Combination of Genetic Programming and Support Vector Machine-Based Prediction of Protein-Peptide Binding Sites With Sequence and Structure-Based Features

Shima Shafiee\(^a\), Abdolhossein Fathi\(^a,\)*

\(^a\)Department of Computer Engineering and Information Technology, Razi University, Kermanshah, Iran.

**ARTICLE INFO.**

**Article history:**
Received: 5 January 2021
Revised: 2 May 2021
Accepted: 14 July 2021
Published Online: 31 July 2021

**Keywords:**
Binding Site, Genetic Programming, Structure Clustering Algorithm, Protein-Peptide Binding Prediction.

**ABSTRACT**

Prediction of the peptide-binding site of proteins is a significant and essential task in different processes such as understanding biological processes, protein functional analysis, comparison of functional sites, comprehension of the transactions mechanism, drug design, cellular signaling, and cancer treatment. Predictive analysis of the protein-peptide binding site is one of the most challenging bioinformatics issues. Experimental methods are time-consuming, costly, and laborious. Therefore, we propose a machine learning-based method for predicting protein-peptide binding sites by utilizing enhanced features vector obtained from three-dimensional protein structure and one-dimensional sequence string data. To this end, the genetic programming technique is applied to the obtained basic features extract a more discriminative feature vector. Then support vector machine is employed to determine the binding residue of each amino acid. Finally, the binding sites are predicted using the structure clustering algorithm on the obtained binding residues. The proposed method was evaluated on the Bio Lip dataset. The prediction rate of 92.76% and 93.09% were achieved when 10-fold cross-validation and independent test set respectively used. The acquired results were compared to the performance of other state-of-the-art methods. The proposed method achieves robust and consistent performance using sequence-based and structure-based features for both 10-fold cross-validation and independent tests.

1 Introduction

Nowadays, machine learning methods are used in various bioinformatics fields. In this regard, several bioinformatics applications can be mentioned, such as improving predictive performance, identifying complex protein structures, predicting protein-ligand interaction, performance acceleration of annotation, and determining the number of amino acid protein-peptide binding residues. These applications focus on vital macromolecules, called proteins. Proteins are complex organic compounds that interact with each other and other molecules through their binding sites. Hence, there are applications for prediction of binding sites such as \(\text{[1,2]}\): gene expression and metabolism, replication, Deoxyribose Nucleic Acid (DNA) repair, drug
Combination of Genetic Programming and Support Vector Machines (SVM) classifier using PSO (Particle Swarm Optimization) [9]. Taherzade et al. [10] also employed SVM algorithm and RF classifier along with Density-Based Spatial Clustering of Applications with Noise clustering (DBSCAN).

C) Some of the current studies focused on a specific ligand called the peptide. Peptides are small polymers composed of amino acids merged in a specific order through peptide bonds. Various computational methods have been proposed such as the "Pep-site" web server predictor [11, 12], the computational method based on DBSCAN [13], and optimized mapping protocols [14]. These studies have been carried out to predict peptide-binding sites due to peptide limitations such as small peptide size, weak binding affinity, and peptide flexibility.

D) Predicting protein-ligand binding sites is critical for understanding biological interactions, protein function analysis, cellular processes, and drug design and discovery. Therefore, computational methods such as 3D structure-based, template similarity-based, traditional machine learning-based [15], dynamic ensemble classifier [16], deep learning-based methods [17], machine learning method with classifier and cluster [18], and online predictor web server based on machine learning method [19] have been applied.

E) Proteins have various types of interactions with other proteins and molecules such as RNA (Ribonucleic Acid) and DNA. Protein-DNA interactions play a crucial role in biological processes such as DNA reproduction, transcription, bond, and repair [20]. Protein-RNA interactions have applications for biological processes such as sequence encoding, RNA transfer, gene regulation at the transcriptional and post-transcriptional levels [20].

F) Predicting the binding site type is important for identifying vital molecule function and interactions. The first prediction type is the protein-protein binding sites prediction [21]. The second prediction type is the protein-carbohydrate binding sites prediction [22, 23]. The third prediction type is the prediction of the RNA-binding protein [24], and DNA binding site prediction on the protein surface [25].

G) The amino acid residue is defined as the surface residue whose solvent accessible surface area is within 3.5 angstroms (Å) per heavy atom in the binding peptide [10, 26]. The binding site amino acid residues are identified using a distance cutoff of 3.5 angstroms (Å) between a heavy atom in protein and peptide. The binding amino acid residue is determined by the structural information and neighboring sequences involved in the interaction. Thus, the traditional machine learning algorithms have tried to predict protein-peptide binding residue focusing on the peptide ligand [5] using extra tree classifier, logistic regression, bagging classifier, gradient boosting classifier, and SVM [27]. Also, various classification algorithms such as decision tree (ID3 and C4.5), gradient boosting, and random forest were used [28]. On the other hand, several studies have been tried to predict protein-ligand binding residue that did not focus on a specific type of ligand using hybrid deep heterogeneous learning method [29].

H) The binding site is defined as a molecule surface area of amino acids like protein that interacts with the binding partner (ligands) to understand the function and behavior of the protein [7].
Therefore, several methods have been proposed to predict protein binding sites by employing different techniques such as two-steps random forest ensemble classifier \[30\], neighboring property \[31\], three steps computational method (local sequence alignment, protein surface detection, and 3D structures comparison) \[32\], 3D convolutional neural network \[33\], structure-based predictor web server \[34\], and deep learning \[35\].

In this study, we developed a machine learning-based method to predict protein-peptide-binding residues and binding sites. To this end, at first, both structural and sequence-based information is enhanced by employing genetic programming; and applied to a support vector machine classifier for prediction of binding residues. Then, the OPTICS (Ordering Points To Identify the Clustering Structure) clustering is employed on the detected binding residues to predict binding sites.

The rest of this paper is organized as follows: Section 2 describes the materials and methods, Section 3 presents the result and discussion and finally, the paper’s conclusion and future work are given in Section 4.

2 Materials and Methods

The binary protein-peptide binding site prediction problem is tried to predict the location of the amino acid residues that are more probably involved in the protein-peptide interaction. The binary means classifying any amino acid residues available in a protein sequence as binding or non-binding classes. To this end, various features include sequence-based and structure-based information can be employed. In this paper, eight basic features such as G-ASA (Group-Accessible Surface Area), CS (Conservation Scores), RSA (Relative Solvent Accessibility), DA (Dihedral Angle), G-HSE (Group-Half Sphere Exposure), G-PF (Sequence Profile Group from PSSM (Position-Specific Scoring Matrix)), and G-SS (Group-Secondary Structure) are extracted. Then Genetic programming (GP) along with the support vector machines is employed to learn a transform for generating new enhanced features based on all possible combinations of basic features. By employing these enhanced GP-based features in the SVM classifier with the RBF (Radial Basis Function) kernel binding residues of protein-peptide are predicted. Furthermore, the OPTICS clustering method is employed to detect the protein-peptide binding sites. The diagram of the proposed method is illustrated in Figure 1.

2.1 The Proposed Method

The proposed method is a machine-learning-based method for predicting protein-peptide binding sites by employing both sequence and structure-based features. The input of the proposed method is the corresponding string data in the form of protein sequences. String data cannot be directly applied to machine learning methods. Analyzing string data types without using automated tools is time-consuming, costly, and error-prone. Automatic acquisition of learning knowledge, generation of features with numerical values, and modification of the problem space provide the possibility of learning appropriate features of a specific task. Providing interpretable features for machine learning algorithms can improve their performance.
feature correlations, enhancing feature representation, and optimizing the number of features are implicitly provided by feature learning techniques. In the proposed method a new GP-based feature learning technique is employed. The details of different steps of the proposed method are given in the next.

2.1.1 Protein Sequence (Input Data)

In the proposed method the input protein sequence is inserted by FastA format designated by the IUPAC-IUB CBN (the International Union of Pure and Applied Chemistry International Union of Biochemistry Commission on Biochemical Nomenclature). In this format, the biochemical composition of a protein is determined by amino acid sequences bonded together through a peptide bond. Hence, a string of letters represents protein sequences. Each letter represents an amino acid residue. Arrangement of the amino acid chain starts at N-terminus and ends at C-terminus. In the FastA format, each sequence name has a unique identifier called PDB ID (Protein Data Bank Identifier). This name starts with symbol “¿” along with a string of unique letters and numbers. An example of the FastA file format related to a protein sequence is presented in Figure 2. As can be observed in this example, the characteristics of this protein sequence are PDB ID=¿ 1dpuA”, protein sequence length is 69 amino acid residues, N-term is A, and C-term is E.

2.1.2 Basic Feature Extraction

Since the objective of this paper is to present the machine learning-based method for protein-peptide binding sites prediction, various useful features, extracted from sequence and structural information, are employed. The list of these features is shown in Table 1. These features are named G-ASA, CS, RSA, DA, G-HSE, G-PF, G-SS. Features of the constituent amino acid residues are achieved in various manners. In the following, the description of each feature group is explained.

- **DA**: Protein backbone structure can be described by dihedral angles (DA). The dihedral angle feature indicates the existence of three angle types of Φ, Ψ, and ω in the amino acid side chain. Φ represents the bond between the central carbon and its adjacent amino group \(N - C_{\alpha}\). And Ψ indicates the bond between each amino acid’s central carbon and its adjacent carboxylic group \(C_{\alpha} - C\). Finally, ω is a bond between an amino group, corresponding amino acid, and a carboxylic group in its adjacent amino acid (C and N). The values of these angles are defined to vary between -180 to 180, which are normalized by the Equation (1) [36]:

\[
\text{Norm}(x) = \frac{x}{360.0}
\]

- **CS**: The conservation scores (CS) feature evaluates the sites of a multiple sequence alignment [36]. The goal is to identify critical amino acid residues for structure or function. This feature determines the number of species in each locus site that are in multiple alignments [37, 38].

- **RSA**: The relative solvent accessibility (RSA) feature represents the amount of the residual amino acids that are in contact with the residue solvent exposure [36]. This feature is computed as Equation (2) [39]:

\[
\text{RSA} = \frac{\text{ASA}}{\text{MAXASA}}
\]

<table>
<thead>
<tr>
<th>Feature</th>
<th>Name</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>DA</td>
<td>Dihedral Angles</td>
<td>Structure-based</td>
</tr>
<tr>
<td>RSA</td>
<td>Relative Solvent Accessibility</td>
<td>Structure-based</td>
</tr>
<tr>
<td>CS</td>
<td>Conservation scores</td>
<td>Sequence-based</td>
</tr>
<tr>
<td>G-SS</td>
<td>Secondary Structure feature group</td>
<td>Structure-based</td>
</tr>
<tr>
<td>G-PF</td>
<td>Sequence-based information</td>
<td>Sequence-based</td>
</tr>
<tr>
<td>G-HSE</td>
<td>Half Sphere Exposure feature group</td>
<td>Structure-based</td>
</tr>
<tr>
<td>G-ASA</td>
<td>Accessible Surface Area feature group</td>
<td>Structure-based</td>
</tr>
</tbody>
</table>

**Figure 2.** FastA Format With Unique PDB ID for the Protein Sequence.
According to Equation (2), the accessible surface area represents the corresponding area in contact with the solvent; moreover, Max ASA (Maximum Accessible Surface Area) indicates the maximum value of this area. The value is defined as the Boolean type of buried amino acids and exposed amino acids. The numerical value 1 is applied for the exposed amino acid and the numerical value 0 is applied for the buried amino acid.

- **G-PF**: The sequence profile group (G-PF) is derived from the position-specific scoring matrix of sequence profiles [10]. The position-specific scoring matrix is a $20 \times L$ dimensional matrix where $L$ is the length of the protein. This matrix is obtained through PSI-BLAST (Position-Specific Iterative Basic Local Alignment Search Tool) [10]. PSI-BLAST uses an E-value threshold of 0.001 in three iterations to extract the 20-dimensional vector for each amino acid in the protein. Then, entropy was calculated by Equation (3) [10].

$$ S_E = \sum_{j=1}^{20} P_{ij} \log(P_{ij}) $$

where $P_{ij}$ is the probability matrix at residue $i$ ($i = 1, \ldots, L$, $L$ is the length of protein)), and $j$ represents the index of 20 amino acids ($j = 1, \ldots, 20$). This entropy is applied to the PSSM matrix’s values to capture the information of protein.

- **G-ASA**: The accessible surface area feature group (G-ASA) is obtained in three steps [3]. The ASA value is initially obtained and normalized for all amino acid residues through DSSP (Dictionary of Protein Secondary Structure or Dictionary of Secondary Structure of Protein) [11]. The normalized value is then converted to a relative ASA. Eventually, the neighbor’s amino acid residues are averaged in a window size ranging from one to the optimal value for each acquired ASA. The value of $r$ is equal to the ASA average.

- **G-HSE**: Half sphere exposure feature group (G-HSE) indicates the degree to which an amino acid residue is buried in the protein’s three-dimensional structure [10]. The upper and lower half-spheres in the structure are acquired by considering a contact number sphere into two halves. These halves are achieved by perpendicular plane into the $C_\beta - C_\alpha$ vector. Finally, the number of structural neighbors of given amino acid in the upper and lower hemispheres is counted. This simple division of the contact number sphere is HSE up and HSE down. They represent the number of $C_\alpha$ atoms in the upper and lower hemisphere, respectively [12].

- **G-SS**: The secondary structure feature group (G-SS) is obtained from the secondary structure through DSSP for each amino acid residue [10]. The segment is obtained by applying a window size filter to the secondary structure. So, a segment is defined as continuous amino acid residues with the same type. The secondary structure is defined in three types such as helix, sheet, and coil [11]. The segment length and the position of the query amino acid residue in the segment are considered features.

The initial assumption in machine learning algorithms is that the data is noise-free, flawless, and valid. Therefore, the data pre-processing phase is an inseparable part of a process that is based on machine learning and data processing. The data pre-processing phase leads to performance optimization, computational complexity reduction, normal distribution for normal feature values, the equal and fair impact of all feature groups on the performance of the proposed method and achieving similar interval and scale for all feature groups. Consequently, the normalization operation (4) is applied to the extracted basic features and mapped their values in the interval of [0, 1] as shown in Equation (4) [13].

$$ x' = \frac{x - \min(x)}{\max(x) - \min(x)} $$

In Equation (4), $x$ and $x'$ denote the original and normalized values of the basic feature.

### 2.1.3 GP Based Feature Enhancement

In classification problems, the extraction of discriminant features has a high effect on the classifier’s performance. To elevate the performance of the classifier, some methods try to define and extract features manually by an expert, while some other methods try to find more discriminative features and discard not relevant features by employing different techniques such as principal components analysis and iterative feature selection. These feature selection techniques only try to find and select more discriminative features from existing features, but for employing more discriminant features it needs to use learning-based methods in the feature extraction process. One of the most prominent learning-based methods is genetic programming (GP). The GP can be used for learning and extracting high discriminative features or learning better mapping (or transfer) functions to elevate the discriminant of the classifier [44].

The feature learning can be explained as a transfer function mathematically. This transfer function can be written as below:

$$ EF_i = G_i \times BF_i $$

(5)
where BF represents the basic feature vector, $EF_i$ is the $i^{th}$ enhanced feature that is obtained from the mapping of basic feature vector using appropriate transfer function $G_i$. In the field of genetic programming-based feature learning, each transfer function $G_i$ is known as a gene (the $i^{th}$ gene), and a set of genes will form a chromosome (or a new mapping solution of the problem space). The GP first randomly creates some chromosomes (K chromosomes), which form an “initial population”. Each of these chromosomes, which is also known as a member of the population, has N distinct genes ($G_1, \ldots, G_N$), which are defined as follows:

$$C_k = \{G_{k,1}, \ldots, G_{k,N}\}$$  

where $C_k$ is the $k^{th}$ chromosome of the initial population, and $G_{k,j}$ is the $j^{th}$ gene of the $k^{th}$ chromosome. In the proposed model, each chromosome transforms the $M$-dimensional basic feature vector ($M$ is dependent on the input window size used for prediction of each residue) to a new $N$-dimensional space ($N$ is the number of genes and is set to 12 by trial and error). Each of these genes calculates one of the features of the new space ($EF_{k,i}$) by applying its specific mathematical operations to the basic feature data. In the GP the representation of gene usually is expressed as a syntax tree. Each tree includes various components such as the root, internal and terminal nodes. In the GP, the variables (basic features) and constants are called terminals. The arithmetic operations are called internal nodes as transfer functions. An example of a tree representation of a gene is shown in Figure 3. In this figure terminal nodes, F1 to F5 represent basic features.

To obtain the mathematical representation of each feature in the new space ($EF_{k,i}$), the corresponding gene tree ($G_{k,i}$) should be parsed using in-order method. An example of parsing is shown in Figure 4. In this figure in addition to in-order parsing method (left-child, node, right-child), pre-order (node, left-child, right-child) and post-order (left-child, right-child, node) parsing methods are shown. The obtained expressions are as: for in-order: $(4-5)+p\ max\ best)$, for pre-order: $(+(-4)\ (max\ p\ best))$, and for post-order: $((-4)\ (p\ best\ max)+)$.

GP is a systematic and domain-independent problem-solving technique that automatically solves problems without requiring the structure of the solution in advance. It is operating with a population of solutions that are iteratively improved using three boosting mechanisms: selection, crossover, and mutation. The pseudo-code of the proposed GP-based feature learning process is shown in Algorithm 1.

In this algorithm at first, initial population, consist of N identical chromosomes, is generated randomly (lines 1-5). Then existing population is evaluated using training basic feature data using SVM classifier (line 6). In the iterative process, the existing population’s performance is improved and new populations are generated by applying selection, crossover, and mutation boosting mechanisms on the current population (lines 10-13). The performance of the new population was also evaluated using an SVM classifier on training data. These processes are repeated until we reach the final generation or the performance of the current population reaches maximum fitness (lines 14-16).

To solve a problem through genetic programming, the details of it must be appropriately selected according to the problem. These details include arithmetic operators, operational parameters (selection, crossover, and mutation), fitness function, and initial population strategy, the number of generations, population size, and maximum tree depth. Table 2 shows the list of arithmetic operators used in this paper.

In Table 2, various types of arithmetic operators, descriptions, and executable operators are listed. For
Algorithm 1 GP Based Feature Learning Procedure.

**Input:** Set of normalized structured and sequence basic features  
**Output:** Return the best Chromosome as the final transfer function

**Begin**

01. **Initialization:** generate initial random population P (Chromosome Set): \( P \leftarrow \{ \} \)
02. **for** \( i : 1 \) to \( N \) **do**
03. generate a random Chromosome \( C_i \)
04. **if** (\( C_i \) is not a member of the initial population \( P \)) **then** \( P \leftarrow \{ P \cup C_i \} \)
05. **end for**
06. \( EP \leftarrow \) Compute fitness of \( P \)
07. \( g \leftarrow 1 \)
08. **While** ((Max(\( EP \)) \( \neq \) maximum fitness) \( \land \) (\( g \) \( \neq \) final generation)) **do**
10. **apply Selection** \( SS \leftarrow SelectTopN(P, T_s) \)
11. **apply Crossover** \( CS \leftarrow Crossover(P, T_c, \sigma) \)
12. **apply Mutation** \( MS \leftarrow Mutation(CS, \mu) \)
13. **update population** \( P \leftarrow \{ SS \cup CS \cup MS \} \)
14. \( EP \leftarrow \) Compute fitness of \( P \)
15. increase generation number \( (g \leftarrow g + 1) \)
16. **end while**
17. \( C_{best} \leftarrow SelectTopN(P, 1) \)
18. **return** \( (C_{best}) \)

**End**

Table 2. List of Arithmetic Operations Used in Genetic Programming.

<table>
<thead>
<tr>
<th>Description</th>
<th>Operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. MUL</td>
<td>Multiplication operator</td>
</tr>
<tr>
<td>2. DIV</td>
<td>Division operator</td>
</tr>
<tr>
<td>3. SIN</td>
<td>Sine operator in radians</td>
</tr>
<tr>
<td>4. COS</td>
<td>Cosine operator in radians</td>
</tr>
<tr>
<td>5. LOG</td>
<td>Logarithm</td>
</tr>
<tr>
<td>6. EXP</td>
<td>Exponential of a value</td>
</tr>
<tr>
<td>7. SUB</td>
<td>Subtraction operator</td>
</tr>
<tr>
<td>8. ADD</td>
<td>Addition operator</td>
</tr>
<tr>
<td>9. CONST</td>
<td>Constant value output</td>
</tr>
<tr>
<td>10. A sin</td>
<td>Arcsine of a value</td>
</tr>
<tr>
<td>11. A cos</td>
<td>Arc cosine of a value</td>
</tr>
<tr>
<td>12. A tan</td>
<td>Arctangent of a value</td>
</tr>
<tr>
<td>13. SQRT</td>
<td>Square root of a value</td>
</tr>
<tr>
<td>14. SQUARE</td>
<td>Power tow of a value</td>
</tr>
<tr>
<td>15. SOFT PLUS</td>
<td>0.2log1.0 + exp a</td>
</tr>
<tr>
<td>16. CUBE</td>
<td>Third power of the operator</td>
</tr>
<tr>
<td>17. HAT</td>
<td>Hat function of operator a</td>
</tr>
<tr>
<td>18. GAUSS</td>
<td>exp(-5.0a2)</td>
</tr>
<tr>
<td>19. ABS</td>
<td>Absolute operator</td>
</tr>
<tr>
<td>20. EQA</td>
<td>Operator a</td>
</tr>
</tbody>
</table>

example, each gene is defined as a mathematical equation of input characteristics. Thus, each gene can be represented as a tree form with mathematical operators. This gene can automatically create a new feature by applying mathematical operators to several features. The GP can generate multi genes to create several sets of new features.

One of the boosting operational parameters of genetic programming is the selection strategy. The selection operator is usually the first execution operator applied to manipulate the population. Its objective is to the production of new populations through the reproduction of existing chromosomes in the current population. Meanwhile, the newly produced population of the next generations must represent a better fit than the current generation population. There are different types of selection operators, such as selection scaling, roulette-wheel selection, tournament selection, and other types. In the present study, tournament selection has been applied. The pseudo-code of the selection parameter is presented in Algorithm 1. Line 01 evaluates the performance of all individuals in the population using the SVM classifier. Then \( T_s \) times a subset of population \( P \) is selected that their obtained performance measure \( EP \) is greater than predefined fitness (lines 4-9). The first top performance of chro-
Algorithm 2 SelectTopN.

**Input:** Current population P and the number of selected top $T_s$ chromosome

**Output:** Set of $T_s$ top chromosome (SS)

**Begin**

01. EP ← Compute fitness of P
02. $t$ ← 0, SS ← {} 
03. While ($t < T_s$) do
04. SP ← Select a subset of P randomly
05. MSP ← Select the Chromosome with the highest fitness on subset SP
06. if (fitness(MSP) $>\theta)$ then
07. SS ← $SS \cup MSP$
08. $t$ ← $t+1$
09. end if
10. end while
11. return (SS)

**End**

Algorithm 3 Crossover Procedure.

**Input:** Current population P, $T_c$ number of crossover chromosomes and crossover rate $\sigma$

**Output:** Set of $T_c$ crossed chromosomes (CS)

**Begin**

01. EP ← Compute fitness of P
02. $t$ ← 0, CS ← {} 
03. While ($t < T_c$) do
04. SP ← Select a subset of P randomly
05. FSP ← Select a chromosomes with the highest fitness on subset SP
06. SSP ← Select a chromosomes with 2nd highest fitness on subset SP
07. for $i=1:N$ do SSP ← Select a chromosomes with 2nd highest fitness on subset SP
08. $q$ ← random (0,1)
09. if ($q > \sigma$) then
10. temp ← SSP(1:i)
11. SSP(1:i) ← FSP(1:i)
12. FSP(1:i) ← temp
13. end if
14. end for
15. CS ← $CS \cup FSP \cup SSP$
16. $t$ ← $t+2$
17. end while
18. return (CS)

**End**

The crossover operator is the second operation applied to two individuals, with the highest performance of each tournament subset. This operator aims to obtain offspring chromosomes by combining two chromosome genes, that demonstrate better characteristics than their parent chromosomes. There are various types of crossover operators such as single-point crossover, multi-point crossover, uniform, and other types. In the current study, the uniform crossover is selected. The pseudo-code of the crossover parameter is presented in Algorithm 3.

Line 01 evaluates the performance of all individuals in the population using SVM classifier. Then $T_c$ times a subset of population P is selected using tournament procedure. The first two top-performance chromosomes are selected from each subset. For each gene, a random number between 0 to 1 is generated and if the generated value is greater than crossover rate $\sigma$ the genes of two chromosomes at this point are exchanged. The final crossed chromosomes are added to the list of new populations.

The mutation is another operation applied to chromosomes from the uniform crossover. The mutation operator modifies the internal node of genes in each chromosome. In other words, it is essential to maintain diversity in the chromosome population and to pre-
Algorithm 4 Mutation Procedure.

Input: Current Crossover population CS, and mutation probability $\mu$

Output: Set of $T_c$ mutated chromosomes (MS)

Begin
01. $\text{MS} \leftarrow \{\}$
02. for each chromosome $C_i \in CS$ do
03. for each gene $G_{i,j} \in C_i$ do
04. for each node $node_k \in G_{i,j}$ do
05. $q \leftarrow \text{random} (0,1)$
06. if $(q > \mu)$ then
07. select operation type or feature type of $\hat{node}_k$ randomly
08. $\hat{G}_{i,j}.\text{node}_{k} \leftarrow \hat{node}_k$
09. else
10. $\hat{G}_{i,j}.\text{node}_{k} \leftarrow G_{i,j}.\text{node}_{k}$
11. end if
12. end for
13. end for
14. $\hat{C}_i \leftarrow C_i$
15. end for
16. return (MS)
End

To prevent early convergence. This parameter provides the randomness and the possibility of escaping from local optimization points. The pseudo-code of the mutation parameter is presented in Algorithm 4.

In this algorithm for each node of each gene in all crossover chromosomes, a random number is generated. This number is between zero and one (lines 2-5). If the generated random number is less than or equal to the mutation probability ($\mu$), the type of that node randomly will be changed to other possible mathematical functions or basic features. In this work, the mutation probability is assumed to be 0.5 (Line 10). Furthermore, line 2 points out that the output is a mutated population in which the uniform crossover and tournament selection executive operators have applied in the previous steps.

Finally, the parameters and tree structures of the best chromosome ($C_{\text{best}} = \{G_{\text{best},1}, \ldots, G_{\text{best},N}\}$) selected by GP are extracted as learning functions to enhance the basic features and produce a higher prediction rate in the test phase:

$$EF_i = G_{\text{best},i}(\overrightarrow{BF})$$ (7)

where $G_{\text{best},i}$ is the $i^{th}$ genome of the best chromosome obtained by GP, and applied it on the basic features to produce the $i^{th}$ enhanced feature. This set of new enhanced features (EF) are employed in the test phase to predict the binding of amino acid residues.

Note: $\overrightarrow{BF}$ represents the enhanced feature of amino acid residues.

### Table 3. The Details of OPTICS Clustering Algorithm Parameters for Prediction of Protein-Peptide Binding Sites.

<table>
<thead>
<tr>
<th>Parameter Value</th>
<th>Parameter Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>$\varepsilon$</td>
</tr>
<tr>
<td>3</td>
<td>MinPts</td>
</tr>
<tr>
<td>2</td>
<td>CoreDistance</td>
</tr>
<tr>
<td>5</td>
<td>Reachability-Distance</td>
</tr>
</tbody>
</table>

2.1.4 Residue Binding Prediction

In the proposed method, the prediction of the binding amino acid residues in the protein-peptide interaction is done by applying the SVM classifier with Radial Basis Function (RBF) kernel on the enhanced features [46]. The RBF kernel is exploited to map the input data to the defined feature space associated with data resolution [47, 48]. This classification is performed for all residues because the purpose was to separate the binding amino acid residues from the non-binding amino acid residues. Although we can only use the enhanced features of the corresponding amino acid, to employ the effect of neighboring amino acids in the binding of each amino acid we use the enhanced features of three adjacent neighbors:

$$EF_{\text{curr}(i)} = \{EF_{i-1} \cup EF_i \cup EF_{i+1}\}$$ (8)
Figure 5. The Schematic Diagram of the Predictive Method for Prediction of the Protein-Peptide Binding Site Using Protein Sequence.

\[ BR_i = SVM \left( EF_{curr(i)} \right) = \begin{cases} 1 & EF_{curr(i)} \in \text{binding residue} \\ 0 & \text{otherwise} \end{cases} \] (9)

2.1.5 Binding Sites Detection Using OPTICS Clustering

After detection of protein-peptide binding residues, the OPTICS clustering algorithm is employed to predict protein-peptide binding sites. This clustering is performed on protein-peptide binding residues. The binding sites contain the most significant binding amino acid residues. The OPTICS’s steps are as follows [49]:

i. Sorting all points in the obtained binding residues (BS) vector based on their density using a statistical distribution called GD (Gaussian Distribution).

ii. Selecting an unvisited sample (a point in the obtained binding residues (BS) vector) such as p.

iii. Examining the neighborhood of sample p according to the radius of epsilon (\(\varepsilon\)) in the BS vector space.

iv. If the algorithm reaches the minimum number of points (MinPts) in the specified epsilon radius for sample p, a cluster will be created with p as the center. Moreover, all the visiting points have belonged to that cluster. The process of visiting points continues until a new point is found in the neighborhood.

v. If p is a boundary point, none of the points with p centrality will belong to a cluster, and other points are visited.

vi. Repeating the process of visiting points until all points are visited.

The OPTICS algorithm depends on two input parameters called neighborhood radius (\(\varepsilon\)) and minimum available points in a cluster (MinPts). Also, there are two criteria of core distance and reachability distance for each sample. The details of the employed OPTICS clustering parameters are listed in Table 3.

In the following, an example of a proposed predictor method aiming to predict protein-peptide binding sites is presented in Figure [5]. As demonstrated in Figure [5] for example, a protein sequence, with protein data bank identification = "1dpuA" that is derived from the dataset [10], was fed to the proposed method as input. Subsequently, the problem output, the protein-peptide binding site, which corresponds to the related input, was obtained by applying the machine learning method. The details of the predictive model (based on machine learning techniques) are illustrated in three phases.

In the following the definition of these parameters is represented.

- \(\varepsilon\) is the neighborhood radius
- MinPts is the minimum number of points in the cluster to form the cluster
- CoreDistance(p) is the minimum required radius to classify non-center points to the center point called p.
- ReachabilityDistance(o,p) indicates the distance between two cluster’s center o and p that can be merged and belong to a similar cluster.

3 Results and Discussion

3.1 Data Set

BioLip [50] is a semi-curated database that derives protein data from the protein data bank. BioLip provides the most comprehensive database for ligand-protein binding interactions, virtual ligand screening, and protein function annotation. Hence, for protein-peptide binding data extraction from BioLip, the following conditions were applied to this dataset like Taherzadeh et al. [10].

i. Considering the peptide as a chain whose length is shorter than 30 amino acid residues.

ii. Removing peptide-binding proteins with a length shorter than three binding amino acid residues.

iii. Considering a cut-off value of 3.5Å to define binding residues.
### Table 4. Summary of the Final Dataset.

<table>
<thead>
<tr>
<th>N_{NBR}</th>
<th>N_{BR}</th>
<th>N_{R}</th>
<th>Number Protein</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>280920</td>
<td>16678</td>
<td>297598</td>
<td>1241</td>
<td>Protein-Peptide Complexes</td>
</tr>
<tr>
<td>251769</td>
<td>14959</td>
<td>266599</td>
<td>1116</td>
<td>TR (training set)</td>
</tr>
<tr>
<td>29151</td>
<td>1719</td>
<td>30870</td>
<td>125</td>
<td>TS (independent test set)</td>
</tr>
</tbody>
</table>

### Table 5. Performance Evaluation Based on Bio and Informatics Criteria.

<table>
<thead>
<tr>
<th>Name of Metric</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>True Positive (TP)</td>
<td>The number of actual binding residues</td>
</tr>
<tr>
<td>False Positive (FP)</td>
<td>The number of actual non-binding residues incorrectly predicted as binding sites</td>
</tr>
<tr>
<td>False Negative (FN)</td>
<td>The number of actual non-binding residues correctly predicted as non-binding sites</td>
</tr>
<tr>
<td>Mathews Correlation</td>
<td>( \frac{(TP+TN)-(FP+FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} )</td>
</tr>
<tr>
<td>F-Measure (F-M)</td>
<td>( \frac{2TP}{2TP+FP+FN} )</td>
</tr>
<tr>
<td>False Positive Rate (FPR)</td>
<td>( \frac{FP}{TP+FP} )</td>
</tr>
<tr>
<td>Specificity (SPE) / True Negative Rate (TNR)</td>
<td>( \frac{TN}{TP+TN} )</td>
</tr>
<tr>
<td>Sensitivity (SEN) / True Positive Rate (TPR)</td>
<td>( \frac{TP}{TP+FN} )</td>
</tr>
<tr>
<td>Receiver operating characteristics (ROC)</td>
<td>A curve created by plotting the true positive rate (TPR or sensitivity) against the false positive rate (FPR)</td>
</tr>
<tr>
<td>Area Under Curve (AUC or AUROC)</td>
<td>The area under the receiver operating characteristics (ROC) curve</td>
</tr>
<tr>
<td>Performance comparison on coverage and precision</td>
<td>Coverage as a function of precision</td>
</tr>
<tr>
<td>ACC</td>
<td>( \frac{TP+TN}{TP+FP+FN+TN+FN} )</td>
</tr>
</tbody>
</table>

iv. Employing the "blastclust" toolkit to cluster the protein chains with at least 30% sequence similarity and proteins that had more than 30% similarity were removed.

v. Randomly selecting a representative chain from each cluster.

vi. The test set included 10% of the protein complexes that were randomly selected. This set is called an independent test set.

vii. Another 10% of the protein complexes were randomly selected and were employed as the training set.

viii. All the subsets have a class ratio of approximately 1:17. This means, on average, there is about 1 amino acid residue in the protein sequence is classified as binding, and there are around 17 amino acid residues in the protein sequence, which are classified as non-binding in a protein segment whose length is 18 amino acid residues.

Subsequently, the final data set was acquired according to the above conditions. The details of the final data set are listed in Table 4. In this table the column N_{BR} represents the total number of amino acid residues (binding and non-binding)), the column NBR is the number of binding residues (which indicates the number of amino acid residues involved in the protein-peptide interaction), and the column NNBR indicates the number of non-binding residues (the number of amino acid residues that are not involved in the protein-peptide interaction).

In this regard, certain items were applied to the final dataset, which are:

i. Applying the cross-validation technique on the training set to overcome limitations such as the lack of similar protein in the test and training sets and reduce the phenomenon of over-training.

ii. Random consideration of 10% of protein complexes as an independent test set. The resulting training and independent test sets were acquired according to the above conditions.
Table 6. Comparison Between Different Machine Learning Methods on the 10-Fold-Cross-Validation With Window Size=3.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Training Data</th>
<th>Test Data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ACC</td>
<td>Time(ms)</td>
</tr>
<tr>
<td>Support Vector Machine</td>
<td>0.74</td>
<td>787</td>
</tr>
<tr>
<td>Random Forest</td>
<td>0.7301</td>
<td>1356</td>
</tr>
<tr>
<td>Decision Tree (C4.5)</td>
<td>0.7299</td>
<td>756</td>
</tr>
<tr>
<td>Decision Tree (ID3)</td>
<td>0.7286</td>
<td>746</td>
</tr>
<tr>
<td>Gradient Boosting</td>
<td>0.7101</td>
<td>599</td>
</tr>
</tbody>
</table>

Figure 6. The Obtained Performances of the SVM Classifier.

3.2 Performance Evaluation

The performance of the proposed method is measured by various performance evaluation metrics listed in Table 5. These metrics have been used to evaluate the performance of the proposed method and to compare it with some of the existing state-of-the-art methods.

To visualize the performance of the binary prediction problem, the proposed method has been evaluated in terms of bio and informatics criteria. The performance of the machine learning-based method has compared to structure-based studies such as "PepSite" [11, 12], "FrustraPocket" [38], "3D-Segmentation" [51], and "SPRINT-Seq" [3] as a sequence-based method. The performance of machine-learning based was evaluated by using datasets [10] from the BioLip database [50]. In the following, the evaluation steps are explained.

3.2.1 Parameters Evaluation

At first, the effect of employing different parameters and techniques was evaluated. The performance of the proposed method was evaluated using 10-fold cross-validation as well as using the independent test. In 10-fold cross-validation, the dataset was divided into $k = 10$ approximately equal subsets. In each iteration, one fold was used for testing, and $k - 1$ folds were employed to train the classifier. This procedure was repeated 10 times that each fold is used as a test set. Finally, the average accuracies of all folds are used as the final prediction. At first, the effect of the classifier was investigated. To this end, five different classifiers including support vector machine, random forest, decision tree (model ID3 and C4.5), and gradient boosting were employed. The obtained results of conventional machine learning algorithms in predicting the protein-peptide binding residues are reported in Table 6. In these experiments for each residue, the basic features of left and right neighbors are also used as a feature vector for residue binding prediction (window size=3).

According to the obtained results (Tables 6), although the execution time of the SVM-based algorithm is longer than other competing algorithms, for both training and test data it has the highest prediction accuracy compared to other competing algorithms. Therefore, in the final proposed method SVM classifier with RBF kernel was used. The obtained results for other evaluation metrics for SVM-based classifier are shown in Figure 6.

Also the effect of employing GP-based feature enhancement is evaluated. The performance of the proposed method was compared with the situations that only SVM (without GP) and GP-based feature enhancement along with simple K nearest neighboring (KNN) classifier (GP& KNN). The obtained results are shown in Table 7. In this paper, the parameters of GP were set as shown in Table 8.
Table 7. Comparison Between Different Machine Learning Methods on Independent Test Set With Window Size-3.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Feature space dimension</th>
<th>Training Data</th>
<th>Test Data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ACC</td>
<td>Time(ms)</td>
</tr>
<tr>
<td>SVM</td>
<td>33</td>
<td>0.74</td>
<td>799.872</td>
</tr>
<tr>
<td>GP &amp; KNN</td>
<td>17</td>
<td>0.8899</td>
<td>31160.103</td>
</tr>
<tr>
<td>GP &amp; SVM</td>
<td>12</td>
<td>0.9276</td>
<td>38866.207</td>
</tr>
</tbody>
</table>

Table 8. Parameter Settings for Genetic Programming Based on a Support Vector Machine.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fitness</td>
<td>Calculate the classification accuracy</td>
</tr>
<tr>
<td>Fitness Function</td>
<td>Based on SVM</td>
</tr>
<tr>
<td>Initial Population Strategy</td>
<td>Ramped half and half-method</td>
</tr>
<tr>
<td>Number of generations</td>
<td>1000</td>
</tr>
<tr>
<td>Population Size</td>
<td>1000</td>
</tr>
<tr>
<td>Cross Over Strategy</td>
<td>Uniform</td>
</tr>
<tr>
<td>Cross Over Rate</td>
<td>0.599</td>
</tr>
<tr>
<td>Mutation Strategy</td>
<td>Mutation</td>
</tr>
<tr>
<td>Mutation Rate</td>
<td>0.5</td>
</tr>
<tr>
<td>Selection Strategy</td>
<td>Tournament</td>
</tr>
<tr>
<td>Size of Tournament</td>
<td>6</td>
</tr>
<tr>
<td>Max Tree Depth</td>
<td>10</td>
</tr>
<tr>
<td>Competition size</td>
<td>5</td>
</tr>
<tr>
<td>Reproduction Rate</td>
<td>0.1220</td>
</tr>
<tr>
<td>Minimum initial tree size</td>
<td>3</td>
</tr>
<tr>
<td>Maximum initial tree size</td>
<td>6</td>
</tr>
<tr>
<td>Terminal condition</td>
<td>Max generation</td>
</tr>
</tbody>
</table>

According to Table 7, the GP-based feature learning algorithm increased the performance of the residue binding detection process. Also employing SVM with RBF kernel shows higher accuracy. It is also clear that employing optimization algorithms such as GP need more training time.

Genetic programming algorithm indicates its robustness and consistency in learning various features with different bioinformatics. Although the optimality of the number of attribute groups is hidden in the GP structure, the SVM fitness function has improved the performance of the GP algorithm. In this paper, the GPTIPS2.0 [45] and the LibSVM [46] packages have been employed for genetic programming and the support vector machine, respectively.

Also, the effect of neighbor amino acids on residue binding detection was investigated. To this end, different window sizes including 1, 3, 6, 9, 12, and 15 for extracting enhanced features were used. The purpose of employing window size technique is to improve the performance of the classifier, and optimize the protein and peptide interaction based on neighboring amino acid residue information. The obtained results for these experiments are shown in Figure 7. Based on the obtained results, when the window size was set to 3 the proposed method has the highest accuracy compared to other window sizes.

3.2.2 Comparison of Residue Prediction Results With Similar Studies

Since the goal in this section is to predict protein-peptide binding residue, the evaluation of the proposed method has been carried out in comparison to the state-of-the-art methods. To this end, the methods based on the structure features including "PepSite" [11, 12], "FrustraPocket" [38], and "3D-segmentation" [51] and a method based on sequence information named "SPRINT-Seq" [3] are compared when the window size was set to 3 for all methods. The obtained results are shown in Figure 7.

As demonstrated in Figure 8, the evaluation of the results on the independent test set has led to the prediction of binding and non-binding residues. Evaluation of the proposed method (SVM + GP) has higher performance in terms of ACC, AUC, F−M, MCC, and SEN measures compared to other existing methods. The value of its SPE measure is also comparable with others.

The ROC (Receiver Operating Characteristic) curve is a curve created by plotting TPR (the true positive
rate) over the FPR (false positive rate). So, for visual evaluation of performance, the ROC curves of all methods were extracted. And depicted in Figure 9. From the obtained results, the proposed method has a higher curve and also has a higher AUC (Area Under Curve) compared to existing sequence-based and structure-based methods. A method that predicts protein-peptide binding residues randomly will have a linear diagonal curve and the area under the ROC curve (AUC) will be 0.5, whereas the best predictor will be higher and have $AUC = 1$. The proposed method has achieved the highest AUC value of 0.6999 compared to the second AUC value of 0.68 obtained for the "SPRINT-Seq" method.

### 3.2.3 Evaluation of OPTICS Based Binding Sites Prediction

A fundamental step in predicting protein function is to recognize the interval of involved amino acids in the protein-peptide interaction. The binding site is a location containing the most important amino acid residues involved in the protein-peptide interaction. Therefore, the OPTICS clustering was employed to identify the amino acid residues sites involved in the interaction. One of the advantages of the proposed method is the prediction of the exact regions of the amino acid residues involved in the interaction. In the current study, two standard clusters named OPTICS and DBSCAN were employed to predict the exact regions of amino acid residues involved in the protein-peptide interaction. The obtained results represent the superiority of the performance of OPTICS clustering. These results for protein sequence with PDB ID="1dpuA" were selected to evaluate these clusters. This protein sequence contains 69 amino acid residues, where 11 of them are actual binding sites. The proposed method using the OPTICS clustering predicted 9 binding sites of them and predicted 8 binding sites using DBSCAN clustering. The outputs of the proposed method with different clustering algorithms are exhibited schematically for the mentioned PDB ID in Figures 10 and 11. According to these figures, the OPTICS clustering was justified and employed as the clustering algorithm in the proposed method.

Also, the coverage rate of the proposed method has been investigated from different perspectives such as bio and informatics. In this section, the bioinformatics evaluation criterion was employed to evaluate the performance of the proposed method.

This bioinformatics criterion is expressed as a coverage metric in terms of precision. The coverage metric is the ratio of the correctly predicted peptide binding sites to all actual binding regions with a given precision:

$$\text{precision} = \frac{Si}{Bi}$$

where $Si$ and $Bi$ represent the number of correct positive amino acid residues in the predicted binding region, and the total number of actual amino acid residues, respectively.

Figure 12 designs coverage as a function of precision. The coverage was gotten decreased along with increasing precision. Coverage and accuracy are in-
versely related. The proposed method has a higher coverage rate in the same precision. For example, the amount of coverage at a constant precision of 0.50 using the proposed method is higher coverage than the "SPRINT-Seq" method. The proposed method has coverage higher than SPRINT-Seq at all precision levels. As can be achieved in Figure 12, the "Pepsite" method was displayed at low precision and the proposed method was displayed at high precision.

Finally, the convergence rate of the proposed method was investigated and depicted in Figure 13. Proper convergence rate causes accelerate the search space properly to reach the optimal point with appropriate speed. Figure 13 shows a convergence diagram based on the SVM classifier error rate with a specified number of iterations using the Equation (11):

\[
Err = \frac{1}{n} \sum_{i=1}^{N} (\text{if} \ classifier \ class(i) = \text{class}(i) \ \text{then} \ 0 \ \text{else} \ 1) \times 100
\]

According to Equation (11), the error rate is defined as the sum of the number of points that are incorrectly classified divided by the total number of points in the data. The proposed method needs more iterations to converge.

4 Conclusions and Future Works

Predicting the binding site is significant in understanding molecular mechanisms for predicting protein function, drug design, cellular signaling, cellular processes, and cancer treatment.

Experimental methods are time-consuming, costly, and challenging. In this work, we have designed a machine learning-based method for the prediction of protein-peptide binding sites directly from the protein sequence. In the proposed method a combination
of seven structure-based and sequence-based feature groups including group-accessible surface area, conservation scores, relative solvent accessibility, dihedral angle, group-half sphere exposure, sequence profile group from PSSM, and group-secondary structure were employed.

Moreover, the values of various features were normalized in the interval of \([0, 1]\). Then GP with the SVM fitness function was employed as the feature learner and SVM (RBF kernel) as the classifier. The output of SVM was presented as protein-peptide binding residues. Additionally, the OPTICS algorithm was employed as a clustering algorithm to predict protein-peptide binding sites. The focus of the current study was based on selecting genetic programming as a feature learner. The most fundamental reasons for this selection are as follows

i. Transferring features through automatic arithmetic operators from one space to another and obtaining new features based on mathematical transformations.
ii. Achieving all the optimal combinations of features with various natures
iii. Reducing the dimensions of the target problem space
iv. Improving the performance of machine learning tasks such as classification and clustering algorithms

Consequently, the proposed method demonstrated a superior performance using the derived dataset from the BioLip database compared to similar studies. Similar studies are structure-based and/or sequence-based methods. These methods are “Pepsite”, “FrustraPocket”, “3D-segmentation”, and “SPRINT-Seq”. Ultimately, the machine-learning-based method demonstrates superior performance to other competitive methods through bio-based and informatics-based evaluations. These evaluations show that machine learning can be employed for the prediction of protein-peptide binding sites effectively.

In the future studies in the bio section we will try to use the autoencoder as a learner. Combining OPTICS clustering algorithm with metaheuristic algorithms to further improvement of results. Also, using a predictive architecture based on ensemble mode can be investigated. Future studies in the informatics section include using the proposed model for other macromolecules such as carbohydrates also can be investigated.

References


Shima Shafiee is currently pursuing a Ph.D. degree at the Department of Computer Engineering and Information Technology, Razi University, Kermanshah, Iran. Shima researches machine learning, bioinformatics, evolutionary computation, optimization, and linear algebra. Nowadays, she is doing computational prediction of protein interactions with deep learning and machine learning-based methods in her Ph.D. thesis at Abdolhossein Fathi’s Lab.

Dr. Abdolhossein Fathi is an Associate Professor at the Department of Computer Engineering and Information Technology, Razi University, Kermanshah, Iran. His research interests include image and video processing, pattern recognition, medical image analysis, biomedical data compression, noise removal algorithms, biometric analysis, design of biometric systems, design of biometric systems, and HDLs (Hardware Description Languages) hardware modeling (VHDL).