

Towards Site-based Protein Functional Annotations

Seak Fei Lei, Jun Huan
School of Electrical Engineering and
Computer Science
University of Kansas
Lawrence, Kansas 66045
Email: {seakfei,jhuan}@ku.edu

Abstract—The exact relationship between protein active centers and protein functions is unclear even after decades of intensive study. To improve the functional prediction ability based on the local protein structures, we proposed three different methods. 1) We used statistical model (known as Markov Random Field) to describe protein active region based on the structure motifs. 2) We developed a filter that considers the local environment around the active sites to remove the false positives. 3) we created multiple structure motifs by extending the motif to neighboring residues for delineating their functions.

Our experimental results, as evaluated in five sets of enzyme families with less than 40% sequence identity, demonstrated that our methods can obtain more remote homologs that could not be detected by traditional sequence-based methods. At the same time, our method could reduce large amount of random matches. Our methods could improve up to 70 % of the functional annotation ability (measured by their Area under the ROC curve) in extended motif method.

I. INTRODUCTION

Understanding structure-function relationship is a fundamental problem in biology. Though sequence based functional annotation has been used for many years, annotating remote homologs — proteins that have similar functions but diverse sequences is a challenging problem. With the fast growing number of protein structures [1], there is a pressing need to perform a *in silico* discovery of proteins' molecular functions using protein structure data, or structure-based functional annotations, which is the focus of this paper.

There are accumulating evidences showing that proteins perform their functions in relative small regions. These local structures are called active sites, which include enzymatic activity centers and protein ligand-receptor binding sites. However, the mappings between an active site and their functions are non-trivial. This is due to a couple of reasons: 1) The sizes of functional regions are typically small (under 20 aa [2]), which causes random matches to unrelated proteins. 2) The shortcomings of current motif models: some of the motif models may not be able to fully describe the active region of a protein family due to its limited search space. For example, a simple sequence model with k residues can only represent 20^k motif instances, which restricts its ability to obtain the optimal result. To conclude, improving active site-based function prediction is necessary.

In this paper, we explored two avenues (total three methods implemented) to improve the prediction results. The first approach involved building a statistical model to describe the

functional site. Specifically, we used Markov Random Field to refine a given active site structure. In second approach, we improved annotations by considering the environments around the functional site. We computed statistical distributions of the environment in order to filter random matches. We created multiple active site representations based on environment information, then we aggregated final results using machine learning techniques such as voting method and feature vectors method.

The rest of the paper is organized as follows. In section II, we review the latest developments on improving prediction results using motifs. In section III, we go through basic graph theory behind our methods. In section IV, we introduce our novel motif refinement algorithm and the filter methods. In section V, we present our experimental study and provide some performance analyses. In section VI, we draw some conclusions from our experiments and discuss the future works.

II. RELATED WORKS

To improve the sensitivity and specificity of an existing functional site, researchers investigate the refinement/filter problem in two directions. First of all, they introduce domain constraints to better identify functional homologs. For example, Geometric Sieving [3] compares the Least Root Mean Squared Distance (LRMSD) distributions between a candidate motif and an external protein set in order to select the optimal motif structure. LRMSD is a similarity measure between two point list representations (the candidate motif and a protein from the data set). The assumption behind Geometric Sieving is that an optimized motif should demonstrate the maximal geometric and chemical differences to all known protein structures. As a result, the researchers first generate a candidate motif set by considering all possible subsets of the original motif, and pick a candidate motif with the highest median LRMSD distribution as the refined motif. Cavity-aware motifs [4] combine structure motifs with a set of spheres known as C-spheres to imitate the protein's active site and its surrounding space for chemical reactions. Other information like structure energy level [5], electric charge or hydrophobicity can also be used for motif improvement.

III. BACKGROUND

This section introduces the concept of labeled graph from the graph theory. We treated all protein structures and their active sites as labeled graphs. Hence, protein functional annotation problem is converted to a graph problem.

A labeled graph is defined as the following,

Definition 1: (Labeled Graph) A labeled graph G is a five elements tuple $G = (V, E, \Sigma_V, \Sigma_E, \lambda)$ where,

- V is a set of vertices or nodes.
- E is set of undirected edges $E = V \times V$.
- Σ_V is disjoint sets of vertex labels
- Σ_E is disjoint sets of edge labels
- λ is a function that assigns labels to the vertices and edges.

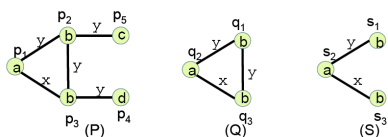


Fig. 1. Examples of labeled graphs. In this paper, protein/active site structures are modeled using labeled graph. Node labels such as a; b; c; d represent amino acids, and the edge labels like x; y are the Euclidean distance between two nodes.

Figure (1) shows an example of a graph database. This representation has been used by many researches, including our previous work [6]. In this paper, we used the following mappings between a protein structure and a labeled graph:

- Nodes \iff amino acids
- Edges \iff chemical / physical interactions among amino acids
- Node labels \iff 20 amino acid types
- Edge labels \iff Euclidean distances of the interactions

To increase the matching efficiency, two types of edges distances are included in the graph — bond edges and proximity edges. Bond edges are polypeptide chains appear in the protein primary sequence. Proximity edges consider the relations of neighbors in its 3D structure. Two residues are treated as connected with a proximity edge if their Euclidean distance is less than a threshold δ . In this paper, we select the δ to be 8.5 \AA .

To identify the function of a protein, we employed the idea of graph matching. Given an active site from a protein family and a protein structure, graph matching determines whether an one-to-one mapping function exists between them. If such a mapping is found, the protein is consider as part of the protein family (i.e. has functions similar to proteins in the same family). Figure (2) shows a flowchart of this graph-based annotation process.

IV. METHODS

In this section, all proposed methods will be discussed in detail. These methods include: Motif refinement with Markov Random Field motif model, the environment filter, and the extended motif filter.

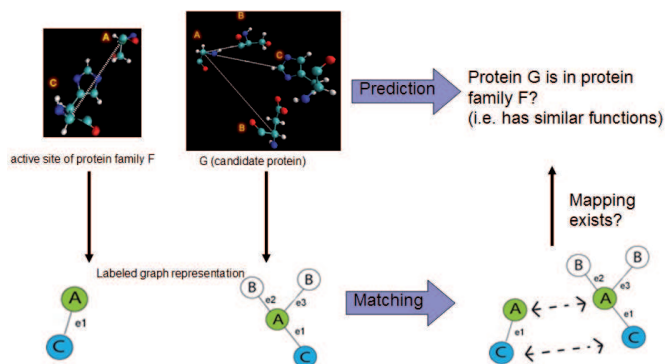


Fig. 2. Breakdown of graph-based protein function annotation

A. Motif refinement with Markov Random Field motif model

Our algorithm takes a functional site of a protein (known as initial motif) and a testing protein dataset as input. The algorithm outputs a new statistical model which can better describe the remote homologs by repeatedly improving the functional site. Figure (3) shows an overview of our algorithm. In general, this algorithm can be divided into three stages:



Fig. 3. Flowchart of motif refinement algorithm.

1) Initial graph matching (Approximate graph matching):

After the functional site and proteins are represented as labeled graphs, we want to determine whether a functional site graph M occurs in a graph G in a flexible manner while still be able to maintain the relevance to the data set. Hence, we introduce a scoring matrix to the node mapping to quantify the degree of the similarity between two graphs. Formally speaking,

Definition 2: (Initial Matching) Graph $G = (V, E, \Sigma_V, \Sigma_E, \lambda)$ is *subgraph isomorphic* to $G' = (V', E', \Sigma_{V'}, \Sigma_{E'}, \lambda')$ if there exists a 1-1 mapping $f: V \rightarrow V'$ such that ,

$$\sum_{u \in V} S(\lambda(u), \lambda'(f(u))) \geq T_1, \text{ and } (1)$$

$$\frac{dRMSD(E, E')}{\sqrt{\frac{\sum_{(u,v) \in E} [\lambda(u,v) - \lambda'(f(u), f(v))]^2}{|V|(|V|-1)/2}}} \leq T'_1 \quad (2)$$

where S is a node matching function that penalizes a node label mismatch, $|V|$ is the size of the graph (i.e. total number of nodes), T_1 is a threshold for node label mismatch, and T'_1 is a threshold for structural differences.

Formula 2 is defined as distance root-mean-square deviation(dRMSD) between G and G' . It is a well-known standard for structural comparison [7]— Larger dRMSD means more

diverse protein structures. In this paper, we set T_1' to be 0.8\AA and S to be the scoring matrix BLOSUM62 [8].

2) *Building refined functional site*: Our new functional site model is defined as follows,

Definition 3: (Pattern Statistical model) Our new functional site model is a triple $(\Theta, \Sigma_E, \lambda)$, where Θ is a Markov Random Field(MRF): $\Theta(n) \rightarrow \mathbb{R}^+$ with $n \in N$. Σ_E is a set of edge labels; and the λ is a function that assigns the edge labels to the corresponding edges in the Markov Random Field graph.

This model not only contains both labeled items and structure components, but also offers large (almost infinite) search space for our algorithm to optimize a functional site. MRF consists of many parameters, including normalization factor Z , and potential functions V of the maximal cliques. To estimate those parameters, we apply Radim Jirousek's Iterative proportional fitting algorithm (IPF) [9]. Given a set of instances from the previous matching, IPF will try to modify the potential function for each clique $V(X_c)$ such that the marginal probability $p(X_c = x_c)$ equals to the maximum likelihood (ML) estimate (in this case ML is the empirical marginal).

3) *Re-matching*: This stage is similar to the initialization stage. Both stages determine if the motif occurs in a protein structure. But in this stage, the newly constructed MRF model from the last stage is used to match with proteins instead of the initial motif M . In other words, criterion (2) in initialization stage will be reused in this stage, and criterion (1) is replaced with the Gibb distribution formula as node matching function,

$$p(x) = \frac{1}{Z} \prod_{c \in C} V_c(x_c)$$

where Z and V_c are the parameters in MRF, and x_c is a subgraph (maximal clique) configuration of the candidate protein. In short, the Gibb distribution formula takes a MRF model and a subgraph as inputs, and outputs the probability that the candidate protein is related to the protein family. See [10] for the detail derivation of the formula.

In our actual implementation, the last two stages (Re-matching and model building) run iteratively until the number of instances captured converges. To carry out the graph matching in both initialization and re-matching stage, we employ J. R. Ullman's occurrence algorithm. For more detail about the proof and implementation of occurrence algorithm, please refer to [11].

B. The environment filter

The environment filter assumes that the local environment around the active site is a determining factor for protein functions. If the surroundings of a probable motif location is very different from other proteins in the same family, then this site may not be functional, and thus having different functions. The environment filter works in two stages. In the first stage, a profile is generated for a particular protein family, which is known as the environment profile. The

environment profile is defined as follows,

Definition 4: (The environment profile) The environment profile P is an ordered list of 20 triples $[(a_1, \mu_1, \sigma_1), \dots, (a_{20}, \mu_{20}, \sigma_{20})]$ where each element represents one amino acid, a_i is the amino acid identifier i , μ_i is the mean frequency of amino acid i , and σ_i is the standard deviation of frequency in amino acid i .

To generate the environment profile in the first stage, it requires a set of proteins from the same protein family, known as protein family set. For each protein in the protein family set, we first collect the neighboring residues around the active site. The neighboring node of an active site is described as following,

Definition 5: (Neighbors of a motif) A node v is considered as the neighbor of active site $G' = (V', E', \Sigma_{V'}, \Sigma_{E'}, \lambda')$ which resides inside a protein structure $G = (V, E, \Sigma_V, \Sigma_E, \lambda)$ if it satisfies the following conditions:

$$v \in V \text{ and } v \notin V', \\ \exists w \in V' \text{ such that } (v, w) \in E \text{ and } \lambda(v, w) \leq 8.5$$

For each protein in the protein family set, the normalized frequency distribution of its neighboring nodes is computed, results in a tuple of twenty numbers. When all the distributions values set are gathered from the family set, we can calculate the environment profile for the protein family,

Given the normalized frequency distributions for N proteins in the protein set: $(d_{1,1}, \dots, d_{20,1}), \dots, (d_{1,N}, \dots, d_{20,N})$, the environment profile $P = [(a_1, \mu_1, \sigma_1), \dots, (a_{20}, \mu_{20}, \sigma_{20})]$ is computed by the following formula,

$$\mu_i = \frac{\sum_{j=1}^N d_{i,j}}{N-1} \quad \sigma_i = \sqrt{\frac{1}{N-1} \sum_{j=1}^N (d_{i,j} - \mu_i)^2}$$

where $i = \{1, 2, 3 \dots 20\}$

To apply the environment profile for improving protein annotations, we need an environment profile $P = [(a_1, \mu_1, \sigma_1), \dots, (a_{20}, \mu_{20}, \sigma_{20})]$, and a candidate protein which is matched to a functional site using approximate matching (see IV-A1 for detail). We calculate the normalized amino acid distribution around matched site of the candidate protein (d_1, \dots, d_{20}) . Then, the difference between the distribution and the profile is obtained using this formula,

$$\sum_{i=1}^{20} \frac{|d_i - \mu_i|}{\sigma_i} \geq T_3$$

where T_3 is called the filter threshold, which is an adjustable value for strictness of the filter. If the result is smaller than the filter threshold, the candidate protein will be considered as a real match .

C. The extended motif filter

Similar to the environment filter, the extended node filter also employs surrounding information of the active site. We randomly embrace one of the neighboring nodes into the functional site, thus enlarging the motif size by one. The definition of the active site neighbor is identical to environment filter.

Although enlarging existing motif can provide stronger discriminative power when doing prediction, larger motif may tend to filter out true samples too. Therefore, we consult multiple extended motifs and combine their matching results. In this study, we used two different ensemble techniques from machine learning: the feature vector method and the voting method.

Feature vector method: Given a set of extended motifs, we apply the approximate matching method (see IV-A1 for detail) to each motif and gather a set of matched protein with their node mismatch scores (as defined by criterion 1). Next, for every protein in the dataset, we form an ordered list of length n , which is the total number of extended motifs (in this experiment $n = 4$) as feature vector. Each feature value represents the matching score obtained from an extended motif. Machine learning approaches like Support vector machine (SVM) will then be utilized to study underlying patterns of the features. The trained model will be used for functional predictions.

Voting method: Given a set of extended motifs which is enlarged by the neighboring residues, we again apply the approximate graph matching method on them. The matched proteins, along with the node mismatch scores from each motif are averaged by their geometric mean.

$$v_p = \left(\prod_{i=1}^n s_{i,p} \right)^{1/n}$$

where v_p is the voting score (averaged score) for protein p , n is total number of extended motifs, and $s_{i,p}$ is the mismatch score for protein p using extended motif i .

The matched proteins will be sorted according to the averaged voting scores. An extra parameter T_4 will be used to determine the number of top-scored results to pass the filter.

V. EXPERIMENTAL STUDY

Each of the proposed methods underwent a series of tests from the real-life protein dataset. In particular, five enzyme families were selected for functional annotation. These enzymes, as one type of proteins, are carefully categorized by Enzyme Commission according to their molecular functions. We compared their performance trade-offs using receiver operating characteristic (ROC) analysis, as well as the area under the ROC curve (AUC) measure. In the followings sections, we will talk about how we construct the training and testing dataset, and discuss the experiment results.

A. Data collection

We randomly picked up five protein functions which span several structural families (SCOP family ID) for protein predictions. For each function (as described by the enzyme[EC]

EC number	Active Region (Query protein)	Source
3.4.21	HIS57-GLY193-SER195 (1mcta)	CSA
3.4.22	CYS25-HIS159-ASN175 (1pppa)	CSA
6.3.2	GLU15-SER150-GLY276 (2dlna)	Fan <i>et al.</i> [12]
1.1.1	ASP201-ARG204-HIS229 (7mdha)	CSA
FAD binding (1.8.1+1.18.1)	GLY11-GLY13-GLY16 -ALA20 (1q1ra)	Hanukoglu <i>et al.</i> [13]

TABLE I

ENZYME FAMILIES USED IN THIS EXPERIMENT. THE *source* COLUMN INDICATES WHERE INITIAL MOTIFS ARE OBTAINED (EITHER FROM CATALYTIC SITE ATLAS [CSA] OR FROM LITERATURE).

family in table I), we followed these steps to create the training and testing dataset:

- Retrieved all protein structures from the EC family from structural database Protein Data Bank.
- Randomly picked one protein as query protein, then obtain its functional site from literature database like PubMed¹ or catalytic residue database such as Catalytic Site Atlas (CSA). The selected query protein from each EC family, along with their active regions and their original sources are shown in table I.
- Identified structural classification of query protein (i.e. SCOP family ID).
- Proteins with the same SCOP ID as query protein were *positive training samples*, other proteins in the same EC family but NOT in the training set were used for *positive testing samples*.
- Random proteins which are not in these five enzyme families were picked as negative *training and testing samples*.
- Both training and testing samples went through pre-processing step.

In the preprocessing step, we first made sure there was no overlap between the training and the testing dataset. Then, we removed all ‘trivial’ proteins by 1) eliminating proteins with sequence identities $> 40\%$. 2) All protein matches that can be done by sequence-based annotation method such as PSI-Blast. The goal of this pre-processing step is to examine if our methods can recognize remote homologs with very different folds.

All 3D coordinate information of the proteins and motifs in this study was obtained from the Protein Data Bank² (PDB). The SCOP database (version 1.71)³ provided information about the protein structure families. Two proteins are considered as functionally related if EC numbers are identical to the third levels, according to the classification scheme defined in ENZYME⁴ database (version 11/2007). To gather the true members from those families, we utilized the list provided by PDBSProtEC⁵ mapping [14]. Preprocessing step was partly done by the Protein Sequence Culling Server (PISCES) [15].

¹<http://www.ncbi.nlm.nih.gov/sites/entrez>

²<http://www.rcsb.org/pdb/home/home.do>

³<http://scop.mrc-lmb.cam.ac.uk/scop/>

⁴<http://ca.expasy.org/enzyme/>

⁵<http://www.bioinf.org.uk/pdbsprotec/>

EC Family	Training		Testing	
	# Positive	# Negative	# Positive	# Negative
3.4.21	10	10	23	1000
3.4.22	8	8	16	1000
6.3.2	9	9	21	1000
1.1.1	7	7	13	1000
FAD binding (1.8.1+1.18.1)	6	6	14	1000

TABLE II
DATASET STATISTICS.

We downloaded a pre-compiled list provided by their server⁶. Table II shows the number of true and false samples for the training and testing dataset after preprocessing.

B. Experiment procedures

We compared the following five methods for functional predictions using the training and testing dataset mentioned in section V-A:

1) For baseline comparison, we approximate matched the original functional site to the proteins in testing set. Proteins would have similar function (i.e. come from the same protein family) if they both contain the same functional site. BLSOUM62 was used to match their nodes and dRMSD was used to match their edges. The matching scores that pass a pre-defined threshold would be considered as related. This method is known as approximate matching method (Approx.) , see section IV-A1 for more detail.

2) Similar to the approximate matching method, except we matched our proposed MRF model to the test proteins instead of the initial functional site. Proteins would have similar function if they match to the MRF model with the joint distribution values larger than a predetermined threshold. The MRF model, on the other hand, is constructed using our proposed motif refinement method. This method is called motif refinement algorithm (MRF) .

3) We first computed environment filter using the positive training samples from each EC family. Then we applied the approximate matching method to the testing samples. Proteins have similar function if they pass the filter threshold with enough node matching scores. This method is known as environment filter method (Env filter).

4) We created four extended motif by randomly adding an extra neighboring residue to the original functional site. The amino acids that were selected to be in the functional site are shown in the Appendix. To aggregate the approximate match results from those extended motif, we took the geometric mean from their node matching scores. Proteins would have similar function if their averaged scores pass the score threshold. We called this method as voting method (voting).

5) Rather than computing the averages like the voting method, we built an ordered list (feature vector) of matching scores for each proteins. Support vector machine (SVM) with radial basis function (RBF) kernel was utilized to study

⁶<http://dunbrack.fccc.edu/PISCES.php>. The parameters used in this list are: resolution=6.0, R factor=0.25.

EC number	Approx.	Env filter	MRF	voting	SVM
3.4.21	0.580	0.671	0.468	0.630	0.678
3.4.22	0.642	0.581	0.488	0.696	0.643
6.3.2	0.332	0.430	0.363	0.564	0.550
1.1.1	0.554	0.701	0.726	0.740	0.701
FAD binding (1.8.1 + 1.18.1)	0.559	0.420	0.442	0.523	0.719

TABLE III

THE AREA UNDER CURVE (AUC) WITH FIVE DIFFERENT METHODS TESTING FIVE EC FAMILIES. THE BOLD NUMBERS ARE THE LARGEST NUMBER OF EACH ROW, WHICH INDICATES THE BEST PERFORMANCE POSSIBLE AMONG ALL FIVE METHODS.

underlying patterns of the features. After using 3-fold cross validation to select the best model parameters from the training data, the trained SVM annotated testing proteins with their feature vectors as inputs.

Because all of our proposed methods were designed on top of the approximate matching algorithm, we fixed its parameters in all of our methods for the ease of performance comparison. And for each of our proposed method, its discrimination threshold was varied to generate the ROC curve. We performed our experiments on a cluster. It has total 128 nodes, 384 Intel Xeon processors and 640 GB of memory. The SVM implementation for the feature vectors method is from the LIBSVM package [16].

C. Experimental results

Table III lists the AUCs of the five methods testing five EC families. Larger area usually indicates better performance. We also summarizes the performance differences using ROC analysis for each EC family, and their results are shown in the Appendix.

In the following sections, we will discuss our observations of each method in detail.

D. Results of approximate match with original functional site

As expected, the approximate matching method did not perform well when compared with other methods. Its ROC curves trended to stay at the bottom half of the graph, and its AUC values were smaller than other proposed methods in most cases. The poor performance was attributed to random matches occurred in various locations of unrelated proteins. The average size of a functional site is very small, and the approximate matching method allows partial matches by introducing scoring functions. These two factors resulted in large number of false positives and false negatives results.

1) Results of approximate match with environment filter:

Compared with the approximate matching method, as seen in table III, three out of five EC families showed a positive response to the filter. And the AUC improvement rate ranged from 15% to about 30%. All of these facts entail that the surrounding distributions of residues may determine the emergence of active regions. However, the introduction of the environment filter also brought us another side effect—the reduction of the true positives. Both EC families 3.4.22 and FAD binding sites exhibited drops on their AUCs after the

environment filter was applied. The implications of diminishing true positives can be attributed to the diversity of the true samples and lack of proteins for profile generation. If a protein had a really different structure than the query protein, its active site environment might also be very different. This situation could be seen when a true protein was filtered through a high threshold. Inaccurate distribution from the profile was another reason for the loss of the TPs.

2) *Results of motif refinement algorithm:* On average, the MRF generation processes converged in 2 to 3 iterations. Overall, our refinement algorithm finished within 3 to 8 iterations as well. Nonetheless, the algorithm did not perform well as expected. Only two (EC 6.3.2 and 1.1.1) out of five experiments had AUC values larger than the baseline method, and their improvement rates were not significant compare to other proposed methods. The refined MRF model did pick up additional remote homologs during the initial and re-matching stages. However, large amount of false positives (FPs) also got included, and those FPs increased as the algorithm iterated. As a result, FPs accumulated at every iteration.

3) *Results of voting method using extended motif filter:* Among all the methods listed in the AUC analysis, voting method had the best performance in terms of the percentage of improvement. In four out of five experiments, ROC analysis showed that voting method outperformed the baseline approximate matching technique (except for the FAD binding family). Its improvement rate can go up to 70% (EC 6.3.2). In addition, three experiments: EC 3.4.22, EC 6.3.2, and EC 1.1.1 showed that voting method formed the largest AUC among all the proposed methods. The geometric mean heavily penalized the proteins to which the extended motifs disagreed. A probable protein would have an averaged score of zero even if one of the extended motifs could not capture that protein — any proteins that did not include in their matching results would have a zero mismatch score. This effect of multiplication in geometric mean computation would potentially filter out all the FPs: only proteins that acquired the consensus from all the extended motifs were remained as functional homologs.

4) *Results of feature vector method using extended motif filter:* Among all the methods listed in the AUC analysis, feature vector method had the best performance in terms of the stability of improvement. In all of our experiments, the feature vector method always provided some degrees of improvement, up to 65% raise in AUC. Even in the FAD binding family experiment, where no proposed method so far could improve the baseline approximate matching results, the feature vector method could raise the AUC by 28.5%. The results shown in the ROC graphs and the AUC table were non-binary features using RBF as SVM kernel.

VI. CONCLUSION

We proposed three different approaches to refine the annotation results based on a query protein with its functional site. The experiments on five sets of enzyme families demonstrated that our algorithms can get up to 70% increase in AUC when compared with the baseline method. This fact illustrates

that our methods obtain remote homologs across diverse global structures using a single query protein. Among all of our approaches, voting method has the best performance in terms of the percentage of improvement, and feature vector method has the best performance in terms of the stability of improvement. In this study, all initial patterns were obtained from the literature/database. For our future works, we can first make use of a subgraph mining tool to gather a set of initial motifs which occur in the input sets frequently. Our algorithm will then take over and refine each of the functional site. Finally, these optimized models will be tested statistically to make sure they are not generated by chance. By using this approach, we can truly perform a large-scale automatic test to construct a more effective functional site.

APPENDIX

Available online at <http://people.eecs.ku.edu/~jhuan/>.

REFERENCES

- [1] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The Protein Data Bank," *Nucl. Acids Res.*, vol. 28, no. 1, pp. 235–242, 2000. [Online]. Available: <http://nar.oxfordjournals.org/cgi/content/abstract/28/1/235>
- [2] Sung-Hou Kim and Dong Hae shin and In-Geol choi and Ursula Schulze-Gahmen, and Shengfeng chen and Rosalind kim, "Structure-based functional inference in structural genomics," *Journal of Structural and Functional Genomics*, vol. 4, no. 2/3, pp. 129–135, 2003.
- [3] D. H. B. Brian Y. Chen, Viacheslav Y. Fofanov, "Geometric sieving: Automated distributed optimization of 3d motifs for protein articlefunction prediction," *Proceedings of The Tenth Annual International Conference on Computational Molecular Biology (RECOMB 2006)*, 2006.
- [4] B. Y. Chen, D. H. Bryant, V. Y. Fofanov, D. M. Kristensen, A. E. Cruess, M. Kimmel, O. Lichtarge, and L. E. Kavasaki, "Cavity-aware motifs reduce false positives in protein function prediction." *Comput Syst Bioinformatics Conf*, 2006.
- [5] A. Kolinski, M. Betancourt, D. Kihara, P. Rotkiewicz, and J. Skolnick, "Generalized comparative modeling (genecomp): A combination of sequence comparison, threading, and lattice modeling for protein structure prediction and refinement," *Proteins*, vol. 44, no. 2, 2001.
- [6] J. Huan, D. Bandyopadhyay, J. Prins, J. Snoeyink, A. Tropsha, and W. Wang, "Distance-based identification of structure motifs in proteins using constrained frequent subgraph mining," *Computational systems bioinformatics / Life Sciences Society. Computational Systems Bioinformatics Conference*, pp. 227–238, 2006.
- [7] B. Zagrovic and V. Pande, "How does averaging affect protein structure comparison on the ensemble level?" *Biophys. J.*, vol. 87, pp. 2240–2246, Oct 2004.
- [8] S. Henikoff and J. Henikoff, "Amino Acid Substitution Matrices from Protein Blocks," *Proceedings of the National Academy of Sciences*, vol. 89, no. 22, pp. 10915–10919, 1992.
- [9] L. R., "Convergence of the iterative proportional fitting procedure," *The Annals of Statistics*, vol. 23, no. 4, pp. 1160–1174, 1995.
- [10] J. Hammersley and P. Clifford, "Markov fields on finite graphs and lattices," 1971, unpublished manuscript.
- [11] J. R. Ullman, "An algorithm for subgraph isomorphism," *Journal of the Association for Computing Machinery*, vol. 23, pp. 31–42, 1976.
- [12] C. Fan, P. Moews, C. Walsh, and J. Knox, "Vancomycin resistance: structure of D-alanine:D-alanine ligase at 2.3 Å resolution," *Science*, vol. 266, pp. 439–443, Oct 1994.
- [13] I. Hanukoglu and T. Gutfinger, "cDNA sequence of adrenodoxin reductase. Identification of NADP-binding sites in oxidoreductases," *Eur. J. Biochem.*, vol. 180, pp. 479–484, Mar 1989.
- [14] A. C. Martin, "PDBSprotEC: a Web-accessible database linking PDB chains to EC numbers via SwissProt," *Bioinformatics*, vol. 20, no. 6, pp. 986–988, 2004.
- [15] G. Wang and R. L. Dunbrack, "PISCES: a protein sequence culling server." *Bioinformatics*, vol. 19:1589–1591, 2003.
- [16] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001.