Predicting Metal-Binding Sites from Protein Sequence

Andrea Passerini, Marco Lippi, and Paolo Frasconi

Abstract—Prediction of binding sites from sequence can significantly help toward determining the function of uncharacterized proteins on a genomic scale. The task is highly challenging due to the enormous amount of alternative candidate configurations. Previous research has only considered this prediction problem starting from 3D information. When starting from sequence alone, only methods that predict the bonding state of selected residues are available. The sole exception consists of pattern-based approaches, which rely on very specific motifs and cannot be applied to discover truly novel sites. We develop new algorithmic ideas based on structured-output learning for determining transition-metal-binding sites coordinated by cysteines and histidines. The inference step (retrieving the best scoring output) is intractable for general output types (i.e., general graphs). However, under the assumption that no residue can coordinate more than one metal ion, we prove that metal binding has the algebraic structure of a matroid, allowing us to employ a very efficient greedy algorithm. We test our predictor in a highly stringent setting where the training set consists of protein chains belonging to SCOP folds different from the ones used for accuracy estimation. In this setting, our predictor achieves 56 percent precision and 60 percent recall in the identification of ligand-ion bonds.

Index Terms—Metal-binding prediction, machine learning, structured-output learning, greedy algorithms.

1 Introduction

Metalloproteins are a large and diverse class of proteins which are crucial for many aspects of the cell life. Their intrinsic metal ions provide catalytic, regulatory, or structural roles critical to protein function [1]. Metals participate in a wide variety of biological processes, from enzyme catalysis, as in the cases of respiration and photosynthesis [2], to functional RNA stabilization [3], [4], or regulation of the catalytic rate of ribozymes [5]. Moreover, metals are implicated in many diseases for which medicine is still seeking an effective treatment, such as Parkinson or Alzheimer [6], and they can also be responsible of DNA damages [7]. Knowing that a functionally uncharacterized protein binds a metal ion in its native conformation is thus a relevant information for understanding its function.

In recent years, high-throughput experimental techniques based on X-ray absorption spectroscopy [8], [9], [10] proved capable of identifying metalloproteins with high reliability. However, these techniques cannot detect the ligands involved in binding the metal ion(s). The simplest *in-silico* solution in this context is *bonding state prediction*, a binary classification task where individual residues are predicted (from sequence alone) as metal binding or not. Bonding state prediction cannot determine whether two residues coordinate the same metal ion but still it has been investigated in several previous studies. The first approaches are based on regular expression mining [11]. The drawback of using regular expressions is that they are usually quite specific but may give a low coverage (many false negatives). Machine learning techniques have been recently applied to predict the metal-bonding state of residues. Existing approaches for metal-bonding state prediction have mainly focused on CYS only [12], CYS and HIS binding transition metals [13], or CYS, HIS, ASP, and GLU binding zinc ions [14], [15]. For CYS, classification in three rather than two states can be made by introducing a class for disulphide bridges, thus discriminating between free, metal binding and disulphide binding [12]. Other works have shown the usefulness of 3D information for predicting bonding state [16]. See [17] for a detailed review of current methods.

Knowing the metal-bonding state of residues, however, is not sufficient to fully characterize binding sites. Many proteins bind multiple ions and predicting the configuration of the sites requires to identify the set of residues coordinating each ion. Few recent works [18], [19], [20] have addressed this challenging task assuming knowledge of the protein 3D structure. This allows to complement experimental evidence, by identifying apo-proteins, i.e., proteins solved in their ion-free form, or detecting experimental artifacts, i.e., binding of metals at adventitious sites. However, its applicability is limited to structurally determined proteins. Our work aims at overcoming this limitation, by predicting the configuration of metal-binding sites from sequence information only. This would allow to extend the applicability of the approach to tasks like: detailed functional annotation of experimentally unsolved proteins, e.g., characterization of active sites in enzymes,

- *A. Passerini is with the DISI - Dipartimento di Ingegneria e Scienza dell'Informazione, Università degli Studi di Trento, Sommarive st. 5, Povo, TN 38100, Italy. E-mail: passerini@disi.unitn.it.*
- *M. Lippi is with the DII - Dipartimento di Ingegneria dell'Informazione, Università degli Studi di Siena, via Roma 56, Siena 53100, Italy. E-mail: lippi@dii.unisi.it.*
- *P. Frasconi is with the DSI - Dipartimento di Sistemi e Informatica, Università degli Studi di Firenze, Via di Santa Marta, 3, Firenze 50139, Italy. E-mail: p-f@dsi.unifi.it.*

Manuscript received 2 Feb. 2011; revised 27 Apr. 2011; accepted 2 May 2011; published online 16 May 2011.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBB-2011-02-0030. Digital Object Identifier no. 10.1109/TCBB.2011.94.

many of which employ metal ions as cofactors [21]; experimental determination of new metalloproteins, as the prediction of metal-binding sites can guide the preparation of samples for in vitro studies [22], [9]. The only approaches [22], [23] we are aware of predicting metal-binding sites from sequence are pattern-based ones, mining motifs such as those recorded in the PROSITE [24] database. However, these motifs are either very specific or very general and cannot be effectively applied to discover novel sites. They can nonetheless produce useful features to be used in combination with more complex techniques, as will be shown in our experimental results.

Predicting the configuration of metal-binding sites is an extremely challenging task: first, the number of admissible configurations is exponential in the number of candidate ligands; second, the participation of a residue to a metal-binding site should not be predicted independently from the other residues: interdependencies between candidates should be taken into account to form a *collective* prediction. Our solution consists of a two-stage approach that takes advantage of structured-output learning [25] at both stages. We first predict the metal-bonding state of candidate residues (positive cases are metal-binding residues, negative cases are the rest, including half cystines, i.e., cysteines involved in disulphide bridges). Residues predicted as metal bonded are then passed to the second stage which outputs the overall configuration by grouping together ligands predicted to bind the same ion. Note that a similar strategy has been often used for the related but simpler problem of disulfide connectivity prediction [26], where the task consists of pairing each half cystine with its partner in the sequence. We address metal-bonding state prediction as a sequence labeling task, collectively assigning the bonding state to all candidate ligands in the sequence. We employ an SVM-HMM [27], a model that can be essentially interpreted as a hidden Markov model with discriminatively learned parameters. The second stage is formalized as a link prediction task in a bipartite graph, where a ligand node is connected to an ion node if and only if the residue coordinates that ion. We show that the problem has the algebraic structure of a matroid, which guarantees the optimality of a greedy search algorithm. Intuitively, we start from the empty structure and incrementally build the output by adding one edge at the time. The search is guided by a scoring function evaluating candidate structures. We adopt an online learning strategy where constraints derived from partial structures are sampled during the greedy search. The greedy approach was originally introduced in [28]. Here, we considerably extend the method by introducing a two-stage predictor and a much richer similarity measure between structures, resulting in substantial performance improvements (see Section 5.1). We also provide a deeper experimental evaluation on highly challenging generalization tasks across Structural Classification of Protein (SCOP) superfamilies and folds. An online service implementing the predictor is available at http://metaldetector.dsi.unifi.it/v2.0/.

The paper is organized as follows: in Section 2, we provide a detailed description of the problem and motivate the requirement for a structured-output approach. Our proposed solution is discussed in Section 3. Section 4 describes the data sets used in the experimental evaluation whose results are reported in Section 5. We finally draw some conclusions in Section 6.

TABLE 1
Percentage and Fraction of Times a Given Amino Acid Type Binds a Specific Metal Ion (or Complex) in Chains Containing a Binding Site for that Ion

Metal	CYS		HIS		ASP		GLU	
Zn	46	(508/1115)	24	(374/1562)	4	(117/3204)	2	(89/3705)
Heme	50	(115/230)	34	(151/450)	1	(5/854)	0	(2/925)
Fe/S	63	(205/326)	3	(10/329)	0	(0/763)	0	(1/886)
Cu	33	(36/108)	32	(86/269)	0	(2/513)	0	(2/455)
Cd	62	(48/77)	32	(25/79)	12	(26/216)	13	(35/262)
Fe	13	(16/122)	18	(59/325)	2	(15/610)	3	(25/745)
Ni	4	(2/46)	16	(18/112)	2	(5/250)	1	(2/271)
Any	48	(930/1923)	25	(723/2942)	3	(169/6045)	2	(156/6836)

2 Problem Description

Given a metalloprotein chain sequence as input, our aim is to predict the number of bound metal ions and, for each ion, the ligands in the sequence. This is not a full 3D characterization of the sites geometry, with angles and distances, but rather the prediction of the coordination relationship between ions and their ligands. With a slight abuse of terminology, in the rest of the paper, we will name metal-binding geometry (MBG) the complete specification of this relationship. Following [13], we focus on proteins binding transition metals, which make up about 66 percent of the PDB metallo-chains and include iron and zinc, the two most abundant metal ions involved in cellular functioning. Transition metals usually form coordinate covalent bonds with the protein ligands, showing much higher binding affinities than the electrostatic interactions typical of alkali and alkaline earth metals. These last metals can also bind protein backbone carbonyls, and virtually any amino acid qualifies as a candidate ligand. On the other hand, by far the most common transition-metal-binding residues are CYS, HIS, ASP, GLU, covering about 92 percent ligands of the structurally known proteins. However, ASP and GLU are rarely found in metal-binding sites, when compared to their natural frequency of occurrence in proteins (see Table 1). By focusing on CYS and HIS only, we cover about 74 percent of transition-metal ligands. Finally, for computational efficiency reasons (see Section 3), we will assume that each ligand binds exactly one ion. This is almost always the case for CYS and HIS, while a full modeling of sites involving ASP and GLU would probably require to admit shared ligands.

Fig. 1 shows an example of a protein kinase C cystein-rich domain (PDB entry 1tbn). It highlights the 3D structure of the binding sites (top) and a graph-based representation of the input sequence together to the desired output (bottom). While showing a rather simple domain, the figure already highlights the complexity of the prediction task. The number of admissible binding configurations for a given protein chain having n candidate ligands is the multinomial coefficient $\frac{n!}{k_1!k_2!\cdots k_m!(n-k_1-\cdots-k_m)!}$ where m is the number of ions and k_i the number of ligands for ion ι_i. In practice, each ion is coordinated by a variable number of ligands (typically ranging from 1 to 4, but occasionally more), and each protein

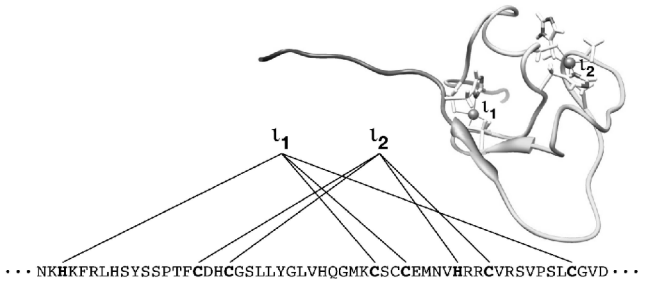


Fig. 1. Protein kinase C cystein-rich domain (PDB entry 1tbn): (top) 3D structure with binding sites highlighted; (bottom) representation of the input sequence and the desired output as a bipartite graph.

chain binds a variable number of ions (typically ranging from 1 to 4). In the small protein of Fig. 1, if we consider CYS, HIS, ASP, and GLU as candidate ligands, we would have $2 \cdot 10^7$ admissible conformations, assuming knowledge of the number of ions and their coordination number. Restricting candidates to CYS and HIS only still generates $9 \cdot 10^4$ conformations. If we consider a more complex example like the small subunit of formate dehydrigenase (PDB code 1h0hB), with three ions coordinated by four residues each, these numbers grow to $7 \cdot 10^{15}$ and $5 \cdot 10^{10}$ for CYS, HIS, ASP, GLU and CYS, HIS candidates, respectively. The actual search space is even larger, as during prediction we do not know the number of ligands nor their respective coordination numbers.

Multiclass classification is not an option in such a huge conformation space. A straightforward approach to apply off-the-shelf predictors to this learning problem would be that of training a pairwise classifier predicting whether two candidate ligands actually bind the same ion. This approach would however easily produce inconsistent predictions, with residue pairs (A,B) and (B,C) predicted as positive and pair (A,C) as negative, and would fail to capture the overall relationship among the set of ligands of a certain ion. Generalizing this strategy to triplets and quartets of residues, up to the maximum possible coordination number k, still requires to solve inconsistencies arising from overlapping sets. Furthermore, it would clearly generate an exponential increase of the number of candidate examples. These grow as $O(k^n)$ with n number of candidate ligands, and only a tiny fraction of them represents true sites, creating a dramatically unbalanced problem. In a word, this task has to be addressed with *collective* strategies, which jointly produce an entire solution and efficiently address the exponential size of the resulting search space.

3 Methods

3.1 Problem Formalization

Let $\mathcal{T}$ denote the set of amino acids used as candidate ligands (in our experiment, only CYS and HIS are included in this set). To simplify presentation, we initially assume that the bonding state of each CYS and HIS is known (see also Section 3.7). We also assume that at most m ions are bound to any given chain. In all our experiments, we fix $m = 4$, covering 97 percent of transition metals in current PDB. Symbols associated with metal ion identifiers are collected in the set $\mathcal{I} = \{\iota_1, \ldots, \iota_m\}$. The goal is to predict the metal-binding geometry of a given chain, which can be formally described by introducing the following binary relation between residues and ion identifiers: $\mathrm{coord}(t, \iota)$ is true if and only if residue at position t (for $t = 1, \ldots, T$, where T is the chain length) coordinates ion $\iota \in \mathcal{I}$. We denote by $\mathcal{C} = \{t : 1 \leq t \leq T, \mathrm{res(t)} \in \mathcal{T}\}$ the set of candidate ligands. The task can be also expressed in a graphical formulation. We are given a protein chain of length T and the goal is to predict the bipartite graph (v, y) with vertex set $v = \mathcal{C} \cup \mathcal{I}$ and edge set $y \subset \mathcal{C} \times \mathcal{I}$. Edges connect residue indices to ion identifiers and the MBG edge set y is a collective truth assignment to every fact $\mathrm{coord}(t, \iota)$ for $t = 1, \ldots, T$ and $\iota \in \mathcal{I}$.

The MBG problem is closely related but not equivalent to the matching problem studied in [29] in the context of disulfide bridge prediction. In the MBG case, more than one edge can be incident to vertices belonging to $\mathcal{I}$.

Note that in our formulation, ion identifiers are mere placeholers and carry no information about the chemical element or prosthetic group. Hence, any two label-isomorphic bipartite graphs (obtained by exchanging two metal ion vertices) are equivalent. Outputs y should be therefore regarded as equivalence classes of structures (where all permutations of $\iota_1, \ldots, \iota_m$ are collected in the same class). For simplicity, we will slightly abuse notation and avoid this distinction in the following.

3.2 Formulation as Structured-Output Prediction

One instance consists of a pair (x, y) where the input portion x contains information about the chain sequence (possibly enriched with multiple alignment profiles) and the output portion y is the associated MBG edge set (see Section 4 for details on data preparation). The structured-output prediction approach can be formulated as follows: first, introduce a joint feature vector $\phi_x(y)$ of inputs and outputs (this can be done explicitly or implicitly via a kernel function as explained in Section 3.6). Second, define a linear *compatibility function* between inputs and outputs as $F_x(y) = w^T \phi_x(y)$, where w is a parameter vector to be estimated from data (see Section 3.4). Third, obtain the prediction $f(x)$ by searching for the best configuration:

$$f(x) = \arg\max_{y \in \mathcal{Y}_x} F_x(y), \tag{1}$$

where $\mathcal{Y}_x$ is the set of admissible output configurations. A clearer intuition can perhaps be gained by interpreting F as a sort of negative energy for the output configuration y in the context of x. Under this interpretation, one could define the conditional distribution of outputs given inputs in the form of a log-linear model: $P(Y = y | X = x) = \frac{1}{Z_x} e^{F_x(y)}$ where Z_x is a partition function ensuring a proper probability normalization. The solution in (1) would then retrieve the maximum-a-posteriori (MAP) binding geometry associated with the protein chain represented by x. Solving the MAP inference problem is usually the most difficult step for many structured-output learning algorithms [25].

3.3 Greedy Inference

The exponential running time required for MAP inference can be sometimes avoided by introducing a generative model so that $\arg\max_y F_x(y)$ can be computed efficiently by dynamic

programming. Models closely related to stochastic regular and context-free grammars have been suggested for this purpose [27]. These approaches work well if the generative model matches or approximates well the domain at hand. Unfortunately, metal binding cannot be even modeled by a context-free grammar (as shown in Fig. 1, the metal-binding graph has crossing edges). While we do not claim that it is impossible to devise a suitable generative model for this task (indeed, this is an interesting direction of research), we argue that handling context sensitiveness is hard.

The core idea of the solution used in this paper is to avoid an underlying generative model of structured outputs and cast the construction of an output structure into a maximum weight problem that can be solved by an efficient greedy algorithm.

In order to allow this algorithmic solution, we introduce the following MBG assumption:

Definition 1 (MBG Property). *Let $\mathcal{C}_x$ and $\mathcal{I}$ be two sets of vertices (associated with candidate ligands and metal ion identifiers, respectively). We say that a bipartite edge set $y \subset \mathcal{C}_x \times \mathcal{I}$ satisfies the metal-binding geometry property if the degree of each vertex in $\mathcal{C}_x$ in the graph $(\mathcal{C}_x \cup \mathcal{I}, y)$ is at most 1.*

Intuitively, this means that *no residue coordinates two different ions in any given chain*. In Nature, there are of course exceptions, the most notable being perhaps due to ASP and GLU, which can coordinate two ions using both oxygen atoms. However, the present study is limited to prediction of sites coordinated by CYS and HIS only. For these two amino acids, cases of dual coordination are rare (in the December 2009 release of PDB, only 0.9 percent HIS and 1.6 percent CYS are found to be within 3 Å of two different ions). As explained below, the MBG property allows us to achieve significant computational benefits at the cost of slightly increased error rate (in our data sets, see Section 5, the additional prediction error rate on edges due to this assumption is only 2 percent).

Definition 2 (Matroid). *A matroid (see, e.g., [30]) is an algebraic structure $\mathcal{M} = (S, \mathcal{Y})$ where S is a finite set and $\mathcal{Y}$ a family of subsets of S such that: 1) $\emptyset \in \mathcal{Y}$; 2) all proper subsets of a set y in $\mathcal{Y}$ are in $\mathcal{Y}$; and 3) if y and y' are in $\mathcal{Y}$ and $|y| < |y'|$, then there exists $e \in y' \setminus y$ such that $y \cup \{e\} \in \mathcal{Y}$.*

Elements of $\mathcal{Y}$ are called *independent sets*. If y is an independent set, then $\mathrm{ext}(y) = \{e \in S : y \cup \{e\} \in \mathcal{Y}\}$ is called the extension set of y. A maximal (having an empty extension set) independent set is called a *base*. In a *weighted* matroid, a local weight function $v : S \mapsto \mathbb{R}^+$ assigns a positive number $v(e)$ to each element $e \in S$. The weight function allows us to compare two structures in the following sense. A set $y = \{e_1, \ldots, e_n\}$ is lexicographically greater than set y' if its monotonically decreasing sequence of weights $(v(e_1), \ldots, v(e_n))$ is lexicographically greater than the corresponding sequence for y'. The following classic result (see, e.g., [30]) is the underlying support for many greedy algorithms:

Theorem 3 (Rado 1957; Edmonds 1971). *For any nonnegative weighting over S, a lexicographically maximum base in $\mathcal{Y}$ maximizes the global objective function $F(y) = \sum_{e \in y} v(e)$.*

Weighted matroids can be seen as a discrete counterpart of concave functions: thanks to the above theorem, if $\mathcal{M}$ is a weighted matroid, then the following greedy algorithm is guaranteed to find the optimal structure, i.e., $\arg\max_{y \in \mathcal{Y}} F(y)$:

GREEDYCONSTRUCT$(\mathcal{M}, F)$
$y = \emptyset$
While $\mathrm{ext}(y) \neq \emptyset$
  $y = y \cup \{\arg\max_{e \in \mathrm{ext}(y)} F(y \cup \{e\})\}$
Return y

This theory shows that if the structured-output space being searched satisfies the property of a matroid, learning structured outputs may be cast into the problem of learning the objective function F for the greedy algorithm. When following this strategy, however, we may perceive the additive form of F as a strong limitation as it would prescribe to predict $v(e)$ independently for each part $e \in S$, while the whole point of structured-output learning is to end up with a *collective* decision about which parts should be present in the output structure. But interestingly, the additive form of the objective function as in Theorem 3 is not a necessary condition for the greedy optimality of matroids. In fact, Helman et al. [31] show that the classic theory can be generalized to so-called *consistent* objective functions, i.e., functions that satisfy the following additional constraints:

$$F(y \cup \{e\}) \geq F(y \cup \{e'\}) \Rightarrow F(y' \cup \{e\}) \geq F(y' \cup \{e'\}) \quad (2)$$

for any $y \subset y' \subset S$ and $e, e' \in S \setminus y'$.

Theorem 4 (Helman et al. 1993). *If F is a consistent objective function then, for each matroid on S, all greedy bases are optimal.*

Note that the sufficient condition of Theorem 4 is also necessary for a slightly more general class of algebraic structures that include matroids, called *matroid embeddings* [31]. We now show that the MBG problem is a suitable candidate for a greedy algorithmic solution.

Theorem 5. *If each $y \in \mathcal{Y}_x$ satisfies the MBG property, then $\mathcal{M}_x = (S_x, \mathcal{Y}_x)$ is a matroid.*

Proof. Suppose $y' \in \mathcal{Y}_x$ and $y \subseteq y'$. Removing an edge from y' cannot increase the degree of any vertex in the bipartite graph, so $y \in \mathcal{Y}_x$. Also, suppose $y \in \mathcal{Y}_x$, $y' \in \mathcal{Y}_x$, and $|y| < |y'|$. Then, there must be at least one vertex t in x having no incident edges in y and such that $(\iota, t) \in y'$ for some $\iota \in \mathcal{I}$. Therefore, $y \cup \{(\iota, t)\}$ also satisfies the MBG property and belongs to $\mathcal{Y}_x$, showing that $\mathcal{M}_x$ is a matroid. □

We can finally formulate the greedy algorithm for constructing the structured output in the MBG problem. Given the input x, we begin by forming the associated MBG matroid M_x and a corresponding objective function $F_x : \mathcal{Y}_x \mapsto \mathbb{R}^+$ (in the next section, we will show how to learn the objective function from data). The output structure associated with x is then computed as

$$f(x) = \mathrm{GREEDYCONSTRUCT}(\mathcal{M}_x, F_x). \quad (3)$$

The following result immediately follows from Definition 1 and Theorem 4:

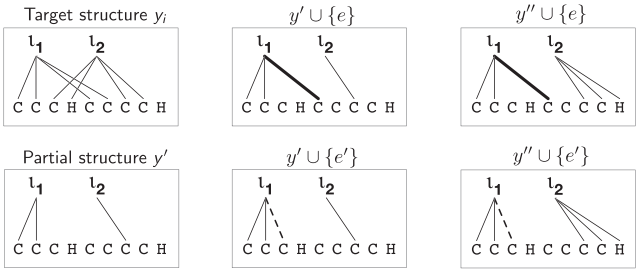


Fig. 2. Illustration of constraints in (5-6). Left: target structure (supervision) y_i and a correct partial structure y' (included in y_i). Middle: Illustration of (5). Adding a correct edge e (thick) should increase the score more than adding an incorrect edge e' (dashed). Equation (5) enforces this constraint with a "large margin" requirement. Right: Illustration of the consistency constraint of (6). If adding the e to y' improves F more than adding e' to y', then adding e to y'' (which strictly includes y') must also improve F more than adding e' to y''. This should be true for all valid edge sets $y' \subset y''$, not only for the substructures of the target.

Corollary 6. *Let (x, y) be an MBG instance. If F_x is a consistent objective function and $F_x(y' \cup \{e\}) > F_x(y' \cup \{e'\})$ for each $y' \subset y$, $e \in \text{ext}(y') \cap y$ and $e' \in \text{ext}(y') \setminus y$, then GREEDYCONSTRUCT$((S_x, \mathcal{Y}_x), F_x)$ returns y.*

3.4 Learning the Greedy Objective Function

A data set for the MBG problem consists of pairs $\mathcal{D} = \{(x_i, y_i); i = 1, \ldots, N\}$ where x_i is a protein sequence and y_i a bipartite graph. The matroid theory outlined above directly suggests the kind of constraints that the objective function needs to satisfy in order to minimize the empirical error of the structured-output problem. For any input string x and (partial) output structure $y \in \mathcal{Y}$, let $F_x(y) = w^T \phi_x(y)$, where w is a weight vector and $\phi_x(y)$ a feature vector for (x, y). The corresponding max-margin formulation is given in

$$\min \frac{1}{2} \|w\|^2, \tag{4}$$

subject to:

$$w^T(\phi_{x_i}(y' \cup \{e\}) - \phi_{x_i}(y' \cup \{e'\})) \geq 1, \tag{5}$$

$$\begin{aligned} &w^T(\phi_{x_i}(y'' \cup \{e\}) - \phi_{x_i}(y'' \cup \{e'\})) \geq 1, \\ &\forall i = 1, \ldots, N,\ \forall y' \subset y_i,\ \forall e \in \text{ext}(y') \cap y_i, \\ &\forall e' \in \text{ext}(y') \setminus y_i,\ \forall y'' : y' \subset y'' \subset S_{x_i}, \end{aligned} \tag{6}$$

where $S_x = \mathcal{C}_x \cup \mathcal{I}$ is the set of possible edges between candidate residues in x and ion identifiers. Intuitively, the first set of constraints (5) ensures that "correct" extensions (i.e., edges that actually belong to the target output structure y_i) receive a higher weight than "wrong" extensions (i.e., edges that do not belong to the target output structure). The purpose of the second set of constraints (6) is to force the learned objective function to obey the consistency property (2). These constraints are illustrated in Fig. 2.

As with classic support vector machines, a regularized variant with soft constraints can be formulated by introducing positive slack variables (one for each constraint) and adding their 1-norm times a regularization coefficient C to (4). The number of resulting constraints in the above formulation grows exponentially with the number of edges in each example; hence, naively solving problem (4-6) is practically unfeasible.

3.5 Solving the Optimization Problem

Our approach seeks an approximate solution to problem (4-5) by leveraging the efficiency of the greedy algorithm also *during* learning. For this purpose, we will use an online active learner that samples constraints chosen by the execution of the greedy construction algorithm.

For each epoch, the algorithm maintains the current highest scoring partial correct output $y'_i \subseteq y_i$ for each example, initialized with the empty MBG structure, where the score is computed by the current objective function F. While there are "unprocessed" examples in $\mathcal{D}$, the algorithm picks a random one and its current best MBG structure y'. If there are no more correct extensions of y', then $y' = y_i$ and the example is removed from $\mathcal{D}$. Otherwise, the algorithm evaluates each correct extension of y', updates the current best MBG structure, and invokes the online learner by calling ADD, which adds a constraint derived from a random incorrect extension (see (5). It also performs a predefined number L of lookaheads by picking a random superset of y'' which is included in the target y_i. The epoch terminates when all examples are processed. In practice, we found that a single epoch over the data set is sufficient for convergence. Pseudocode for one epoch is given below, where subroutine ADD adds an individual constraint.

Given that only a subset of the consistency constraints is sampled, the learning algorithm is no more guaranteed to find a consistent scoring function. In order to compensate for this approximation, a beam search can be introduced in the greedy procedure. Section 5.5 will show that this modification slightly improves the performance of the predictor.

There are several suitable online learners implementing the interface required by the above procedure. Possible candidates include perceptron-like or ALMA-like update rules like those proposed in [32] for structured-output learning. An alternative online learner is the LaSVM algorithm [33] equipped with obvious modifications for handling constraints between pairs of examples. LaSVM is an SMO-like solver for the *dual* version of problem (4-6) that optimizes one or two coordinates at a time, alternating *process* (on newly acquired examples, generated in our case by the ADD procedure) and *reprocess* (on previously seen support vectors or patterns) steps. The ability to work efficiently in the dual is the most appealing feature of LaSVM in the present context and advantageous with respect to perceptron-like approaches. Our unsuccessful preliminary experiments with simple feature vectors confirmed the necessity of flexible design choices for developing rich feature spaces. Kernel methods are clearly more attractive in this case, as detailed in the following.

3.6 The Metal-Binding Kernel

Generalizing the standard case of kernel methods for scalar outputs, the objective function F can be rewritten using a kernel $k(z, z') = \langle \phi_x(y), \phi_{x'}(y') \rangle$ between two structured instances $z = (x, y)$ and $z' = (x', y')$, so that $F_x(y) = F(z) = \sum_j \alpha_j k(z, z_j)$. Here, α_j are the parameters of the model and

GREEDY-EPOCH($\mathcal{D}, L$)
for $i = 1, \ldots, |\mathcal{D}|$
  $y'_i = \emptyset$
while $\mathcal{D} \neq \emptyset$
  pick a random example $(x_i, y_i) \in \mathcal{D}$
  $y' = y'_i$
  $y'_i = \emptyset$
  if $\mathrm{ext}(y') \cap y_i = \emptyset$
    $\mathcal{D} = \mathcal{D} \setminus (x_i, y_i)$
  else
    for each $e \in \mathrm{ext}(y') \cap y_i$
      if $F_{x_i}(y'_i) < F_{x_i}(y' \cup \{e\})$
        $y'_i = y' \cup \{e\}$
        ADD-CONSTRAINTS(x_i, y'_i, y_i, y', e)

ADD-CONSTRAINTS(x_i, y'_i, y_i, y', e)
pick randomly $e' \in \mathrm{ext}(y') \setminus y_i$
ADD($F_{x_i}(y' \cup \{e\}) - F_{x_i}(y' \cup \{e'\}) \geq 1$)
for $l = 1, \ldots, L$
  pick randomly $y'' : y' \subset y'' \subset y_i \wedge e, e' \in S_x \setminus y''$
  ADD($F_{x_i}(y'' \cup \{e\}) - F_{x_i}(y'' \cup \{e'\}) \geq 1$)

Fig. 3. Pseudocode for one epoch of the greedy search algorithm.

z_j are the input-output pairs contained in the constraints added during the learning stage (see subroutine ADD in Fig. 3). The kernel function k can be seen as a similarity measure between candidate MBG structures (belonging to the same or different chains), implicitly encoding them via the joint feature vector $\phi_x(y)$.

In designing a kernel for candidate MBG structures, one has to keep in mind that structures at quite different stages of refinement will need to be compared. These range from a simple initial structure consisting of a single edge up to a complete assignment of candidate ligands. A preliminary investigation showed that imposing hard constraints on compatibility between (partial) structures, as detailed in the following, dramatically improved the quality of the results.

Let $\sigma_i(z)$ denote the set of edges incident to an ion identifier ι_i and $n(z)$ the number of ion identifiers that have at least one incident edge. A top-down definition of the designed kernel is given in

$$k(z, z') = k_{\mathrm{glob}}(z, z') \sum_{i=1}^{n(z)} \sum_{j=1}^{n(z')} \frac{k_{\mathrm{mbs}}(\sigma_i(z), \sigma_j(z'))}{n(z)n(z')}, \quad (7)$$

$$k_{\mathrm{glob}}(z, z') = \delta(n(z), n(z'))k_{\min}(|x|, |x'|)k_{\mathrm{trans}}(z, z'), \quad (8)$$

$$k_{\mathrm{mbs}}(\sigma_i(z), \sigma_j(z')) = \delta(|\sigma_i(z)|, |\sigma_j(z')|) \sum_{\ell=1}^{|\sigma_i(z)|} k_{\mathrm{res}}(x_i(\ell), x'_j(\ell)), \quad (9)$$

where $\delta(a, b) = 1$ if $a = b$ and zero otherwise, $x_i(\ell)$ denotes the ℓth residue in $\sigma_i(z)$, taken in increasing order of sequential position in the protein, and $k_{\mathrm{res}}(x_i(\ell), x'_j(\ell))$ is simply the dot product between the feature vectors describing residues $x_i(\ell)$ and $x'_j(\ell)$ (details on these features are given in Section 5). k_{mbs} measures the similarity between individual sites (two sites are orthogonal if they have a different number of ligands, a choice that is supported by protein functional considerations). k_{glob} ensures that two structures are orthogonal unless they have the same number of sites and downweights their similarity when their number of candidate ligands differs ($k_{\min}(a, b) = 2\min(a, b)/(a + b)$ is a normalized minimum kernel[1]). k_{trans} is a *transition* kernel measuring the similarity between patterns of binding in terms of transitions between different ions along the protein sequence. The geometry of metal-binding sites can be encoded into a string s by restricting to metal-binding residues and representing each residue with the identifier of the ion it binds. We call a *transition* in such an encoding the case in which a pair of contiguous residues has different identifiers (i.e., bind different ions). We then let t be a string of indicator variables, $t_i = 1$ if a transition occurs at position i. Fig. 4 shows some examples of string representations for known metal-binding geometries. For 1a1tA (top), the two metal-binding sites are nonoverlapping in sequence ($s = 11112222$), and the only transition occurs between the last ligand of ion ι_1 and the first of ion ι_2, i.e., $t = 0001000$. In the case of 1chc (bottom), the two sites overlap ($s = 11221122$) and there are multiple transitions: between the second ligand of ι_1 and the first of ι_2, the second of ι_2 and the first of ι_1, the last of ι_1 and the third of ι_2, i.e., $t = 0101010$. Note that the encoding is a simplified version of the move-to-front transformation [36] widely employed as a component of compression algorithms. Most metal-binding geometries in our data set tend to have a quite regular behavior in terms of number and patterns of transitions. We developed a transition kernel aimed at modeling these regularities:

$$k_{\mathrm{trans}}(t, t') = k_{\min}(L(t), L(t'))k_{\mathrm{kgram}}(t, t'),$$

where $L(t) = |\{i : t_i = 1\}|$. The kernel is the product of two parts: a transition number kernel downweighting the similarity of geometries having a different number of transitions, and a transition pattern kernel, which is a k-gram string kernel [37] on the transition string encoding. We used $\mathrm{k} = 2$ in our experiments.

Moreover, it is well known [11] that metal-binding sites often follow regular patterns and show regularities in their amino acid composition (for example, motif CxxCH is very

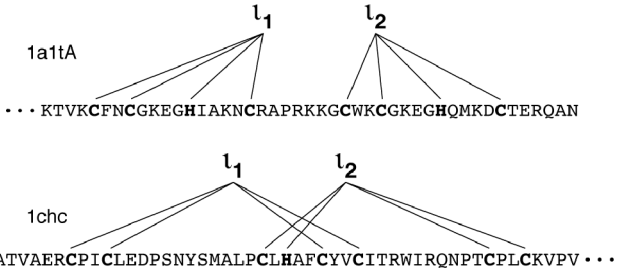


Fig. 4. Two examples of string encoding of metal-binding geometries. In the case of 1a1tA (top) $s = 11112222$ and $t = 0001000$; in the case of 1chc (bottom) $s = 11221122$ and $t = 0101010$.

1. See, e.g., [34], pag. 318 for a proof that $\min(a, b)$ is a valid kernel. The normalized version we employ here corresponds to the similarity coefficient discussed in [35]. See the case of all dichotomous variates in its appendix for the proof of positive semidefiniteness.

frequent in heme-binding conformations). For this reason, additional features were employed to describe the metal-binding sites:

- A vector of Booleans $b = b_1, \ldots, b_k$ where b_j indicates whether the metal-binding site conformation matches a particular motif m_j, taken from a list $m = m_1, \ldots, m_k$ of frequent motifs. These features are used to compute an additional term for the global kernel defined in (8): $k_{\text{motifs}}(z, z') = \sum_{j=0}^{k}(m_j(z)m_j(z'))$.
- The sequence of amino acids composing the site (e.g., CCH, CCCC, ...), as well as the percentage of CYS/HIS in the site. These features are used within k_{mbs} defined in (9), to add a second term: $k_{\text{comp}}(\sigma_i(z), \sigma_j(z'))$.

In the cases in which the bonding state of residues is known (see Section 5.3), this information can be used to produce a better similarity measure between geometries. The overall kernel is thus multiplied by $k_{\min}(|mbr(x)|, |mbr(x')|)$ where $mbr(x)$ is the set of true ligands in sequence x. The kernel downweights the similarity of sequences having a different number of ligands.

3.7 Bonding State Prediction

In principle, the structured-output learning approach described so far can be easily extended to predict metal-bonding state of CYS and HIS as a byproduct (just add a dummy identifier ι_0 to $\mathcal{I}$ so that free residues are linked to ι_0 in the output structure). However, predicting the metal-binding geometry is considerably harder than predicting bonding state and employing an ad hoc bonding state predictor cascaded to the geometry predictor gives better accuracy (see Section 5).

Support vector machines have been employed in the past for predicting the bonding state of CYS and HIS [13], [38]. One problem with a straightforward application of SVM is that the bonding state of different residues are predicted independently, while the corresponding random variables are in fact correlated. In [13], [38], the authors used a bidirectional recurrent neural network and Viterbi decoding with a simple probabilistic automaton to refine local predictions. This allowed to obtain a collective bonding state assignment for all CYS and HIS in a given chain, starting from the margins predicted by a binary classification SVM trained on individual residues. In the experiments reported in Section 5, we use SVM-HMM [27] for the metal-bonding state prediction. SVM-HMM is a structured-output learning algorithm that in this context receives as input a sequence of CYS and HIS residues (where each residue is represented by a vector of features x_t) and outputs a sequence of binary labels corresponding to the bonding state of all residues. The approach is genuinely collective and takes into account correlations between bonding states of different residues in the same chain (but chains are assumed to be mutually independent). Compared to the approach in MetalDetector [38], SVM-HMM offers a simplified strategy (MetalDetector is a combination of several models, is designed to predict also disulfide-bound CYS, and requires a more sophisticated selection of training examples), faster training time, and no significant penalty in terms of prediction accuracy (see Section 5).

The feature vector x_t used by SVM-HMM to predict metal-bonding state consists of both residue and sequence features, which are described in the following.

3.7.1 Residue Features

The set of residue features employed by SVM-HMM consists of: 1) a profile window of 15 amino acids centered around the target residue, with profile scores discretized into 50 bins and represented with the unary code[2] to simulate the min-kernel: profiles were generated for each sequence by running one iteration of PSI-BLAST on the nonredundant (nr) NCBI data set, with an e-value cutoff of 0.005; and 2) distance separation discretized in nine bins as in [13], with respect to previous CYS/HIS in sequence.

3.7.2 Sequence Features

The set of global sequence features consists of:

1. a global descriptor of the sequence encoding the amino-acidic composition, each entry being computed as $\log(\frac{N_j^i}{N_j})$, where N_j^i is the number of occurrences of the jth amino acid in the ith chain, while N_j is the number of occurrences of the jth amino acid in the whole training set;
2. the ratio between the length of the sequence and the average length in the data set;
3. the relative number of CYS/HIS with respect to the sequence length;
4. the relative number of CYS/HIS with respect to the average number of CYS/HIS in the data set; and
5. a parity bit for the number of CYS.

4 Data Preparation

We performed experiments on two distinct data sets, obtained using different criteria of redundance elimination to select the chains. The first data set was built using sequence similarity as the removal criterion: starting from the data set in [13], where chains were selected using UniqueProt [39] to remove redundancy, we discarded all chains having metal-binding sites with residues different from CYS and HIS, as well as few very rare cases of chains with metal-binding sites with coordination greater than 4. The final data set consisted of 199 metal-binding chains, containing 1,235 HIS and 1,147 CYS. 63 percent of the resulting chains were bonded to a zinc (Zn) ion, 8 percent to heme (HEM/HEC) groups, 3 percent to cadmium (Cd), 8 percent to iron (Fe), 19 percent to iron-sulfur groups (FS4,SF4,Fe/S), and 7 percent to copper (Cu).

The second data set was built using a more stringent criterion to remove redundancy, by taking into account the Structural Classification of Proteins hierarchy [40]: in this case, in fact, we aim at measuring the ability of the predictor in identifying metal-binding sites within proteins belonging to SCOP superfamilies—or folds—which are not observed in the training set. First, we extracted from the December 2009 release of PDB 17,783 protein chains with at least a CYS or HIS bonded to a metal ion.[3] We detected ligands

2. The unary code for the discretized score s is a vector of 50 bits, where the first s are set to 1, and the remaining to 0.

3. We considered the same transition metals used in [13].

TABLE 2
Results on the UniqueProt-Based Data Set (Averaged on 30 Train/Test Random Splits)

	Bonding State Predictor		Bonding State + Geometry Predictor			
N	P_B	R_B	P_E	R_E	H_T	H_F
199	79±4	88±4	68±4	74±4	93±4	10±3

TABLE 3
Results of Leave-K-Superfamilies-Out on the First SCOP-Based Data Set (Including Multidomain Chains)

		Bonding State Predictor		Bonding State + Geometry Predictor			
m^*	N	P_B	R_B	P_E	R_E	H_T	H_F
1	1281	64±4	65±8	63±5	62±8	71±8	24±5
2	413	67±13	64±16	55±14	47±8	70±14	13±7
≥3	130	78±12	50±24	62±16	35±18	55±24	9±10
all	1824	62±5	71±10	61±6	57±7	70±9	19±4

using a cutoff of 3 Å on the distance between the metal ion (or complex) and the sulfur or nitrogen atoms for cysteines and histidines, respectively. We then discarded 6,090 entries not mapped in the 1.75 release (June 2009) of the SCOP database. We also removed very few cases in which the number of metal-binding sites was greater than 5. Finally, we obtained a sequence-unique subset of 1,824 protein chains by running CD-HIT v4.0 [41] with sequence identity threshold set to 0.9 (default value). The data set contained 12,323 HIS and 8,290 CYS. 54 percent of the resulting chains were bonded to zinc, 14 percent to heme groups, 7 percent to cadmium, 7 percent to iron, 7 percent to iron-sulfur groups, and 5 percent to copper. Following the procedure described above, we found 122 CYS and 12 HIS coordinating multiple ions. In these cases, we kept in the data set only the closest ligand-ion pair.

The data sets, together to the splits employed in the validation procedures discussed further on, are available in the Supplementary Material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TCBB.2011.94.

5 Results and Discussion

For all the experiments, we report several performance measures:

- P_B and R_B indicate precision and recall for the residue bonding state, computed as $P_B = \frac{TP}{TP+FP}$ and $R_B = \frac{TP}{TP+FN}$ where TP is the number of correctly identified metal-bonded residues, FP the number of false positives (free residues wrongly predicted as metal bonded), and FN the number of false negatives (metal-bonded residues predicted as free).
- P_E and R_E are precision and recall for the correct assignment between a residue and the metal ion identifier: P_E is the number of correctly predicted coordinations, relative to the total number of predicted coordinations, while R_E is the fraction of correctly identified coordinations over the number of actual coordinations.
- H_T and H_F are true-positive and false-positive hit rates, where a hit is counted whenever the intersection between the predicted and the actual site is nonempty: H_T is therefore the percentage of sites having at least one correctly identified ligand, and H_F is the fraction of predicted sites having no correctly identified residues.

In our preliminary experiments on bonding state prediction, we also compared SVM-HMM against MetalDetector [38] on the same data set of 2,727 chains (365 metalloproteins) used in [13]. Precision and recall of MetalDetector were 73.5 and 61.6, while SVM-HMM achieves 74.3 and 59.2. We deemed these differences too small to justify the inclusion of a much more complex system such as MetalDetector in the current predictor. Moreover, on that data set, SVM-HMM correctly predicts as nonmetalloproteins 96 percent of the 2,362 chains having no metal-bonded CYS/HIS.

A lookahead $L = 10$ and a beam search width $b = 2$ were employed in all the reported experiments. See Section 5.5 for a discussion on the impact of these search parameters on the prediction accuracy.

5.1 UniqueProt-Based Data Set

As a first set of experiments on the UniqueProt-based data set, we run 30 different train/test random splits, always in a ratio of 80/20. Table 2 shows the results obtained by our predictor.

On this data set, we also tested a different architecture (a setting previously adopted in [28]), where the bonding state assignment was jointly performed with the prediction of metal-binding site geometry, using the dummy-ion trick described in Section 3.7. At the same level of precision, we observed an improvement of 20 points in bonding state recall, using the two-stage architecture; similarly, precision/recall on edges improves from 63/52 to 68/74.

5.2 SCOP-Based Data Set

When using the SCOP-based data set, we employed a different strategy to perform the experiments: in this case, the goal is to measure the performance of the predictor on SCOP superfamilies—or folds—which are not observed in the training set. We refer to this procedure as *leave-k-superfamilies-out*, or *leave-k-folds-out*, where *folds* here are intended as SCOP hierarchy folds, and should not be confused with folds of the standard k-fold-cross-validation procedure. We partitioned the data set in $k = 10$ subsets of chains, maintaining the same average percentage of ligands in each subset, and with the additional constraint that no pair of chains in different subsets belonged to the same SCOP superfamily. We also prepared a second version of this data set, where we considered SCOP folds instead of superfamilies: in this case, we discarded multidomain chains, as building the partition would have been otherwise unfeasible. This version of the data set was therefore reduced to 1,466 chains.

Tables 3 and 4 show the results on this second data set, including the breakdown of performance measures for proteins binding different numbers of metal ions. Performance measures are averaged on 10 different splits, according to a leave-k-superfamilies-out (Table 3) or a

TABLE 4
Results of Leave-K-Folds-Out on the Second SCOP-Based Data Set (Single-Domain Chains)

		Bonding State Predictor		Bonding State + Geometry Predictor			
m^*	N	P_B	R_B	P_E	R_E	H_T	H_F
1	1043	61±6	69±11	58±7	64±11	76±11	27±8
2	317	65±7	65±16	53±8	49±11	71±15	17±6
≥3	106	65±16	57±22	49±15	40±15	65±23	17±19
all	1466	60±4	74±10	56±6	60±10	74±10	22±5

TABLE 5
Detailed Results on the Most Frequent Metal Ions, on the Leave-K-Folds-Out (Left) and Leave-K-Superfamilies-Out (Right) Experiments

	leave-k-folds-out					leave-k-superfamilies-out				
metal	N	P_B	R_B	P_E	R_E	N	P_B	R_B	P_E	R_E
Zn	817	63	70	57	62	1,019	64	67	58	59
Heme	234	67	77	62	71	258	68	65	73	60
Fe/SF4	202	68	67	60	56	290	66	68	59	59
Cu	87	57	64	51	58	106	61	57	56	53
Cd	83	64	46	57	39	96	65	40	58	32

N is the number of chains containing at least one ion of that type.

leave-k-folds-out (Table 4) procedure. Results are grouped according to the number m^* of actual metal-binding sites and N is the number of chains in each group.

The leave-k-superfamilies-out and—even more—the leave-k-folds-out procedures ensure an extremely challenging task: different superfamilies/folds often bind different metal ions and show very different metal-binding sites conformations, ruling out homology-based techniques. Experiments on the SCOP-folds data set are on one hand more challenging, since chains in training and test sets are more "distant" within the SCOP hierarchy than in the case of superfamilies; yet, on the other hand, the SCOP-superfamilies data set also contains multidomain chains, which are much more difficult to predict (on multidomain chains, we obtained $P_B = 63$, $R_B = 55$, $P_E = 57$, and $R_E = 48$ to be compared with the first row of Table 3).

In Table 5, results on the most frequent ions in the data set are also detailed, both in the case of the SCOP-superfamilies data set, and in the case of the SCOP-folds data set. Heme is the group where the predictor performance is the highest, followed by Zn and Fe/SF4. All these cases tend to have quite regular binding patterns. For heme, an additional advantage is the fact that most heme-bonded chains are single domain (234 out of 258).

5.3 Results with Known Bonding State

In order to assess the accuracy of the second stage alone, we also tested the geometry predictor starting from perfect knowledge of bonding state (rather than using the first stage). For the UniqueProt-based data set, we obtained $P_E = R_E = 90 \pm 3$; for the SCOP-based data set, we obtained $P_E = R_E = 89 \pm 3$ in the leave-k-superfamilies-out setting and $P_E = R_E = 75 \pm 6$ in the leave-k-folds-out setting.

5.4 Predicting the Number of Metal Ions

As a byproduct of the predicted geometry, we can obtain a prediction m of the number m^* of metal ions bound to the input chain. In the leave-k-folds-out setting, the prediction accuracy, counting a correct prediction if $m = m^*$, for $m, m^* = 0, \ldots, 5$ was 57.0 percent. When counting a correct prediction if $|m - m^*| \leq 1$, the accuracy was 92.4 percent.

5.5 Effects of Kernels and Search Parameters

Additional experiments were run in order to observe the impact of the search algorithm parameters and the kernel employed on the accuracy of the predictions. With no lookahead, in the known bonding state setting, the precision/recall on edges decreased by 1 percent, while the training time was reduced by a factor of 6.3. Removing the beam search had the effect of losing another 0.5 percent in precision/recall, with the training time reduced by a half. In the kernel used in [28], neither the transition kernel, nor motifs/residue features were employed: in those conditions, the precision/recall on edges in the known bonding state setting decreased by 2 percent, other search parameters being equal (no beam search was used in [28]).

6 Conclusions

Prediction of metal-binding geometry was never attempted before starting from protein sequence alone. The approach presented in this paper is based on structured-output learning, with a novel inference algorithm that exploits a reasonable assumption (ligands coordinate a single metal ion) to achieve computational efficiency using a greedy algorithm. As expected, results strongly depend on the nonredundancy criteria used to define the learning task: generalization across SCOP folds is far more difficult than that across sequentially distant proteins. A careful design of the discriminant features, encoded in the kernel between candidate geometries, is crucial to the quality of results. We expect that further research in this direction can provide a major contribution in tackling this extremely challenging learning task.

Our results assuming known bonding state show that this knowledge allows to substantially improve the prediction of metal-binding sites. The overall quality of the predictions can significantly benefit from advancements in predicting bonding state. From this viewpoint, relational discriminative learning techniques, which jointly provide labeling for a set of entities exploiting their relations, are a promising direction for automatic annotation of protein sequences, as our results on SVM-HMM seem to suggest.

Finally, we currently ignored the protein quaternary structure in our experiments. A number of metal-binding sites lie at the protein interface, with the ion(s) coordinated by ligands from multiple chains. We leave the handling of these cases as a direction for further research.

References

[1] K. Degtyarenko, "Bioinorganic Motifs: Towards Functional Classification of Metalloproteins," *Bioinformatics*, vol. 16, pp. 851-864, 2000.

[2] I. Bertini, A. Sigel, and H. Sigel, "Handbook on Metalloproteins," *J. Am. Chemical Soc.*, vol. 123, no. 50, p. 12748, http://pubs.acs.org/doi/abs/10.1021/ja015322x, 2001.

[3] J. Muller, "Functional Metal Ions in Nucleic Acids," *Metallomics*, vol. 2, no. 5, pp. 318-327, http://dx.doi.org/10.1039/c000429d, 2010.

[4] D.E. Draper, "A Guide to Ions and RNA Structure," *RNA,* vol. 10, no. 3, pp. 335-343, http://www.biomedsearch.com/nih/guide-to-ions-RNA-structure/14970378.html, 2004.

[5] E. Freisinger and R.K. Sigel, "From Nucleotides to Ribozymes-A Comparison of Their Metal Ion Binding Properties," *Coordination Chemistry Rev.,* vol. 251, nos. 13/14, pp. 1834-1851, http://www.sciencedirect.com/science/article/B6TFW-4N9DK1X-1/2/6566239afcd0cc18464374985dba075a, 2007.

[6] K.J. Barnham and A.I. Bush, "Metals in Alzheimer's and Parkinson's Diseases," *Current Opinion in Chemical Biology,* vol. 12, no. 2, pp. 222-228, http://www.sciencedirect.com/science/article/B6VRX-4S9R226-2/2/730f58d2a562f3703d907abd8fd3cfa0, 2008.

[7] D. Beyersmann and S. Hechtenberg, "Cadmium, Gene Regulation, and Cellular Signalling in Mammalian Cells," *Toxicology and Applied Pharmacology,* vol. 144, no. 2, pp. 247-261, http://www.sciencedirect.com/science/article/B6WXH-45KN3J7-5/2/dc811bb0 fb069b66bc3a1148f1f5bbae, 1997.

[8] W. Shi, C. Zhan, A. Ignatov, B.A. Manjasetty, N. Marinkovic, M. Sullivan, R. Huang, and M.R. Chance, "Metalloproteomics: High-Throughput Structural and Functional Annotation of Proteins in Structural Genomics," *Structure,* vol. 13, no. 10, pp. 1473-1486, http://www.sciencedirect.com/science/article/B6VSR-4H9GRCF-D/2/61c1815e c8906707b782983bd4ac5f90, 2005.

[9] M.R. Chance and W. Shi, "Metallomics and Metalloproteomics," *Cellular and Molecular Life Sciences,* vol. 65, no. 19, pp. 3040-3048, 2008.

[10] W. Shi, M. Punta, J. Bohon, J.M. Sauder, R. D'Mello, M. Sullivan, J. Toomey, D. Abel, M. Lippi, A. Passerini, P. Frasconi, S.K. Burley, B. Rost, and M.R. Chance, "Characterization of Metalloproteins by High-Throughput X-Ray Absorption Spectroscopy," *Genome Research,* vol. 21, no. 6, pp. 898-907, http://www.ncbi.nlm.nih.gov/pubmed/21482623, Apr. 2011.

[11] C. Andreini, I. Bertini, and A. Rosato, "A Hint to Search for Metalloproteins in Gene Banks," *Bioinformatics,* vol. 20, no. 9, pp. 1373-1380, 2004.

[12] F. Ferrè and P. Clote, "DiANNA 1.1: An Extension of the DiANNA Web Server for Ternary Cysteine Classification," *Nucleic Acids Research,* vol. 34, pp. W182-W185, 2006.

[13] A. Passerini, M. Punta, A. Ceroni, B. Rost, and P. Frasconi, "Identifying Cysteines and Histidines in Transition-Metal-Binding Sites Using Support Vector Machines and Neural Networks," *Proteins,* vol. 65, no. 2, pp. 305-316, 2006.

[14] A. Passerini, C. Andreini, S. Menchetti, A. Rosato, and P. Frasconi, "Predicting Zinc Binding at the Proteome Level," *BMC Bioinformatics,* vol. 8, p. 39, 2007.

[15] N. Shu, T. Zhou, and S. Hovmoller, "Prediction of Zinc-Binding Sites in Proteins from Sequence," *Bioinformatics,* vol. 24, no. 6, pp. 775-782, 2008.

[16] A.J. Bordner, "Predicting Small Ligand Binding Sites in Proteins Using Backbone Structure," *Bioinformatics,* vol. 24, no. 24, pp. 2865-2871, Dec. 2008.

[17] I. Bertini and G. Cavallaro, "Bioinformatics in Bioinorganic Chemistry," *Metallomics,* vol. 2, pp. 39-51, http://dx.doi.org/10.1039/B912156K, 2010.

[18] M. Babor, S. Gerzon, B. Raveh, V. Sobolev, and M. Edelman, "Prediction of Transition Metal-Binding Sites from Apo Protein Structures," *Proteins,* vol. 70, no. 1, pp. 208-217, 2007.

[19] J.C. Ebert and R.B. Altman, "Robust Recognition of Zinc Binding Sites in Proteins," *Protein Science,* vol. 17, no. 1, pp. 54-65, http://www.biomedsearch.com/nih/Robust-recognition-zinc-binding-sites/18042678.html, 2008.

[20] J.A. Capra, R.A. Laskowski, J.M. Thornton, M. Singh, and T.A. Funkhouser, "Predicting Protein Ligand Binding Sites by Combining Evolutionary Sequence Conservation and 3D Structure," *PLoS Computational Biology,* vol. 5, no. 12, p. e1000585, Dec. 2009.

[21] G. Bartlett, C. Porter, N. Borkakoti, and J. Thornton, "Analysis of Catalytic Residues in Enzyme Active Sites," *J. Molecular Biology,* vol. 324, no. 1, pp. 105-121, 2002.

[22] C. Andreini, L. Banci, I. Bertini, and A. Rosato, "Counting the Zinc-Proteins Encoded in the Human Genome," *J. Proteome Research,* vol. 5, no. 1, pp. 196-201, http://pubs.acs.org/doi/abs/10.1021/pr050361j, 2006.

[23] C. Andreini, L. Banci, I. Bertini, and A. Rosato, "Occurrence of Copper Proteins through the Three Domains of Life: A Bioinformatic Approach," *J. Proteome Research,* vol. 7, no. 1, pp. 209-216, http://pubs.acs.org/doi/abs/10.1021/pr070480u, 2008.

[24] N. Hulo, A. Bairoch, V. Bulliard, L. Cerutti, E. De Castro, P.S. Langendijk-Genevaux, M. Pagni, and C.J.A. Sigrist, "The PROSITE Database," *Nucleic Acids Research,* vol. 34, no. suppl. 1, pp. D227-D230, http://nar.oxfordjournals.org/cgi/content/abstract/34/suppl_1/D227, 2006.

[25] G.H. Bakir, T. Hofmann, B. Schölkopf, A.J. Smola, B. Taskar, and S.V.N. Vishwanathan, *Predicting Structured Data.* The MIT Press, 2007.

[26] A. Vullo and P. Frasconi, "Disulfide Connectivity Prediction Using Recursive Neural Networks and Evolutionary Information," *Bioinformatics,* vol. 20, no. 5, pp. 653-659, http://bioinformatics.oxfordjournals.org/cgi/content/abstract/btg463v1, 2004.

[27] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large Margin Methods for Structured and Interdependent Output Variables," *J. Machine Learning Research,* vol. 6, pp. 1453-1484, 2005.

[28] P. Frasconi and A. Passerini, "Predicting the Geometry of Metal Binding Sites from Protein Sequence," *Proc. Neural Information Processing Systems (NIPS),* pp. 465-472, 2008.

[29] B. Taskar, V. Chatalbashev, D. Koller, and C. Guestrin, "Learning Structured Prediction Models: A Large Margin Approach," *Proc. Int'l Conf. Machine Learning (ICML '05),* pp. 896-903, 2005.

[30] E. Lawler, *Combinatorial Optimization: Networks and Matroids.* Holt, Rinehart, and Winston, 1976.

[31] P. Helman, B.M.E. Moret, and H.D. Shapiro, "An Exact Characterization of Greedy Structures," *SIAM J. Discrete Math.,* vol. 6, pp. 274-283, 1993.

[32] H. Daumé III and D. Marcu, "Learning as Search Optimization: Approximate Large Margin Methods for Structured Prediction," *Proc. Int'l Conf. Machine Learning (ICML '05),* pp. 169-176, 2005.

[33] A. Bordes, S. Ertekin, J. Weston, and L. Bottou, "Fast Kernel Classifiers with Online and Active Learning," *J. Machine Learning Research,* vol. 6, pp. 1579-1619, 2005.

[34] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis.* Cambridge Univ. Press, http://www.loc.gov/catdir/toc/cam051/2003069590.html, 2004.

[35] J.C. Gower, "A General Coefficient of Similarity and Some of Its Properties," *Biometrics,* vol. 27, no. 4, pp. 857-871, 1971.

[36] J. Bentley, D. Sleator, R. Tarjan, and V. Wei, "A Locally Adaptive Data Compression Scheme," *Comm. ACM,* vol. 29, no. 4, pp. 320-330, 1986.

[37] C. Leslie, E. Eskin, and W. Noble, "The Spectrum Kernel: A String Kernel for Svm Protein Classification," *Proc. Pacific Symp. Biocomputing,* pp. 564-575, 2002.

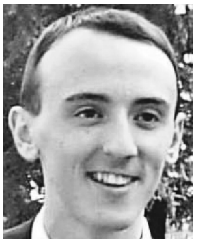
[38] M. Lippi, A. Passerini, M. Punta, B. Rost, and P. Frasconi, "MetalDetector: A Web Server for Predicting Metal-Binding Sites and Disulfide Bridges in Proteins from Sequence," *Bioinformatics,* vol. 24, no. 18, pp. 2094-2095, 2008.

[39] S. Mika and B. Rost, "Uniqueprot: Creating Representative Protein Sequence Sets," *Nucleic Acids Research,* vol. 31, no. 13, pp. 3789-3791, http://www.rostlab.org/papers/2003_narweb_unique/, 2003.

[40] A.G. Murzin, S.E. Brenner, T. Hubbard, and C. Chothia, "Scop: A Structural Classification of Proteins Database for the Investigation of Sequences and Structures," *J. Molecular Biology,* vol. 247, no. 4, pp. 536-540, Apr. 1995.

[41] W. Li and A. Godzik, "Cd-Hit: A Fast Program for Clustering and Comparing Large Sets of Protein or Nucleotide Sequences," *Bioinformatics,* vol. 22, no. 13, pp. 1658-1659, July 2006.

Andrea Passerini graduated in computer science from the University of Florence in 2000 and received the PhD degree from the same university in 2004. He is currently an assistant professor at the Department of Information Engineering and Computer Science of the University of Trento. His main research interests are in the area of machine learning, with a special emphasis on bioinformatics applications. In recent years, he developed techniques aimed at combining statistical and symbolic approaches to learning, via the integration of inductive logic programming and kernel machines. He is also pursuing a deeper integration of machine learning approaches and complex optimization techniques. He coauthored more than 40 scientific publications.



Marco Lippi received the bachelor's degree in computer engineering, the Doctoral degree in computer engineering, and the PhD degree in computer and automation engineering from the University of Florence in 2004, 2006, and 2010, respectively. He is currently a postdoc fellow at the Department of Computer Engineering, University of Siena. His main research fields are machine learning and artificial intelligence, with a focus on statistical relational learning and inductive logic programming. He has specific interests in bioinformatics applications, in particular concerning protein structure prediction and RNA secondary structure prediction.

Paolo Frasconi is a professor of computer science at the University of Florence. His research interests are in the area of machine learning, with particular emphasis on algorithms for structured and relational data, and applications to bioinformatics. He is an associate editor of the *Artificial Intelligence Journal* and an action editor of the *Machine Learning Journal*. He cochaired the AAAI 2010 Special Track on AI and Bioinformatics, the 20th International Conference on Inductive Logic Programming 2010, and the Fifth International Workshop on Mining and Learning with Graphs (2007).