Position Papers of the Federated Conference on
Computer Science and Information Systems pp. 83–87

DOI: 10.15439/2015F391
ACSIS, Vol. 6

Predicting Metal-Binding Sites of Protein Residues

Serkan R. Küçükbay, Hasan Oğul
Department of Computer Engineering,
Başkent University, Ankara, Turkey
Email: skucukbay@gmail.com,
hogul@baskent.edu.tr

Abstract—**Metal ions in protein are critical to the function, structure and stability of protein. For this reason accurate prediction of metal binding sites in protein is very important. Here, we present our study which is performed for predicting metal binding sites for histidines (HIS) and cysteines from protein sequence. Three different methods are applied for this task: Support Vector Machine (SVM), Naive Bayes and Variable-length Markov chain. All these methods use only sequence information to classify a residue as metal binding or not. Several feature sets are employed to evaluate impact on prediction results. We predict metal binding sites for mentioned amino acids at 35% precision and 75% recall with Naive Bayes, at 25% precision and 23% recall with Support Vector Machine and at 0.05% precision and 60% recall with Variable-length Markov chain. We observe significant differences in performance depending on the selected feature set. The results show that Naive Bayes is competitive for metal binding site detection.**

I. Introduction

Protein plays a crucial role in all biological processes. *And* they consist of one or more long chains of amino acid residues. In the frame of this perspective, amino acids are important ligands with nitrogen and oxygen as the donor, constituent of many biological important molecules [1].

It is estimated that approximately half of all *proteins* contain a metal [2]. A significant fraction (about one third) of all known proteins is believed to bind metal ions as cofactors in their native conformation [3]. The biological activities of proteins require these cofactors to assist their daily routines. For this reason, a metal ion in a protein and prediction of its binding point is very important to understand the function of proteins in biological activities. Metal ions in proteins are responsible for multiple tasks. They help stabilizing protein structure [4], induce conformational changes [5–7], and assist protein functions (e.g. electron transfer, nucleophilic catalysis).

There are many related studies about predicting metal binding sites, however, machine learning techniques have been recently applied to predict the metal binding sites of residues.

Predicting metal binding sites by using non-computational methods has some drawbacks. X-ray absorption spectroscopy (HT-XAS) has been recently proved to be capable of identifying metalloproteins with high reliability [8, 9]. However, the specific ligands involved in binding metal ion(s) cannot be identified by these techniques [9]. Motif-based system has also been developed by using regular expressions but since regular expressions can be quite specific, their results have many false negatives. To overcome these drawbacks, many computational learning techniques have been developed to predict metal binding sites. Early approaches can be found in the work of Nakata et al. (1995). In this study, they focused on predicting zinc-finger DNA-binding proteins with a neural network. In this approach, applicable results were generated by a method for certain types of zinc-binding protein in spite of limitation about scarcity of data at that time. Recently-developed approaches for metal-binding sites prediction have mainly focused on CYS only [10], CYS and HIS binding transition metals [3] or CYS, HIS, ASP, and GLU binding zinc ions [12, 13].

In addition, recent studies in predicting metal binding sites indicate that Support Vector Machine is a popular machine learning technique in this area. In many works, Support Vector Machine was employed as a single solution of a problem or it was used with some other techniques to predict metal binding sites. For example, developed architecture consists of two stages. In the first stage of this study, Support Vector Machine was employed for local classification and these outputs were used as inputs for second stage to refine these local predictions [13].

In this study, we employed three different methods to predict metal binding for CYS and HIS by using only sequence information and amino acid composition: Support Vector Machine, Naive Bayes and Variable-length Markov Chain. Obtained results were compared with each other to give information for future works. Furthermore, we used some different feature compositions to train our model and prediction results were compared to give some clues about used features which were valuable for metal binding sites prediction.

This paper is organized as follows: in Section 2, we provide detailed description of materials and methods. Our obtained results are discussed in Section 3. We finally draw some conclusions in Section 4.

II. Materials and Methods

A. Feature Representation

To obtain feature vector for SVM and Naïve Bayes, we found all target residues (CYS and HIS) for each protein sequence. Each target residue refers individual instances for our input vectors. These individual instances were obtained by building a window. This window consisted of flanking residues of target residue and target residue was located in the middle of the window. Each position in the window was represented by the corresponding value of PAM 120 (Table I. values of pam are shown) substitution matrix. By this way, each position of window was represented by 23 finite numbers. In other words, the total size of window equals to (2w-1) X 23 because the target residue value was ignored for input vectors. Also some characteristic values of selected window were obtained by running script version of Protein Composition Server [14] and these values were added to input vectors. Additionally, we used four different types of amino acid compositions. These are Pseudo Amino Acid Composition (**PAAC**), Amphiphilic Pseudo Amino Acid Composition (**APAAC**), Five Factor Score of Amino Acid Composition (**5FSS**) and PhysioChemical Properties (**PC**). By using different types of amino acid compositions, we made four different individual instances for each target residues. Hence, we could compare their contributions to prediction results. Finally, one additional feature was coded for the relative position of the target residue with respect to the sequence length for each individual instance. Input vector generation steps are illustrated in Fig 1.

To obtain features for Variable-length Markov chain, first, we found all target residues for each protein sequence. Then, we built a window which consisted of flanking residues of the target residue and this target residue was located in the middle of the window. Window size was set to 15. After building the window, it was split at its center. After splitting process, four different feature sets were created. This feature sets contained different parts of the window according to the metal binding state of target residue. Content of these feature sets differed from each other. The mentioned content was constructed via the following steps.

If target residue is metal-bonded and flanking amino acids are located at the left side of target residue, one part of the window is considered as a left-side metal-bonded sample.

If target residue is metal-bonded and flanking amino acids are located at the right side of target residue, one part of the window is considered as a right-side metal-bonded sample. If target residue is not metal-bonded and flanking amino acids are located at the right side of target residue, one part of the window is considered as a right-side non-metal bonded sample. If target residue is not metal-bonded and flanking amino acids are located at the left side of target residue, one part of the window is considered as a left-side non-metal bonded sample.

B. Classification

Support Vector Machines: In this study, we applied three different methods for our problem. One of these is Support Vector Machine (SVM).

SVM learning has been applied to many prediction problems in bioinformatics. Furthermore, obtained prediction results are acceptable for many problems in this search area. For this reason, we chose SVM for our solution. For classification, publicly-available at github libsvm package was used to implement SVM. The radial basis kernel was selected for this process (gamma = 0.05, cost = 0.1).

In this process, SVM was used to train feature vectors of

TABLE I.
PAM 120 MATRIX

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	3	-3	-1	0	-3	-1	0	1	-3	-1	-3	-2	-2	-4	1	1	1	-7	-4	0	0	-1	-1	-8
R	-3	6	-1	-3	-4	1	-3	-4	1	-2	-4	2	-1	-5	-1	-1	-2	1	-5	-3	-2	-1	-2	-8
N	-1	-1	4	2	-5	0	1	0	2	-2	-4	1	-3	-4	-2	1	0	-4	-2	-3	3	0	-1	-8
D	0	-3	2	5	-7	1	3	0	0	-3	-5	-1	-4	-7	-3	0	-1	-8	-5	-3	4	3	-2	-8
C	-3	-4	-5	-7	9	-7	-7	-4	-4	-3	-7	-7	-6	-6	-4	0	-3	-8	-1	-3	-6	-7	-4	-8
Q	-1	1	0	1	-7	6	2	-3	3	-3	-2	0	-1	-6	0	-2	-2	-6	-5	-3	0	4	-1	-8
E	0	-3	1	3	-7	2	5	-1	-1	-3	-4	-1	-3	-7	-2	-1	-2	-8	-5	-3	3	4	-1	-8
G	1	-4	0	0	-4	-3	-1	5	-4	-4	-5	-3	-4	-5	-2	1	-1	-8	-6	-2	0	-2	-2	-8
H	-3	1	2	0	-4	3	-1	-4	7	-4	-3	-2	-4	-3	-1	-2	-3	-3	-1	-3	1	1	-2	-8
I	-1	-2	-2	-3	-3	-3	-3	-4	-4	6	1	-3	1	0	-3	-2	0	-6	-2	3	-3	-3	-1	-8
L	-3	-4	-4	-5	-7	-2	-4	-5	-3	1	5	-4	3	0	-3	-4	-3	-3	-2	1	-4	-3	-2	-8
K	-2	2	1	-1	-7	0	-1	-3	-2	-3	-4	5	0	-7	-2	-1	-1	-5	-5	-4	0	-1	-2	-8
M	-2	-1	-3	-4	-6	-1	-3	-4	-4	1	3	0	8	-1	-3	-2	-1	-6	-4	1	-4	-2	-2	-8
F	-4	-5	-4	-7	-6	-6	-7	-5	-3	0	0	-7	-1	8	-5	-3	-4	-1	4	-3	-5	-6	-3	-8
P	1	-1	-2	-3	-4	0	-2	-2	-1	-3	-3	-2	-3	-5	6	1	-1	-7	-6	-2	-2	-1	-2	-8
S	1	-1	1	0	0	-2	-1	1	-2	-2	-4	-1	-2	-3	1	3	2	-2	-3	-2	0	-1	-1	-8
T	1	-2	0	-1	-3	-2	-2	-1	-3	0	-3	-1	-1	-4	-1	2	4	-6	-3	0	0	-2	-1	-8
W	-7	1	-4	-8	-8	-6	-8	-8	-3	-6	-3	-5	-6	-1	-7	-2	-6	12	-2	-8	-6	-7	-5	-8
Y	-4	-5	-2	-5	-1	-5	-5	-6	-1	-2	-2	-5	-4	4	-6	-3	-3	-2	8	-3	-3	-5	-3	-8
V	0	-3	-3	-3	-3	-3	-3	-2	-3	3	1	-4	1	-3	-2	-2	0	-8	-3	5	-3	-3	-1	-8
B	0	-2	3	4	-6	0	3	0	1	-3	-4	0	-4	-5	-2	0	0	-6	-3	-3	4	2	-1	-8
Z	-1	-1	0	3	-7	4	4	-2	1	-3	-3	-1	-2	-6	-1	-1	-2	-7	-5	-3	2	4	-1	-8
X	-1	-2	-1	-2	-4	-1	-1	-2	-2	-1	-2	-2	-2	-3	-2	-1	-1	-5	-3	-1	-1	-1	-2	-8
*	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	1

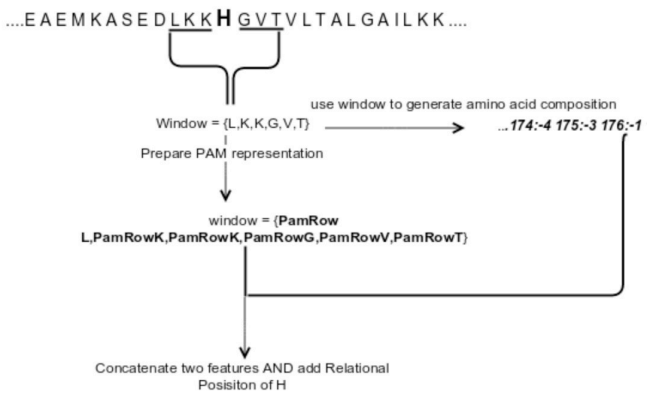

Fig.1 Input vector generation steps

the training data set and to make the prediction on the test data set. Detailed information about data set is mentioned in Section 2.3.

Naïve Bayes: As mentioned above, we applied different methods on our test sets and another one of these methods is Naive Bayes. The Naïve Bayes classifier has proved to be very effective in many real data applications [17]. Naive Bayes classifiers are of the family of simple probabilistic classifiers. It is based on applying Bayes' theorem with strong independence assumptions between the features.

$$p(x = v|c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{\frac{-(v-\mu_c)^2}{2\sigma_c^2}} \quad (1)$$

For classification, the publicly-available MATLAB Naive Bayes package was used to train and predict our data set. We chose Gaussian Naïve Bayes because our data set consisted of continuous data. Applied kernel function is shown in Equation 1.

C. A Generative Approach: Variable Length Markov Chain

Markov chains are used to model sequential data in terms of the order of individual letters. In zero-order Markov Chain, the likelihood of a sequence S_1^N is given by the probability that is obtained by multiplying the probabilities of each symbols contained, i.e.,

$$P(S_1^N) = \prod_{j=1}^{N} P(S_j = s_j) \quad (2)$$

where P(.) refers to probability, S_j is the random variable representing the letter position j with s_j as its realization.

A more flexible version of higher order Markov models allows a variable length that depends on the preceding subsequence to given position such that the order of the model becomes a function the context at each position. This model is called as Variable Length Markov Chain (VLMC) built on the sequence likelihood defined as:

$$P(S_1^N) = \prod_{j=1}^{N} P\left(S_j = s_j \mid S_{j-L_j}^{j-1} = s_{j-L_j}^{j-1}\right) \quad (3)$$

where L_j is the optimal length preceding subsequences respectively and $S_{j-L_j}^{j-1}$ is that sub-sequences.

An efficient implementation of VLMC can be realized using Probabilistic Suffix Trees (PST). The PST method was introduced by Bejerano and Yona to model the protein families [15]. The original PST model was based on identifying significant short segments among the many input sequences, regardless of the relative position of these segments within the different proteins [16]. In this study, to classify a sequence into one of the families, a separate PST is constructed for each family in the data set, and according to the probability distribution over PST, a probability that the sequence belongs to that family is assigned to the query sequence. By comparing this probability score the sequence is determined as belonging to that family or not.

For this approach, we created four different train data sets for training processes as mentioned in feature representation section. We trained each data set to obtain probabilistic suffix trees(PST) so we created four different PST (PST1 consists of data such as flanking amino acids that are located at the left side of metal-bonded CYS or HIS; PST2 consists of data such as flanking amino acids that are located at the left side of CYS or HIS which are not bonded by a metal; PST3 consists of data such as flanking amino acids that are located at the right side of metal-bonded CYS or HIS; PST4 consists of data such as flanking amino acids that are located at the right side of CYS or HIS which are not bonded by a metal.) for each train data set. After PST generation, we built a window for each CYS and HIS from test sets. Then, for each created window, we ran prediction processes for all obtained PSTs. Finally, the outputs of the prediction processes were compared with each other and we marked predicted CYS or HIS as metal bonded or not by evaluating comparison results. However, before comparison, we multiplied outputs of metal-bonded and nonmetal-bonded ones between each other.

D. Dataset

We used a non-redundant set of PDB containing 2727 protein sequences to test our methods. The used data set was prepared by [3] for their research. The detailed and well defined explanation can be found in the mentioned paper. In Table II, we listed some information about this data set.

Table III.
NUMBER OF CYS & HIS AND THEIR STATE OF METAL BOUNDED

	Metal Bounded	Non-Metal Bounded
CYS	933	4702
HIS	678	12982

TABLE IVII.
CHANGE OF SVM PREDICTION RESULTS ACCORDING TO SELECTED FEATURE FOR TRAINING

PAM	Relative Position	5FSS	APAAC	PAAC	PC	RECALL	PRECISION	AUC
X	-	-	-	-	-	0.22	0.24	0.58
X	-	X	-	-	-	0.23	0.25	0.58
X	X	X	-	-	-	0.22	0.24	0.59
X	-	-	X	-	-	0.11	0.18	0.51
X	X	-	X	-	-	0.11	0.18	0.61
X	-	-	-	X	-	0.24	0.27	0.60
X	X	-	-	X	-	0.24	0.27	0.45
X	-	-	-	-	X	0.11	0.14	0.45
X	X	-	-	-	X	0.11	0.13	0.45
X	X	-	-	-	-	0.23	0.25	0.60

TABLE IIIV
CHANGE OF NAIVE BAYES PREDICTION RESULTS ACCORDING TO SELECTED FEATURE

PAM	Relative Position	5FSS	APAAC	PAAC	PC	RECALL	PRECISION	AUC
X	-	-	-	-	-	0.65	0.45	0.78
X	-	X	-	-	-	0.75	0.35	0.80
X	X	X	-	-	-	0.72	0.36	0.76
X	-	-	-	X	-	0.65	0.44	0.77
X	X	-	-	X	-	0.66	0.45	0.77
X	-	-	-	-	X	0.51	0.11	0.59
X	X	-	-	-	X	0.50	0.13	0.64
X	-	-	X	-	-	0.43	0.18	0.62
X	X	-	X	-	-	0.43	0.18	0.61

TABLE V
PREDICTION RESULT OF VARIABLE LENGTH MARKOV CHAIN

PRECISION	RECALL	AUC
0.05	0.60	0.39

III. EVALUATION CRITERIA

In this work, we use precision, recall and area under the curve as performance measurements. The precision was defined as **TP/(TP + FP)**, where **TP** (true positives) was referred to the number of correctly-identified positive examples (metal binding residues); **FP** (false positive) was the number of negative examples (residues predicted to bind metal, although they do not bind to a metal according to PDB) that were incorrectly predicted as positive. The recall was defined as **TP/(TP + FN)**, where **FN** (false negative) was the number of positive examples that were incorrectly predicted as negative. In this study, the negative examples were far more than the positive examples. For such an unbalanced dataset, Area Under Curve (AUC) can present an overly optimistic view of the performance of a method. To

obtain AUC values, we used publicly available MATLAB package.

IV. Results

In this study, we created ten different feature vectors to train with SVM and Naive Bayes. Also we evaluated the predictions for Variable-length Markov chain. All obtained scores are listed in Table III, Table IV and Table V.

The obtained results show us that Naive Bayes is competitive for metal binding site detection.

On the other hand, we used varied feature combinations and they give us a chance to evaluate their prediction score changes according to feature type. For ex; using pam matrix representation is very smart way to identify amino acid for classification because the result is really acceptable and length of this feature limited by number of amino acid count in nature. Also using global descriptors as a feature is practicable for this area.

As a result, we presented a method to predict metal binding sites from amino acid sequences by SVM, Naive Bayes and Variable-length Markov chain .We obtained many results for different feature sets and we reached higher results with Naive Bayes(used features were PAM and 5FSS). The mentioned case predicted CYS/HIS with 35% precision at 75% recall level and 80% AUC value, when tested on a non-redundant set of PDB containing 2727 unique protein chains.

V. Conclusion

Predicting metal-binding conformations of proteins through computational techniques is a favorable effort in the wake of estimating final protein structures. In this study, we evaluate different feature representation schemes and implement different methods to predict metal binding sites of protein residues. Obtained results are compared with each other and valuable feature types are observed. The results justify that Naive Bayes approach can produce acceptable predictions for residue classification. We believe that this study is going to lead our future works and our approach can have an impact on metal binding site detection. We will use Naive Bayes classification for large data set using big data technologies such as spark and storm.

Acknowledgments

We would like to thank those who publicly share their developed codes, scripts and experimental data and those who maintain these useful sharings.

References

[1] J. Reedijk, “Comprehensive Coordination Chemistry”, vol. 2, chp. 13.2, Pergamon, Oxford, pp. 73-98, 1987.

[2] A. J. Thomson and H. B. Gray “Bio-inorganic chemistry”, Current Opinion in Chemical Biology 2: 155–158.

[3] 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.

[4] L. Bancini et. al., “A prokaryotic superoxide dismutase paralog lacking two Cu ligands: from largely unstructured in solution to ordered in the crystal”, Proc Natl Acad Sci USA, 102:7541-7546, 2005.

[5] M. Akke, T. Drakenberg and WJ. Chazin, “Three-dimensional solution structure of Ca(2+)-loaded porcine calbindin D9k determined by nuclear magnetic resonance spectroscopy”, 31:1011-1020, 1992.

[6] H. M. Greenblatt, H. Feinberg, PA. Tucker and G. Shoham, “Carboxypeptidase A: native, zinc-removed and mercury-replaced forms”, 54:289–305, 1998.

[7] H. Sun, H. Li and PJ. Sadler, “Transferrin as a metal ion mediator”, Chem Rev., 99: 2817-2842, 1999.

[8] M. R. Chance and W. Shi, “Metallomics and metalloproteomics.”, Cell Mol. Life Sci., 65, 3040-3048,2008.

[9] W. Shi et. al., “Characterization of metalloproteins by high-throughput X-ray absorption spectroscopy”, Genom Res., 21(6):898-907, 2011.

[10] A. Passerini, M. Lippi and P. Frasconi, “MetalDetector v2.0: predicting the geometry of metal binding sites from protein sequence”, Nucleic Acids Res., 39(Web Server issue):W288-92, 2011.

[11] F. Ferre 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.

[12] 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.

[13] 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.

[14] L. Rishishwar, N. Mishra, B. Pant, K. Pant, and K. R. Pardasani, ProCoS - PROtein COmposition Server, Bioinformation, 5(5): 227. PMC: 3040505, 2010.

[15] G. Bejenora and G. Yona, “Variations on probabilistic suffix trees: statistical modelling and prediction of protein families”, Bioinformatics Vol.17 No.1, pp. 23-43, 2000.

[16] H. Oğul and E. Mumcuoğlu, “SVM-based detection of distant protein structural relationships using pairwise probabilistic suffix trees”, Computational Biology and Chemistry Vol.30, pp. 292-299, 2006.

[17] M. Boulle, “Parsimonious Naïve Bayes”, 2014 Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 355-359, 2014.