between the feature vectors and has the form:

$$f(x_{n+1}) = \text{sign}\left(\sum_{i=1}^{n} \alpha_i y_i \langle \phi(x_{n+1}), \phi(x_i) \rangle + b \right)$$

$$= \text{sign}\left(\sum_{i=1}^{n} \alpha_i y_i K(x_{n+1}, x_i) + b \right) \quad (1)$$

with $\alpha_i \geq 0$. In practice, a kernel approach really depends on two independent modules: (1) a module for computing the kernel and the Gram matrix; and (2) a module for computing the optimal manifold (usually a hyperplane in classification problems) in feature space, typically using techniques from quadratic convex optimization. Because relatively standard packages exist to cover the second module, for conciseness we focus on the description of the kernels and refer the readers to Schölkopf and Smola[32] for additional details about kernel methods and the origin and solution of Equation 1. We use six different kernels (Spectrum, Mismatch, Profile, Smith–Waterman, Local Alignment, and Fisher) to classify protein chains, according to whether they contain at least one disulphide bridge or not.

Spectrum kernel

Spectrum kernels for sequences are derived by constructing, for each sequence x, the feature vector $\phi_k(x)$ counting the occurrences of all possible substrings of length k.[38] Similarity between spectral vectors can be computed by simple scalar product, or further processed using Gaussian exponentials or other positive convex functions.[32]

Mismatch kernel

Mismatch kernels[39] are a variation of spectrum kernels which allows inexact matching of substrings. Specifically, mismatch kernels count co-occurrences of substrings of length k, allowing up to $m \leq k$ mismatches, between the two input sequences. Like spectrum kernels, mismatch kernels can be computed efficiently by using trie data structures (suffix trees).

Profile kernel

Profile kernels[39] are another variation on mismatch kernels which use a position-dependent mutation neighborhood for inexact matching of subsequences of length k. The local neighborhood is defined using probabilistic profiles, such as those produced by the PSI-BLAST algorithm, by aligning the sequences to the NR database. A given substring is in the local neighborhood if its negative log probability, according to the profile, is smaller than some threshold σ. Once the profiles have been derived, profile kernels can also be computed efficiently using a trie data structure. In the simulations, for each sequence in the SP41 or SPX dataset, profiles are derived using two iterations of PSI-BLAST[40] against the NR database with default parameter settings. Software for computing spectrum, mismatch, and profile kernels is available from http://www1.cs.columbia.edu/compbio/string-kernels/.

Smith–Waterman kernel

The SW kernel is an empirical kernel technique,[32] which uses the E-values of a Smith–Waterman alignment score

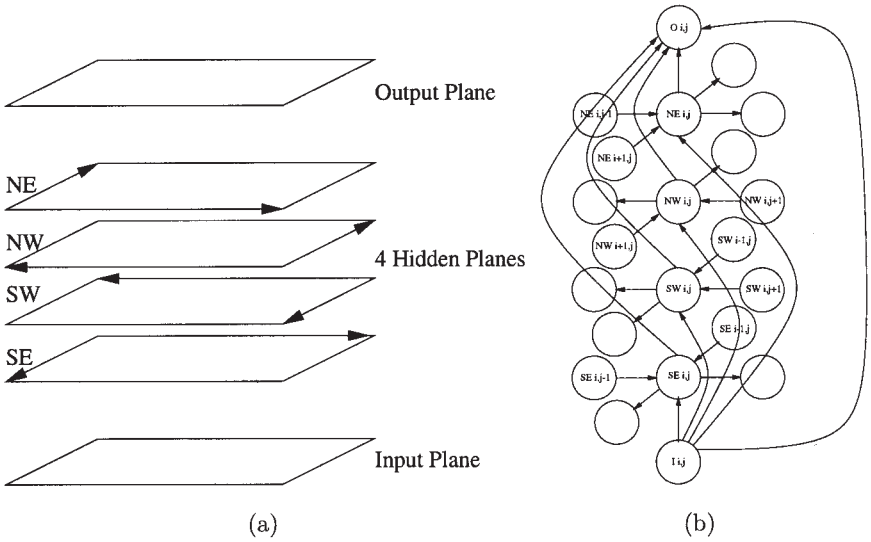

Fig. 3. (a) General layout of a DAG for processing two-dimensional objects such as disulphide contacts, with nodes regularly arranged in one input plane, one output plane, and four hidden planes. In each plane, nodes are arranged on a square lattice. The hidden planes contain directed edges associated with the square lattices. All the edges of the square lattice in each hidden plane are oriented towards one of the four possible cardinal corners: NE, NW, SW, SE. Additional directed edges run vertically within a vertical column from the input plane to each hidden plane, and from each hidden plane to the output plane. (b) Connections within a vertical column (i, j) of the DAG. I_{ij} represents the input, O_{ij} the output, and NE_{ij} represents the hidden variable in the North–East hidden plane. Similarly for the other hidden variables.

as a measure of similarity and maps each sequence into a feature vector consisting of its SW scores with all the input training sequences.

Local alignment kernel

The local alignment kernel is a variation on the SW kernel constructed to ensure positivity conditions (Mercer kernel conditions[32]) using dynamic programming. It is described in detail in Saigo and colleagues.[41]

Fisher kernel

Fisher kernels are derived from probabilistic generative models of the data. Here, we train a protein HMM[42] on the positive examples using the SAM software package available at http://www.cse.ucsc.edu/research/compbio/ sam.html (with local alignment option "SW 2" and Dirichlet prior option "recode3.20comp"). The feature vector (Fisher score) is obtained from the derivatives of the likelihood of a sequence with respect to the HMM parameters (see Jaakkola and colleagues[33] for details). It is obtained using the "get_fisher_scores" function of SAM with the "fisher_feature_match" option which ensures that only matching states are considered for scoring. The Fisher scores are then fed into a Gaussian kernel.

Kernel combinations

Basic algebraic operations such as addition, multiplication, and exponentiation preserve the positive properties of a kernel matrix and provide a mechanism for combining information from different kernels. We experimented with convex linear combinations of kernels, in particular with simple additivity where a new kernel matrix $K(x, y) = K_1(x, y) + K_2(x, y)$ is constructed using two normalized kernel matrices K_1 and K_2 with the corresponding feature vector defined by $\phi(x) = [\phi_1(x), \phi_2(x)]$. These kernel combinations did not seem to lead to significant improvements and therefore are not discussed in detail.

Recursive Neural Networks to Predict Cysteine Pairing Probabilities

To predict disulphide connectivity patterns, we use the 2D DAG-RNN (Directed Acyclic Graph-Recursive Neural Network) approach described in Baldi and Pollastri,[43] whereby a suitable Bayesian network is recast, for computational effectiveness, in terms of recursive neural networks. Local conditional probability tables in the underlying Bayesian network are replaced by deterministic relationships between a variable and its parent node variables. These functions are parameterized by neural networks using appropriate weight sharing, as described below. Here the underlying DAG for disulphide connectivity has six two-dimensional-layers: input, output, and four hidden layers [Fig. 3(a)]. Vertical connections, within an (i, j) column, run from input to hidden and output layers, and from hidden layers to output [Fig. 3(b)]. In each one of the four hidden planes, square lattice connections are oriented towards one of the four cardinal corners. Detailed motivation for these architectures can be found in Baldi and Pollastri[43] and a mathematical analysis of their relationships to Bayesian networks in Baldi and Rosen-2vi.[44] The essential point is that they combine the flexibility of graphical models with the deterministic propagation and learning speed of artificial neural networks. Unlike traditional neural networks with fixed-size input, these architec-

tures can process inputs of variable structure and length, and allow lateral propagation of contextual information over considerable length scales.

In a disulphide contact map prediction, the (i, j) output represents the probability of whether the ith and jth cysteines in the sequence are linked by a disulphide bridge or not. This prediction depends directly on the (i, j) input and the four-hidden units in the same column, associated with omnidirectional contextual propagation in the hidden planes. Hence, using weight sharing across different columns, the model can be summarized by five distinct neural networks in the form

$$\begin{cases} O_{ij} = N_o(I_{ij}, H^{NW}_{ij}, H^{NE}_{ij}, H^{SW}_{ij}, H^{SE}_{ij}) \\ H^{NE}_{ij} = N_{NE}(I_{ij}, H^{NE}_{i-1,j}, H^{NE}_{ij-1}) \\ H^{NW}_{ij} = N_{NW}(I_{ij}, H^{NW}_{i+1,j}, H^{NW}_{ij-1}) \\ H^{SW}_{ij} = N_{SW}(I_{ij}, H^{SW}_{i+1,j}, H^{SW}_{ij+1}) \\ H^{SE}_{ij} = N_{SE}(I_{ij}, H^{SE}_{i-1,j}, H^{SE}_{ij+1}) \end{cases} \quad (2)$$

where N denotes NN parameterization. In the simulations, these five NNs have a single hidden layer containing 9 hidden units. The number of output units in each of the four NNs associated with the four cardinal corners is also 9. Weights are initialized randomly using a uniform distribution over the $[-0.1, 0.1]$ interval. Because of the acyclic nature of the underlying graph, learning can proceed by gradient descent (backpropagation). We use a stochastic form of gradient in the sense that training examples are used online and in randomized order after each training epoch. The learning rate is set to 0.008.

The input information is based on the sequence itself or rather the corresponding profile derived by multiple alignment methods to leverage evolutionary information, possibly augmented by secondary structure and solvent accessibility information derived from the PDB files with DSSP and/or the SCRATCH suite of predictors available at www.igb.uci.edu/servers/ psss.html and described in Pollastri and colleagues,[46] Baldi and Pollastri,[43] and Cheng and colleagues.[46] For a sequence of length N and containing M cysteines, the output layer contains $M \times M$ units. The input and hidden layer can scale like $N \times N$ if the full sequence is used, or like $M \times M$ if only fixed-size windows around each cysteine are used, as in the experiments reported here. It is also possible to use one-dimensional DAG-RNN to locally encode the input, as described in Baldi and Pollastri.[43]

It is essential to remark that the same DAG-RNN approach can be trained and applied in two different modes. In the first mode, we can assume that the bonded state of the individual cysteines is known, for example through the use of a specialized predictor for residue classification. Then if the sequence contains M cysteines, $2K$ ($2K \leq M$) of which are intrachain disulphide bonded, the prediction of the connectivity can focus on the $2K$ bonded cysteines exclusively and ignore the remaining $M - 2K$ cysteines that are not bonded. In the second mode, we can try to solve both prediction problems— residue and bridge classification—at the same time by focusing on all cysteines in a given sequence. In both cases, the output is an array of pairwise probabilities from which the overall disulphide connectivity graph must be inferred. In the first case, the total number of bonds or edges in the connectivity graph is known (K). In the second case, the total number of edges must be inferred. In the Results section, we show that sum of all probabilities across the output array can be used to effectively estimate the number of disulphide contacts.

Input Specifications

The results reported here are obtained using local windows of size 5 around each cysteine, as in Vullo and Frasconi.[13] To improve prediction by exploiting evolutionary information and conserved sequence patterns encoded in homologous protein sequences, we derive position-specific profiles (also called Position Specific Scoring Matrix) from multiple sequence alignments by aligning all proteins against the NR database using PSI-BLAST[48] according to the same protocol for creating profiles described in Pollastri and colleagues.[45] Gaps are treated as if "-" corresponded to one additional amino acid. Thus, the position-specific profile for each position in a sequence is a real vector of length 21, representing the probability of the 20 amino acids plus gap. For a window of five amino acids centered around two cysteines, the profile-component of the input consists of 210 ($21 \times 5 \times 2$) numbers. One extra input encodes the linear sequence separation between the two cysteines. To study how secondary structure (SS) and solvent accessibility (SA) information affect prediction accuracy, we also add SS and SA information to the input in all four possible combinations.

Graph Matching to Derive Connectivity from Pairing Probabilities

In the case where the bonded state of the cysteines is known, one has a graph with $2K$ nodes, one for each cysteine. The weight associated with each edge is the probability that the corresponding bridge exists, as computed by the predictor. The problem is then to find a connectivity pattern with K edges, where each cysteine is paired uniquely with another cysteine. This can be solved using Edmond's maximum weight matching algorithm,[47] which has $O(V^4)$ time complexity on a graph with V edges, or rather the faster $O(V^3)$ implementation derived by Gabow,[48], with linear $O(V) = O(K)$ space complexity beyond the storage of the graph. Note that because the number of bonded cysteines in general is not very large, it is also possible in many cases to use an exhaustive search of all possible combinations. Indeed, the number of possible combinations is $1 \times 3 \times 5 \times \ldots \times (2K - 1)$, which in the case of 10 cysteines with five disulphide bridges result in only 945 possible connectivity patterns.

The case where the bonded state of the cysteines is not known is slightly more involved and the Gabow algorithm cannot be applied directly because the graph has M nodes but only a subset of $2K < M$ nodes may participate in the final maximum weighted matching. However, we can still use Gabow's algorithm as follows: Assume first that we can get a good estimate of the total number K of bonds. In general, it is still not possible to try all ($M2K$) possible

TABLE I. Statistics Relating Proportion of Bonded and Nonbonded Cysteines in the SPX Dataset to Secondary Structure (SS) and Relative Solvent Accessibility

Bonding State	Number	Helix	Strand	Coil	Exposed	Buried
Nonbonded Cys	901	0.30	0.31	0.39	0.15	0.85
Bonded Cys	5082	0.19	0.32	0.49	0.21	0.79
SS Pairs	HH	HE	HC	EE	EC	CC
Bonded pairs	0.07	0.10	0.15	0.13	0.28	0.28
Random pairs	0.04	0.12	0.19	0.10	0.31	0.24

H, helix; E, strand; C, coil.
The first two rows correspond to percentages of individual cysteines and the last two rows to percentages of pairs of cysteines. Random values correspond to the product of the individual frequencies.

subsets and run Gabow's algorithm on each one of them, but one can use a good heuristic approximation. If M is even ($M = 2R$) we apply Gabow algorithm to the $2R$ nodes and then prune down the final result by removing, from the final set of R edges, the $R - K$ edges with lowest probabilities. If M is odd, $M = 2R + 1$ we apply the same strategy as above $2R + 1$ times, each time removing one of the cysteines. We then select the matching with K edges that has the highest probability. In practice this procedure gives very good results although it is not guaranteed to find the global optimum and, furthermore, it relies on a good estimate of the total number K of bonds. In the results section, we show that the total number K of bonds can be estimated from the sum of all the probabilities produced by the predictor using a simple regression approach. Although this may seem surprising, we have observed similar effects in contact map prediction, where the sum of the probabilities along a diagonal band is closely related to the total number of contacts in that band.

Alternatively, it is also possible to use a slightly different greedy algorithm to derive the connectivity pattern using the estimate of the total number of bonds. First, we order the edges in decreasing order of probabilities. Then we pick the edge with the highest probability, followed by the edge with the next highest probability that is not incident to the first edge, and so forth, until K edges have been selected. Because this greedy procedure is not guaranteed to find the global optimum, it is useful to repeat it L times. In each run $i = 1, \ldots, L$, the first edge selected is the ith most probable edge. This is based on the observation that in practice the optimal solution always contains one of the top L edges and, for L reasonably large, the optimal connectivity pattern is usually found. We have compared this method with Gabow's algorithm in the case where the bonding state is known and observed that when $L = 6$, this greedy heuristic yields results that are as good as those obtained by Gabow's algorithm which, in this case, is guaranteed to find a global optimum. Thus the simulation results we report are derived using the greedy procedure with $L = 6$. The advantage of the greedy algorithm is its low $O(M^2 \log M + LKM)$ time complexity. This is because it takes $O(M^2 \log M)$ steps to sort all the pairing probabilities, and at most $O(KM)$ steps to derive a matching, starting from one of the L most promising edges.

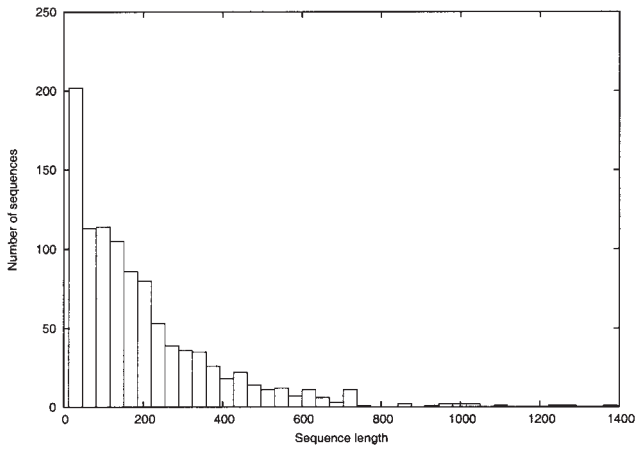

Fig. 4. Distribution of sequence lengths in redundancy-reduced dataset (SPX) of sequences containing disulphide bridges.

RESULTS

Statistical Analysis

Basic statistics extracted from the larger SPX dataset are shown in Figures 4, 5, 6, and 7 and Table I. Figure 4 provides the distribution of sequence lengths. As the number of disulphide bridges in a protein chain increases, the number of possible disulphide connectivity patterns increases exponentially. Thus, it is important to study the distribution of the number of bridges per protein and investigate how connectivity prediction deteriorates with the number of bridges. Figure 5 shows that most sequences have less than five disulphide bridges, but there are exceptions, and a fraction of the sequences contains over 10 disulphide bridges. In SPX, the average number of disulphide bridges per chain is 2.5, with a standard deviation of 2.14. Figure 6 illustrates the distribution of disulphide bridge densities measured by the number of bridges divided by the sequence length. Figure 7 shows the distribution of disulphide bridge lengths measured in terms of the number of intervening amino acids. A very significant fraction of bridges is long-ranged with lengths above 30, far exceeding the scale of local secondary structure. This is the dual signature of the important stabilizing role of disulphide bridges and the challenge they pose for prediction methods.

To analyze the relationship between disulphide bridges and secondary structure and relative solvent accessibility,

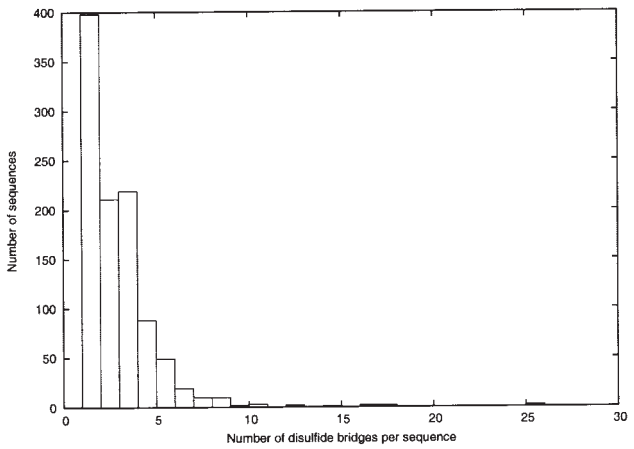

Figure 5. Distribution of the number of disulfide bridges per sequence in SPX.

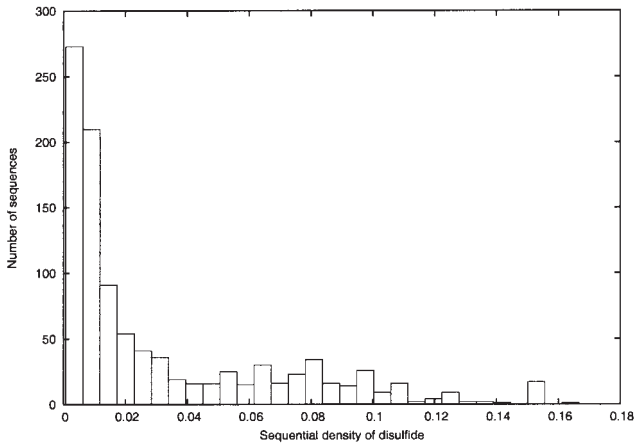

Fig. 6. Distribution of sequential density of disulfide bridges (number of bridges divided by sequence length) in SPX.

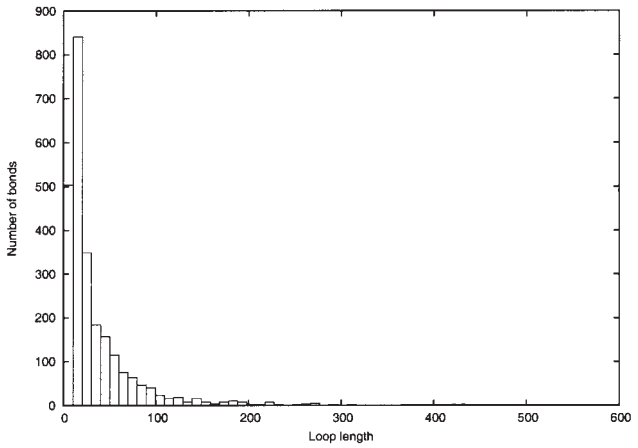

Fig. 7. Distribution of disulphide bridge lengths ion SPX.

we compute the empirical distribution of secondary structure classes (Helix, Beta Strand, or Coil) and relative solvent accessibility classes (Exposed or Buried with respect to a 25% cutoff) for both bonded and non-bonded cysteines (Table I). We observe several statistical relationships between the oxidized state of cysteines and their secondary structure and solvent accessibility. For example, 30% of nonbonded cysteines are found in helices, versus only 19% of bonded cysteines. About half of bonded cysteines (49%) are found in coils, versus only 39% for nonbonded cysteines. Most cysteines tend to be buried, however, bonded cysteines have a slight tendency towards solvent exposure, compared to nonbonded cysteines. The last two rows of Table I show slight pairing biases. For example, 13% of disulphide bridges are established between two beta strands (EE) versus 10% if the secondary structure of the pairs were selected at random (0.32×0.32). Taken together, these statistics suggest that secondary structure information, and to a lesser extent solvent accessibility information, may be useful for predicting disulphide bridges and worth incorporating in the inputs.

Protein Chain Classification

Results obtained on the problem of separating protein chains containing disulphide bridges from those that do not contain any bridges using kernel methods are shown in Tables II and III for the SP41 and the SPX datasets, respectively. Each kernel is assessed in terms of sensitivity, specificity, accuracy, and ROC score. The ROC score is the normalized area under the curve relating true positives as a function of false positives. Each performance metric is averaged across the 10 folds. Overall the results are consistent across both datasets and confirm expected trends. In general, the more complex and flexible kernels (e.g., profile, mismatch, Fisher) tend to perform better than the simpler kernels (spectrum of fixed length) and combinations of spectrum (respectively mismatch) kernels outperform individual spectrum (respectively mismatch) kernels. On the SP41 dataset, for example, the profile kernel achieves the best overall performance with accuracy of 85% and ROC score of 0.9. The superiority of more complex kernels is less pronounced on the SPX datasets and a few other differences are observed between the two datasets, probably resulting from the fact that SP41 has 966 positive examples and 506 negative examples, and SPX has 897 positive examples and 1650 negative examples, with much greater variability in the negative examples. In general, the results are weaker on the SPX datasets and higher sensitivity is observed on SP41 versus higher specificity on SPX. On the SPX dataset, the profile kernel is still among the best but is slightly outperformed by the Fisher and combined-mismatch kernels. The combined mismatch kernel, for example, achieves 75% accuracy and 0.75 ROC score.

Disulphide Bridge Classification and Connectivity Prediction Assuming Knowledge of Bonded Cysteines

To compare with previous methods, most of which assume that the bonding state of each cysteine is known, we first train and test two-dimensional DAG-RNN architectures using the SP39 dataset under the same assumption. Thus, the output pairing probabilities are predicted only

TABLE II. Protein Classification Results Using Kernel Methods on the SP41 Dataset

Kernel	Sensitivity	Specificity	Accuracy	Mean ROC
Spectrum ($k = 2$)	0.78	0.60	0.72	0.76
Spectrum ($k = 3$)	0.82	0.51	0.71	0.77
Spectrum ($k = 4$)	0.88	0.36	0.70	0.77
Spectrum ($k = 5$)	0.88	0.25	0.66	0.72
Spectrum ($k = 2, 3, 4, 5$)	0.85	0.69	0.80	0.85
Mismatch ($k = 3; m = 1$)	0.82	0.58	0.74	0.79
Mismatch ($k = 4; m = 1$)	0.86	0.58	0.76	0.83
Mismatch ($k = 5; m = 1$)	0.90	0.49	0.76	0.82
Mismatch ($k = 6; m = 1$)	0.93	0.08	0.64	0.76
Mismatch ($k = 3, 4, 5, 6; m = 1$)	0.86	0.74	0.82	0.87
Fisher kernel	0.75	0.87	0.79	**0.88**
SW kernel	0.83	0.81	0.82	**0.88**
LA kernel	0.89	0.76	**0.84**	0.87
Profile kernel ($k = 6; \sigma = 9.0$)	0.87	0.82	**0.85**	**0.90**

Top two accuracy and mean ROC scores are in bold face. Spectrum ($k = 2, 3, 4, 5$) corresponds to the sum of the four spectrum kernels from $k = 2$ to $k = 5$. Mismatch ($k = 3, 4, 5, 6; m = 1$) corresponds to the sum of the four mismatch kernels from $k = 3$ to $k = 6$ while $m = 1$ is kept unchanged. For the LA and SW kernels, alignments are derived using the BLOSUM 62 matrix with gap open and extension penalties of 12 and 2, respectively. The scaling parameter β of the LA kernel is set to $\beta = 0.5$.

TABLE III. Protein Classification Results Using Kernel Methods on the SPX Dataset

Kernel	Sensitivity	Specificity	Accuracy	mean ROC
Spectrum ($k = 2$)	0.63	0.68	0.66	0.71
Spectrum ($k = 3$)	0.56	0.72	0.66	0.67
Spectrum ($k = 4$)	0.39	0.86	0.70	0.66
Spectrum ($k = 5$)	0.25	0.91	0.68	0.62
Spectrum ($k = 2, 3, 4, 5$)	0.54	0.83	**0.73**	0.74
Mismatch ($k = 3; m = 1$)	0.57	0.66	0.63	0.67
Mismatch ($k = 4; m = 1$)	0.57	0.77	0.70	0.71
Mismatch ($k = 5; m = 1$)	0.49	0.87	**0.73**	0.71
Mismatch ($k = 6; m = 1$)	0.25	0.94	0.70	0.66
Mismatch ($k = 3, 4, 5, 6; m = 1$)	0.56	0.83	**0.74**	**0.75**
Fisher kernel	0.55	0.82	0.72	**0.76**
SW kernel	0.46	0.80	0.68	0.66
LA kernel	0.45	0.88	**0.73**	0.72
Profile kernel ($k = 6; \sigma = 9.0$)	0.49	0.86	**0.73**	0.71

Top two accuracy and mean ROC scores are in bold face. Spectrum $k = 2,3,4,5$) corresponds to the sum of the four spectrum kernels from $k = 2$ to $k = 5$. Mismatch ($k = 3, 4, 5, 6; m = 1$) corresponds to the sum of the four mismatch kernels from $k = 3$ to $k = 6$ while $m = 1$ is kept unchanged. For the LA and SW kernels, alignments are derived using the BLOSUM 62 matrix with gap open and extension penalties of 12 and 2, respectively. The scaling parameter β of the LA kernel is set to $\beta = 0.5$.

for the cysteines known to participate in a disulphide bridge. The precision percentages at the level of both individual pairs and entire connectivity patterns are reported in Table IV as a function of the number K of disulphide bridges in the chain. In all but one case, the results are better than those previously reported in the literature.[11, 13] In some cases, the results are substantially better. For example, for three disulphide bridges ($K = 3$), the precision reaches 0.61 and 0.51 at the pair and pattern levels respectively, whereas the best results reported in the literature on the same dataset are 0.51 and 0.41. Note that SP39 contains only sequences with five bridges or less and thus only results for $K \leq 5$ are reported here. The observed improvement in performance is likely to result from the architectural differences between that approach described in Vullo and Frasconi[13] and the one introduced here.

TABLE IV. Disulphide Connectivity Prediction with Two-Dimensional DAG-RNN Assuming the Cysteine Bonding State Is Known Derived on the SP39 Dataset for Comparison PurPoses

K	Pair Precision	Pattern Precision
2	0.74*(0.73)	0.74*(0.73)
3	0.61*(0.51)	0.51*(0.41)
4	0.44*(0.37)	0.27*(0.24)
5	0.41*(0.30)	0.11(0.13)
2 . . . 5	0.56*(0.49)	0.49*(0.44)

Last row reports performance on all test chains. Asterisque indicates level of precision exceeding best previously reported results given in parentheses.[13]

TABLE V. Cysteine Bonding State Sensitivity and Specificity with Different Combinations of Secondary Structure and Solvent Accessibility Information on the SPX Dataset

	No SS No SA	SS	SA	SS and SA	PSS and PSA
Bond. state sens.	0.886	0.883	0.884	0.894	0.889
Bond. state spec.	0.876	0.878	0.872	0.878	0.879

SS, = secondary structure; SA, solvent accessibility; PSS, predicted secondary structure; PSA, predicted solvent accessibility.

Disulphide Connectivity Prediction from Scratch

In this set of experiments, we do not assume any knowledge regarding whether individual cysteines are disulphide bonded or not and apply the two-dimensional DAG-RNN approach to predict pairing probabilities for *all* pairs of cysteines in each sequence. Thus, for each chain we predict the number of disulphide bridges, and address the residue and bridge classification problems, as well as the global connectivity problem.

Prediction of cysteine bonding states (residue classification)

Prediction of the bonding state of individual cysteines is assessed in Table V using the larger SPX dataset. Specificity and sensitivity of bonding state predictions are close to 87% and 89% in the absence of additional secondary structure or relative solvent accessibility information, with at best a small improvement when this information is added.

Prediction of the number of disulphide bridges

Analysis of the prediction results shows that there is a relationship between the sum $S(p)$ of all the probabilities in the graph (or the output layer of the 2D DAG-RNN) and the total number of bonded cysteines. Using both SS and SA as inputs, the correlation coefficient between $2K$ and $S(p)$ is 0.89, the correlation coefficient between $2K$ and M is 0.87, and the correlation coefficient between $2K$ and $S(p)$ log M is 0.94, where M is the total number of cysteines in the sequence being considered. Thus, we estimate the total number of bonded cysteines using this linear regression approach and rounding off the result, making sure that the total number of bonded cysteines is even and does not exceed the total number of cysteines in the sequence. Figure 8 represents the plot of predicted bond numbers against true bond numbers on the SPX dataset. As shown in the plot, the bond number prediction is rather accurate for most $K > 1$ cases, with few exceptions for very large K ($K > 20$). For $K = 1$, the method tends to overpredict the number of bridges. Table VI reports the accuracy for predicting the number of bridges. The total number of disulphide bridges in 0.68 of chains is correctly predicted with no additional inputs, with a standard error (mean square root of residuals) of 1.06. With true SS and SA input information the performance reaches 71% of correct predictions, with a standard error of 1.04. In more than 94% of the cases, the predicted number of bridges is within one from the correct value. With predicted SS and SA there is no noticeable improvement (68% accuracy).

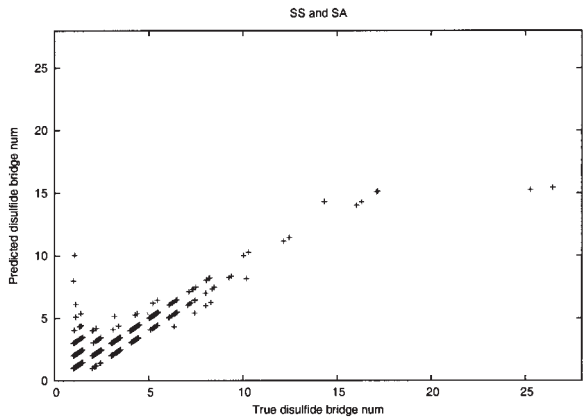

Fig. 8. Predicted bond number is plotted against the true bond number using both profiles, SS, and SA as inputs. With a total 1018 protein chains in the SPX dataset, the number of disulphide bridges of 71% of these sequences are predicted correctly using 10-fold cross validation. Uniform random noise in the range of [0,0.5] is added to both the true bond number and the predicted bond number to improve readability.

Prediction of disulphide bridges (bridge classification)

Table VII reports the specificity and sensitivity for the prediction of individual bridges. The sensitivity for chains with one disulphide bridge is around 71%, while the specificity is around 47%. Both specificity and sensitivity for chains with two or three disulphide bridges using true SA and SS information in the inputs fall in the range of 62% to 67%. The specificity and sensitivity for chains with four disulphide bridges using true SA and SS information are 55% and 50%, respectively.

When the number of disulphide bridges increases in chains, the performance decreases in general. The overall specificity and sensitivity using four different input schemes are around 51% to 55%. The variation of the performance for chains with many disulphide bridges ($K > 6$) is large because there are very few such examples in the dataset. Thus, for proteins with a large number of disulphide bridges ($K > 6$), predictions must be used with caution. The results also show that secondary structure information improves prediction accuracy of disulphide bridges by two percentage points on average. Solvent accessibility alone does not help much, but when used in combination with secondary structure the best results are achieved in most cases. Predicted SS and SA do not seem to help.

Table VIII reports the results of disulphide bridge classification on the SP41 dataset. On this dataset, only

TABLE VI. Prediction Accuracy for the Number of Disulphide Bridges on the SPX Dataset

	No SS No SA	SS	SA	SS and SA	PSS and PSA
Accuracy (number of bridges)	0.68	0.68	0.67	0.71	0.68
Mean square root of residual	1.06	1.05	1.09	1.04	1.05

TABLE VII. Specificity and Sensitivity for the Disulphide Bridge Classification Problem Derived on the SPX Dataset, as a Function of the Number *K* of Bridges in the Chain from 1 to 26, and with Different Combinations of Input Information

	No SS No SA		SS		SA		SS and SA		PSS and PSA	
K	Sensitivity	Specificity	Sensitivity	Specificity	Sensitivity	Specificity	Sensitivity	Specificity	Sensitivity	Specificity
1	0.71	0.47	0.71	0.47	0.70	0.46	0.71	0.48	0.71	0.48
2	0.59	0.59	0.63	0.63	0.59	0.59	0.63	0.63	0.59	0.60
3	0.59	0.65	0.61	0.67	0.58	0.64	0.62	0.67	0.55	0.61
4	0.44	0.49	0.48	0.53	0.46	0.52	0.50	0.55	0.44	0.48
5	0.33	0.37	0.33	0.37	0.31	0.35	0.37	0.41	0.32	0.35
6	0.24	0.28	0.32	0.37	0.29	0.34	0.29	0.33	0.32	0.36
7	0.26	0.31	0.30	0.36	0.21	0.25	0.31	0.36	0.29	0.32
8	0.16	0.18	0.21	0.25	0.26	0.30	0.30	0.32	0.20	0.22
9	0.50	0.59	0.56	0.64	0.55	0.63	0.61	0.71	0.44	0.52
10	0.40	0.43	0.27	0.29	0.40	0.43	0.37	0.40	0.33	0.36
12	0.38	0.44	0.54	0.61	0.46	0.58	0.50	0.55	0.38	0.39
14	0.71	0.83	0.50	0.54	0.42	0.50	0.57	0.62	0.79	0.85
16	0.19	0.20	0.19	0.20	0.31	0.33	0.22	0.23	0.13	0.13
17	0.35	0.40	0.41	0.47	0.38	0.43	0.35	0.40	0.53	0.60
25	0.08	0.13	0.24	0.40	0.28	0.47	0.24	0.40	0.32	0.53
26	0.38	0.67	0.31	0.53	0.23	0.40	0.42	0.73	0.31	0.51
Overall	0.52	0.51	0.54	0.53	0.52	0.51	0.55	0.54	0.52	0.51

TABLE VIII. Prediction of Disulphide Bridges with Two-Dimensional DAG-RNN on All the Cysteines, without Assuming Knowledge of the Bonding State on the SP41 Dataset

K	1	2	3	4	5	6	7	8	9	10	11	12	15	16	17	18	19
Sensitivity	.74	.61	.54	.52	.33	.27	.36	.27	.23	.30	.34	.17	.27	.11	.22	.06	.11
Specificity	.39	.51	.45	.59	.42	.34	.55	.41	.35	.45	.47	.23	.50	.13	.33	.09	.20

sequence information and profiles are used in the RNN input. While the accuracy on the SP41 dataset is lower than that on the SPX dataset, it follows the same pattern and in general deteriorates with the number of bridges.

Prediction of disulphide bridge connectivity patterns

It is very difficult to correctly predict the entire disulphide connectivity pattern because the number of connectivity patterns increases exponentially with *K*. Not knowing the bonding states of individual cysteines makes the prediction even harder. Table IX reports the pattern prediction accuracy for *K* between 1 and 4, and the overall accuracy for all the chains in the SPX dataset. The overall accuracy with no additional input information is 48% and reaches 51% using true SS and SA as inputs. Thus, for about half of all the chains, we can predict the entire pattern of disulphide bridges correctly. Consistently with our other experiments and with recent results in Ferre and Clote,[49] using true SS and SA information slightly improves performance, however predicted SS or SA information seems too noisy at this stage to be helpful.

CONCLUSION

We have presented a framework for disulphide bridge predictions that addresses all four subproblems in this area: chain classification, residue classification, bridge classification, and connectivity prediction. Table X summarizes the motivation for the initial chain classification step, by comparing overall results on residue and bridge classification derived with and without the chain classification step. In all cases, as expected, the chain classification step increases the specificity but reduces the sensitivity. The tradeoff, assessed by the F measure of information retrieval, is in favor of having the chain classification step ($F = 0.62$ versus $F = 0.54$ for residue classification, and $F = 0.38$ versus $F = 0.33$ for bridge classification). Thus, retaining the chain classification step in the overall pipeline is justified both in terms of overall performance, and because the classification can be of biological interest as well. Furthermore, our web server reports the results of each prediction stage separately.

Beyond the chain classification step, the prediction pipeline we have described presents several advantages

TABLE IX. The Prediction Accuracy of Disulphide Bridge Connectivity on the SPX Dataset

Bridge Number	No SS No SA	SS	SA	SS and SA	PSS and PSA
1	0.58	0.60	0.58	0.59	0.59
2	0.55	0.58	0.55	0.59	0.56
3	0.50	0.53	0.50	0.54	0.47
4	0.27	0.33	0.28	0.34	0.22
Overall (1–26)	0.48	0.50	0.48	0.51	0.48

TABLE X. Performance Comparison of the Disulphide Prediction Pipeline (Residue and Bridge Classification) with and without the Initial Chain Classification Step

	Residue			Bridge		
	Sensitivity	Specificity	*F*	Sensitivity	Specificity	F
Without chain classification	0.89	0.39	0.54	0.55	0.24	0.33
With chain classification	0.52	0.77	0.62	0.32	0.48	0.38

The F measure here is defined by the harmonic mean of sensitivity and specificity, $F = 2 \times \text{sensitivity} \times \text{specificity}/(\text{sensitivity} + \text{specificity})$.

over other approaches. First, assuming knowledge of cysteine bonding states, the method outperforms existing ap proaches on the same validation data. Second, the method can easily cope with chains containing an arbitrary number of bonded cysteines, overcoming the limitation of previous approaches which restrict predictions to chains containing at most 10 oxidized cysteines ($K = 5$). As an added bonus, larger training and testing sets can now be used. Third, the method proposed can deal with ab initio predictions and, in particular, it does not require pre-existing knowledge or prediction of cysteine bonding states. Good specificity and sensitivity on connectivity predictions are achieved even when the bonding state of individual cysteines is not known. Equally important, for previous methods that rely on predicting the cysteine bonding state first, false predictions are fatal. Once a false prediction has been made at the residue level, the corresponding disulphide bridges cannot be recovered (false negative) or eliminated (false positive) during subsequent bridge classification or connectivity prediction. When used in ab initio mode, the method presented here delays the prediction of the bonding state by first predicting the total number of disulphide bridges in a cooperative, robust fashion, and then globally predicting the overall connectivity from which cysteine bonding states are trivially inferred. The same fundamental idea of combining pairing probabilities with graph matching algorithms to enforce global constraints has now been expanded and applied to the problem of beta-sheet topology prediction.[50] Fourth, the method can leverage true secondary structure and relative solvent accessibility information. Results demonstrate the role secondary structure and solvent accessibility can play in disulphide bridge prediction. Overall, inclusion of SS and SA information leads to small, but noticeable improvements in the range of 1%. This is demonstrated by appropriately encoding the corresponding information in the input layer of the architecture. Predicted SS or SA information, however, is currently not accurate enough to improve performance and therefore is not retained in our implementation. Finally, while training can take days, once trained predictions can be carried on a proteomic or protein engineering scale to sift through large numbers of proteins. The resulting disulphide bridge prediction server (DIpro), software, and datasets are available through http://www.igb.uci.edu/servers/psss.html.

REFERENCES

1. Abkevich VI, Shankhnovich EI. What can disulfide bonds tell us about protein energetics, function and folding: simulations and bioinformatics analysis. J Mol Biol 2000;300:975–985.
2. Matsumura M, Signor G, Matthews BW. Substantial increase of protein stability by multiple disulfide bonds. Nature 1989;342:291–293.
3. Clarke J, Fersht AR. Engineered disulfide bonds as probes of the folding pathway of barnase–increasing stability of proteins against the rate of denaturation. Biochemistry 1993;32:4322–4329.
4. Klink TA, Woycechosky KJ, Taylor KM, Raines RT. Contribution of disulfide bonds to the conformational stability and catalytic activity of ribonuclease A. Eur J Biochem 2000;267:566–572.
5. Betz S. Disulfide bonds and the stability of globular proteins. Proteins 1993;21:167–195.
6. Doig A, Sternberg M. Side chains, conformational entropy in protein folding. Protein Sci 1995;4:2247–2251.
7. Wedemeyer WJ, Welkler E, Narayan M, Scheraga HA. Disulfide bonds and protein-folding. Biochemistry 2000;39:4207–4216.
8. Demetrius L. Thermodynamics and kinetics of protein folding: an evolutionary perspective. J Theor Biol 2000;217:397–411.
9. Harrison PM, Sternberg MJE. Analysis and classification of disulfide connectivity in proteins. J Mol Biol 1994;244:448–463.
10. Fariselli P, Casadio R. Prediction of disulfide connectivity in proteins. Bioinformatics 2001;17:957–964.
11. Fariselli P, Martelli PL, Casadio R. A neural network-based method for predicting the disulfide connectivity in proteins. 6th International Conference on Knowledge-Based Intelligent and Engineering Systems 2002.
12. Klepeis JL, Floudas CA. Prediction of β-sheet topology and disulfide bridges in polypeptides. J Comput Chem 2003;24:191–208.
13. Vullo A, Frasconi P. Disulfide connectivity prediction using recursive neural networks and evolutionary information. Bioinformatics 2004;20:653–659.
14. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig, H, Shindyalov IN, Bourne PE. The Protein Data Bank. Nucl Acids Res 2000;28:235–242.
15. Frasconi P, Gori M, Sperduti A. A general framework for adaptive processing of data structures. IEEE Trans Neural Networks 1998;9:768–786.

16. Fariselli P, Riccobelli P, Casadio R. Role of evolutionary information in predicting the disulfide-bonding state of cysteine in proteins. Proteins 1999;36:340–346.
17. Fiser A, Simon I. Predicting the oxidation state of cysteines by multiple sequence alignment. Bioinformatics 2000;16:251–256.
18. Martelli PL, Fariselli P, Malaguti L, Casadio R. Prediction of the disulfide-bonding state of cysteines in proteins at 88% accuracy. Protein Sci 2002;11:2735–2739.
19. Frasconi P, Passerini A, Vullo A. A two stage SVM architecture for predicting the disulfide bonding state of cysteines. In Proceedings of IEEE Neural Network for signal processing conference. Piscataway, NJ: IEEE Press; 2002. p 287–295.
20. Ceroni A, Frasconi P, Passerini A, Vullo A. Predicting the disulphide bonding state of cysteines with combinations of kernel machines. VLSI Signal Processing 2003;35.
21. Bairoch A, Apweiler R. The SWISS-PROT protein sequence database and its supplement TrEMBL. Nucleic Acids Res 2000;28: 45–48.
22. Hobhom U, Scharf M, Schneider P, Sander C. Selection of representative protein data sets. Prot Sci 1992;1:409–417.
23. Smith TF, Waterman MS. Identification of common molecular subsequences. J Mol Biol 1981;147:195–197.
24. Abagyan, Batalov. J Mol Biol 1997;273:355–368.
25. Hobohm U, Sander C. Adaptive mixtures of local experts. Protein Sci 1994;3.
26. Westbrook J, Fitzgerald PM. The pdb format, mmcif formats and other data formats. Structural Bioinformatics 2003.
27. Mika S, Rost B. Uniqueprot: creating representative protein-sequence sets. Nucleic Acids Res 2003;31:3789–3791.
28. Sander C. Schneider R. Database of homology-derived protein structures and the structural meaning of sequence alignment. Proteins 1991;9:56–68.
29. Kabsch W, Sander C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 1983;22:2577–2637.
30. Cristianini N and Shawe-Taylor J. An introduction to support vector machines. Cambridge, UK: Cambridge University Press; 2000.
31. Müller, K-R, Rätsch G, Mika S, Tsuda K, Schölkopf B. An introduction to kernel-based learning algorithms. IEEE Neural Networks 2001;12:181–201.
32. Schölkopf B, Smola AJ. Learning with kernels: support vector machines, regularization, optimization, and beyond. Cambridge, MA: MIT Press; 2002.
33. Jaakkola T, Diekhans M, Haussler D. A discriminative framework for detecting remote protein homologies. J Computa Biol 2000; 7(1,2):95–114.
34. Leslie C, Eskin E, Cohen A, Weston J, Noble W. Mismatch string kernels for discriminative protein classification. Bioinformatics 2004;20:467–476.
35. Liao L, Noble WS. Combining pairwise sequence similarity and support vector machines for remote protein homology detection. In Proceedings of the Sixth International Conference on Computational Molecular Biology. New York: ACM Press; 2002.
36. Schölkopf B, Tsuda K, Vert J-P. Support vector machine applications in computational biology. Cambridge, MA: MIT Press; 2004.
37. Lanckriet GRG, Cristianini N, Jordan MI, Noble WS. Kernel-based integration of genomic data using semidefinite programming. In kernel methods in computational biology. Cambridge, MA: MIT Press, 2004.
38. Leslie C, Eskin E, Noble WS. The spectrum kernel: A string kernel for SVM protein classification. In: Altman RB, Dunker AK, Hunter L, Lauerdale K, Klein T, editors. Proceedings of the Pacific Symposium on Biocomputing 2002. River Edge, NJ: World Scientific; 2002. p564–575.
39. Kuang R, Ie E, Wang K, Siddiqi M, Freund Y, Leslie C. Profile-based string kernels for detection of remote homologs and discriminative motifs. J Bioinform Computa Biol 2005;3:527–550.
40. Altschul SF, Madden TL, Schaffer AA. Gapped blast and psi-blast: a new generation of protein database search programs. Nucleic Acid Res 1997;25:3389–3402.
41. Saigo H, Vert J-P, Ueda N, Akutsu T. Protein homology detection using string alignment kernels. Bioinformatics 2004;20:1682–1689.
42. Baldi P, Brunak S. Bioinformatics: the machine learning approach. Cambridge, MA: MIT Press;2001.
43. Baldi P, Pollastri G. The principled design of large-scale recursive neural network architectures-DAG-RNNs and the protein structure prediction problem. J Machine Learning Res 2003;4:575–602.
44. Baldi, P. Rosen-Zvi, M. On the relationship between deterministic and probabilistic directed graphical models: from Bayesian networks to recursive neural networks and back. Neural Networks 2005. Forthcoming.
45. Pollastri G, Przybylski D, Rost B, Baldi P. Improving the prediction of protein secondary strucure in three and eight classes using recurrent neural networks and profiles. Proteins 2001;47:228–235.
46. Cheng J, Randall, AZ, Sweredoski M, Baldi P. SCRATCH: a protein structure and structural feature prediction server. Nucleic Acids Res 2005;33:W72–W76.
47. Edmonds J. Paths, trees, and flowers. Canadian J Mathematics, 1965;17:449–467.
48. Gabow HN. An efficient implementation of Edmond's algorithm for maximum weight matching on graphs. J ACM 1976;23:221–234.
49. Ferre F, Clote P. Disulfide connectivity prediction using secondary structure information and diresidue frquencies. Bioinformatics 2005;21:2336–2346.
50. Cheng J, Baldi, P. Three-stage prediction of protein beta-sheets by neural networks, alignments, and graph algorithms. Bioinformatics 2005;21(Suppl. 1):175–184.