Prediction of interface residue based on features of residue interaction network extracted using shapley value centrality measure

Babita Pandey¹, Sarneet K Chawla¹, Devendra K Pandey Corresp.²

¹ School of Computer Application, Lovely Professional University, Punjab, India
² Department of Biotechnology, Lovely Professional University, Phagwara, Punjab, India

Corresponding Author: Devendra K Pandey
Email address: devendra.15673@lpu.co.in

Protein-Protein interaction plays an important role in the life processes. Molecular mechanisms of the related processes can be better understood with the help of interface prediction. In this work, we use game theory concept of Shapley value to analyse the spatial relationship between residues in residue interaction network. Four features are extracted from network using shapley value and given as input to ACO for optimization. Our experiment shows that optimized feature set, significantly improves the result of normal classifier and accuracy from 80% to 85%. These findings are useful for identifying protein-like complex networks. The presented results suggest that the feature selection by Shapley value and optimization by ACO improves the classification of protein structure at great extent less computational complexity.
Prediction of interface residue based on features of residue interaction network extracted
using shapley value centrality measure

Babita Pandey*, School of Computer Application, Lovely Professional University
Sarneet K. Chawla, School of Computer Application, Lovely Professional University
Devendra K. Pandey, School of Bio-Sciences, Lovely Professional University,

*Corresponding author

Abstract

Protein-Protein interaction plays an important role in the life processes. Molecular mechanisms of the related processes can be better understood with the help of interface prediction. In this work, we use game theory concept of Shapley value to analyse the spatial relationship between residues in residue interaction network. Four features are extracted from network using shapley value and given as input to ACO for optimization. Our experiment shows that optimized feature set, significantly improves the result of normal classifier and accuracy from 80% to 85%. These findings are useful for identifying protein-like complex networks. The presented results suggest that the feature selection by Shapley value and optimization by ACO improves the classification of protein structure at great extent less computational complexity.

Keywords: Ant Colony Optimization, Residue Interaction Network, Shapley Value, Classification, Naïve Baye’s classifier

1. Introduction

Protein-Protein interaction plays very important role in various domain such as to understand the molecular mechanisms of cellular functions (Szilagy & Zhang, 2014; Jiao & Ranganathan, 2017), to identify the possible drug targets and to design the drug, to decode of the structure-function relationship. Although there exist many experiments such as: X-ray crystallography (Kendrew, 1960; Perutz, 1960) and multi dimensional magnetic resonance (Travers, 1990; Bjorkman & Parham, 1990) to identify the interacting protein pair, however the use of these methods are limited due to their high cost and high time consuming nature. Currently, computational methods are very popular in residue interface prediction (Jiao & Ranganathan, 2017). These computational methods take as an input features of either sequence information (Jia et al., 2016; Ofran & Rost, 2003), the physicochemical properties of amino acids (Jia et al., 2015; Jia et al., 2016), or combination of information such as: gene ontology information and pseudo-amino acid composition (PseAA) (Chou & Cai, 2006), structural information or combination
of shape complementarity, hydrophobicity, charge of residues and evolutionary conservation.

Various authors have deployed the residue interaction network (RIN) information (Pavlopoulos et al., 2011; Jiao and Ranganathan, 2017) for interface prediction. In network the centrality identifies the most prominent nodes in a network. The centrality measure is valued by several parameters such as: degree, shortest path, closeness, betweenness, clustering coefficient, eigenvector centrality, eccentricity, average nearest neighbour degree and edge betweenness centrality (Chakrabarty & Parekh, 2016). Nan and Zhang (Nan & Zhang, 2013) used degree and the community characteristics, del Sol & O’Meara (del Sol & O’Meara, 2005) used betweenness and Pons et al., (Pons et al., 2016) used closeness to predict the hot spot on the interface. Brinda and Vishveshwara (Brinda & Vishveshwara, 2005) predicted the hot spot on the interface by comparing the interface hubs of RIN model with the non-interface hub and no-hub area of the interface. Yan et al.(Yan et al, 2016) predicted the hot spot on the interface by deploying minimum cut tree algorithm on a sub-graph.

Although, lot of work have been reported by various authors for interface prediction using conventional centrality measure such as: degree, shortest path, closeness and betweenness etc, the reminder and caution from these works is that these centrality measure consider the failure of individual nodes and completely ignores the situation where multiple nodes can fail simultaneously, each node is treated separately therefore the hidden assumption is that node failures occur independently of each other. In order to remove these restrictions game based Shapley value (SV) is used as centrality measure. In network domain SV is used to measure the importance of individual nodes (Pauling & Corey, 1951; Pauling, Corey, Branson, 1951). These SV is treated as network features for interface prediction.

Besides the extraction of network features, choosing a proper prediction algorithm is another important step in interface prediction. Various authors have deployed various techniques for prediction like: random forest (Jia et al., 2016), neural network (Ofran & Rost, 2003; Bendell et al., 2014), support vector machine (Rost & Sander, 1993a; Dong et al. 2007; Zellner et al., 2012; Cheng & Yan, 2017), logistic regression, Bayesian Networks, and decision tree (Bendell et al., 2014). The performance of prediction and classification methods can significantly improved by feature optimization algorithm.
In this work, we have applied the Shapley Value based centralities measure as features of RIN. These features are optimized with ant colony optimization (ACO) and then given as input to Naive Baye’s classifier for classifying the interface residue.

This remainder of the paper is organized as follows. Section 2 describes the network representations using amino acid residues, various algorithms for computing SV based centralities measure for RIN, ACO and Naïve Baye’s classifier. Results are discussed in Section 3. Discussion is done in section 4. Conclusions are drawn in Section 5.

2. Materials and methods

2.1. Network Representation

In C-alpha network, C-alpha atom is considered as a hub in the system and an edge is built if the separation between a couple of C-alpha atoms is inside the lower and upper thresholds characterized by the client (default upper threshold = 7 Å; lower threshold = 0 Å).

i. Unweighted: consider all edges equally is important.

ii. Weighted: for a C-alpha weighted network the Edge weight is given by:

$$ W_{ij} = \frac{1}{d_{ij}} $$

where, $d_{ij}$ is the euclidean distance between C-alpha atoms of $i^{th}$ and $j^{th}$ residues (Chakrabarty and Parekh, 2016).

2.2. Shapley value (SV) based network centrality

In a network domain unlike traditional centrality measure, SV is used to find the contribution of individual node (Aadithya et al., 2010).

Let any subset of amino acid be $S$ from any given RIN $G(V;E)$. For every such $S$, assign a value $u(S)$ given by

$$ u(S) = \sum_{x \in S} \frac{1}{1 + \min_{y \in S \setminus S} d(x,y)} $$

where, $d(x,y)$ is the distance among amino acids $x$ and $y$ which is measured as the shortest length between $x$ and $y$ in $G$ graph.

Given a graph $G(V,E)$ with the amino acids set $V$ and the linked amino acids set $E$, where $G$ is used to define coalitional game $g(G,V,u)$ having characteristic function $u$.

$$ u: 2^{V(G)} \to R $$

Where graph $G$ is dependent, satisfying condition $u(\emptyset)=0$. 
υ(S) define the sphere of influence of the coalition S over the other nodes. Aadithya et. al. (2010) has defined the υ(S) in four ways: in first game formulation υ(S) is the set of all nodes immediately reachable (within one hop) from S (Algorithm 1). The second game formulation defines υ(S) set of only those nodes which are immediately reachable in at least k different ways from S (Algorithm 2). The third game defines υ(S) as the set of all nodes within a cut off distance of S (as measured by shortest path lengths on the weighted graph) (Algorithm 3). The fourth formulation defines the υ(S) to be an arbitrary function ƒ(;) of the distance between S and the other nodes (Algorithm 4).

Algorithm 1: Computation of υ1(S) = 1 degree away hop amino acids as shown in Fig 1a.

Step 1: Consider the undirected protein network g (V, E).

Step 2: Set “fringe” of a subset S ⊆ V(g) that has at most one hop reachability, having set
\{ y∈V(g): y∈S (or)∃x∈S so that (x,y)∈E(g) \}.

Step 3: Defining its characteristic function υ1:2^V(G)→R given by
υ1(S)=\begin{cases} 
0 & \text{if } S = \emptyset \\
\text{size (fringe (S))} & \text{else} 
\end{cases}

Step 4: Selecting fringe of amino acids are u∈V(g) as v, the amino acid in fringe are picked one by one and we define deg_g (y) is equal to the number of amino acids which are one hop away from y i.e. y∈V(g),

\text{Shapley value } |y| = \frac{1}{1 + \text{deg}_g(y)}.

Step 4.1: N_g(y) is the one hop neighbour, where x∈N_g(y).

\text{Shapley value } |y| = \frac{1}{1 + \text{deg}_g(x)}.

Step 5: Obtaining the Shapley value of each amino acid.

Algorithm 2: υ2(S) = agents with k neighbours at least in, S as shown in Fig 1b.

Step 1: Consider an unweighted graph g(V,E) including number of agents having k least neighbours in S.

Step 2: Formally, considering function characteristics υ2:2^V(g)→R, where
υ2(S)=\begin{cases} 
0 & \text{if } S = \emptyset \\
\{ y; y∈S(\text{or})\text{N}_g(y) \cap S \geq k \} & \text{else} 
\end{cases}

Step 3: If deg_g(y_i)<k, else then

E[B_{y,y_j}] = \delta(y_i,y_j)
E[B_{y_j,y_j}] = 1 \text{ for } y_i=y_j ,0 \text{ otherwise}

Step 4: If deg_g(n_i) ≥ k, two cases are discussed.

Step 4.1: If y_j âˆ‰ y_j, then the marginal contribution condition is (k-1) neighbours of y_j i.e. y_j âˆ‰ S_i.

E[B_{y,y_j}] = \frac{1 + \text{deg}_g(y_j) - k}{\text{deg}_g(y_j)(1 + \text{deg}_g(y_j))}
Step 4.2: If $y_j = y_j$, then the marginal contribution condition follows only if $S_j$ consist of $(k-1)$ neighbours of $y_j$ at most.

$$E[B_{y_j}] = \frac{1}{1 + deg_g(y_j)}$$

Step 5: Substituting the above given equation for SV expression,

$$SV(y_j) = \frac{k(y_j)}{1 + deg_g(y_j)} + \sum_{y_j \in N_g(y_j)} \frac{1 + deg_g(y_j) + k(y_j)}{deg_G(y_j)(1 + deg_G(y_j))}$$

Algorithm 3: $\nu_3(S) = \text{agents with d cutoff at_most away, as shown in Fig 1c.}$

Step 1: Consider an unweighted graph $g(V,E,W)$ including the weight function is $W:E \rightarrow R^+$.  
Step 2: For function $S \subseteq y(g)$,

$$\nu_3(S) = \begin{cases} 0 & \text{size}(\{y_i : \exists y_j \in S \mid distance(y_i, y_j) \leq d_{\text{cutoff}}\}) \text{ if } S = \emptyset \\ \text{else} \\ \end{cases}$$

Step 3: Extended neighbourhood notation is introduced for driving SV formula,

$$N_g(y_j, d_{\text{cutoff}}) = \{y_k : \forall y_j : distance(y_k, y_j) \leq d_{\text{cutoff}}\}$$

Step 3.1: Size of $N_g(y_j, d_{\text{cutoff}})$ is denoted by $deg_g(y_j, d_{\text{cutoff}})$ which is computed by using Dijkstra’s algorithm.

Step 4: An unique value $d_{\text{cutoff}}(y_i)$ is assigned to each amino acid $y_i \in y(g)$, where $\nu(S)$ is within a distance $d_{\text{cutoff}}(y_i)$ from $S$.

Step 5: Obtained SV from below given expression,

$$SV(y_j) = \sum_{y_j:distance(y_i, y_j) \leq d_{\text{cutoff}}(y_j)} \frac{1}{1 + deg_g(y_i, d_{\text{cutoff}}(y_j))}$$

Algorithm 4: Computation of $\nu_3(C) = \sum_{v_i \in V(G)} f(distance(v_i, C))$, as shown in Fig 1d.

Step 1: Considering weighted graph $G(V,E,W)$, for function $f:R^+ \rightarrow R^+$.

Step 2: Defining the coalition $C$ value is given by
\[
v_d(C) = \begin{cases} 
0 & \text{if } C = \emptyset \\
\sum_{v_i \in V(G)} f(d(v_i,C)) & \text{else}
\end{cases}
\]

Where \(d(v_i,C)\rightarrow \min \{distance(v_i,v_j) \mid v_j \in C\} \).

Step 3: Dijkstra algorithm is used to compute a distance vector for each vertex \(v\).

Step 4: The backward cumulative sum \(\sum_{k=0}^{\frac{f(d_k+1)}{(k+1)(k+2)}}\) is computed by transverse of the vector in reverse where according to \(E[MC(w,v)]\) equation, SV of the appropriate amino acid is updated.

Step 5: For \(v_i=v_j\), producing similar analysis

\[
E[MC(v_i,v_i)] = f(0) - 2 \sum_{k=0}^{\frac{f(d_k+1)}{(k+1)(k+2)}}
\]

Step 6: According to \(E[MC(v,v)]\) equation, SV of \(v\) updates itself after the transverse.

Step 7: Computed the SVs are given by:

\[
SV(v_i) = \sum_{v_j \in V(G)} E[MC(v_i,v_j)]
\]

The four centrality measures extracted from network using game theory algorithm 1 to 4 are used as network features. Fig 1a), 1b) 1c) & 1d) shows the extraction of four features using Shapley value algorithm.

### 2.3 Naive Baye’s and Ant Colony Optimization (ACO)

#### 2.3.1 Naive Baye’s

In this work, we have chosen the Bayesian classifiers because they have given a good precision and intuitive graphical representation in many applications (Pons et al., 2011). The Naive Baye’s classifier (Pons et al., 2011) applies the Bayes theorem to predict for each unseen instance \(x\), the class \(c \in C\) for which it has a higher a posterior probability. The posteriori probability is computed as eq. (1)

\[
p(c|x) \propto p(c, x) = p(c)\Pi_{i=1}^{n}p(xi/c)
\]
where, \( p(x_i | c) \) is the conditional probability of \( X_i = x_i \) given that \( C = c \) when all variables have discrete values. As a result, the Naïve Baye’s classifier follows the following approach as shown in eq (2):

\[ c = \arg \max_c p(c) \prod_{i=1}^n p(x_i | c) \] (2)

### 2.3.2 Ant Colony Optimization (ACO)

ACO is fundamentally roused by the genuine ant settlements conduct called artificial framework. Through the charts the ACO calculation is utilized for taking care of computational problems and discovering great way. Like ant conduct, looking for way between food source and their colony to look through an ideal way comparative is the principle point of this calculation. For the \( k^{th} \) ant the change probability at the time step \( t \) from city \( x \) to city \( y \) in the TSP problem:

\[ \text{PROB}_{xy}^k(t) = \begin{cases} \frac{[\tau_{xy}(T)]^\alpha [\eta_{xy}]^\beta}{\sum_{y \in I_k^x} [\tau_{xy}(T)]^\alpha [\eta_{xy}]^\beta} & \text{if } j \in I_k^x \\ 0 & \text{Otherwise} \end{cases} \]

Where

\( \eta_{xy} \leftrightarrow \) priority heuristic information,

\( \tau_{xy} \leftrightarrow \) pheromones trail amount on the edge \((x, y)\) at the time \( T \),

The pheromone trail and heuristic information relative effects are identified by two factors i.e., \( \alpha \) and \( \beta \). And the city’s neighbourhood set that are reasonable is denoted by \( I_k^x \).

After a visit is finished by every ant, a constant dissipation rate at first bringing down them which refreshed the pheromone trail. Inferable from which every ant is permitted effective pheromone affidavit on curves which is its visit part as appeared in the condition underneath:

\[ \tau_{xy} = (1 - \rho) \tau_{xy} + \sum_{k=1}^N \Delta \tau_{yx}^k \]

Where

\( \rho \leftrightarrow \) pheromones rate of trail evaporation,

\( N \leftrightarrow \) no. of ants,

The no. of cycles declining the pheromone quality related on circular segments which ants don't choose. \( \Delta \tau_{yx}^k \), the trail substance quality per unit length which lays nervous \((y, x)\) is given as takes
\[ \Delta \tau_{yx}^k = \begin{cases} Q & \text{if ant k in its tour uses edge (y,x)} \\ 0 & \text{Otherwise} \end{cases} \]

Where

\[ Q \leftarrow \text{constant that is predefined}, \]

\[ L_k \leftarrow \text{length of the tour}. \]

In order to get better results from Naive Baye’s Classifier, we must adjust the weights of features. Therefore in this works, we have chosen the ACO to adjust the parameters. The reason that we chose ACO is that it is simple and effective. The process of integrated ACO and Naive Baye’s is following:

**Step 1:** Initializing ants, where for each ant \( n = 1,2,3 \ldots \ldots N \).

**Step 2:** In ant \( n \), each variable \( a_n^d \), \( d=1,2,3 \ldots \ldots D \).

**Step 3:** Updating pheromones by choosing \( \mu_x^d \) from the pheromone table with probability, where \( i \in \{1,2,3 \ldots \ldots K \} \).

**Step 4:** If minimum error is obtained, then it has higher probability.

**Step 5:** Generating a standard deviation \( \sigma_x^d \), if \( rv \leq a_1 \) with the use of uniform distribution \( U(0,1) \), where \( rv \) is the random value lies between \( x_1 \), the predefined threshold \( 0 \) and \( 1 \).

**Step 6:** Generating a new value for variable \( a_n^d \), if \( rv \leq a_2 \), by normal distribution \( N(\mu_x^d, \sigma_x^d) \).

**Step 7:** Else, uniform distribution generates random value, and generating random solution for \( a_n^d \).

**Step 8:** Obtained variable \( a_n^d \) denoting the observed attribute values to certain class label \( c \).

**Step 9:** Computing probability for each class: \[ p(b_y | a_n^d) = \frac{p(b_y)p(a_n^d | b_y)}{\sum_{i=1}^{c} p(b_y)p(a_n^d | b_y)} \quad y=1,2 \ldots \ldots c \]

Where

\[ p(b_y) \] is the \( y_i \) prior probability,

\[ p(x_n^d | b_y) \] is conditional class probability density function.

**Step 10:** Calculate probability distribution over the set of features:

\[ p(a) = \prod_{i=1}^{k} p(c_x)p(a_n^d | c_x) \]

Where

\( k \leftarrow \text{number of classes}, \)

\( c_x \leftarrow x^{th} \) class.

**Step 11:** Calculate precision, accuracy, and recall.
3. Experiments and Result

Complete process of implementation is shown in Fig 2. The complete process is divided into four phases (1) network construction given in section 2.1, (2) Features extraction from network using game based SV algorithm as explained in section 2.2. (3) Weight optimization using ACO classifier and classification using Bayes algorithm explained in section 2.3. Complete procedure is shown in Fig 2.

3.1 Data sets

In this work the oldest non homologous dataset RS126 developed by Rost and Sander (1993a) is used. It has 126 protein sequences that comprises of 26,846 residues. The average sequence identity is less than 31% and the average sequence length is 185 residues (Rost and Sander, 1993b). Few proteins from RS126 are in Table 1.

3.2 Performance evaluation parameters

In this work, result was evaluated using accuracy, precision, specificity, F-measure, accuracy.

Accuracy:- For predictive model quality analysis, as well as an obvious prediction criterion the accuracy is often the starting point. It measures the ratio of the total number of evaluated cases to predictions that are correct.

\[
\text{Accuracy} = \frac{TP+TN}{(TP+TN+FP+FN)}
\]

Where TP, TN, FP and FN are true positive, true negative, false positive and false negative respectively. For an interface residue, TP means interface is predicted as interface (positive), FN means interface is predicted as non interface, TN means non interface is predicted as non interface and FP non interface is predicted as interface.

Precision:- It is defined by the true positives number (TP) over the true positives number (TP) plus false positive number (FP).

\[
\text{Precision} = \frac{TP}{(TP+FP)}
\]

Specificity:- It is defined by the true positives number (TN) over the true negative number plus false positives number (FP).

\[
\text{Specificity} = \frac{TN}{(TN+FP)}
\]
Sensitivity/Recall: It is defined as the true positives number (TP) over the true positive number (TP) plus the false negative number (FP).  
Recall = TP / (TP + FN).

F-Measure: The F-measure is a measure of a test's accuracy and is defined as the weighted harmonic mean of the precision and recall of the test.  
F-Measure = 2 * (Recall * Precision) / (Recall + Precision)

3.3 Experimental results

The complete program is developed in Java. In our experiment, ACO is trained and tested with data shown in Table 2. The data set consist of input as four features extracted from shapley value as shown 2nd, 3rd and 4th column, and output is two labels: interface residue (denoted as +1) and not interface residue (denoted as -1). 70% data are used for training and 30% is used for testing. ACO optimizes the weight of the feature, i.e., pheromones updating and transient probability prediction. The ACO is learned by Naïve Baye’s generative classifier and classified the protein structure according to four features. In this experiment we adopted k fold cross-validation where k = 7. The complete R126 dataset is divided in equal subset of 18 proteins. In each step, we have selected one subset of protein as the test set, and the rest 6 subsets were used for training. The same procedure was repeated 7 times with all other subsets (see Figs. 3-5). The complete GUI of the software is shown in Fig 6. Fig: 6 a), b), c), d) interfaces for PDB file selection from PDB database; e), f) interface showing the training, testing and classification results.

4. Discussion

The accuracy (ACC) of Naïve Baye’s and Naïve Baye’s with ACO based on the RIN features are 0.8 and 0.85 respectively, the F-measure is about 0.74 and 0.79 respectively, recall is 0.7 and 0.75 respectively, and the precision is 0.75 and 0.8 respectively. There are many works about the interface prediction. Based on the input information all of the work can be grouped in three
categories: sequence, structure and mixed. In the present work, we have selected only that work which performs prediction based or partly structural knowledge for comparison, and the results are shown in Table 3. With the data in Table 3 and Fig 7, we can get an overall impression that the predictor based on RIN features extracted using game theory based SV is feasible, and the result is comparable to that of other works. It is observed from the Fig. 7 that the precision, recall, accuracy and F-measure are high for ACO with Naïve Baye’s classification as compared to other methods. This improvement is due to the network feature set extracted using SV and further optimized by optimization algorithm ACO.

5. Conclusion

In the present work, SV based network centrality concept is applied to RIN for detecting its features in polynomial time. The features are optimized with ACO then naïve Bayes applied to classify the interface residue. This paper compares the optimized naïve Bayes with normal naïve Bayes and with other interface residue prediction methods. In result, optimized Naive Baye’s shows significant higher accuracy 85%, precision 80%, recall 75% and F-measure 79% as compared to Naive Baye’s and other prediction methods. In future we will propose and apply more feature extraction methods, learning methods, new computational intelligence or machine learning methods to predict interface residue.

REFERENCES


Wei ZS, Han K, Yang J-Y, Shen HB, Yu DJ. 2016. Protein-protein interaction sites prediction by


Table 1 (on next page)

List of RS126 dataset used in protein structure prediction lab
<table>
<thead>
<tr>
<th>PDB ID</th>
<th>Chain</th>
<th>PDB ID</th>
<th>Chain</th>
<th>PDB ID</th>
<th>Chain</th>
<th>PDB ID</th>
<th>Chain</th>
<th>PDB ID</th>
<th>Chain</th>
</tr>
</thead>
<tbody>
<tr>
<td>1BMV</td>
<td>1</td>
<td>2UTV</td>
<td>A</td>
<td>4SGB</td>
<td>1</td>
<td>1PYP</td>
<td>-</td>
<td>3CD4</td>
<td>-</td>
</tr>
<tr>
<td>4RHV</td>
<td>1</td>
<td>3GAP</td>
<td>A</td>
<td>1MCP</td>
<td>L</td>
<td>1RBP</td>
<td>-</td>
<td>3CLA</td>
<td>-</td>
</tr>
<tr>
<td>1BMV</td>
<td>2</td>
<td>3HMG</td>
<td>A</td>
<td>2OR1</td>
<td>L</td>
<td>1RHD</td>
<td>-</td>
<td>3CLN</td>
<td>-</td>
</tr>
<tr>
<td>1R09</td>
<td>2</td>
<td>3TIM</td>
<td>A</td>
<td>1GD1</td>
<td>O</td>
<td>1S01</td>
<td>-</td>
<td>3EBX</td>
<td>-</td>
</tr>
<tr>
<td>1LMB</td>
<td>3</td>
<td>4SDH</td>
<td>A</td>
<td>2TMV</td>
<td>3</td>
<td>1SH1</td>
<td>-</td>
<td>3ICB</td>
<td>-</td>
</tr>
<tr>
<td>4RHV</td>
<td>3</td>
<td>4TS1</td>
<td>A</td>
<td>2WRP</td>
<td>R</td>
<td>1UBQ</td>
<td>-</td>
<td>3PGM</td>
<td>-</td>
</tr>
<tr>
<td>2MEV</td>
<td>4</td>
<td>4XIA</td>
<td>A</td>
<td>5CYT</td>
<td>R</td>
<td>2AAT</td>
<td>-</td>
<td>3RNT</td>
<td>-</td>
</tr>
<tr>
<td>4RHV</td>
<td>4</td>
<td>5HVP</td>
<td>A</td>
<td>1ACX</td>
<td>-</td>
<td>2ALP</td>
<td>-</td>
<td>4BP2</td>
<td>-</td>
</tr>
<tr>
<td>1BBP</td>
<td>A</td>
<td>7CAT</td>
<td>A</td>
<td>1AZU</td>
<td>-</td>
<td>2CAB</td>
<td>-</td>
<td>4CMS</td>
<td>-</td>
</tr>
<tr>
<td>1CDT</td>
<td>A</td>
<td>9API</td>
<td>A</td>
<td>1BDS</td>
<td>-</td>
<td>2CYP</td>
<td>-</td>
<td>4CPV</td>
<td>-</td>
</tr>
<tr>
<td>1FXI</td>
<td>A</td>
<td>9WGA</td>
<td>A</td>
<td>1CBH</td>
<td>-</td>
<td>2FOX</td>
<td>-</td>
<td>4GR1</td>
<td>-</td>
</tr>
<tr>
<td>1GP1</td>
<td>A</td>
<td>1WSY</td>
<td>B</td>
<td>1CC5</td>
<td>-</td>
<td>2FXB</td>
<td>-</td>
<td>4PFK</td>
<td>-</td>
</tr>
<tr>
<td>1L8</td>
<td>A</td>
<td>2LTN</td>
<td>B</td>
<td>1CRN</td>
<td>-</td>
<td>2GBP</td>
<td>-</td>
<td>4RXN</td>
<td>-</td>
</tr>
<tr>
<td>1OVO</td>
<td>A</td>
<td>2SOD</td>
<td>D</td>
<td>1ECA</td>
<td>-</td>
<td>2GCR</td>
<td>-</td>
<td>5LDH</td>
<td>-</td>
</tr>
<tr>
<td>1TNF</td>
<td>A</td>
<td>3HMG</td>
<td>B</td>
<td>1ETU</td>
<td>-</td>
<td>2GNS</td>
<td>-</td>
<td>5LYZ</td>
<td>-</td>
</tr>
<tr>
<td>1WSY</td>
<td>A</td>
<td>9API</td>
<td>B</td>
<td>1FDX</td>
<td>-</td>
<td>2IB</td>
<td>-</td>
<td>6ACN</td>
<td>-</td>
</tr>
<tr>
<td>256B</td>
<td>A</td>
<td>9INS</td>
<td>B</td>
<td>1FKF</td>
<td>-</td>
<td>2LHB</td>
<td>-</td>
<td>6CPA</td>
<td>-</td>
</tr>
<tr>
<td>2AK3</td>
<td>A</td>
<td>1FC2</td>
<td>C</td>
<td>1FND</td>
<td>-</td>
<td>2MHU</td>
<td>-</td>
<td>6CPP</td>
<td>-</td>
</tr>
<tr>
<td>2CCY</td>
<td>A</td>
<td>5ER2</td>
<td>E</td>
<td>1GDJ</td>
<td>-</td>
<td>2PCY</td>
<td>-</td>
<td>6CTS</td>
<td>-</td>
</tr>
<tr>
<td>2GLS</td>
<td>A</td>
<td>6TMN</td>
<td>E</td>
<td>1HIP</td>
<td>-</td>
<td>2PHH</td>
<td>-</td>
<td>6DFR</td>
<td>-</td>
</tr>
<tr>
<td>2HMZ</td>
<td>A</td>
<td>1FDL</td>
<td>H</td>
<td>1L58</td>
<td>-</td>
<td>2SNS</td>
<td>-</td>
<td>6HIR</td>
<td>-</td>
</tr>
<tr>
<td>2LTN</td>
<td>A</td>
<td>1CSE</td>
<td>I</td>
<td>1LAP</td>
<td>-</td>
<td>2STV</td>
<td>-</td>
<td>7ICD</td>
<td>-</td>
</tr>
<tr>
<td>2PAB</td>
<td>A</td>
<td>1TGS</td>
<td>I</td>
<td>1MRT</td>
<td>-</td>
<td>3AIT</td>
<td>-</td>
<td>7RSA</td>
<td>-</td>
</tr>
<tr>
<td>2RSP</td>
<td>A</td>
<td>2TGP</td>
<td>I</td>
<td>1PAZ</td>
<td>-</td>
<td>3BSC</td>
<td>-</td>
<td>8ABP</td>
<td>-</td>
</tr>
<tr>
<td>2TSC</td>
<td>A</td>
<td>4CPA</td>
<td>I</td>
<td>1PPT</td>
<td>-</td>
<td>3BLM</td>
<td>-</td>
<td>8ADH</td>
<td>-</td>
</tr>
<tr>
<td>9PAP</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Table 2 (on next page)

Value of four Features of Residue Interaction network for some protein sequence
Table 2. Value of four Features of Residue Interaction network for some protein sequence

<table>
<thead>
<tr>
<th>PDB ID</th>
<th>Feature 1</th>
<th>Feature 2</th>
<th>Feature 3</th>
<th>Feature 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>3CD4</td>
<td>11.2668</td>
<td>0.0487</td>
<td>23892.69</td>
<td>4.83207E+15</td>
</tr>
<tr>
<td>3CLA</td>
<td>10.4667</td>
<td>0.002</td>
<td>2920.2745</td>
<td>4.00701E+13</td>
</tr>
<tr>
<td>3CLN</td>
<td>11.1537</td>
<td>0.0248</td>
<td>1966.0017</td>
<td>9.19202E+12</td>
</tr>
<tr>
<td>3EBX</td>
<td>31.0715</td>
<td>0.0619</td>
<td>5536.7767</td>
<td>3.29095E+13</td>
</tr>
<tr>
<td>3ICB</td>
<td>12.3955</td>
<td>0.0218</td>
<td>49168144</td>
<td>8.16792E+16</td>
</tr>
<tr>
<td>3PGM</td>
<td>54.7417</td>
<td>0.0854</td>
<td>6969.6901</td>
<td>2.79167E+12</td>
</tr>
<tr>
<td>3RNT</td>
<td>11.0356</td>
<td>0.03</td>
<td>4509.52</td>
<td>7.22275E+11</td>
</tr>
<tr>
<td>4BP2</td>
<td>29.6864</td>
<td>0.0133</td>
<td>1376.0607</td>
<td>5.52819E+12</td>
</tr>
<tr>
<td>4CMS</td>
<td>10.619</td>
<td>0.0289</td>
<td>14773.6618</td>
<td>9.18271E+15</td>
</tr>
<tr>
<td>4CPV</td>
<td>11.3347</td>
<td>0.0149</td>
<td>1413.5552</td>
<td>7.00924E+14</td>
</tr>
<tr>
<td>4GR1</td>
<td>18.7158</td>
<td>0.025</td>
<td>1966.0017</td>
<td>9.19202E+12</td>
</tr>
<tr>
<td>4PKF</td>
<td>10.0633</td>
<td>0.0406</td>
<td>6359.9036</td>
<td>4.86256E+13</td>
</tr>
<tr>
<td>4RXN</td>
<td>14.6973</td>
<td>0.0196</td>
<td>2049.1769</td>
<td>7.07581E+13</td>
</tr>
<tr>
<td>5LDH</td>
<td>40.1529</td>
<td>0.0268</td>
<td>4439.0002</td>
<td>5.03938E+11</td>
</tr>
<tr>
<td>5LYZ</td>
<td>17.695</td>
<td>0.1164</td>
<td>4616.841</td>
<td>2.02463E+14</td>
</tr>
</tbody>
</table>
Table 3 (on next page)

The comparison with the results of the related prediction work
Table 3: The comparison with the results of the related prediction work

<table>
<thead>
<tr>
<th>The knowledge source of the prediction model</th>
<th>The method used in the prediction</th>
<th>SN (Recall)</th>
<th>SP</th>
<th>ACC</th>
<th>AUC</th>
<th>Precision</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>The knowledge source of the prediction Structural neighbor, contacting frequencies, solvent accessible surface areas (Zhang et al., 2011)</td>
<td>SVM</td>
<td>0.575</td>
<td>-</td>
<td>0.575</td>
<td>0.73</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Evolutionary conservation, hydrophobic property, solvent accessibility features (Wei et al., 2016)</td>
<td>SVM, RF</td>
<td>0.581</td>
<td>0.697</td>
<td>0.679</td>
<td>0.711</td>
<td>-</td>
<td>0.63</td>
</tr>
<tr>
<td>Both sequence and 3D structure information (Guo et al., 2016)</td>
<td>Clustering method</td>
<td>0.63</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Physicochemical property of the spatially neighboring residues (Ezkurdia et al., 2009)</td>
<td>Clustering method</td>
<td>0.453</td>
<td>0.64</td>
<td>0.694</td>
<td>-</td>
<td>-</td>
<td>0.53</td>
</tr>
<tr>
<td>Weighting the input physicochemical features of the spatially neighboring residues based on distance (Ezkurdia et al., 2009)</td>
<td>SVM</td>
<td>0.586</td>
<td>0.634</td>
<td>0.715</td>
<td>-</td>
<td>-</td>
<td>0.60</td>
</tr>
<tr>
<td>Sequence features of the spatially neighboring residues (Ezkurdia et al., 2009)</td>
<td>SVM</td>
<td>0.496</td>
<td>0.578</td>
<td>0.589</td>
<td>-</td>
<td>-</td>
<td>0.53</td>
</tr>
<tr>
<td>RIN parameter, B-factor (Jiao &amp; Ranganathan, 2017)</td>
<td>SVM</td>
<td>0.69</td>
<td>0.62</td>
<td>0.64</td>
<td>0.72</td>
<td>-</td>
<td>0.65</td>
</tr>
<tr>
<td>This paper game based shapely value centrality measure of RIN and Naïve bayes+ ACO</td>
<td>Naïve Bayes</td>
<td>0.7</td>
<td>-</td>
<td>0.8</td>
<td>-</td>
<td>0.75</td>
<td>0.74</td>
</tr>
<tr>
<td></td>
<td>Naïve Bayes + ACO</td>
<td>0.75</td>
<td>-</td>
<td>0.85</td>
<td>-</td>
<td>0.8</td>
<td>0.79</td>
</tr>
</tbody>
</table>
Figure 1 (on next page)

Four Feature extraction a) first feature, b) second feature c) third feature, d) fourth feature
Fig 1: Four Feature extraction a) first feature, b) second feature c) third feature, d) fourth feature
Figure 2 (on next page)

Flowchart of complete process
Fig 2. Flowchart of complete process
Figure 3 (on next page)

Accuracy computed at various iteration of 7 cross validation
Fig 3: Accuracy computed at various iteration of 7 fold cross validation
Figure 4 (on next page)

Precision computed at various iteration of 7 cross validation
Fig 4. Precision computed at various iteration of 7 cross validation
**Figure 5** (on next page)

Recall computed at various iteration of 7 cross validation
Fig 5 Recall computed at various iteration of 7 cross validation
Figure 6 (on next page)

a) PDB option selection, b) selecting folder, c) selecting txt file, d) train and test model, e) showing trained successfully, f) classification as primary structure
Fig: 6  a) PDB option selection, b) selecting folder, c) selecting txt file, d) train and test model, e) showing trained successfully, f) classification as primary structure
Figure 7 (on next page)

The comparison with the results of the related prediction work
Methods proposed by various author

Fig 7. The comparison with the results of the related prediction work.