PROTEINS: Structure, Function, and Bioinformatics 67:262–270 (2007)

Predicting Disulfide Connectivity Patterns

Chih-Hao Lu,[1] Yu-Ching Chen,[1] Chin-Sheng Yu,[2] and Jenn-Kang Hwang[1,2,3]*
[1]*Institute of Bioinformatics, National Chiao Tung University, Hsinchu 30050, Taiwan*
[2]*Department of Biological Science and Technology, National Chiao Tung University, Hsinchu, Taiwan*
[3]*Core Facility for Structural Bioinformatics, National Chiao Tung University, Hsinchu, Taiwan*

ABSTRACT **Disulfide bonds play an important role in stabilizing protein structure and regulating protein function. Therefore, the ability to infer disulfide connectivity from protein sequences will be valuable in structural modeling and functional analysis. However, to predict disulfide connectivity directly from sequences presents a challenge to computational biologists due to the nonlocal nature of disulfide bonds, i.e., the close spatial proximity of the cysteine pair that forms the disulfide bond does not necessarily imply the short sequence separation of the cysteine residues. Recently, Chen and Hwang (Proteins 2005;61:507–512) treated this problem as a multiple class classification by defining each distinct disulfide pattern as a class. They used multiple support vector machines based on a variety of sequence features to predict the disulfide patterns. Their results compare favorably with those in the literature for a benchmark dataset sharing less than 30% sequence identity. However, since the number of disulfide patterns grows rapidly when the number of disulfide bonds increases, their method performs unsatisfactorily for the cases of large number of disulfide bonds. In this work, we propose a novel method to represent disulfide connectivity in terms of cysteine pairs, instead of disulfide patterns. Since the number of bonding states of the cysteine pairs is independent of that of disulfide bonds, the problem of class explosion is avoided. The bonding states of the cysteine pairs are predicted using the support vector machines together with the genetic algorithm optimization for feature selection. The complete disulfide patterns are then determined from the connectivity matrices that are constructed from the predicted bonding states of the cysteine pairs. Our approach outperforms the current approaches in the literature. Proteins 2007;67:262–270.** **© 2007 Wiley-Liss, Inc.**

Key words: disulfide bond; disulfide connectivity pattern; support vector machine; genetic algorithm; feature selection

INTRODUCTION

In recent years, there is an increasing interest in developing *ab initio* approaches[1–6] to predict disulfide connectivity directly from protein sequences. Fariselli and Casadio[1] converted the problem of disulfide connectivity to a graph matching (GM) problem. In their approach, the graph vertices are equivalent to the cysteines that form disulfide bridges, and the weight edges are equivalent to contact potentials. They used the Monte-Carlo (MC) simulated annealing (SA) method to optimize the weights, with which they identified disulfide bridges through the maximal weight perfect matching. We will refer to this method as MCSA. Later the same group improved on their own approach with the neural networks (NN), instead of MCSA, to determine the cysteine pairwise interactions.[2] They were able to yield much better prediction accuracies for the case of two disulfide bonds. This method will be referred to as NNGM. Vullo and Frasconi[3] applied an *ad hoc* recursive neural network (RNN) to improve the overall protein-based prediction accuracy by around 10% for the same test data set. Baldi et al.[4] applied two-dimensional recursive neural networks (2D-RNN) to this problem and obtained a protein-based prediction accuracy of 49%. Recently, Chen and Hwang[5] treated each disulfide connectivity pattern as a separate class and treated it as a multiclass classification problem. They used the support vector machines[7] (SVM) based on a variety of sequence features and the combinations of them. They found that the sequence separation between the cysteine pair that forms a disulfide bridge is one of the most important sequence features related to disulfide connectivity. This is consistent with the recent reports[8,9] that the cysteine separations and the disulfide patterns are closely related. Chen and Hwang[5] were able to obtain an overall protein-based prediction accuracy of 55% for the dataset sharing less than 30% sequence identity. However, since the number of possible disulfide patterns N_B is related to disulfide bonds B by the relation: $N_B = (2B - 1)!! = (2B - 1)(2B - 3)\ldots1$, the number of disulfide patterns grows rapidly when the number of disulfide bonds increases [Fig. 1(A)]. Their method performs unsatisfactorily for the cases of large number of disulfide bonds. In

Grant sponsor: National Science Council (NSC), University System of Taiwan and the Veteran General Hospital (UST-VGH), Ministry of Education (MOE).

*Correspondence to: Jenn-Kang Hwang, Department of Biological Science and Technology, National Chiao Tung University, Hsinchu 30050, Taiwan. E-mail: jkhwang@faculty.nctu.edu.tw

Received 17 May 2006; Revised 11 September 2006; Accepted 23 October 2006

Published online 6 February 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/prot.21309

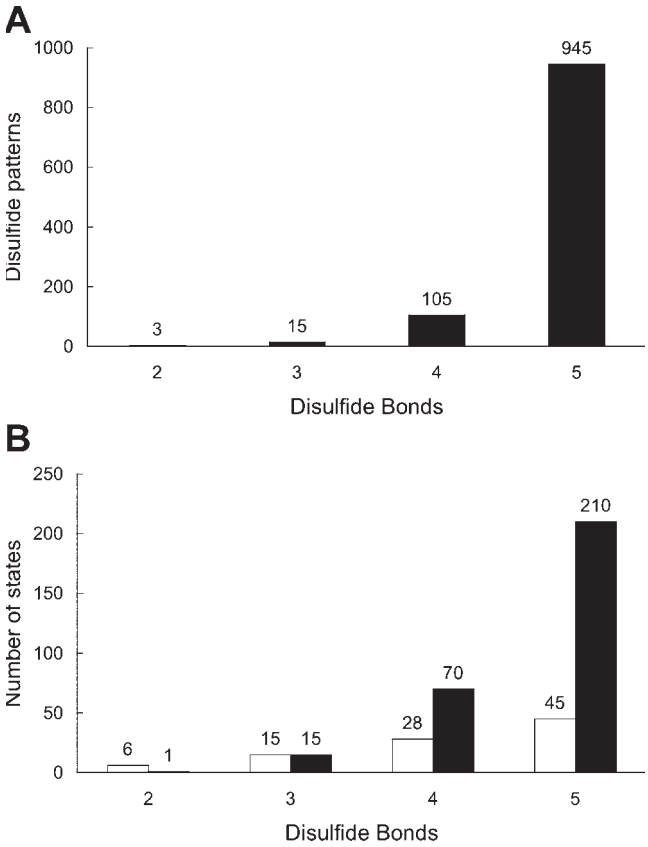

Fig. 1. (**A**) The number of disulfide patterns, N_p, versus that of disulfide bonds, *B*. (**B**) The number of bonding states of disulfide bonds in the CP_1 (empty bar) and CP_2 (solid bar) representations.

this work, we use the cysteine pairs to define disulfide connectivity, thereby reducing the number of classification classes. In addition, we use the genetic algorithm (GA) for feature selection to remove noisy or irrelevant features. We are able to obtain an overall protein-based prediction accuracy over 70%.

Disulfide Patterns in Cysteine-Pair Representation

We use the notation $\Im = \{C_1, C_2\}$ to denote the cysteine pair comprising C_1 and C_2. For each cysteine pair, there are two possible bonding states: $\sigma_1 = C_1 \oplus C_2$, where $\oplus$ denotes a disulfide bridge between C_1 and C_2, and $\sigma_2 = C_1 \otimes C_2$, where $\otimes$ denotes no disulfide bridge between C_1 and C_2. In this way, we can define the disulfide connectivity patterns in terms of the bonding states. For example, for a sequence with two disulfide bonds denoted by $[C_1C_2, C_3C_4]$, which means that C_1 and C_2 form the first disulfide bridge, and C_3 and C_4 form the second one, this disulfide pattern can be uniquely defined by the following states: $C_1 \oplus C_2$, $C_1 \otimes C_3$, $C_1 \otimes C_4$, $C_2 \otimes C_3$, $C_2 \otimes C_4$, and $C_3 \oplus C_4$. We will refer to this type of representation of the disulfide pattern as the CP_1 representation. Likewise, we can use two cysteine pairs to define the disulfide pattern, i.e. $\Im = \{C_1,C_2,C_3,C_4\}$, where

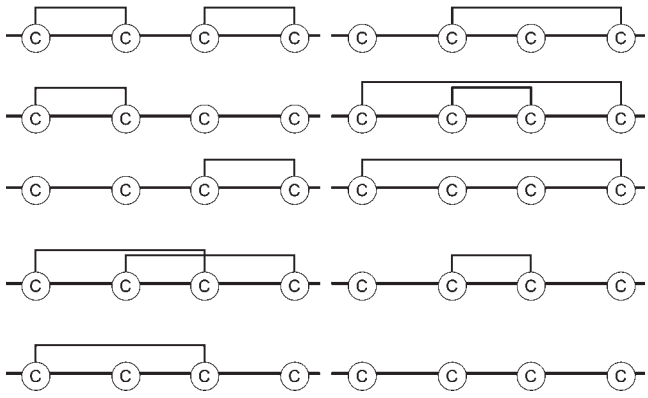

Fig. 2. Ten bonding states in the CP_2 representation with each cysteine denoted by the letter "C." The line connecting two cysteines represents the disulfide linkage between them.

C_1, C_2, C_3, and C_4 are any four distinct cysteines. For four cysteines, there are 10 possible bonding states: $\sigma_1 = (C_1 \oplus C_2, C_3 \oplus C_4)$, $\sigma_2 = (C_1 \oplus C_2, C_3 \otimes C_4)$, $\sigma_3 = (C_1 \otimes C_2, C_3 \oplus C_4)$, $\sigma_4 = (C_1 \oplus C_3, C_2 \oplus C_4)$, $\sigma_5 = (C_1 \oplus C_3, C_2 \otimes C_4)$, $\sigma_6 = (C_1 \otimes C_3, C_2 \oplus C_4)$, $\sigma_7 = (C_1 \oplus C_4, C_2 \oplus C_3)$, $\sigma_8 = (C_1 \oplus C_4, C_2 \otimes C_3)$ $\sigma_9 = (C_1 \otimes C_4, C_2 \oplus C_3)$, and $\sigma_{10} = (C_1 \otimes C_2, C_3 \otimes C_4)$. Note that σ_{10} is a shorthand notation of the state that has no disulfide bond between any pair of the cysteines. We will refer to this type of representation of the disulfide pattern as the CP_2 representation. Figure 1(B) plots the numbers of bonding states in terms of the CP_1 and CP_2 representations as a function of the number of disulfide bonds. Figure 2 schematically shows the 10 bonding states in the CP_2 representation. It is possible to use more disulfide pairs (i.e. $n \geq 3$) to define the disulfide connectivity patterns, but the number of bonding states will increase rapidly.

The Support Vector Machines

The SVM has recently found many applications[5,10–16] in computational biology. Here, we will give only a brief sketch of the SVM, since the SVM has been reviewed in many excellent textbooks.[7,17] The basic idea of the SVM is simple: given training vectors $\mathbf{x}_i$ and a vector $\mathbf{y} = (y_1,\ldots,y_l)$ defined as: $y_i = 1$ if $\mathbf{x}_i$ is in one class, and $y_i = -1$ if $\mathbf{x}_i$ is in the other class. What the SVM tries to do is to locate the separating hyperplane $\mathbf{w}^T\mathbf{x}_i + b = 0$ with the largest distance between two classes, measured along a line perpendicular to this hyperplane. This requirement is equivalent to minimizing the following equation:

$$\min_{w,b,\xi} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i$$

under the constraints

$$y_i\left[\left(\mathbf{w}^T\Phi(\mathbf{x}_i)\right) + b\right] \geq 1 - \xi_i, \quad i = 1,\ldots,l.$$

where C is the penalty parameter to be optimized. If the penalty parameter C is large enough and the data is line-

arly separable, all ξ_i will be zero.[18] In practice, we need to calculate only the kernel function given by

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^{\mathrm{T}}\Phi(\mathbf{x}_j)$$

In this work, we use the radial basis function (RBF) kernel given by $e^{-\gamma\|x_i - x_j\|^2}$, where γ is the kernel parameter. All the SVM calculations are performed using LIBSVM.[19] For SVM training, both the penalty parameter C and the kernel parameter γ must be determined in advance. In this work, we perform the grid-search on C and γ using cross-validation for the model selection.[20] Although the grid-search is a time-consuming process, it can be easily parallelized to save computational cost.

The Feature Vectors

We used the feature vectors previously proposed by Chen and Hwang[5]: the cysteine separations, the amino acid composition, and the coupling between the local sequence environments of cysteine pairs. A brief explanation of these feature vectors is given in the following sections. More details of the feature codings can be found in Ref. 5.

Cysteine separation

Let $\{c_ic_j,\ldots\}$ denote the set of cysteine pairs forming a disulfide bridges, where c_i and c_j form a disulfide bond. The cysteine separation feature vector D is given by $\{d_{ij},\ldots\}$, where $d_{\mathrm{ij}} = |c_i - c_j|$ is the sequence separation between c_i and c_j. We will use CS to denote the cysteine-separation coding scheme. The sequence separations between cysteine pairs have been recently shown to be closely related to the disulfide patterns.[8,9]

Amino acid composition

The amino acid composition of the whole protein chain is an effective global sequence feature in fold assignment[12,21] and in the prediction of protein subcellular localization.[14,22] The amino acid composition of the whole protein chain is a 20-component vector, each component representing the relative occurrence of a given amino acid type. We will use AA to denote the amino acid-composition coding scheme.

Cysteine–cysteine coupling

The cysteine–cysteine coupling feature vector $\mathbf{s}_{ij}$ describes the correlation between the sequences environments of cysteine i and cysteine j that form a disulfide bond. The flanking sequences of the cysteine are described by a sequence window, which incorporates evolutionary information through the use of the position specific substitution matrix (PSSM) computed by PSI-BLAST.[23] We will use CC to denote the cysteine–cysteine coupling coding scheme.

Feature Selection

We use GA to optimize feature selection. The basic procedure is as follows: N solutions (S_i, $i = 1,\ldots, N$) are randomly generated as the starting population, where S_i is represented as a set of three feature vectors $S_i = (\Phi^i,\mathrm{X}^i,\Gamma^i)$. The first feature vector $\Phi^i = (f_1^i,\ldots,f_m^i)$, an m-dimensional vector, represents the binary representations of m features: If $f_j^i = 1$, the j^{th} feature is kept; if $f_j^i = 0$, the j^{th} feature is eliminated. The other two vectors $\mathrm{X}^i = (C_1^i,\ldots,C_{20}^i)$ and $\Gamma^i = (\gamma_1^i,\ldots,\gamma_{10}^i)$ are the binary representations of the penalty parameter C and the kernel parameter γ of the SVM, respectively. The fitness function is defined as the prediction accuracy of disulfide connectivity. In the initial population, N solutions are randomly divided into two halves. The "Father" α and the "Mother" β of the population are defined as

$$\alpha = (\Phi^\alpha, \mathrm{X}^\beta, \Gamma^\alpha) = \max\{S_1,\ldots,S_{N/2}\},$$

$$\beta = (\Phi^\beta, \mathrm{X}^\beta, \Gamma^\beta) = \max\{S_{N/2+1},\ldots,S_N\}.$$

Three basic mechanisms driving the evolutionary processes in one generation are selection, mutation, and crossover.

Selection operator

In the τ^{th} generation, the selection operators are defined as:

$$\alpha^\tau = \max\{S_1^{\tau-1},\ldots,S_{N/2}^{\tau-1},\alpha^{\tau-1}\},$$

$$\beta^\tau = \max\{S_{N/2+1}^{\tau-1},\ldots,S_N^{\tau-1},\beta^{\tau-1}\}.$$

Note that for the special case of $\tau = 0$, α^0, and β^0 are defined to be 0. A new solution S_i^τ is set to α^τ if i is odd, while S_i^τ is set to β^τ if i is even.

Mutation operator

We apply two types of mutation to the solution S_i. In the case of $i = 1,\ldots, N/2$, every bit b of the vectors (i.e. Φ^i) is subject to mutation: $b = \sim b$, if the mutation rate is less than a mutation threshold $\mu_0 = 0.1$. In the case of $i = N/2{+}1,\ldots, N$, we randomly choose a bit from each of the vectors (hence, a total of three bits per each solution S_i). These bits are then subject to mutation without any mutation threshold.

Crossover operators

The crossover operations are carried out between S_{2p-1} and S_{2p}, where $p = 1,\ldots, N/2$. The crossover operations are as follows: one-point crossover is performed between Φ^{2p-1} and Φ^{2p} if the crossover rate is less than the crossover threshold $\mu_1 = 0.5$. Similar one-point crossover operations are also applied to the vectors X and Γ.

A

Disulfide patterns	Predicted states	Matrix elements
c_1, c_2, c_3, c_4	$\sigma_1 = (C_1 \oplus C_2, C_3 \oplus C_4)$	$M_{12} = M_{12} + 1, M_{34} = M_{34} + 1$
c_1, c_2, c_3, c_5	$\sigma_2 = (C_1 \oplus C_2, C_3 \otimes C_4)$	$M_{12} = M_{12} + 1, M_{35} = M_{35}$
c_1, c_2, c_3, c_6	$\sigma_1 = (C_1 \oplus C_2, C_3 \oplus C_4)$	$M_{12} = M_{12} + 1, M_{36} = M_{36} + 1$
c_1, c_2, c_4, c_5	$\sigma_2 = (C_1 \oplus C_2, C_3 \otimes C_4)$	$M_{12} = M_{12} + 1, M_{45} = M_{45}$
c_1, c_2, c_4, c_6	$\sigma_4 = (C_1 \oplus C_3, C_2 \oplus C_4)$	$M_{14} = M_{14} + 1, M_{26} = M_{26} + 1$
c_1, c_2, c_5, c_6	$\sigma_1 = (C_1 \oplus C_2, C_3 \oplus C_4)$	$M_{12} = M_{12} + 1, M_{56} = M_{56} + 1$
c_1, c_3, c_4, c_5	$\sigma_9 = (C_1 \otimes C_4, C_2 \oplus C_3)$	$M_{15} = M_{15}, M_{34} = M_{34} + 1$
c_1, c_3, c_4, c_6	$\sigma_7 = (C_1 \oplus C_4, C_2 \oplus C_3)$	$M_{16} = M_{16} + 1, M_{34} = M_{34} + 1$
c_1, c_3, c_5, c_6	$\sigma_3 = (C_1 \otimes C_2, C_3 \oplus C_4)$	$M_{13} = M_{13}, M_{56} = M_{56} + 1$
c_1, c_4, c_5, c_6	$\sigma_9 = (C_1 \otimes C_4, C_2 \oplus C_3)$	$M_{16} = M_{16}, M_{45} = M_{45} + 1$
c_2, c_3, c_4, c_5	$\sigma_9 = (C_1 \otimes C_4, C_2 \oplus C_3)$	$M_{25} = M_{25}, M_{34} = M_{34} + 1$
c_2, c_3, c_4, c_6	$\sigma_7 = (C_1 \oplus C_4, C_2 \oplus C_3)$	$M_{26} = M_{26} + 1, M_{34} = M_{34} + 1$
c_2, c_3, c_5, c_6	$\sigma_3 = (C_1 \otimes C_2, C_3 \oplus C_4)$	$M_{23} = M_{23}, M_{56} = M_{56} + 1$
c_2, c_4, c_5, c_6	$\sigma_1 = (C_1 \oplus C_2, C_3 \oplus C_4)$	$M_{24} = M_{24} + 1, M_{56} = M_{56} + 1$
c_3, c_4, c_5, c_6	$\sigma_2 = (C_1 \oplus C_2, C_3 \otimes C_4)$	$M_{34} = M_{34} + 1, M_{56} = M_{56}$

B

	c_1	c_2	c_3	c_4	c_5	c_6
c_1	–	5	0	1	0	1
c_2		–	0	1	0	2
c_3			-	6	0	1
c_4				–	1	0
c_5					–	4
c_6						–

Fig. 3. An example illustrating the construction of the connectivity matrix and the determination of disulfide connectivity therefrom. (**A**) For a sequence with six cysteines, disulfide connectivity can be represented in terms of 15 disulfide patterns in CP_2 representation. The matrix elements of the connectivity matrix are constructed from the predicted bonding states using Eq. (1). (**B**) The constructed connectivity matrix. The predicted disulfide connectivity pattern is $[C_1C_2, C_3C_4, C_5C_6]$, which has maximal score $\Omega_T = M_{12} + M_{34} + M_{56} = 15$.

The Connectivity Matrix

The connectivity matrix M is defined in terms of the bonding states $c_p\tilde{O}c_q$. The initial matrix elements M_{pq} are set to 0, and the rules to construct the matrix are:

$$M_{pq} = M_{pq} + 1, \text{ if } \tilde{O} = \oplus \tag{1a}$$

$$M_{pq} = M_{pq}, \text{if } \tilde{O}_1 = \otimes \tag{1b}$$

The score Ω_T of the disulfide connectivity pattern T is computed from M by

$$\Omega_T = \sum_{i<j}^{\mathrm{T}}{}' M_{ij} \tag{2}$$

where Σ' indicates that any two index pairs (i,j) and (i',j') under the summation sign should satisfy the requirements $i \neq i'$ and $j \neq j'$. The disulfide pattern with the maximal score, i.e. $\max\{\Omega_T\}$, is taken as the prediction.

The notation M_χ is used to denote the connectivity matrix based on the χ coding scheme (for example, M_{CP_1} denotes the connectivity matrix based on CP_1). We also define the hybrid connectivity matrix as: $M_{\mathrm{CP}_1+\mathrm{CP}_2+\mathrm{GA}} = wM_{\mathrm{CP}_1+\mathrm{GA}} + (1-w)M_{\mathrm{CP}_2+\mathrm{GA}}$, where w is the weight. The weight is numerically determined using the grid-search method.

In Figure 3, we show an example to illustrate the construction of the connectivity matrix for a sequence with three disulfide bonds. In CP_2 representation, the disulfide connectivity pattern is represented in terms of 15 disulfide patterns [Fig. 3(A)]. The connectivity matrix constructed from these bonding states is shown in Figure 3(B). The scores of all possible types of disulfide connectivity are then computed by Eq. (2). In this particular example, the predicted disulfide pattern is $[C_1C_2,C_3C_4,C_5C_6]$. The complete flowchart of our method is schematically shown in Figure 4. Optimizing feature selection is the most time consuming step, which will take several hours to up to 2 days computational time on a 3-GHz Pentium processor, depending on the size of the data set. The typical number of generations is around 3000 and the features selected are highly reproducible. The same features are used for both training and testing processes.

Performance Indices

Following the previous studies,[1,5] we use two assessment indices to evaluate the performance of the classifiers. The first one is the cysteine pair-based index Q_c, which is the fraction of the correctly predicted disulfide bridges and is defined as

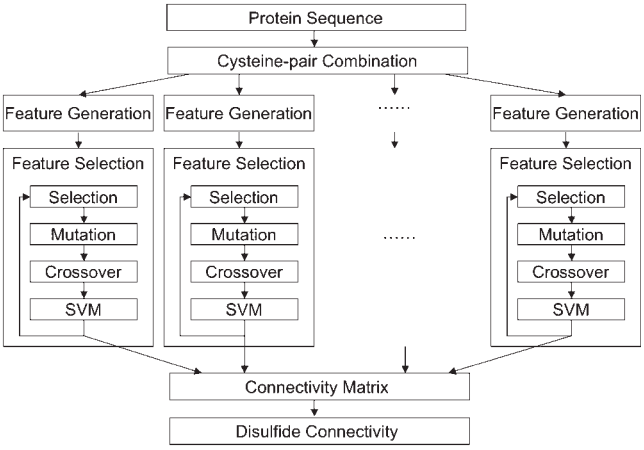

Fig. 4. The flowchart of our procedure to predict disulfide connectivity from the protein sequence.

$$Q_c = \frac{1}{N_c}\sum_{i=1}^{N_c}\delta_{c_i} \tag{3}$$

where N_c is the total number of disulfide bridges, and δ_{c_i} is 1 if the i^{th} predicted disulfide bridge is correct, and 0 otherwise. The second assessment measure is the pattern-based index Q_p, which is the fraction of proteins whose global disulfide pattern is correctly predicted and is defined as

$$Q_p = \frac{1}{N_p}\sum_{i=1}^{N_p}\delta_{p_i} \tag{4}$$

where N_p is the total number of disulfide proteins and δ_{p_i} is 1 if the predicted connectivity pattern of the i^{th} protein is correct, and 0 if the predicted connectivity pattern of the ith protein is incorrect.

Datasets

We use the same dataset as used in the previous studies.[1,3,5] This dataset, extracted from the SWISS-PROT database release no. 39,[24] contains only the sequences with experimentally verified intrachain disulfide bond annotations and excludes those with the *hypothetical* disulfide bonds, that is, those disulfide bonds designated as "probable," "potential," or "by similarity." Also, the interchain disulfide bonds are not considered in this study. In the SWISS-PROT data set, the sequences with disulfide bonds from two to five account for about 80% of the total disulfide sequences. As a result, our dataset comprises only the sequences with 2–5 disulfide bonds: 168 sequences with two disulfide bonds ($B = 2$), 177 three ($B = 3$), 95 four ($B = 4$), and 42 five ($B = 5$). The total number of the sequences is 482. All sequences of the data set have pairwise sequence identities less than 30%. All results reported in this work are based on the fourfold cross validation procedures.

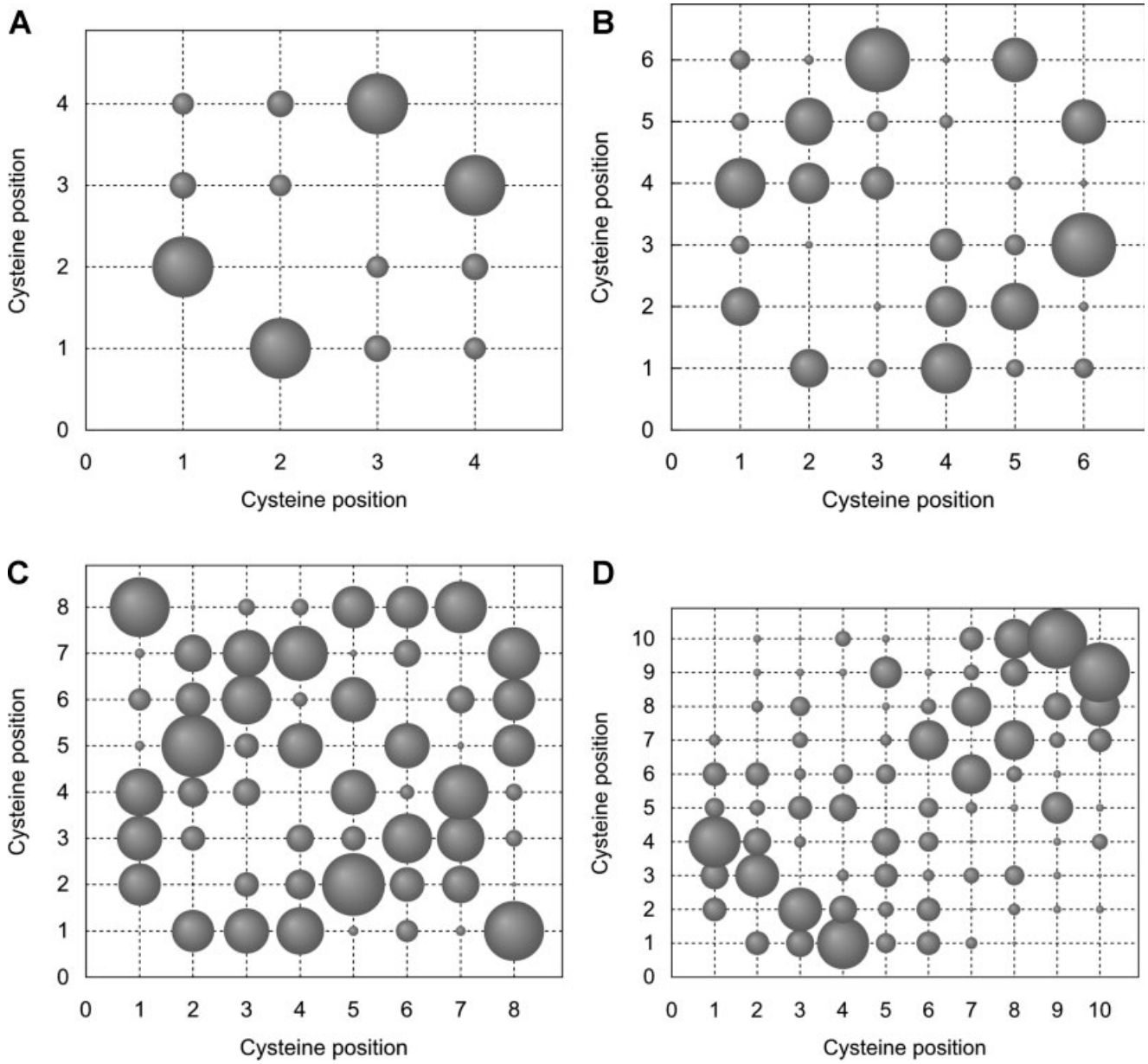

Fig. 5. The distribution patterns of disulfide connectivity for (**A**) $B = 2$, (**B**) $B = 3$, (**C**) $B = 4$, (**D**) $B = 5$.

RESULTS

Distribution of Cysteine Pairs Forming Disulfide Bridges

Figure 5 shows the distribution of disulfide bridges for $B = 2,\ldots,5$ in the dataset. In the simplest case of $B = 2$ [Fig. 5(A)], two of the most populated disulfide bridges are C_1C_2 and C_3C_4, and the dominant connectivity pattern is $[C_1C_2, C_3C_4]$. The distribution of the connectivity patterns becomes more complicated when the number of disulfide bonds increases [Fig. 5(B–D)]. We notice that there are distinct distribution patterns of the connectivity patterns for different Bs. For example, for $B = 3$ [Fig. 5(B)], the disulfide bridges C_3C_6 and C_1C_4 are the dominant ones; for $B = 4$ [Fig. 5(C)], the dominant disulfide bridges are C_2C_5 and C_1C_8; for $B = 5$ [Fig. 5(D)], the most dominant ones are C_1C_4 and C_9C_{10}.

Results Based on Different Coding Schemes

Table I compares the prediction results based on different coding schemes. All methods perform well for $B = 2$, but their performances vary significantly when B increases. CP_1 + GA and CP_2 + GA in general outperform CP_1 and CP_2. The performance gain is more significant for larger B. For example, CP_1 + GA yields a prediction accuracy Q_p that is 13% higher than CP_1 for $B = 2$ and

TABLE I. Comparison of the Prediction Results (in %) Based on Different Coding Schemes of Disulfide Connectivity

	$B = 2$		$B = 3$		$B = 4$		$B = 5$		Overall	
Method	Q_p	Q_c	Q_p	Q_c	Q_p	Q_c	Q_p	Q_c	Q_p	Q_c
CP_1	72.0	72.0	42.4	52.4	15.8	26.8	4.8	12.4	44.2	44.5
CP_2	73.8	73.8	49.7	60.6	27.4	41.3	7.1	36.2	50.0	55.1
CP_1 + GA	85.1	85.1	63.8	71.0	59.0	73.7	45.2	68.6	68.7	74.6
CP_2 + GA	81.0	81.0	66.1	73.3	39.0	51.8	16.7	48.6	61.6	66.0
CP_1 + CP_2 + GA	85.7	85.7	74.6	79.7	63.2	77.1	47.6	71.4	73.9	79.2

21% higher for $B = 3$, indicating that the effect of inherent noise in the original feature vector becomes more pronounced for larger B. Our results show that feature extraction using GA is quite effective in extracting information relevant to disulfide connectivity. The hybrid method is able to achieve a prediction accuracy over 70% in both Q_p and Q_c. Although CP_1 and CP_2 codings are theoretically equivalent to each other, but since they use different numbers of bonding states to describe the same disulfide pattern [Fig. 1(B)], their performances are substantially different in practice. Such difference also reflects that, though the performance of CP_2 is better than that of CP_1, upon feature extract CP_1 + GA outperforms CP_2 + GA. The hybrid method can obviously take advantage of the complementary natures of both CP_1 and CP_2 codings to give the best performance. Note that the pattern-based indices Q_p monotonously decrease when B increases. This is because Q_p is a very strict assessment index, which requires perfect prediction of all disulfide bonds of a sequence. On the other hand, the downward trends of the cysteine pair-based indices Q_c are less pronounced, which suggests that the predictions of cysteine pairs are more or less independent of each other.

It will be instructive to examine the amino acid types being selected by GA. We will first explain the term called *selection percentage*: in the case of CP_1, there are six possible combinations of cysteine pairs for $B = 2$, i.e., $C_1 - C_2$, $C_1 - C_3$, $C_1 - C_4$, $C_2 - C_3$, $C_2 - C_4$, and $C_3 - C_4$, 15 for $B = 3$, 27 for $B = 4$ and 45 for $B = 5$. Since each combination has its own feature selection sets, there are a total of $6 + 15 + 28 + 45 = 94$ selection sets. If the selection percentage for the amino acid type F is 50%, it means that there are $94 \times 0.5 = 46$ selection sets being selected by F. The higher the selection percentage of a particular amino acid type, the more significant it is related to the disulfide connectivity. Figure 6 shows the selection percentages of amino acid types by GA for AA and CC codings. The trends of the selection percentages of C_1 and C_2 for the AA coding are in general quite similar. Amino acids like F, W, N, K, R, M, C, and P are among the most selected, while S, E, L, A, I, and G among the least selected. In the CC coding, the CP_1 + GA model prefers F, S, V, Q, P, H, R, and G, while the CP_2 + GA model does not show particular preferences for any amino acid types.

It will be interested to check how the model performance depends on the training set size. In the case of $B = 2$, the model CP_1 + CP_2 + GA yielded an overall prediction accuracy of 0.82, 0.85, 0.86 and 0.85 for the training-set size ratio of 0.6, 0.8, 0.9, and 1, respectively. These results indicate that the performance of our model gets better when the training size increases and appears to reach a plateau when the size increases further.

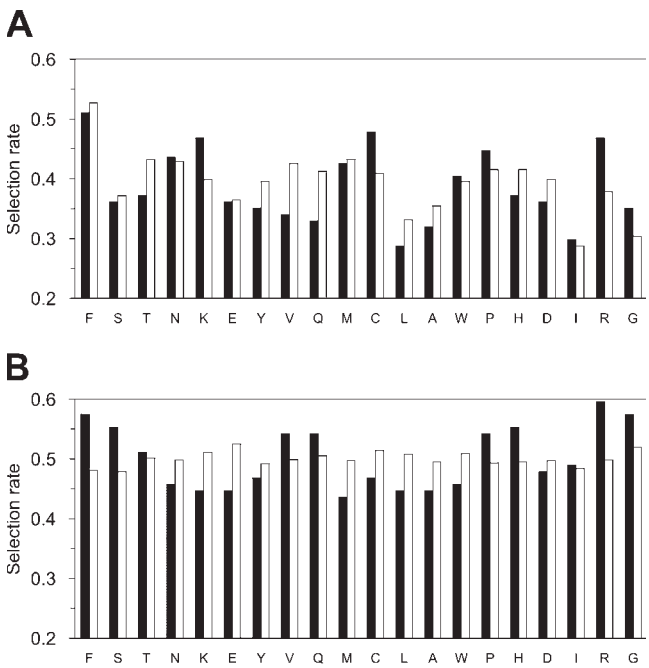

Fig. 6. The selection percentages of amino acid types by GA for (**A**) AA and (**B**) CC coding schemes. The CP_1 is in solid bar and the CP_2 is in empty bar.

Comparison With Other Methods

In Table II, we compare our results with those of other methods in the literature. Our method outperforms other approaches in all cases from $B = 2$ to $B = 5$. For example, for $B = 2$, our prediction accuracy is more than 12–30% higher than others[1–5] in both Q_p and Q_c (with the exception of the results of Tsai et al.,[25] which will be discussed later). For $B = 3$, our prediction accuracy is 24–54% higher in Q_p and 11–42% higher in Q_c than others. Our overall prediction accuracy is 19–45% higher in Q_p and 22–41% higher in Q_c than other methods. Here we like to comment on the recent work by Tsai et al.[25] Although they were able to obtain good overall prediction accuracy,

TABLE II. Comparison of Predictive Results (in %) of Different Approaches to Predict Disulfide Connectivity

	$B = 2$		$B = 3$		$B = 4$		$B = 5$		Overall	
Method	Q_p	Q_c	Q_p	Q_c	Q_p	Q_c	Q_p	Q_c	Q_p	Q_c
This work	85.7	85.7	74.6	79.7	63.2	77.1	47.6	71.4	73.9	79.2
Tsai et al.[25]	79	79	53	62	55	70	58	71	63	70
Chen and Hwang[5]	74	74	61	69	30	40	12	31	55	57
2D-RNN[4]	74	74	51	61	27	44	11	41	49	56
RNN[3]	73	73	41	51	24	37	13	30	44	49
NNGM[2]	68	68	22	37	20	37	2	26	34	42
MCGM[1]	56	56	21	36	17	37	2	21	29	38

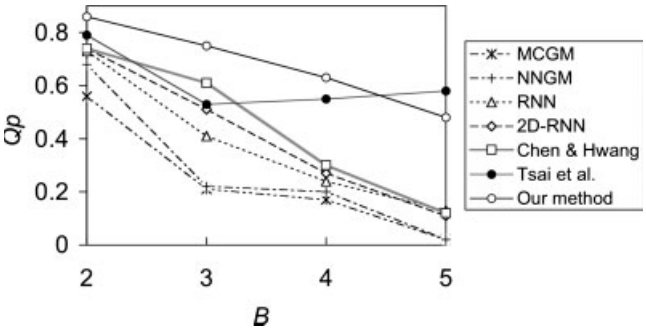

Fig. 7. Comparison of the protein-based assessment index Q_P of different methods versus disulfide bonds B.

their Q_p gets higher when B increases, as shown in Figure 7. Such an anomalous upward trend of Q_p is in sharp contrast to that of other approach. Obviously, further efforts are needed to clarify this issue.

DISCUSSION

In this work, we represent the disulfide connectivity patterns in terms of cysteine pairs, the bonding states of which are then predicted using SVM. In addition, GA is used to optimize feature selection. From the bonding states of the cysteine pairs, we are able to build the connectivity matrix, through which the disulfide connectivity patterns are predicted. Our method outperforms other methods in the literature and achieves over 70% overall prediction accuracy in both pattern-based and cysteine pair-based assessment indices. Our results indicate that it is possible to reliably predict disulfide connectivity from protein sequences. Our method may provide a useful tool for structural modeling and functional analysis of disulfide proteins.

ACKNOWLEDGMENTS

We are grateful to both hardware and software support from the Structural Bioinformatics Core Facility at National Chiao Tung University.

REFERENCES

1. Fariselli P, Casadio R. Prediction of disulfide connectivity in proteins. Bioinformatics 2001;17:957–964.
2. Fariselli P,Riccobelli P,Casadio R, editors.A neural network-based method for predicting the disulfide connectivity in proteins. In:Damiiani E,Jain LC,Howlett RJ,Ichalkaranje N, editors. Knowledge based intelligent information engineering systems and allied technologies (KES 2002), Vol.1. Amsterdam: IOS Press; 2002.
3. Vullo A, Frasconi P. Disulfide connectivity prediction using recursive neural networks and evolutionary information. Bioinformatics 2004;20:653–659.
4. Baldi P, Cheng J, Vullo A. Advances in neural information processing systems. In: Saul LK, Weiss Y, Bottou L, editors. Large-scale prediction of disulfiphide bond connectivity. Cambridge, MA: MIT press; 2005. pp97–104.
5. Chen YC, Hwang JK. Prediction of disulfide connectivity from protein sequences. Proteins 2005;61:507–512.
6. Cheng J, Saigo H, Baldi P. Large-scale prediction of disulphide bridges using kernel methods, two-dimensional recursive neural networks, and weighted graph matching. Proteins 2006;62:617–629.
7. Vapnik V. The nature of statistical learning theory. New York: Springer; 1995.
8. Chuang CC, Chen CY, Yang JM, Lyu PC, Hwang JK. Relationship between protein structures and disulfide-bonding patterns. Proteins 2003;53:1–5.
9. van Vlijmen HW, Gupta A, Narasimhan LS, Singh J. A novel database of disulfide patterns and its application to the discovery of distantly related homologs. J Mol Biol 2004;335:1083–1092.
10. Hua S, Sun Z. A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. J Mol Biol 2001;308:397–407.
11. Ward JJ, McGuffin LJ, Buxton BF, Jones DT. Secondary structure prediction with support vector machines. Bioinformatics 2003;19:1650–1655.
12. Yu CS, Wang JY, Yang JM, Lyu PC, Lin CJ, Hwang JK. Fine-grained protein fold assignment by support vector machines using generalized n-peptide coding schemes and jury voting from multiple-parameter sets. Proteins 2003;50:531–536.
13. Chen YC, Lin YS, Lin CJ, Hwang JK. Prediction of the bonding states of cysteines using the support vector machines based on multiple feature vectors and cysteine state sequences. Proteins 2004;55:1036–1042.
14. Yu CS, Lin CJ, Hwang JK. Predicting subcellular localization of proteins for Gram-negative bacteria by support vector machines based on n-peptide compositions. Protein Sci 2004;13:1402–1406.
15. Huang SW, Hwang JK. Computation of conformational entropy from protein sequences using the machine-learning method—application to the study of the relationship between structural conservation and local structural stability. Proteins 2005;59:802–809.
16. Lei Z, Dai Y. An SVM-based system for predicting protein subnuclear localizations. BMC Bioinformatics 2005;6:291.
17. Scholkopf B, Smola AJ. Learning with kernels—support vector machines, regularization, optimization, and beyond. Cambridge, MA: MIT Press; 2002.
18. Lin C-J. Formulations of support vector machines: a note from an optimization point of view. Neural Comput 2001;13:307–317.

19. Chang C-C, Lin C-J. LIBSVM v. 2.81: a library for support vector machines. 2005. Available athttp://www.csie.ntu.edu.tw/~cjlin/libsvm
20. Duan K, Keerthi SS, Poo AN. Evaluation of simple performance measures for tuning SVM hyperparameters. Neurocomputing 2003;51:41–59.
21. Nakashima H, Nishikawa K, Ooi T. The folding type of a protein is relevant to the amino acid composition. J Biochem 1986;99:152–162.
22. Hua S, Sun Z. Support vector machine approach for protein subcellular localization prediction. Bioinformatics 2001;17:721–728.
23. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 1997;25:3389–3402.
24. Bairoch A, Apweiler R. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. Nucleic Acids Res 2000;28:45–48.
25. Tsai CH, Chen BJ, Chan CH, Liu HL, Kao CY. Improving disulfide connectivity prediction with sequential distance between oxidized cysteines. Bioinformatics 2005;21:4416–4419.