Functional Site Prediction by Exploiting Correlations between Labels of Interacting Residues

Saradindu Kar
Ericsson Research India
Chennai-600 016, India
saradindu.kar@ericsson.com

Deepak Vijayakeerthi
Department of Computer Science and Engineering
IIT Madras
Chennai-600 036, India
vdeepak@cse.iitm.ac.in

Balaraman Ravindran
Department of Computer Science and Engineering
IIT Madras
Chennai-600 036, India
ravi@cse.iitm.ac.in

Ashish V. Tendulkar
School of Technology and Computer Science
Tata Institute of Fundamental Research
Mumbai-400 005, India
ashishvt@tifr.res.in

ABSTRACT

Functional site prediction is an important problem in the structural genomics era where we have a large number of experimentally determined protein structures with unknown function. The functional sites provide useful insights into protein function. In this paper, we propose a method for prediction of functional residues in a given protein from its three-dimensional (3D) structure. Our method exploits correlation between labels of interacting residues to obtain significant performance improvements over the existing methods on the benchmark dataset. We represent each protein as a weighted undirected residue interaction network, where spatially proximal residues in terms of their Van der Waal radii are connected by an edge. The edge weight captures correlation between the labels of interacting residues. The correlation is estimated based on the features of interacting residues. We then obtain a label assignment by minimizing combined cost of residue-wise label misclassification and violation of label correlation constraints. We solve this problem in two stages, where the first stage minimizes residue-wise label misclassification cost followed by an iterative collective inference scheme that adjusts the labels predicted in the first stage so as to minimize the correlation constraint violations. Our approach significantly outperforms state of the art methods on standard benchmark dataset. It achieves 23.06% precision at 69% recall and 87.78% recall at 18% precision, which translates to an improvement of 5.06 percentage points in the precision at 69% recall and 18.78 percentage point improvement in recall at 18% precision.

ACM-BCB '12, October 7-10, 2012, Orlando, FL, USA

Categories and Subject Descriptors

J.3 [**Life and Medical Sciences**]: Biology and genetics; I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms

Keywords

Collective Inference, Functional site prediction, L_1 regularized logistic regression, Protein structure

1. INTRODUCTION

Proteins play a vital role in cellular functions of living organisms. Amino acids are building blocks of proteins. The amino acid sequence of the protein determines its three dimensional structure, which in turn determines its function. A small number of spatially proximal amino acid residues (typically between three to six) are directly involved in protein function. Such residues are known as functional residues. These residues interact with one another and form an entity called functional site. There may be one or more functional sites in a protein depending on its functionality. Knowledge of functional sites is essential for understanding mechanism of protein function and discovery of effective drugs. The functional sites are determined via experimental techniques in which biologists have to evaluate enormous number of potential sites. Clearly such an exercise is infeasible due to time and resource constraints. Hence computational methods are required for obtaining a small number of high quality leads from a large number of potential sites. These methods are highly relevant in the current situation; where worldwide structural genomic projects are producing a large number of novel structures with unknown function and functional sites.

A popular way of identifying functional residues is based on the conservation scores of related sequences. Though this method holds well in theory, it doesn't perform well in practice. One of the main disadvantages of this technique

is that, even after selecting related sequences, which by itself is a herculean task, the conservation method returns too many false positives [9]. Even though various methods have been developed in the past for addressing the problem, they struggle to achieve substantial performance. This reflects the challenging nature of the problem and calls for sophisticated techniques to handle it. We refer the readers to a survey by Xin et al. [38] for a detailed account of these methods.

In this paper, we present a new method for functional residue prediction using three dimensional (3D) structure information. Our method takes 3D structure of protein as an input and returns a list of functional residues. Formally, the problem can be stated as follows: Given a protein three dimensional (3D) structure, $\mathcal{S}$, containing n amino acid residues $\langle a_1, a_2, \ldots, a_n \rangle$, label each residue as either functional or non-functional. This binary classification problem has been addressed by many different methods in literature [38]. We will review these methods in section 2. The problem is challenging due to extremely skewed distribution of functional and non-functional residues in any given protein.

In the past, Machine Learning(ML) techniques have shown promising results for this problem. These methods classify residues into functional and non-functional classes based on their structural, physicochemical, evolutionary and electrostatic features. The following are some of the examples of ML techniques used for addressing the problem: Wei et al. [37] used naïve Bayes classifier, Gutteridge et al. [11] used neural network, Youn et al. [40] used support vector machines (SVMs) and Sankararaman et al. [30] used L_1 logistic regression classifier. Among all the existing methods, the L_1 logistic regression classifier achieves the best performance on standard benchmark datasets. The L_1 regularization enabled them to overcome the problem of overfitting, which adversely affects the performance of ML techniques due to lack of availability of large number of training examples. On CATRES-FAM benchmark [30], it achieves 18% precision at 69% recall.

Traditional machine learning techniques do not exploit the inherent structure (correlation between the labels of interacting residues) in the problem, since they assume the data to be i.i.d. (independent and identically distributed). Collective classification is one of the popular methods which extends the traditional machine learning techniques to exploit the structure in the problem by jointly classifying the related instances. We propose a *cautious collective inference* scheme that exploits the inherent structure in the problem to provide a better performance than the existing techniques. Of the existing techniques the closest to that of ours is the usage of CRFs [30] to predict the functional residues. CRF is one of the popular collective classification models. Our work's major difference is that we use a cautious scheme, which helps us to prevent cascading errors due to misclassifications.

Given a 3D structure of a protein, we construct a Residue Interaction Network (RIN), where nodes represent amino-acid residues and the edges capture the interaction among them. An edge is added between two nodes, when the corresponding residues are spatially proximal to each other based on their Van der Waal radii. Our method then obtains a label assignment for a given residue interaction network by minimizing the combined cost of residue-wise label misclassification and violations of interaction polarity constraints. The problem is solved in two stages. The first stage involves the use of statistical models to predict the label of each residue and the polarity of interaction. **Polarity of interaction**, is either positive (+1, when the labels of the interacting residues are same) or negative (-1, when the labels of the interacting residues are different). Polarity of interaction helps us to capture the correlation among the labels. The second stage uses a cautious collective inference scheme, to obtain a label assignment by adjusting the labels of the residues to satisfy the polarity constraints. The cautious scheme prevents the cascading errors due to misclassifications by allowing to update the label of a residue only when the model is highly confident of its prediction.

Organization: Section II gives detailed account of the existing functional residue prediction techniques. The section III describes problem formulation and the proposed strategy. Section IV describes results on a benchmark dataset and comparison with state of the art techniques. Section V discusses the salient features of the method and the improvement it provides over the existing methods.

2. RELATED WORK

Several methods have been proposed in literature to address the problem [38]. Broadly these methods fall into two classes namely (i) template based methods; and (ii) residue based methods.

The **template based methods** search for a user defined three dimensional pattern of amino acids or their properties in the protein structure. The amino acid residues from the matching constellation are labeled as functional residues. The user defined templates are specified in various forms. For example, TEmplate Search and Superposition (TESS) method uses a reference frame based on amino acid side chains and positions of atoms in the vicinity as specified by a distance threshold [34]. Gregory et al. [10], Wallace et al. [35], Russell [29], Camer et al. [5] define templates using a set of inter-residue or inter-atomic distance between functional residues. The templates are also defined using physicochemical properties of local structural neighborhood [32, 13]. PDBSiteScan [14] uses PDB SITE record to define templates. Several methods have been proposed for automatic identification of templates from a set of related structures [38]. For example, SuMo uses triangle of chemical groups to represent protein structures [16]. GASPS employs genetic algorithm strategy to create templates consisting of 3-10 conserved residues in a protein family [27]. DReSPat models each protein structure as a graph. It then applied graph theoretic algorithms to enumerates potential functional site templates consisting 3 to 6 residues [36]. It discovers functional sites by clustering the templates and selecting the ones that recur in proteins performing similar function. The template matching is performed via superposition transformations or geometric hashing [24, 8]. DReSPat proposed an efficient scheme for template matching using a set of geometric invariant descriptors [33].

The **residue based methods** usually learn a model to identify functional residues from a set of training examples. The model is based on structural, evolutionary, electrostatic and physicochemical features of a residue or its structural neighborhood. The evolutionary conservation property of residues is exploited by several methods like Evolutionary Trace [21], Aloy et al. [1], Consurf [3], PHUNCTIONER [26]

etc. The idea of conservation is also extended to the structural neighborhood of amino acids in order to predict functional residues [25, 19]. Supervised machine learning techniques like naïve Bayes classifier and support vector machines have been applied to predict functional residues from their features mentioned earlier. FEATURE [37], WebFEATURE [20], S-Blast [22], Xin et al. [39], Bhardwaj et al. [4], Discern [30] are a few examples of methods which apply machine learning techniques for addressing the problem. Amitai et al. [2] use network centrality features derived from a network of residue interactions in a given protein to identify functional sites.

3. METHODS

In this section, we describe the proposed approach in detail. First we will describe the problem formulation followed by details of each stages used in solving the problem.

3.1 Problem Formulation

The proposed method predicts functional residues for a given protein structure $\mathcal{S}$. We represent $\mathcal{S}$ as a weighted undirected residue interaction network (RIN) $\mathcal{G}$, where nodes represent amino-acid residues and the edges capture the interaction among them. Thus, $\mathcal{G}$ has n nodes corresponding to n amino acid residues in $\mathcal{S}$. Let $A = \{a_1, a_2, \ldots, a_n\}$ be the set of nodes, where a_i is amino acid residue at i–th position in protein sequence. We add an edge between a_i and a_j if they are spatially proximal to each other based on their Van der Waal radii. We represent the contact information in a $n \times n$ matrix $\mathbf{E}$, whose (i,j)–th entry e_{ij} is as follows:

$$e_{ij} = \begin{cases} 1 & \text{if } a_i \text{ and } a_j \text{ are in contact} \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Note that a_i and a_j can be in contact without being close in the sequence. Each residue $a_i \in A$ takes a label y_i as follows:

$$y_i = \begin{cases} +1 & \text{if } a_i \text{ is a functional residue} \\ -1 & \text{otherwise} \end{cases}$$

Let $\mathbf{y} \in \{+1, -1\}^n$ be the label vector. The i–th component of $\mathbf{y}$ corresponds to label y_i of i–th residue a_i. Let $w_{ij} \in \{+1, -1\}$ be the polarity of interaction between any two residues in contact. Let $\mathbf{W}$ be $n \times n$ matrix, whose (i,j)–th entry stores w_{ij}. We will refer to $\mathbf{W}$ as an interaction polarity matrix.

$$w_{ij} = \begin{cases} +1 & y_i = y_j \text{ and } e_{ij} = 1 \\ -1 & y_i \neq y_j \text{ and } e_{ij} = 1 \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

We are interested in predicting a label vector $\hat{\mathbf{y}}$ for $\mathcal{G}$ that not only has minimum residue-wise misclassification cost but also violates as few polarity constraints as possible. Minimizing only the residue-wise misclassification cost is equivalent to applying any traditional machine learning technique that does not exploit the correlation among the residues. On the other hand, minimizing only the number of violations in the polarity constraints alone is not sufficient as it will lead to a label vector that satisfies these constraints without minimizing residue wise misclassification cost. For example, the interaction polarity constraint $\tilde{w}_{ij} = -1$ is satisfied by the following two label configurations: (i) $\tilde{y}_i = -1$ and $\tilde{y}_j = +1$ and (ii) $\tilde{y}_i = +1$ and $\tilde{y}_j = -1$, with only one of them being correct. The residue wise label misclassification cost helps to overcome this limitation in order to achieve accurate predictions. Since the actual label correlations are not available, its estimated based on the features of interacting residues. Let $\widehat{\mathbf{W}}$ be the estimated interaction polarity matrix.

We obtain the label assignment using a two stage approach where the first stage focuses on minimization of residue-wise label misclassification cost. It uses L_1 regularized logistic regression classifier that predicts a label vector $\hat{\mathbf{y}}$. We will refer to this classifier as a local classifier in the subsequent text. According to the local classifier, a_i is a functional residue if $\hat{y}_i = +1$ and is non-functional otherwise. The second stage uses an iterative collective inference algorithm to obtain a label assignment by adjusting $\hat{y}_i \in \hat{\mathbf{y}}$ in order to satisfy the polarity constraints specified in $\widehat{\mathbf{W}}$. The labels $\hat{y}_i \in \hat{\mathbf{y}}$ are carefully adjusted in order to avoid significant escalation in the label misclassification cost, which was minimized by the local classifier. Upon convergence, the iterative scheme returns a label vector $\tilde{\mathbf{y}}$. We provide details of both the stages in the next section.

3.2 Functional residue prediction by collective inference

The broad steps in our approach for functional residue prediction are shown in figure 1 and the scheme is listed in *Algorithm 1*. Here we provide an overview of the steps involved in the process leaving out the details to the subsequent subsections.

Algorithm 1 Algorithm for prediction of functional residues

Input: Protein structure $\mathcal{S}$
Output: Label vector $\tilde{\mathbf{y}}$

1: $\mathcal{G} \leftarrow$ formRIN($\mathcal{S}$)
2: $\vartheta \leftarrow$ applyInteractionPolarityClassifier($\mathbf{X_{ij}}; \theta$)
3: $\varphi \leftarrow$ applyLocalClassifier($\mathbf{X}; \rho$)
4: $\tilde{\mathbf{y}} \leftarrow$ applyCollectiveInference $(\varphi, \vartheta, \mathcal{G}, \alpha, \beta, \gamma)$

First of all, we construct a RIN $\mathcal{G}$ for the input 3D structure $\mathcal{S}$ as described in the previous section (line – 1). Then, we estimate the interaction polarity matrix $\widehat{\mathbf{W}}$ based on features of the interacting residues. We use L_1-regularized logistic regression classifier parametrized by θ to obtain probability of each $\hat{w}_{ij}$ being $+1$. Let $\mathbf{x_{ij}}$ be the feature vector constructed by concatenating features of a pair of interacting residues a_i and a_j, then $\vartheta_{ij} = \Pr(\hat{w}_{ij} = +1 | \mathbf{x_{ij}}; \theta)$. Let ϑ be the matrix containing ϑ_{ij} for all the interacting pairs. (line – 2). Similarly we use L_1 regularized logistic regression parametrized by ρ as a local classifier to estimate probability of each residue being functional. Let φ be the label probability vector and $\varphi_i = \Pr(\hat{y} = +1 | \mathbf{x_i}; \rho)$ is the probability corresponding to residue a_i being functional (line – 3). Finally, we use an iterative collective inference scheme to obtain $\tilde{y}$ (line – 4). It has three tunable parameters namely polarity threshold α, local classifier threshold β and message strength threshold γ. These parameters enable us to obtain label vector predictions of desired quality and also enable us to understand effects of individual components on the overall prediction accuracy. One of the main advantages of the proposed scheme is that it is independent of the classifier being used to estimate the labels of the residue and

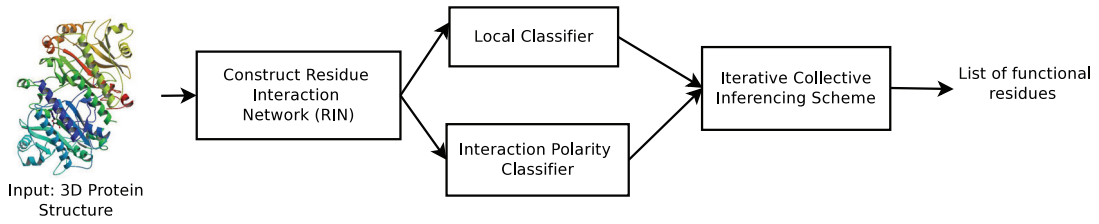

Figure 1: Flow-chart of the proposed scheme

the interactions polarities. We use a L_1 regularized logistic regression[18] as mentioned previously.

The performance of our method for functional residue prediction (Algorithm 1) is evaluated using the test set as follows: We use Algorithm 1 to obtain probability estimates for $+1$ class label for residues and polarity of interaction from the best local and interaction polarity classifier models. We obtain different models by varying all three tunable parameters α, β and γ of iterative collective inference algorithm. For each model, we obtain a precision-recall (PR) curve using class labels predicted by the model. We select the model with the best AUC-PR. We repeat the process for different test sets. Finally, we average the PR curves across different test sets and obtain PR curve for our method.

3.2.1 *Interaction Polarity Classifier*

In this subsection, we will describe an interaction polarity classifier, which estimates probability of a polarity of interaction being $+1$. Let $\hat{w}_{ij}$ be the predicted polarity of interaction between a residue pair a_i and a_j in the given structure and ϑ_{ij} be the corresponding probability. Let ϑ be the matrix that stores such probabilities for all interacting residue pairs. We estimate ϑ_{ij} from the features of residues a_i and a_j. Let $\mathbf{x_{ij}}$ be the augmented feature vector for the interacting pair. We will call it as interaction feature vector. Let $\text{cat}(\mathbf{x_i}, \mathbf{x_j})$ be the function that concatenates features vectors $\mathbf{x_i}$ and $\mathbf{x_j}$ in that order and returns a vector $\mathbf{x_{ij}} \in \mathbb{R}^{2k}$. We use $\text{cat}(\mathbf{x_i}, \mathbf{x_j})$ function to obtain interaction feature vector as follows:

$$\mathbf{x_{ij}} = \begin{cases} \text{cat}(\mathbf{x_i}, \mathbf{x_j}) & \text{if } i < j \text{ and } e_{ij} = 1 \\ \text{cat}(\mathbf{x_j}, \mathbf{x_i}) & \text{if } i > j \text{ and } e_{ij} = 1 \end{cases} \tag{3}$$

We predict ϑ_{ij} for an interaction vector $\mathbf{x_{ij}}$ using L_1 regularized logistic regression classifier.

$$\vartheta_{ij} = \Pr(\hat{w}_{ij} = +1 | \mathbf{x_{ij}}; \theta) = \frac{1}{1 + \exp(-\theta^{\mathrm{T}} \mathbf{x_{ij}})} \tag{4}$$

Note that $\Pr(\hat{w}_{ij} = -1 | \mathbf{x_{ij}}; \theta) = 1 - \vartheta_{ij}$, since the polarity can be either $+1$ or -1. The parameter vector $\theta \in \mathbb{R}^{2k+1}$ has $2k+1$ parameters. The first parameter θ_0 is a regularization term that controls tradeoff between false positives and false negatives. The remaining $2k$ parameters θ contain weights of the corresponding interaction features in $\mathbf{x_{ij}}$.

For each protein $\mathcal{S} \in S$ with n residues, there will be c interaction feature vectors, where $c = \sum_{1 \le i,j \le n} e_{ij}$. Let C be the set of ordered pairs of interaction feature vectors and the corresponding polarity $\{(\mathbf{x}^l_{ij}, w^l_{ij})\}_{l=1}^{c}$ for $\mathcal{S}$. We estimate the parameters of the model (weights) by **5-fold cross validation**. In order to choose the best polarity classifiers among five models obtained from the cross-validation procedure, we plot receiver operating characteristics (ROC) curve for each model after obtaining its specificity and sensitivity on proteins in the test set. For each model, we vary polarity threshold α on its ϑ estimates to obtain a prediction for interaction polarity.

$$\hat{w}_{ij} = \begin{cases} +1 & \text{if } \vartheta_{ij} \ge \alpha \\ -1 & \text{otherwise.} \end{cases} \tag{5}$$

Based on these predictions, we calculate sensitivity or recall and specificity as a fraction of number of true negatives (in this case number of -1 labels) and the sum of the number of true negatives and the number of false positives. We calculate area under ROC curve (AUC-ROC) for each model and select one with the best AUC-ROC. Such a model achieves the highest performance while predicting negative polarity of interaction which is a minority class in terms of distribution of the two classes. Note that more than 99% interactions in proteins have positive polarity. The best ROC curves across different test sets are averaged to obtain ROC curve for the interaction polarity classifier. The polarity threshold α helps us to handle the issue of skewness in the class distribution by allowing a flexible choice of decision thresholds.

3.2.2 *Local Classifier*

The local classifier takes a protein structure $\mathcal{S}$ as an input and predicts a class label for each residue a_i in the structure. We use a logistic regression classifier to estimate probability of each residue being functional φ_i.

$$\varphi_i = \Pr(\hat{y}_i = +1 \mid \mathbf{x_i}; \rho) = \frac{1}{1 + \exp(-\rho^{\mathrm{T}} \mathbf{x_i})} \tag{6}$$

where $\rho \in \mathbb{R}^{k+1}$ is a parameter vector that whose components from ρ_1 to ρ_k contains weights for the corresponding residue features, while ρ_0 is a bias term that controls the rate of false positives and false negatives. Note that we can calculate the probability of residue being non-functional as $1 - \varphi_i$, since a residue can be either functional or non-functional. One of the primary challenges in solving the problem is the skewness of the class distribution. The local classifier threshold β helps us to alleviate the issue by allowing a flexible choice of decision thresholds. We estimate the parameters of the model (weights) by **5-fold cross validation**. Each model estimates probability of each residue in proteins from the test set being functional. For each model, we obtain a precision-recall (PR) curve by varying local classifier threshold β from 0 to 1.

$$\hat{y}_i = \begin{cases} +1 & \text{if } \varphi_i \ge \beta \\ -1 & \text{otherwise.} \end{cases} \tag{7}$$

At each value of probability threshold, we compute precision $= \frac{TP}{\text{TP+FP}}$ and recall $= \frac{TP}{\text{TP+FN}}$, measures. Here TP is the number of residues whose class label is correctly predicted, FP is the number of residues predicted to be functional, but are

actually non-functional and FN is the number of residues predicted to be non-functional but are actually functional. We calculates area under PR curve (AUC-PR) of each model and select the models with the best values of AUC-PR. The PR-curves across different test sets are averaged to obtain PR-curve for the local classifier.

The local classifier used in our approach closely follows L_1-regularized logistic regression classifier used by DISCERN. The only difference is in the feature vector: while our classifier solely uses features of individual residue, the one employed by DISCERN operates on a feature vector constructed from features of individual residues along with its neighbours.

3.2.3 Iterative collective inference scheme

In this subsection, we will describe an iterative collective inference scheme. It operates on residue interaction network $\mathcal{G}$ of protein structure $\mathcal{S}$ in order to predict a label vector $\tilde{\mathbf{y}}$. The scheme also takes estimated label probability vector φ and estimated polarity probability matrix ϑ along with α, β and γ. Algorithm 2 provides stepwise account of the scheme.

We use polarity threshold α to obtain $\widehat{\mathbf{W}}$ (line – 1) entry-wise given by equation (5). We use local classifier threshold β to obtain $\hat{\mathbf{y}}$ (line – 2) given by equation (7). We initialize $\tilde{\mathbf{y}}$ to $\hat{\mathbf{y}}$ (line – 3) We define a message vector $\mathbf{q} \in \mathbb{R}^{|A|}$, where $|A|$ is the number of nodes in $\mathcal{G}$. The i–th component q_i stores a message of residue a_i that signifies fraction of positive votes received from its neighbours:

$$q_i = \frac{\sum_{\forall j: e_{ij}=1} \max(0, \hat{y}_j \times \widehat{W}_{ij})}{\sum_j e_{ij}} \tag{8}$$

The entire vector $\mathbf{q}$ can be calculated in one step as follows:

$$\mathbf{q} = \hat{\mathbf{y}}^{\mathbf{T}} \widehat{\mathbf{W}} \mathbf{o} \tag{9}$$

where $\mathbf{o}$ stores inverse of number of neighbors of nodes in G_i. Specifically,

$$o_i = \begin{cases} \frac{1}{\sum_{j=0}^{|A|} e_{ij}} & \text{if } \sum_{j=0}^{|A|} e_{ij} > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

We initialize $\mathbf{q}$ to 1. In each iteration, we update $\mathbf{q}$ with messages from the neighbouring nodes as per eq. (9) (line–5). The message vector $\mathbf{q}$ is used to update the label vector $\tilde{\mathbf{y}}$(line – 6). We take a *cautious* approach in updating the label vector $\tilde{\mathbf{y}}$, as misclassified instances will have an negative influence on the accuracy of the method due to cascading inference errors. We update the label $\tilde{y}_i$ only when the confidence of the prediction (q_i) is greater than the threshold γ (confidence factor).

$$\tilde{y}_i = \begin{cases} +1 & \text{if } q_i > \max(0.5, \gamma) \\ -1 & \text{if } q_i < 1 - \max(0.5, \gamma) \end{cases} \tag{11}$$

If the confidence of the prediction (q_i) is not sufficiently high, we either retain or flip the label $\tilde{y}_i$ based on the confidence of the local classifier (φ_i) on its prediction. The confidence is measured using a local classifier threshold β as given by equation (7). The iterative scheme terminates whenever $\mathbf{q}$ ceases to change. This is the point when it is no longer possible to satisfy polarity constraints without increasing labelwise misclassification cost. Upon convergence, the label vector $\tilde{y}$ contains predictions for each residue in $\mathcal{S}$.

Algorithm 2 Iterative collective inference scheme

Input: (i) Interaction network $\mathcal{G}$, (ii) Local classifier probability vector φ, (iii) Interaction polarity probability matrix ϑ, (iv) Polarity threshold α (v) Local classifier threshold β (vi) Message strength threshold γ
Output: Label vector $\tilde{\mathbf{y}}$

1: $\widehat{\mathbf{W}} \leftarrow$ getPolarity(ϑ, α)
2: $\hat{\mathbf{y}} \leftarrow$ getLabel(φ, β)
3: $\tilde{\mathbf{y}} \leftarrow \hat{\mathbf{y}}$
4: **repetir**
5: $\quad \mathbf{q} \leftarrow \hat{\mathbf{y}}^{\mathbf{T}} \widehat{\mathbf{W}} \mathbf{o}$
6: $\quad \tilde{\mathbf{y}} \leftarrow$ updateLabel $(\mathbf{q}, \hat{\mathbf{y}}, \varphi, \gamma)$
7: **hasta q** stops showing any change

4. RESULTS

4.1 Benchmark

We evaluated our method on CATRES-FAM benchmark dataset [30]. The dataset contains functional site annotations for 140 proteins. The annotations are derived from catalytic site atlas database [28]. The proteins in the dataset are non-redundant at SCOP [23] superfamily level. The dataset consists of 49180 amino acid residues, out of which 472 are labeled as functional and the rest are labeled as non-functional. Thus, merely 0.95% residues in the dataset are labeled as functional. The 3D structures for all the proteins are available from protein data bank[1]. The proteins are partitioned into five folds, each containing 28 proteins and used for five-fold cross-validation as described in section 3.2.

Each protein is converted into a RIN as described in section 3. As part of RIN generation, we calculate 48 features for each residue. These features were proposed by Discern [30] and their detailed description can be found in [30]. We will provide a brief description of these features here. The features can be broadly divided into three types: sequence conservation, amino acid properties and structure features. Three real-valued sequence conservation features were used: (i) GLOBAL-JS [6], (ii) INTREPID-JS [31]; and (iii) INTREPID-LO [31]. We use twenty three features based on amino acid properties. Out of these, we use twenty binary features to capture amino acid sidechain information. Only one of these features will be 1 for each residue. Three more binary features denote the group of the amino acid out of charged, polar and hydrophobic [2]. We use twenty-two structure based features. The twelve of these are real valued features and the rest are binary features. The real value features including B-factor, closeness centrality calculated from residue interaction network, five each relative and absolute solvent accessibility values calculated by NACCESS [12]. Out of ten binary features, seven features capture secondary structure type of the residue as defined in DSSP [17] and the remaining three features capture the location of residues in one of the three largest pockets on the surface.

4.2 Performance of our method

We evaluated the performance of our method as described

[1]http://www.rcsb.org/

[2]Charged amino acid includes {D, E, H, K, R}, polar includes {Q, T, S, N, C, Y} and hydrophobic includes {A, F, G, I, L, M, P, V, W}

in section 3.2. It is of interest to understand how different components in our algorithm impact the functional residue prediction task. First we will first present our results about performance of individual modules and then present our analysis of their impact, individual as well as collective, on the performance of iterative collective inference scheme.

The performance of the local classifier is measured using PR-AUC as described in section 3.2. We found that PR-AUC is maximized at $\beta = 0.5$. At this threshold, we obtain recall of 58% at 18% precision and precision of 14.38% at recall of 69%. The precision and recall of our local classifier is lower than Discern [30], which has precision of 18% at 69% recall. The drop in performance in our method is due to the fact that our local classifier solely rely on the features of the residue and does not take into account the features of its neighbors as in Discern [30]. We do not take into account the features of the neighbors as it has been shown that these are not useful in collective classification framework [7].

The performance of the compatibility classifier is measured in terms of ROC-AUC as described in section 3.2. We found that ROC-AUC is maximized at $\alpha = 0.1$. At this threshold, we obtain specificity of 88.5%.

The PR-curve of the collective inference scheme is shown figure 2. We obtain PR-curves for various values of tuning parameter α, β and γ. We analyzed number of iterations required for convergence of the iterative scheme and found that the iterative scheme converges in 15 to 20 iterations for most of the parameter settings. We found that the best PR-AUC is obtained for the following values of the tunable parameter: $\alpha = 0.1$, $\beta = 0.5$ and $\gamma = 0.9$. Under these parameter settings, our method achieves recall of 87.78% at 18% precision and precision of 23.06% at 69% recall. It is interesting to analyze the amount of improvement iterative collective scheme provides over the local classifier. Across different parameter settings, we observe that the iterative inference consistently improves performance of local classifier. At the best parameter settings mentioned earlier, we obtain improvement of 8.5 percentage point in precision (at 69% recall local classifier has precision of 14.38%) and almost 30 percentage point improvement in recall of the local classifier.

Now we will provide performance comparison of our method with the existing methods. Our method significantly outperforms existing methods for functional site prediction both in terms of precision and recall (Table 1). In comparison with the existing ML methods for functional site prediction, we obtained 18.78 percentage point improvement over relational classifier and CRF used by Sankararaman et al.[30], 39 percentage point over neural network based predictor [11] and 31 percentage point improvement over SVM based predictor [40] at 18% precision. The recall obtained by our method is 70 percentage point more than the methods using sequence conservation features alone [31, 3, 21]. INTREPID [31] has recall of 19%, ET [21] has recall of 2% [30] at the same level of precision. At 69% recall, we obtain precision of 23.06%, which is 5 percentage point more than Discern [30] and 15 percentage point more than Consurf [3] at the same level of recall [30]. In addition, we also compared our results by implementing iterative classification algorithm (ICA) [15]. It achieves 8% precision at 69% recall, which is 15 percentage point lesser than our method. The maximum precision achieved by ICA was 10% at 85% recall. ICA iteratively adjusts labels of residues using the labels of neighbors without taking into account the interaction polarity. The ICA results shows importance of using polarity of interaction while labeling label of a residue based on its neighbors. We reported performance numbers at specific levels of precision and recall in order to enable comparison with earlier methods.

Method	**Precision$_{69}$ (%)**	**Recall$_{18}$(%)**
Collective Inference	**23.06**	**87.78**
Discern [30]	18	69
CRF [30]	18	69
Local classifier	14.38	58

Table 1: Comparative performance of various classification techniques for predicting functional site residues in CATRES-FAM dataset. We have not included neural network [11] and support vector machine [40] predictors in this table as their precisions are not reported at 69%**.**

Finally, we analyze how well we can do using a perfect interaction polarity classifier. We conducted this experiment by using actual interaction polarity matrix $\mathbf{W}$ corresponding to the protein structure instead of $\widehat{\mathbf{W}}$ in iterative collective inferencing scheme. Using the best tuning parameters mentioned above, our method achieves 80% precision and 94.35% recall on CATRES-FAM benchmark data. The results indicate that we could profitably focus more effort on improving the polarity classifier.

5. DISCUSSION

In this paper, we have proposed a novel method for prediction of functional residues from a given 3D structure. Our method has the following contributions:

1. Unlike existing methods, our approach exploits correlation between labels of interaction residues in the form of interaction polarity constraints. These constraints act as regularizers forcing the labels of neighbouring residues to conform to the correlations typically observed. This helps our method to achieve significantly better performance than existing methods that obtain labels by minimizing only the residue-wise misclassification costs.

2. This is the first instance when the collective inference techniques have been applied for functional residue prediction problem. The collective inference techniques have been state of the art for many different problems in the domain of computer vision, web and network classification. The previous best method, Discern by Sankararaman et al. [30], use relational classifier that labels a residue based on its own features as well as features from its neighbors. The collective inference scheme used in this paper labels a residue based on its features, labels of its interaction partners in RIN and the polarity of their interaction. We do not take into account the features of the neighbors as it has been shown that these are not useful in collective classification framework [7].

3. We have provided two stage approach to solve the problem of predicting functional residues. It has got three main components namely interaction polarity classifier, local classifier and iterative collective inference

scheme. Since iterative collective inference scheme adjusts labels predicted by the local classifier iteratively in collective inference scheme, we get results that are consistently better than that of the local classifier. Our iterative collective inference scheme works well in practice and rapidly converges to steady state within 15 to 20 iterations across different values of tunable parameters.

4. Our method achieves significant improvement over the existing methods in terms of precision and recall on CATRES-FAM benchmark dataset.

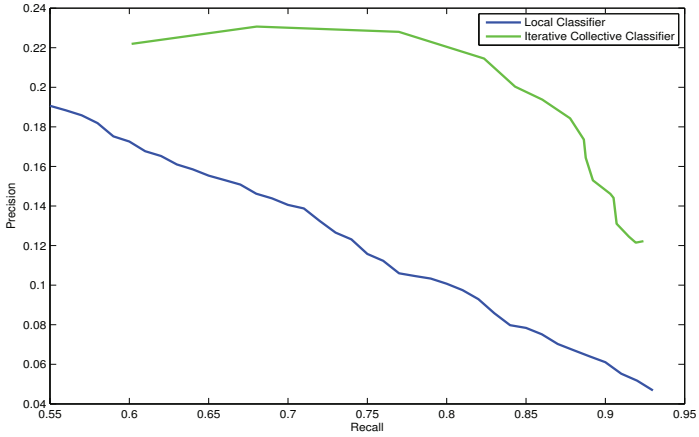

Figure 2: Precision-recall (PR) curve for iterative collective inferencing scheme. We have also shown PR curve of local classifier for the comparison purpose.

The proposed method can be applied for predicting functional residues in novel proteins since the method does not require information about homologous proteins.

6. ACKNOWLEDGMENTS

This work is partially supported by Innovative Young Biotechnologist Award grant from Department of Biotechnology of Government of India to Ashish Tendulkar.

7. REFERENCES

[1] P. Aloy, E. Querol, F. X. Aviles, and M. J. E. Sternberg. Automated structure-based prediction of functional sites in proteins: applications to assessing the validity of inheriting protein function from homology in genome annotation and to protein docking. *Journal of Molecular Biology*, 311(2):395 – 408, 2001.

[2] Gil Amitai, Arye Shemesh, Einat Sitbon, Maxim Shklar, Dvir Netanely, Ilya Venger, and Shmuel Pietrokovski. Network analysis of protein structures identifies functional residues. *Journal of Molecular Biology*, 344(4):1135 – 1146, 2004.

[3] A. Armon, D. Graur, and N. Ben-Tal. Consurf: an algorithmic tool for the identification of functional regions in proteins by surface mapping of phylogenetic information. *Journal of Molecular Biology*, 307(1):447 – 463, 2001.

[4] N. Bhardwaj, R. Langlois, G. Zhao, and Lu. H. Structure based prediction of binding residues on dna-binding proteins. In *Conference Proceedings of IEEE Engineering in Medicine and Biology Society*, pages 2611–2614, 2005.

[5] Stephen A. Cammer, Brian T. Hoffman, Jeffrey A. Speir, Mary A. Canady, Melanie R. Nelson, Stacy Knutson, Marijo Gallina, Susan M. Baxter, and Jacquelyn S. Fetrow. Structure-based active site profiles for genome analysis and functional family subclassification. *Journal of Molecular Biology*, 334(3):387 – 401, 2003.

[6] John A. Capra and Mona Singh. Predicting functionally important residues from sequence conservation. *Bioinformatics*, 23:1875–1882, August 2007.

[7] Soumen Chakrabarti, Byron E. Dom, and Piotr Indyk. Enhanced hypertext categorization using hyperlinks. In *International Conference of Management of Data*, volume 27, pages 307–318, 1998.

[8] Daniel Fischer, Haim Wolfson, Shuo L. Lin, and Ruth Nussinov. Three-dimensional, sequence order-independent structural comparison of a serine protease against the crystallographic database reveals active site similarities: Potential implications to evolution and to protein folding. *Protein Science*, 3(5):769–778, 1994.

[9] Richard A. George, Ruth V. Spriggs, Gail J. Bartlett, Alex Gutteridge, Malcolm W. MacArthur, Craig T. Porter, Bissan Al-Lazikani, Janet M. Thornton, and Mark B. Swindells. Effective function annotation through catalytic residue conservation. *Proceedings of the National Academy of Sciences of the United States of America*, 102(35):12299–12304, August 2005.

[10] David S. Gregory, Andrew C.R. Martin, Janet C. Cheetham, and Anthony R. Rees. The prediction and characterization of metal binding sites in proteins. *Protein Engineering*, 6(1):29–35, 1993.

[11] Alex Gutteridge, Gail J. Bartlett, and Janet M. Thornton. Using a neural network and spatial clustering to predict the location of active sites in enzymes. *J Mol Biol*, 330:719–734, 2003.

[12] S. Hubbard and J. Thornton. A computer algorithm to calculate surface accessibility. In *Department of Biochemistry and Molecular Biology, University College, London*, 1993.

[13] C.Axel Innis, A.Prem Anand, and R. Sowdhamini. Prediction of functional sites in proteins using conserved functional group analysis. *Journal of Molecular Biology*, 337(4):1053 – 1068, 2004.

[14] Vladimir A Ivanisenko, Sergey S Pintus, Dmitry A Grigorovich, and Nickolay A Kolchanov. Pdbsite: a database of the 3d structure of protein functional sites. *Nucleic Acids Research*, 33(Database Issue):D183–D187, 2005.

[15] D. Jensen J. Neville. Iterative classification in relational data. In *In Proc. AAAI*, pages 13–20. AAAI Press, 2000.

[16] Martin Jambon, Anne Imberty, Gilbert DelÃl'age, and Christophe Geourjon. A new bioinformatic approach

to detect common 3d sites in protein structures. *Proteins: Structure, Function, and Bioinformatics*, 52(2):137–145, 2003.

[17] W Kabsch and C Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637, 1983.

[18] Kwangmoo Koh, Seung-Jean Kim, and Stephen Boyd. A method for large-scale l1-regularized logistic regression. In *Proceedings of the 22nd national conference on Artificial intelligence - Volume 1*, pages 565–571. AAAI Press, 2007.

[19] R. Landgraf, I. Xenarios, and D. Eisenberg. Three-dimensional cluster analysis identifies interfaces and functional residue clusters in proteins. *Journal of Molecular Biology*, 307(5):1487 – 1502, 2001.

[20] M. P. Liang, D. R. Banatao, T. E. Klein, D. L. Brutlag, and R. B. Altman. Webfeature: an interactive web tool for identifying and visualizing functional sites on macromolecular structures. *Nucleic Acids Research*, 31(13):3324–3327, 2003.

[21] O. Lichtarge. An evolutionary trace method defines binding surfaces common to protein families. *Journal of Molecular Biology*, 257(2):342–358, 1996.

[22] S. D. Mooney, M. H. Liang, R. DeConde, and R. B. Altman. Structural characterization of proteins using residue environments. *Proteins: Structure, Function, and Bioinformatics*, 61(4):741–747, 2005.

[23] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *Journal of molecular biology*, 247(4):536–540, 1995.

[24] R Nussinov and H J Wolfson. Efficient detection of three-dimensional structural motifs in biological macromolecules by computer vision techniques. *Proceedings of the National Academy of Sciences of the United States of America*, 88(23):10495–10499, 1991.

[25] Anna R. Panchenko, Fyodor Kondrashov, and Stephen Bryant. Prediction of functional sites by analysis of sequence and structure conservation. *Protein Science*, 13(4):884–892, 2004.

[26] F. Pazos and M. J. E. Sternberg. Automated prediction of protein function and detection of functional sites from structure. *Proceedings of the National Academy of Sciences of the United States of America*, 101(41):14754–14759, 2004.

[27] Benjamin J. Polacco and Patricia C. Babbitt. Automated discovery of 3d motifs for protein function annotation. *Bioinformatics*, 22(6):723–730, 2006.

[28] Craig T. Porter, Gail J. Bartlett, and Janet M. Thornton. The catalytic site atlas: a resource of catalytic sites and residues identified in enzymes using structural data. *Nucleic Acids Research*, 32(Database-Issue):129–133, 2004.

[29] R. B. Russell. Detection of protein three-dimensional side-chain patterns: new examples of convergent evolution. *Journal of Molecular Biology*, 279(5):1211 – 1227, 1998.

[30] Sriram Sankararaman, Fei Sha, Jack F. Kirsch, Michael I. Jordan, and Kimmen Sjölander. Active site prediction using evolutionary and structural information. *Bioinformatics*, 26(5):617–624, 2010.

[31] Sriram Sankararaman and Kimmen Sjölander. Intrepid;information-theoretic tree traversal for protein functional site identification. *Bioinformatics*, 24:2445–2452, November 2008.

[32] Alexandra Shulman-Peleg, Ruth Nussinov, and Haim J. Wolfson. Recognition of functional sites in protein structures. *Journal of Molecular Biology*, 339(3):607 – 633, 2004.

[33] Ashish V. Tendulkar, Pramod P. Wangikar, Milind A. Sohoni, Vivekanand V. Samant, and Chetan Y. Mone. Parameterization and classification of the protein universe via geometric techniques. *Journal of Molecular Biology*, 334:157–172, 2003.

[34] Andrew C. Wallace, Neera Borkakoti, and Janet M. Thornton. Tess: A geometric hashing algorithm for deriving 3d coordinate templates for searching structural databases. application to enzyme active sites. *Protein Science*, 6(11):2308–2323, 1997.

[35] Andrew C. Wallace, Roman A. Laskowski, and Janet M. Thornton. Derivation of 3d coordinate templates for searching structural databases: Application to ser-his-asp catalytic triads in the serine proteinases and lipases. *Protein Science*, 5(6):1001–1013, 1996.

[36] Pramod P. Wangikar, Ashish V. Tendulkar, S. Ramya, Deepali N. Mali, and Sunita Sarawagi. Functional sites in protein families uncovered via an objective and automated graph theoretic approach. *Journal of Molecular Biology*, 326(3):955 – 978, 2003.

[37] L. Wei and R. B. Altman. Recognizing protein binding sites using statistical descriptions of their 3d environments. In *Pacific Symposium on Biocomputing*, pages 497–508, 1998.

[38] F. Xin and P. Radiojac. Computational methods for identification of functional residues in protein structures. *Current Protein and Peptide Science*, 12(6):465–469, 2011.

[39] Fuxiaao Xin, Steven Myers, Yong Fuga Li, David N. Cooper, Sean D. Mooney, and Predrag Radivojac. Structure-based kernels for the prediction of catalytic residues and their involvement in human inherited disease. *Bioinformatics*, 26(16):1975–1982, 2010.

[40] Eunseog Youn, Brandon Peters, Pregrad Radivojas, and Sean Mooney. Evaluation of features for catalytic residue prediction in novel folds. *Protein Science*, 16:216–226, February 2007.