Published in final edited form as:
Int J Data Min Bioinform. 2010 ; 4(4): 452–470.

Towards Site-based Protein Functional Annotations

Seak Fei Lei and **Jun Huan***
School of Electrical Engineering and Computer Science, University of Kansas, Lawrence, Kansas 66045

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 developd 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.

Keywords

Protein function prediction; Markov Random Field

1 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. Experimental methods like ChIP on chip, on the order hand, can accurately discover protein functions. Nevertheless, most of them engage in expensive and lengthy processes. With the fast growing number of protein structures (Berman et al. 2000), 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 in length as mentioned in Kim et al. (2003)], 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

*Corresponding author, slei@eecs.ku.edu jhuan@ku.edu

k residues can only represent 20^k motif instances, which restricts its ability to obtain the optimal result. Some motif models may also impose strict assumptions, e.g. independence among residues. As a result, the motif model requires multiple motif instances to describe a single protein family. Exact matching from maximum cliques would limit the number of annotations since the composition of the active site trends to vary slightly within the protein family. 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 to refine a given active site structure. Markov Random Field, as one of the statistical graph models, can preserve overall topology of the active site while maintaining information about its amino acid composition. In second approach, we improved annotations by considering the environments around the functional site. We computed statistical distributions of environment, in this case, the surrounding residues in 3D space in order to filter random matches. If a candidate protein had a very surrounding profile than the other proteins from the same family, it would be considered as unrelated. We also 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 2, we review the latest developments on improving prediction results using motifs. In section 3, we go through basic graph theory behind our methods. In section 4, we introduce our novel motif refinement algorithm and the filter methods. In section 5, we present our experimental study and provide some performance analyses. In section 6, we draw some conclusions from our experiments and discuss the future works.

2 Related Works

To improve the sensitivity and specificity of protein functional annotations using 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 (Brian Y. Chen 2006) 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 (Chen et al. 2006) 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 from Kolinski et al. (2001), electric charge or hydrophobicity can also be used for motif improvement.

In data mining community, some studies focus on summarizing patterns using statistical models or machine learning algorithms. Yan et al. (2005) construct pattern profile on a frequent itemsets based on Bernoulli distributions with clustering techniques. Wang & Parthasarathy (2006) reduce the number of frequent itemset patterns by building a model at each level iteratively. In protein analysis, Berger & Singh (1997) implement another iterative method that uses randomness and statistical techniques to improve the motif recognition on coiled coils proteins. Shah et al. (2008) and Xiao & Segal (2008), on the other hand, use iterative method on top of multiple classifiers. They both tackle this imbalanced dataset by training classifiers

with unlabeled data. The unknown data is selected based on the closeness of the positive/negiative data points. During the re-training process at each iteration, the classifiers become more accurate and thus more proteins will be annotated correctly.

3 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.

Figure (1) shows an example of a graph database. This representation has been used by many researches, including our previous work(Huan et al. 2006). In this paper, we used the following mappings between a protein structure and a labeled graph:

- Nodes ⇔ amino acids
- Edges ⇔ chemical / physical interactions among amino acids
- Node labels ⇔ 20 amino acid types
- Edge labels ⇔ 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 Å.

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.

4 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.

4.1 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:

4.1.1 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,*

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

$$dRMSD\left(E, E'\right) = \sqrt{\frac{\sum_{(u,v)\in E}\left[\lambda(u,v) - \lambda'(f(u), f(v))\right]^2}{|V|(|V|-1)/2}} \leq T_1' \tag{2}$$

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(Zagrovic & Pande 2004)— Larger dRMSD means more diverse protein structures. In this paper, we set T_1' to be 0.8$\mathring{A}$ and S to be the scoring matrix BLOSUM62(Henikoff & Henikoff 1992).

4.1.2 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):* $\Theta(n) \rightarrow \Re^+$ *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. At the same time, our model enforces certain restrictions on the edges labels and nodes labels (i.e. the potential functions) such that dependencies among neighboring elements can be preserved.

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)(R. 1995). 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).

4.1.3 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 (Hammersley & Clifford 1971) 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. As a result, our algorithm goes through the MRF search space iteratively so that optimal parameters are found for a given functional site. 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 Ullman (1976).

4.2 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 $[(a_1, \mu_1, \sigma_1), ..., (a_{20}, \mu_{20}, \sigma_{20})]$ 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:*

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

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: $(d_{1,1}, ...d_{20,1})$, ..., $(d_{1,N}, ...d_{20,N})$, the environment profile $P = [(a_1, \mu_1, \sigma_1), ..., (a_{20}, \mu_{20}, \sigma_{20})]$ is computed by the following formula,

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

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

To apply the environment profile for improving protein annotations, we need an environment profile $P = [(a_1, \mu_1, \sigma_1), ..., (a_{20}, \mu_{20}, \sigma_{20})]$, and a candidate protein which is matched to a functional site using approximate matching (see 4.1.1 for detail). We calculate the normalized amino acid distribution around matched site of the candidate protein $(d_1, ...d_{20})$. 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 .

4.3 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 4.1.1 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.

5 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.

5.1 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] family in table 1), 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 PubMed [a] 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 1.
- 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 Bank [b] (PDB). The SCOP database (version 1.71)[c] provided information about

[a] http://www.ncbi.nlm.nih.gov/sites/entrez
[b] http://www.rcsb.org/pdb/home/home.do

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 ENZYME[d] database (version 11/2007). To gather the true members from those families, we utilized the list provided by PDB-SProtEC [e] mapping (Martin 2004). Preprocessing step was partly done by the Protein Sequence Culling Server (PISCES) (Wang & Dunbrack 2003). We downloaded a pre-complied list provided by their server [f]. Table 2 shows the number of true and false samples for the training and testing dataset after preprocessing.

5.2 Experiment procedures

We compared the following five methods for functional predictions using the training and testing dataset mentioned in section 5.1:

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 4.1.1 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 Table 3. 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 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. All of our proposed algorithms are implemented as a serial

[c] http://scop.mrc-lmb.cam.ac.uk/scop/
[d] http://ca.expasy.org/enzyme/
[e] http://www.bioinf.org.uk/pdbsprotec/
[f] http://dunbrack.fccc.edu/PISCES.php. The parameters used in this list are: resolution=6.0, R factor=0.25.

application using C++. For the C++ compilation environment, we used gcc compiler with 03 optimization. The SVM implementation for the feature vectors method is from the LIBSVM package (Chang & Lin 2001).

5.3 Experimental results

Figure (4) summarizes the performance differences using ROC analysis for EC 3.4.21. In the ideal case, a perfect method should have a curve that passes through the coordinate (0,1) point, which reveals that it can achieve 100% true positive rate (100% precision) and 0 % false positive rate (100% recall). On the other hand, a method which forms a diagonal line from (0,0) to (1,1) would imply a performance of random guesses. In short, a good method should have a ROC curve close to the top left corner of the graph. Figure (5,6,7,8) show the ROC analysis for the rest of the EC families. Generally, other ROC graphs follow the similar trend as Figure (4).

Table 4 lists the AUCs of the five methods testing five EC families. Larger area usually indicates better performance.

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

5.4 Results of approximate match with original functional site

All families were reported with reasonable amount of true matches. And since ‘trivial’ matches were removed during preprocessing, the approximate matching technique recovered more remote homologs than PSI-Blast. The reason why sequence-based annotation methods like PSI-Blast cannot detect those true positives is that they focus on the global configuration of the proteins. For instance, BLAST tries to align the query sequence to the database, protein which has a longer matches with the query sequence normally has higher probability to get picked during the statistical computation of e-values. The advantage of this approach is that it has very few false positives, as the matches are somehow similar to the query protein. Nonetheless, proteins (especially enzymes) can retain their functions largely due to their active regions, not the rest of the protein structures. As a result, although PSI-Blast can pick up homologs accurately, it has difficulties to recover remote homologs which have diverse structures. In short, the search space of sequence-based method is restricted, and in this case, it may stuck with the local optimal solution.

However, the approximate matching method did not perform well when compared with other proposed 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. In fact, some matches were found within a single true positive in different locations, meaning that this method could not event locate the active site correctly. 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.

5.4.1 Results of approximate match with environment filter—Compared with the approximate matching method, as seen in Table 4, three out of five EC families showed a positive response to the filter. And the AUC improvement rate raged from 15% to about 30%. All of these facts entail that the surrounding distributions of residues can determine the emergence of active regions. But 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 true positives. Some amino acids in the profile have very low standard deviation. During the profile difference calculation, the quotient became very large. Including more proteins with distinct structures (e.g. different SCOP families) during the profile generation should alleviate the problem.

5.4.2 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. Both facts indicate that MRF model became more generalize to a particular protein family as it converged. 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. Although the refined MRF model did pick up additional remote homologs during the initial and re-matching stages, 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. This phenomenon is known as propagation effect and is very common in iterative algorithms such as PSI-Blast. To avoid propagation effect, one have to make sure the quality of initial matching results so that FPs cannot retain and propagate. We have already applied the environment filter to filter out some of the FPs in the initial matching stage. One may either crate an additional filter, or manually gather a list of TPs to bypass the first stage of the algorithm.

5.4.3 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. As multiplication is used to aggregate the proteins' mismatch scores, 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. This voting method actually makes biological sense because every extended motif includes different additional features from the active site environment. If a protein that satisfies all the characteristics described by the motifs, that protein will have a high probability to be related to the query protein.

5.4.4 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. To further enhance the annotation ability of the feature vector method, one may try other machine learning techniques such as feature selections, feature extractions, or different classifiers. One can also use more extended motifs to enlarge the size of the feature vector so that better results

can be achieved. The main purpose of this experiment is just a proof-of-concept — we just want to show that it is feasible to use our extended motif to achieve better annotation results.

6 Conclusion

We proposed three different approaches to refine the annotation results based on a query protein with its functional site. One of which involves the reconstruction of the motif model using a statistical model MRF, and the rest of them utilize the active center surroundings and multiple extended motifs to eliminate the false positives. 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. To summarize, using machine learning techniques with active site surrounding information resulted in the best annotation performance. Model-based methods, in this case, did not perform well among other proposed methods. In this study, all initial patterns were obtained from the literature/database. For our future works, we can first make use of subgraph mining tools 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.

Biography

Seak Fei Lei received his Master degree in Computer Science from the University of Kansas under supervision of Dr. Jun Huan.

Dr. Jun Huan has been an assistant professor in the Electrical Engineering and Computer Science department at the University of Kansas since 2006. He received his Ph.D. in Computer Science from the University of North Carolina at Chapel Hill in 2006. Before joining KU, he worked at Argonne National Laboratory (with Ross Overbeek), Glaxo-SmithKline Inc. (with Nicolas Guex), and Nortel Networks. His research interests include Bioinformatics and Data Mining. He is a recipient of the Scholar of Tomorrow Fellowship and the Alumni Fellowship from the University of North Carolina at Chapel Hill.

References

Berger, B.; Singh, M. An iterative method for improved protein structural motif recognition; RECOMB ’97: Proceedings of the first annual international conference on Computational molecular biology; 1997; p. 37-46.

Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE. The Protein Data Bank. Nucl. Acids Res 2000;28(1):235–242. **URL:**http://nar.oxfordjournals.org/cgi/content/abstract/28/1/235. [PubMed: 10592235]

Chen, Brian Y.; Fofanov, Viacheslav Y. 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; D. H. B.

Chang, C-C.; Lin, C-J. LIBSVM: a library for support vector machines. 2001.

Chen, BY.; Bryant, DH.; Fofanov, VY.; Kristensen, DM.; Cruess, AE.; Kimmel, M.; Lichtarge, O.; Kavraki, LE. Cavity-aware motifs reduce false positives in protein function prediction; Comput Syst Bioinformatics Conf.; 2006;

Fan C, Moews P, Walsh C, Knox J. Vancomycin resistance: structure of D-alanine:D-alanine ligase at 2.3 A resolution. Science 1994;266:439–443. [PubMed: 7939684]

Hammersley, J.; Clifford, P. Markov fields on finite graphs and lattices. 1971. Unpublished manuscript

Hanukoglu I, Gutfinger T. cDNA sequence of adrenodoxin reductase. Identification of NADP-binding sites in oxidoreductases. Eur. J. Biochem 1989;180:479–484. [PubMed: 2924777]

Henikoff S, Henikoff J. Amino Acid Substitution Matrices from Protein Blocks. Proceedings of the National Academy of Sciences 1992;89(22):10915–10919.

Huan, J.; Bandyopadhyay, D.; Prins, J.; Snoeyink, J.; Tropsha, A.; Wang, W. Distance-based identification of structure motifs in proteins using constrained frequent subgraph mining; Computational systems bioinformatics / Life Sciences Society.Computational Systems Bioinformatics Conference; 2006; p. 227-238.

Kim SH, D SH, Choi IG, Ursula SG, Chen SF, Kim R. Structure-based functional inference in structural genomics. Journal of Structural and Functional Genomics 2003;4(2/3):129–135. [PubMed: 14649297]

Kolinski A, Betancourt M, Kihara D, Rotkiewicz P, Skolnick J. Generalized comparative modeling (genecomp): A combination of sequence comparison, threading, and lattice modeling for protein structure prediction and refinement. Proteins 2001;44(2)

Martin AC. PDBSprotEC: a Web-accessible database linking PDB chains to EC numbers via SwissProt. Bioinformatics 2004;20(6):986–988. [PubMed: 14764547]

R L. Convergence of the iterative proportional fitting procedure. The Annals of Statistics 1995;23(4): 1160–1174.

Shah A, Oehmen C, Webb-Robertson B. SVM-HUSTLE-an iterative semi-supervised machine learning approach for pairwise protein remote homology detection. Bioinformatics 2008;24:783–790. [PubMed: 18245127]

Ullman JR. An algorithm for subgraph isomorphism. Journal of the Association for Computing Machinery 1976;23:31–42.

Wang, C.; Parthasarathy, S. Summarizing itemset patterns using probabilistic models; KDD ’06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining; 2006; p. 730-735.

Wang G, Dunbrack RL. PISCES: a protein sequence culling server. Bioinformatics 2003;19:1589–1591. [PubMed: 12912846]

Xiao Y, Segal M. Biological sequence classification utilizing positive and unlabeled data. Bioinformatics 2008;24:1198–1205. [PubMed: 18344247]

Yan, X.; Cheng, H.; Han, J.; Xin, D. Summarizing itemset patterns: a profile-based approach; KDD ’05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining; 2005; p. 314-323.

Zagrovic B, Pande V. How does averaging affect protein structure comparison on the ensemble level? Biophys. J 2004;87:2240–2246. [PubMed: 15454426]

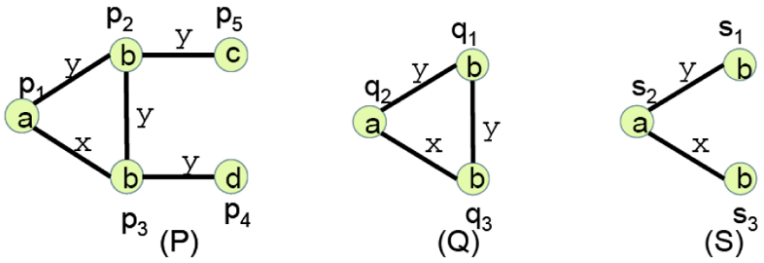

Figure 1.
Examples of labeled graphs. In this paper, protein/acitve site structures are modeled using labeled graph. Node labels such as a; b; c; d represent amino acids, and the edge labels like x; y are the Euclidean distance between two nodes.

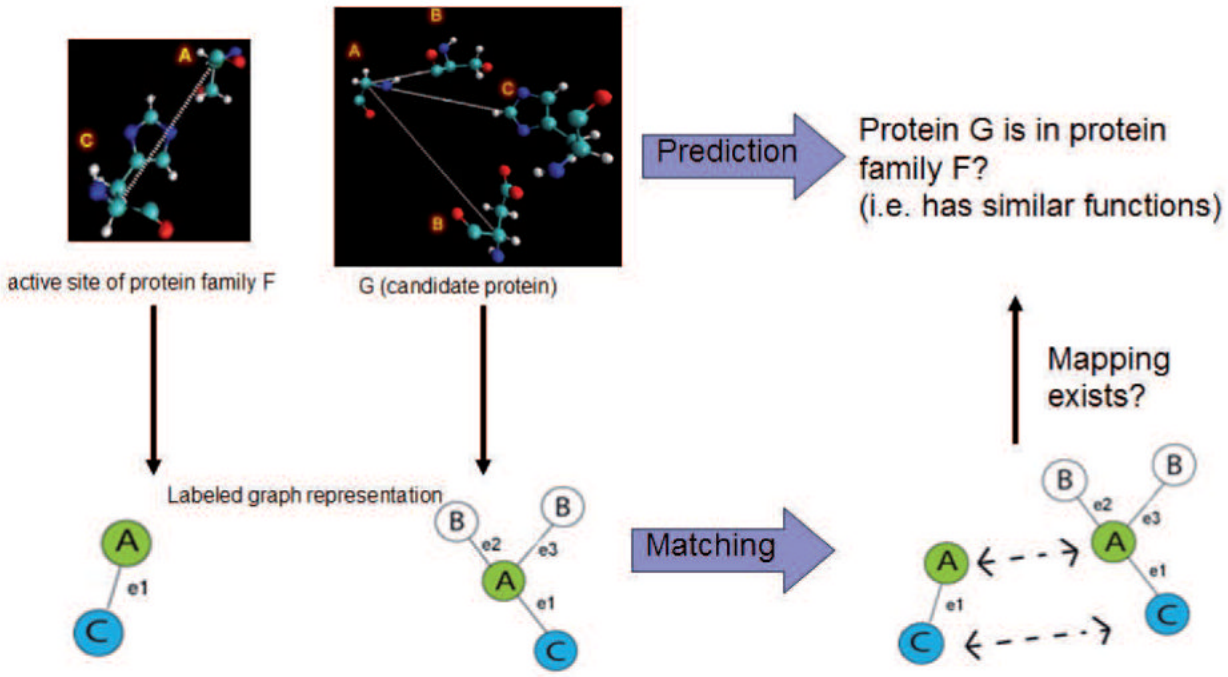

Figure 2.
Breakdown of graph-based protein function annotation

Figure 3.
Flowchart of motif refinement algorithm.

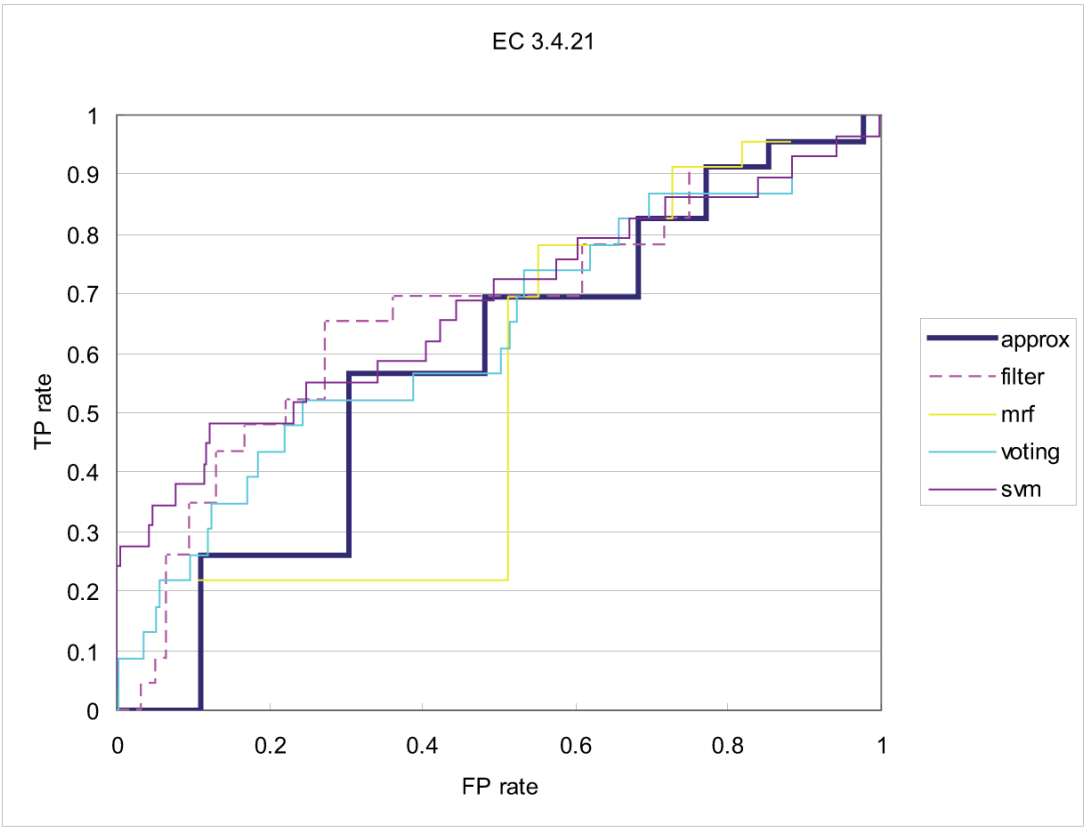

Figure 4.
The ROC analysis of EC 3.4.21 using five proposed methods.

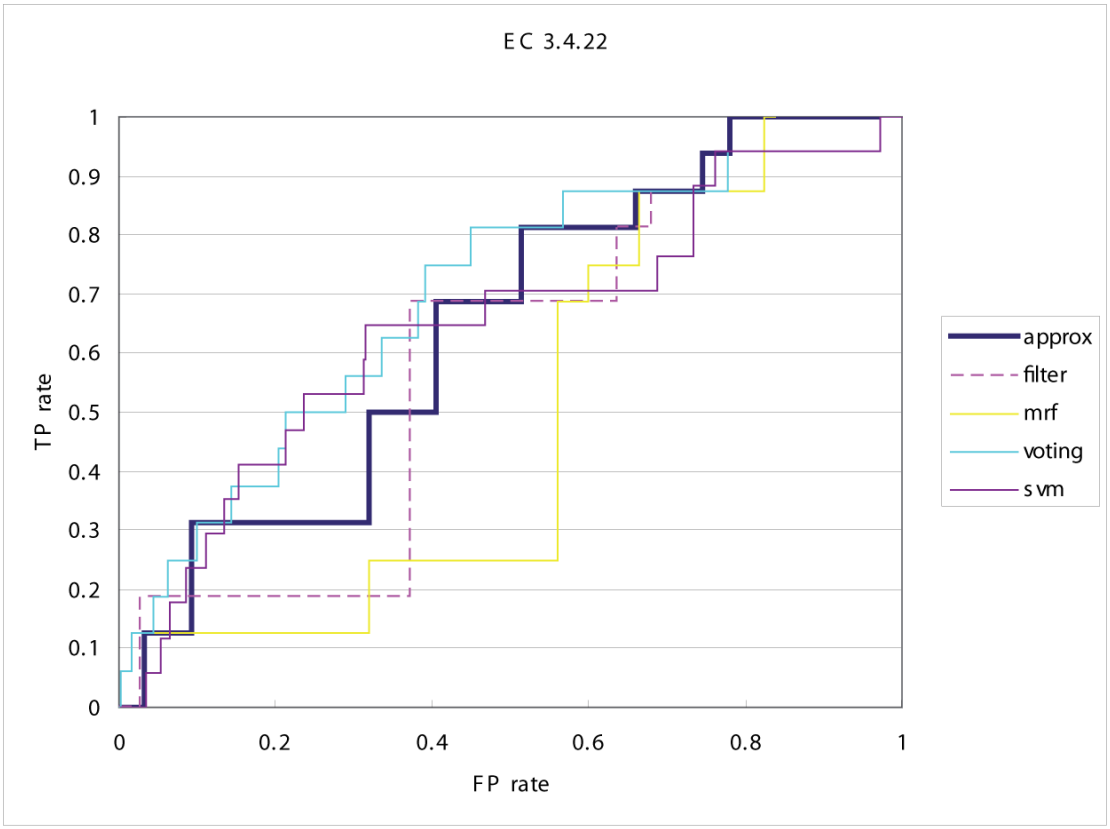

Figure 5.
ROC analysis of enzyme family EC 3.2.22.

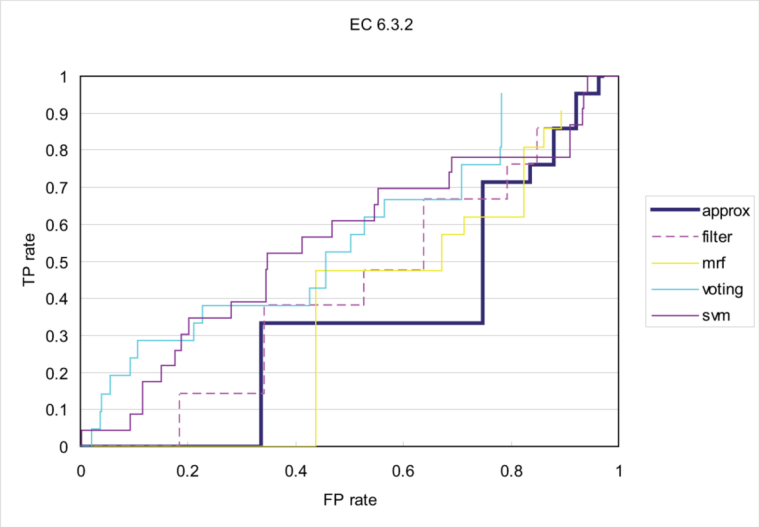

Figure 6.
ROC analysis of enzyme family EC 6.3.2.

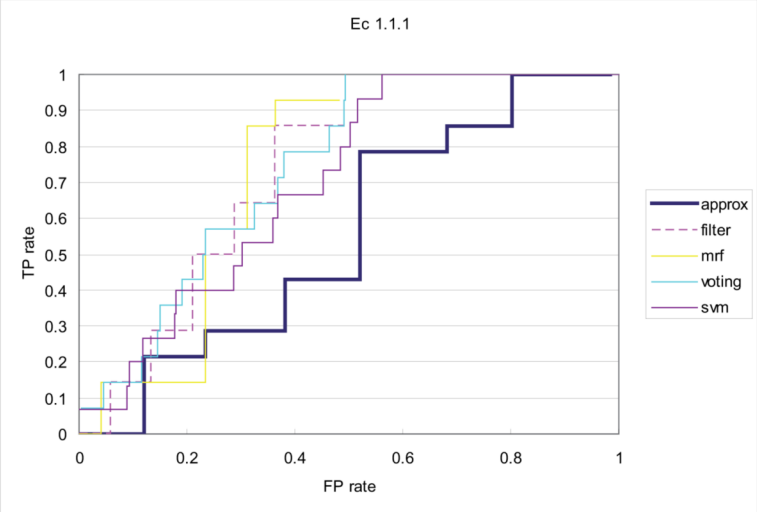

Figure 7.
ROC analysis of enzyme family EC 1.1.1.

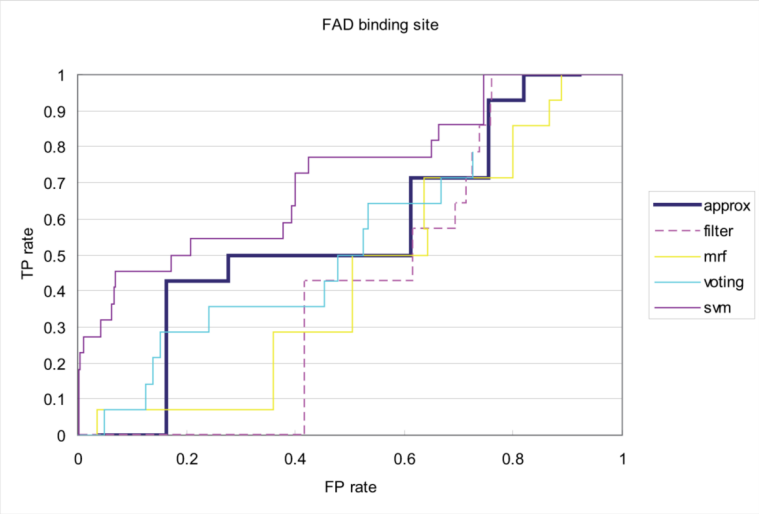

Figure 8.
ROC analysis of FAD binding families (1.8.1 + 1.18.1).

Table 1

Enzyme families used in this experiment. The *source* column indicates where initial motifs are obtained (either from Catalytic Site Atlas [CSA] or from literature)

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 et al. (1994)
1.1.1	ASP201-ARG204-HIS229 (7mdha)	CSA
FAD binding (1.8.1+1.18.1)	GLY11-GLY13-GLY16-ALA20 (1q1ra)	Hanukoglu & Gutfinger (1989)

Table 2

Dataset statistics after preprocessing

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 3

The list of the extended motifs for different EC families, which are used for the voting method and the feature vector method. The bold residues indicate some additional nodes added to the initial motifs. Note that even though the additional nodes have the same type (e.g. both the second and third extended motifs in 3.4.21 include CYS residue), their edge labels can still be diRerent. As a result, their approximate matching results will vary

EC number	Query protein	Extended motifs
3.4.21	1mcta	**ASN95**-HIS57-GLY193-SER195
		CYS42-HIS57-GLY-193-SER195
		CYS58-HIS57-GLY-193-SER195
		ILE212-HIS57-GLY-193-SER195
3.4.22	1pppa	**ALA160**-CYS25-HIS159-ASN175
		SER29-CYS25-HIS159-ASN175
		VAL157-CYS25-HIS159-ASN175
		ALA27-CYS25-HIS159-ASN175
6.3.2	2dlna	**SER19**-GLU15-SER150-GLY276
		HIS63-GLU15-SER150-GLY276
		LEU62-GLU15-SER150-GLY276
		THR278-GLU15-SER150-GLY276
1.1.1	7mdha	**LEU200**-ASP201-ARG204-HIS229
		VAL265-ASP201-ARG204-HIS229
		GLY227-ASP201-ARG204-HIS229
		ASN173-ASP201-ARG204-HIS229
FAD binding (1.8.1 + 1.18.1)	1q1ra	**HIS43**-GLY11-GLY13-GLY16-ALA20
		PRO42-GLY11-GLY13-GLY16-ALA20
		ALA38-GLY11-GLY13-GLY16-ALA20
		GLY111-GLY11-GLY13-GLY16-ALA20

Table 4

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

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**