Zhang et al. BMC Bioinformatics (2016) 17:323
DOI 10.1186/s12859-016-1201-8

BMC Bioinformatics

RESEARCH ARTICLE Open Access

Identification of DNA-binding proteins using multi-features fusion and binary firefly optimization algorithm

Jian Zhang[1], Bo Gao[1], Haiting Chai[1], Zhiqiang Ma[1] and Guifu Yang[1,2*]

Abstract

Background: DNA-binding proteins (DBPs) play fundamental roles in many biological processes. Therefore, the developing of effective computational tools for identifying DBPs is becoming highly desirable.

Results: In this study, we proposed an accurate method for the prediction of DBPs. Firstly, we focused on the challenge of improving DBP prediction accuracy with information solely from the sequence. Secondly, we used multiple informative features to encode the protein. These features included evolutionary conservation profile, secondary structure motifs, and physicochemical properties. Thirdly, we introduced a novel improved Binary Firefly Algorithm (BFA) to remove redundant or noisy features as well as select optimal parameters for the classifier. The experimental results of our predictor on two benchmark datasets outperformed many state-of-the-art predictors, which revealed the effectiveness of our method. The promising prediction performance on a new-compiled independent testing dataset from PDB and a large-scale dataset from UniProt proved the good generalization ability of our method. In addition, the BFA forged in this research would be of great potential in practical applications in optimization fields, especially in feature selection problems.

Conclusions: A highly accurate method was proposed for the identification of DBPs. A user-friendly web-server named iDbP (identification of DNA-binding Proteins) was constructed and provided for academic use.

Keywords: DNA-binding proteins, Binary firefly algorithm, Feature selection, Parameters optimization

Background

DNA-binding proteins (DBPs) are fundamental in many biological processes, such as recognition of specific nucleotide sequence, regulation of gene, transcription and translation, and DNA replication and repair [1, 2]. Thus, it is highly desirable to develop effective DBP identification methods. Traditionally, experimental techniques, which include filter binding assays [3], X-ray crystallography [4] and genetic analysis [5], are used to identify DBPs. Although these techniques can produce detailed information and provide confident assertion of the DBPs, they are both expensive and time-consuming. This spurred the development of computational methods to tackle this problem.

These computational methods can be divided into two categories: structure-based methods [6–8] and sequence-based methods [9–15]. Many of the early methods are structure based. Gao et al. [6] developed a knowledge-based method named DNA-binding Domain Hunter for identifying DBPs and associated binding sites using structural comparison. Zhao et al. [7] proposed a template-based prediction method by employing both structural similarity and binding affinity. Nimrod et al. [8] recruited random forests to identify DBPs by detecting evolutionarily conserved regions and using electrostatic features. However, the number of proteins with well annotation and good resolution structure are very limited. The structure-based methods may break down when homogeneous structures of a query protein is not available. Hence, many sequence-based methods had been proposed to deal with this problem. Kumar et al. [9] utilized various SVM modules and evolutionary information to forge the DNA-binder method. Kumar

* Correspondence: guifuyang.nenu@gmail.com
[1]School of Computer Science and Information Technology, Northeast Normal University, Changchun 130117, People's Republic of China
[2]Office of Informatization Management and Planning, Northeast Normal University, Changchun 130117, People's Republic of China

et al. [10] employed random forest to predict DBPs. Lin et al. [11] proposed the iDNA-Prot predictor by incorporating the features into the general form of pseudo amino acid composition that were extracted from protein sequence via the grey model and adopting the random forest operation engine. Song et al. [12] and Xu et al. [13] both applied the ensemble learning technique combined with hybrid features to predict DBPs. Zou et al. [14] conducted a comprehensive feature analysis of four categories of protein properties and three different feature transformation methods to find an optimal prediction model. Lou et al. [15] predicted DBPs by performing feature ranking with random forest and feature selection with forward best-first strategy. The features comprised properties from primary sequence, predicted structures and sequence alignment.

Although many efforts were put on the computational identification of DBPs, the prediction performance was still far from satisfactory. There are some possible reasons: (i) structure-based methods can provide reliable results in recognizing specific proteins. However, the insufficiency in known DBP structures leads to limited applications of these methods. Sequence-based methods are featured by their widely application, while the performance of these predictors are usually not as good as expected; (ii) the complexity of DBPs. The DBPs span over many protein families from enzymes to transcription factors [16], which makes it very difficult to describe DBPs discriminatively using mathematical models; (iii) A common approach to describe a protein in DBP prediction is by forming a feature vector, but the redundancy and contradiction among these features may seriously deteriorate the predication and generalization ability of the model.

In light of the aforementioned problems, we proposed a novel sequence-based predictor, named iDbP (identification of DNA-binding Proteins), to identify DBPs in this study. Firstly, instead of developing a narrow-application structured-based method, we focused on the challenge of sequenced-based methods. Secondly, a number of discriminative features, including evolutionary conservation, secondary structure motifs and physicochemical properties, were constructed to encode the proteins. These informative features have been proved to be associated with DNA binding interactions. Thirdly, a novel improved binary firefly algorithm (BFA) was introduced to remove redundant and noisy features as well as select optimal parameters for the classifier. In the proposed BFA, we used normalized Hamming distance to calculate attractiveness for fireflies, which greatly improved the converging rate. We also added a dynamic mutation operator to increase the diversity of fireflies. Based on the effective BFA, our predictor produced promising performance on the main dataset and two benchmark datasets. Tests on an independent testing dataset collected from PDB and a large-scale DBP dataset collected from UniProt database demonstrated the good generalization ability of iDbP.

Methods

Datasets

In this study, experimentally verified DBPs were collected from the Protein Data Bank (PDB, http://www.rcsb.org) by specifying keyword "DNA binding protein" and release date "before 2015-05-01" through "Advanced Search", and 1248 sequences were obtained. Then, these sequences were pre-processed through the following procedures: (1) Sequences which contained unknown residues were discarded. (2) Sequences with less than 50 amino acid residues or belonged to fragments were removed [17]. (3) Sequences with multi-bindings were removed to avoid other influences. (4) Sequence similarity among the dataset was reduced to less than 30 % by using PISCES [18]. As a result, 455 experimentally verified DBPs were obtained as positive samples. Similarly, 455 experimentally verified non-binding proteins were also extracted from PDB with "Does not contain: DNA binding protein" as key words with less than 30 % identity. Finally, a main dataset was obtained by combining the 455 DBPs and 455 non-DBPs. This main dataset was used to find the optimal feature subset and train the iDbP prediction model. To construct the training dataset, 355 sequences were randomly picked from positive and negative samples of the main dataset, respectively. The remaining positive and negative samples were used for testing. In order to ensure unbiased and objective results, the process of under-sampling was performed 20 times. The final performance was the average prediction results of 20 experiments on different training and testing datasets.

To evaluate the effectiveness of the proposed method as well as to perform fair comparisons with previous methods [9–15], two benchmark training and testing datasets were adopted: (i) PDB594 and PDB186 [15]. The training dataset PDB594 contained 297 DBPs and 297 non-DBPs, and the testing dataset PDB186 contained 93 DBPs and 93 non-DBPs. Both PDB594 and PDB186 shared sequence similarity of less than 25 %; (ii) DNAdset and DNAiset [14]. DNAdset included 231 DBPs and 231 non-DBPs, and DNAiset contained 80 DBPs and 192 non-DBPs. The sequence similarity in DNAdset and DNAiset was less than 30 %.

In real life, the number of DBPs is much less than that of non-DBPs. To further test the generalization ability of our method, a new-compiled independent testing dataset (named DBP189) was introduced in this work. All the predictors that we compared with in this research were built before May 2015. Therefore, proteins released in PBD after May 2015 would be less likely to be used to train these models. DBP189 contained 21 DBPs and 167 non-DBPs, which were deposited in PDB between 2015-05-01 and 2016-05-01. None of these proteins shared more than 30 % sequence similarity with the main dataset. The main dataset and DBP189 were provided in Additional file 1.

Feature vector

Evolutionary conservation profile

Highly conserved regions are often required for basic cellular function, stability or reproduction. Thus, evolutionary conservation analysis are often indicative of structural or functional importance [19, 20]. The position specific scoring matrix (PSSM), which carries evolutionary information of proteins, was widely used in various bioinformatics researches. In this study, the PSSM of each protein was generated by using PSI-BLAST [21] to search against the non-redundant database (ftp://ftp.ncbi.nlm.nih.gov/blast/db/nr.tar.gz) through 3 iterations with E-value of 0.0001. A $L \times 20$ PSSM was generated for each protein, where L was the length of the sequence.

$$PSSM = \begin{bmatrix} E_{1,1} & E_{1,2} & \cdots & E_{1,20} \\ E_{2,1} & E_{2,2} & \cdots & E_{2,20} \\ \vdots & \vdots & \cdots & \vdots \\ E_{L,1} & E_{L,2} & \cdots & E_{L,20} \end{bmatrix} \quad (1)$$

Each score in PSSM represents whether the related substitution exceed or beneath expected frequency, and indicates whether this substitution would be favored in the process of evolution. Here, these preferences are statistical classified and analyzed by using the following formula:

$$P_{m,n} = \sum\nolimits_{m=1}^{L} E_{m,n} \times \delta \begin{cases} \delta = 1, \ R_m = a_n \\ \delta = 0, \ R_m \neq a_n \end{cases} \quad (2)$$

where R_m indicates the m-th ($m \in \{1, 2, \ldots, L\}$) residue in the protein sequence, and a_n ($n \in \{1, 2, \ldots, 20\}$) indicates the type of amino acid. To eliminate the influences of sequence length, $P_{m,n}$ is normalized into the [0, 1] interval by using logistic function:

$$E_{R_i \to a_i} = \frac{1}{1 - e^{-P_{m,n}}} \quad (3)$$

Finally, feature vector $\{E_{R_i \to a_i} | R \in [1, L], i \in \{1, 2, \ldots, 20\}\}$ was generated to construct the features of evolutionary conservation profile.

Secondary structure motifs

Secondary structure plays an important role in the function of DBPs [22]. Many DBPs show obvious preference of certain secondary structure motifs, such as helix-turn-helix and coil-helix-coil. These structures are usually solvent exposed and hydrophilic, which grant high probabilities in interaction with DNA segments [23]. Shown in Fig. 1 are the examples of DBP complexes. The secondary structure motifs repeat regularly in DBPs, and this phenomenon could be utilized to discriminate DBPs from non-binding proteins. Figure 2 shows the distributions of the secondary structure motifs on the main dataset. The over-expression of "CXC", "HCX" and "ECX" confirms the experimental observation of enrichments of a series of helices or strands in DBPs.

To obtain secondary structure motifs, firstly, the predicted secondary structure for each residue was calculated as a probability matrix using PSIPRED [24] (Eq. (4)).

$$ss\ probMarix = \begin{bmatrix} P_{1 \to H} & P_{1 \to E} & P_{1 \to C} \\ P_{2 \to H} & P_{2 \to E} & P_{2 \to C} \\ \vdots & \vdots & \vdots \\ P_{L \to H} & P_{L \to E} & P_{L \to C} \end{bmatrix} \quad (4)$$

where $P_{i \to H/E/C}$ ($i \in \{1, 2, \ldots, L\}$) is the probability of the i-th residue to be part of a helix (H), strand (E) or coil (C). Next, $max(P_{i \to H/E/C})$ for each position would be selected as the corresponding secondary structure, and secondary structure segments were generated to represent the secondary structure distribution for the protein. Then, the secondary structure motifs were obtained from the segments:

$$ss\ motif = \sum \left\{ seg_\alpha seg_\beta seg_\gamma \right\} \quad (5)$$

where $seg_{\alpha/\beta/\gamma}$ indicates continuous secondary structure segments of the same type and $\alpha, \beta, \gamma \in \{H, E, C\}$. Finally, a protein was encoded by a 12-dimentional feature vector.

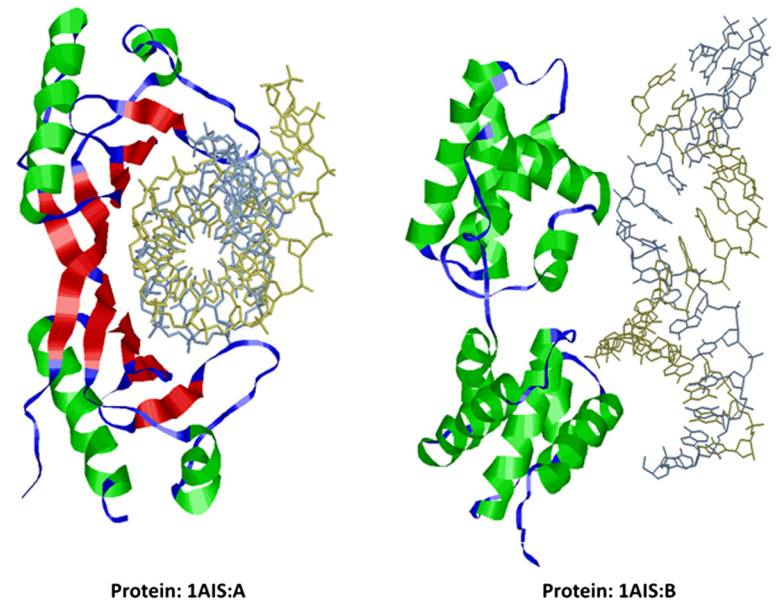

Fig. 1 An example that illustrates the preferences of certain secondary structure motifs of a protein complex. Panel (**a**) is a TATA-binding protein (PDB ID: 1AIS_A). The binding surface is composed of strands (*red*) while the outer region is composed of helices (*green*). The general secondary structure pattern of this protein is strand-helix-strand-helix-strand-helix-strand-helix. Panel (**b**) is a transcription initialization protein (PDB ID: 1AIS_B) that is mainly composed of helices (*green*) and turns (*blue*)

Physicochemical properties

Physiochemical properties reveal macroscopic phenomena among atoms and molecules such as motions, energy, force and dynamics [25]. For instance, Surendra et al. [26] pointed out that hydrophobic and polar residues contributed the bonds across the interfaces and binding residues were strongly correlated with exposed surface area. Solvation free energy [27] and transfer free energy [28], which helped to form small paths, were vital free energy to the hot spots. In addition, graph shape also played an important role in deciding the functional sites on the protein surface [29]. In this study, fourteen physiochemical properties, namely net charge [30],

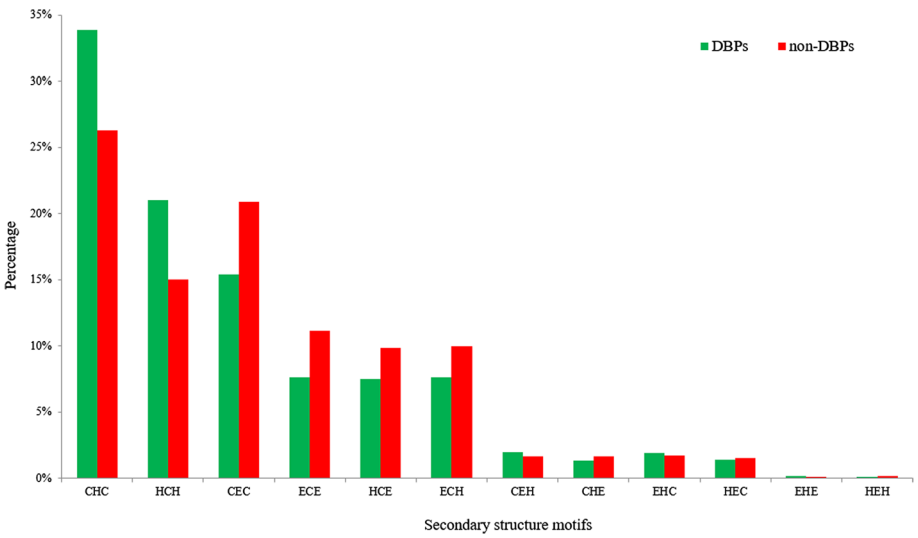

Fig. 2 The distribution of secondary structure motifs

hydrophobicity [31], hydrophilicity [27], polarity [32], polarizability [33], solvation free energy [27], graph shape index [34], transfer free energy [28], amino acid composition [35], correlation coefficient in regression analysis [36], residue accessible surface area [37], partition coefficient [38], entropy of formation [39], and pKa values of side chain [40], were collected and used. In this encoding scheme, each property were first calculated by taking the sum of its value over the residues on the whole sequence. Then, the summarized value of each property was divided by the length of the sequence [41].

Support vector machine

Support vector machine (SVM) is a machine learning technique derived from statistical learning theory first proposed by Vapnik [42]. It was successfully applied in many bioinformatics problems and yielded promising results. In this study, we utilized the LIBSVM toolset [43] and chose Radial Basis Function (RBF) as the kernel function. Two parameters c and γ of SVM were optimized using BFA. All feature descriptors were normalized into the [0, 1] interval by using logistic function.

The proposed binary firefly algorithm

Continuous firefly algorithm

The continuous Firefly Algorithm (FA) is a swarm-intelligence and meta-heuristic optimization algorithm developed by Xin-She Yang in 2007 [44]. FA is based on the idealized behavior of the flashing characteristics of the fireflies. It is featured by its efficiency as well as robustness. As a novel meta-heuristic algorithm, FA has been proved to be able to find almost optima in continuous problems [45]. In essence, the idea of FA can be abstracted into the following three rules [46]:

(i) Every firefly has its own lightness and could be attracted by other fireflies;
(ii) The brightness and distance determine the attractiveness. That is, a brighter firefly will always attract its adjacent less bright ones. The attractiveness will decline if the distance between two fireflies increases. If a firefly cannot find a brighter firefly within the designated distance, it will make random movements;
(iii) The brightness of a firefly is referred as light intensity (I), which is defined as:

$$I = F(f(x), \beta) \tag{6}$$

where $f(x)$ is the objective function. The attractiveness β is proportional to I, and is defined as:

$$\beta = \beta_0 e^{-\gamma r^2} \tag{7}$$

where β_0 is the attractiveness at $r = 0$; γ denotes the light absorption coefficient; and r represents the distance between any two fireflies. The movement of a firefly x_i attracted to another firefly x_j is defined as:

$$x_i = x_i + \beta\left(x_j - x_i\right) + \alpha\varepsilon_i \tag{8}$$

where α is the randomization parameter, and ε_i is an element of a vector drawn from random Gaussian or uniform distributions.

Binary firefly algorithm

The original FA is designed for continuous problems, which means that the outcome of the objective function (i.e. the brightness of a firefly) must lie in a continuous interval. Recently, several BFA were developed to solve discrete problems, such as scheduling, timetabling and combination. Compared with the original FA, BFA obeyed similar fundamental principles while redefined distance, attractiveness, or movement of the firefly [47–49]. Palit et al. [47] applied BFA to discover the plaintext from the cipher text. Sayadi et al. [48] defined a new firefly position and applied BFA to manufacture cell formation. Poursalehi et al. [49] introduced a new form of movement of fireflies to global best in each iteration, and applied BFA on fuel reload design of nuclear reactors. In this study, a novel improved BFA was proposed for feature selection as well as parameter optimization.

The feature selection task is a typical combination problem in essence. That is, to select an optimal combination of features from a given feature space. By using this optimal subset, the machine learning algorithm could produce the best predictive performance. Every feature must be either in or not in this subset. Theoretically, for an n-dimensional feature space, there will be 2^n possible solutions (NP-hard problem). Empirically, meta-heuristic algorithms will perform better than traditional filter or wrapper methods [50]. In BFA, every firefly represents a subset of the feature space and a group of parameters (i.e., a possible solution for the problem). The effectiveness of BFA is determined by two factors: the ability to converge to the potential global optimum rapidly and the capability of jumping out of local optima. In this work, normalized Hamming distance was used to calculate attractiveness and improve converging rate in feature selection; dynamic mutation operator was introduced to increase the diversity of fireflies. The pseudo code of BFA is provided in Algorithm 1.

Algorithm 1 The pseudo code of the BFA.

```
Begin
Initialize the population P; Initialize the algorithm parameters.
While stop criteria is not met do
    Evaluate the lights of the population
    Find the brightest as the current best
    For i=1 to population size
        For j=1 to population size
            If firefly j is brighter than firefly i then
                Determine the similarity parameter r
                Determine the β
                For each bit in firefly i and j
                        If two bits are different then
                    The bit in i changes to bit in j with probability of β
                        End if
                        If rand(0,1)< α
                  Mutate the bit in i
                        End if
                End for
            End if
        End for
    End for
    Update the population
End while
End
```

a. Firefly representation

In BFA, a binary string is used to encode a firefly. Every element in the string is either 0 or 1, the length and interpretation for the string are both problem specific. That is, a firefly X is defined as the following:

$$X = x_1 x_2 x_3 \ldots x_n \quad where\ x_i \in \{0, 1\} \tag{9}$$

Figure 3 shows an instance of the definition of a firefly X with a length of n. The string is divided into three parts. The first part (t elements) and second part (t elements) are used to represent the values of parameters c and γ of SVM, respectively. The third part represents the features. Its length w is the same as the dimension of the feature space. In this part, 1 denotes the corresponding feature is selected, and 0 indicates the opposite.

b. The attractiveness of a firefly

Similar to FA, a firefly in BFA is also attracted by brighter fireflies. However, the attractiveness is not only determined by the brightness but also greatly affected by the similarity between fireflies. In BFA, the attractiveness β between a pair of fireflies is defined as $\beta = \beta_0 e^{-\gamma r^2}$. Here, γ controls the impact of β in the movement function; r determines the stride of the firefly movement. For two fireflies X_i and X_j, r is defined based on the similarity ratio of the two fireflies (or the normalized Hamming distance of two vectors) as follows:

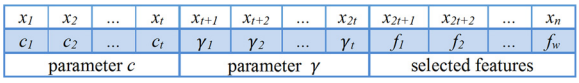

Fig. 3 The coding scheme for a firefly

$$r = 1 - \frac{\sum_{k=1}^{n} \left| X_i^k \oplus X_j^k \right|}{n} \tag{10}$$

where ⊕ denotes the XOR operation, n is the length of X. Mathematically, the less identical bits two fireflies share, the greater stride a firefly would take and the more likely it would move towards the brighter one. β is the probability of a hetero-bit in the moving firefly changes to the corresponding bit in the brighter firefly ($0 \rightarrow 1$ or $1 \rightarrow 0$). Compared with Cartesian distance and Euclidean distance, the normalized Hamming distance performs best in keeping good feature as well as removing bad ones, and also made the algorithm converge fast. Figure 4 demonstrates an example of calculating parameter r.

c. The movement of a firefly

When a firefly moves, every bit in its representation string will make a decision to move (change its value) or not. The decision is determined after two actions: the attraction, which is regulated by the attractiveness (β); and the mutation, which is controlled by a parameter (α). The movement of a bit X_i^k in firefly X_i moving towards the corresponding bit X_j^k in firefly X_j is defined as follows:

$$X_i^k = g\left(f\left(X_i^k, X_j^k, \beta\right), \alpha\right) \tag{11}$$

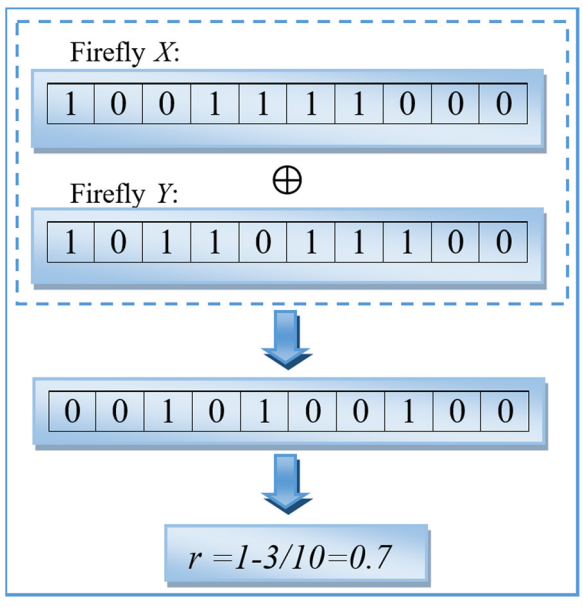

Fig. 4 An Example of calculating parameter r. Firefly X = {1 0 0 1 1 1 1 0 0 0}, Firefly Y = {1 0 1 1 0 1 1 1 0 0}. The distance or difference is calculated by $X \oplus Y$ operation and equals {0 0 1 0 1 0 0 1 0 0}. Finally, the similarity ratio of between X and Y is r. -(3/10) = 0.7

$$f\left(X_i^k, X_j^k, \beta\right) = \begin{cases} X_j^k, & \text{if } X_i^k \neq X_j^k \text{ and } rand(0,1) < \beta \\ X_i^k, & \text{otherwise} \end{cases} \tag{12}$$

$$g\left(X_i^k, \alpha\right) = \begin{cases} 1 - X_i^k, & \text{if } rand(0,1) < \alpha \\ X_i^k, & \text{otherwise} \end{cases} \tag{13}$$

$$\alpha = 0.5 - \frac{0.5 \times Iteration}{Max\ Iteration} \tag{14}$$

where the inner function $f(x,y,x)$ of (Eq.11) regulates the attracted movement of bit X_i^k to X_j^k, and the outer function $g(x, \alpha)$ regulates the random moving behavior (mutation) of X_i^k. It should be noted that an attracted movement would incur only when the two corresponding bits are different, while the mutation might occur on every bit with the same probability. The introduction of dynamic mutation operator grants the firefly the ability to escape from a local optimum and check nearby regions while flying. In this work, parameter α controls the probability of mutation. The mutation probability is high in initial iterations, which makes BFA focus on exploration. As the number of iteration increases, the mutation probability will decrease, and BFA will accelerate its converging pace gradually. Figure 5 demonstrates an example of firefly movement. If a firefly is attracted by another, each different bit in the attracted firefly would change with probability β. Then each bit in the new firefly mutates with probability α.

Statistic inference and performance evaluation

Five indices were employed to measure the performance of our method. These indices included sensitivity (SN), specificity (SP), accuracy (ACC), and Matthews's correlation coefficient (MCC):

$$SN = \frac{TP}{TP + FN} \tag{15}$$

$$SP = \frac{TN}{TN + FP} \tag{16}$$

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{17}$$

$$MCC = \frac{TP \times TN - FN \times FP}{\sqrt{(TP + FN) \times (TP + FP) \times (TN + FP) \times (TN + FN)}} \tag{18}$$

where TP, FP, TN, and FN were the abbreviations of true positive, false positive, true negative, and false negative, respectively. The area under the receiver operating characteristic curve (ROC-AUC) was carried out when we assessed our method with other feature selection methods. The performance was evaluated by using leave-one-out cross-validation on the main dataset and

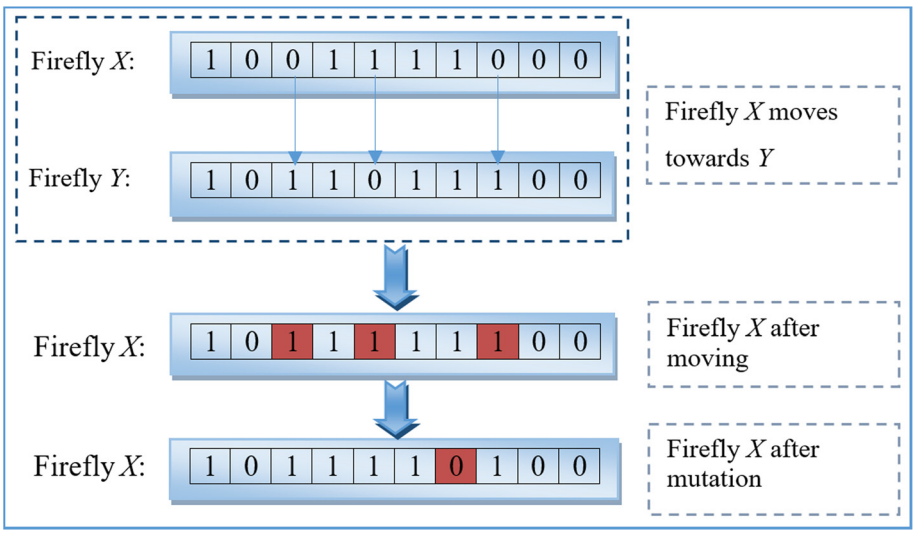

Fig. 5 An example of movement and mutation for a firefly

selected optimal feature subset and parameters. Finally, the workflow of our method is shown in Fig. 6.

Results and discussion

The performance of the proposed method

The proposed method was implemented by combining informative features and optimizing parameters using BFA based on SVM. The settings of BFA were tuned as the following: the number of fireflies was set to 30; the visibility γ was set to 1; and the maximum iterations was set to 500. The light intensity was defined as follows:

$$I = \omega \times MCC + (1-\omega) \times \left(1-\frac{n}{N}\right) \quad (19)$$

where n was the number of selected features, N was the total number of features, and ω was the weighting coefficient that controlled the trade-off between the prediction accuracy and the selected features. Usually, the weighting coefficients of an algorithm are determined empirically. In our research, ω was set as 0.55. Here, MCC was used as the key criterion to evaluate the performance of a feature subset, as it could provide balanced and unbiased measurement of the prediction ability of the model. $\frac{n}{N}$ was used to assess the number of selected features. This experiment was repeated 20 times. The final performance was the average of the 20 results. The experiment with the medium value of MCC were chosen and the corresponding optimal feature subset and parameters were used to build the iDbP prediction model. The following experiments were all based on the selected optimal feature subset and parameters. Finally, the proposed method achieved a promising performance with the mean MCC of 0.595, ACC of 0.795, SN of 0.863, SP of 0.726 on the main dataset.

Comparison with other feature selection techniques

Feature selection is an important technique in predictive modeling. By removing redundant features, it can considerably improve the prediction accuracy. In this section, we compared BFA with several popular feature selection techniques: binary particle swarm optimization (BPSO) [50], genetic algorithm (GA) [51], minimum redundancy maximum relevance [52] combined with incremental feature selection (mRMR + IFS) [41], the original FA [44], and the straightforward method with all features.

PSO is a meta-heuristic algorithm that optimizes a problem by searching optimal particle (candidate solution). The position and velocity of the particle vary in each iteration to approach the best position (global optimum). BPSO is the binary version of PSO. GA is a classic intelligent algorithm that emulates genetic evolution. It uses binary representation in nature and is good at discrete optimizations. mRMR + IFS is a combined feature selection scheme. It firstly sorts the features with criteria of minimum redundancy maximum relevance. Then, it iteratively uses the first n ranked features to build models to find the best feature subset. For the original FA, which should only be used in continuous problems, the binary string of the feature vector was transferred to decimal values. All these methods were embedded with SVM and run 20 times on the main dataset using exactly the same procedure. The final performance for each method were the average performance of 20 results.

Table 1 lists the detailed results of five feature selection methods and the straightforward method with all features. Compared with simple feature fusion or filter feature selection, the meta-heuristic algorithms were

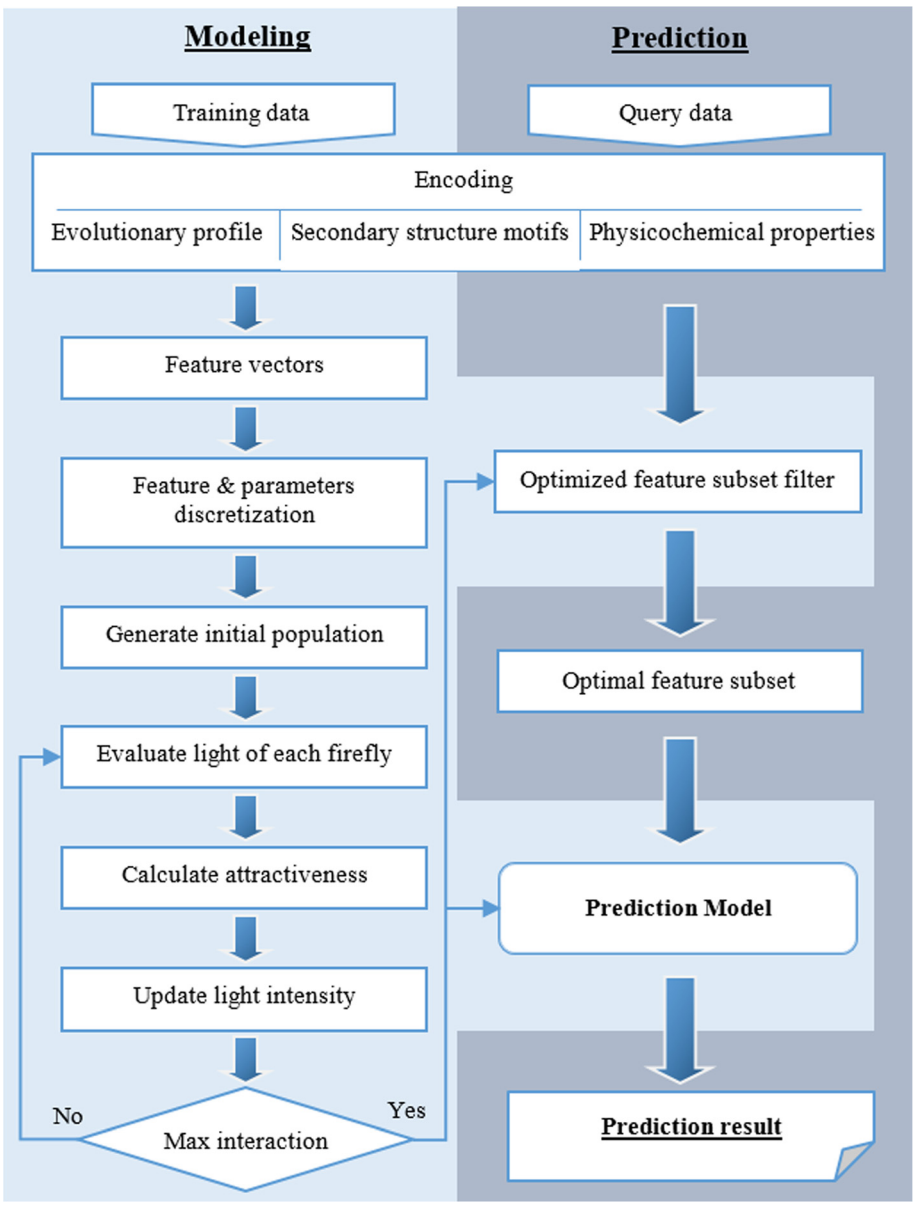

Fig. 6 The flowchart of proposed method

Table 1 Comparison of BFA with different feature selection methods

Method	SN	SP	ACC	MCC
BFA	0.863	0.726	0.795	0.595
BPSO	0.830	0.710	0.770	0.544
GA	0.840	0.680	0.760	0.527
FA	0.720	0.610	0.665	0.332
mRMR + IFS	0.790	0.640	0.715	0.435
All features	0.680	0.760	0.600	0.365

more effective in selecting the optimal feature subsets. In addition, the FA produced an unsatisfactory performance, which proved that it was not suitable for discrete problem. Among the three meta-heuristic algorithms, BFA outperformed other methods with the highest MCC of 0.595.

To assess the robustness of our BFA, we further drawn ROC curves for each method using the leave-one-out cross-validation on the main dataset. With all features, the predictor gave an AUC of 0.727. The mRMR + IFS scheme gave an AUC of 0.767. Additionally, the heuristic feature selection algorithms achieved better performance, an AUC of 0.747 for FA, an AUC of 0.768 for GA

and an AUC of 0.779 for BPSO (Fig. 7). The newly proposed BFA produced an AUC values of 0.791, which was the highest among these feature selection methods. In our research, the BFA takes about 90 min to complete one entire experiment on a PC with a 3.20 GHz Intel Xeon CPU and 8GB RAM. Further improvement can be achieved by parallel computation, which is almost 4 times faster by computing 6 fireflies concurrently.

Comparison with existing methods

Comparison with other predictors on benchmark datasets

In recent years, several methods were proposed to identify DBPs. These methods included DNAbinder [9], iDNA-Prot [11], enDNA-Prot [13], nDNA-Prot [12], DBPPred [15], DBD-Threader [53] and Zou's method [14]. Among these methods, DNAbinder, iDNA-Prot, enDNA-Prot, nDNA-Prot, DBPPred and Zou's method were sequence-based methods. To ensure a fair comparison with previous studies, the training dataset PDB594 of DBPPred was adopted to train iDbP and the independent testing dataset PDB186 was used to evaluate our predictor and compare with previous studies. Listed in Table 2 are the results of the comparison. Our iDbP achieved the highest SN of 0.894, ACC of 0.809 and MCC of 0.625. Additionally, we also compared the AUC value of iDbP with these predictors. As the AUC scores for iDNA-Prot, DNA-Prot, enDNA-Prot, nDNA-Prot, and DBD-Threader were unavailable, the comparisons were performed among DBPPred, DNAbinder, DNABIND and iDbP. The DBPPred, DNAbinder, DNABIND produced the AUC scores of 0.791, 0.607 and 0.694. Our iDbP yielded the highest AUC score of 0.803, which was slightly better than DBPPred.

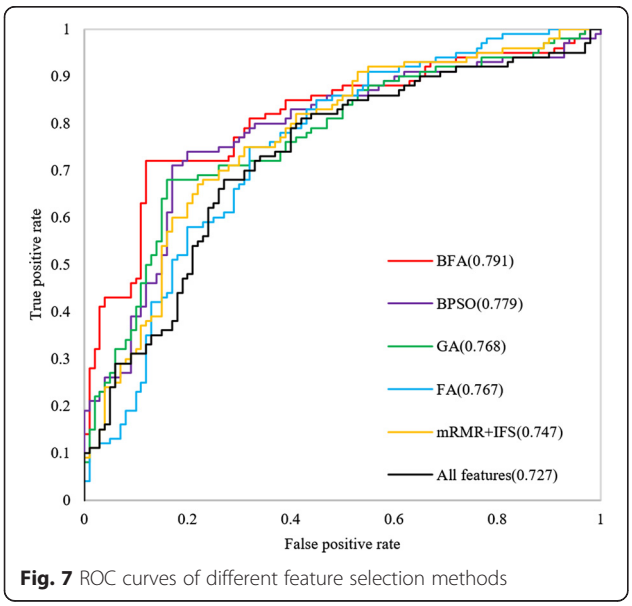

Fig. 7 ROC curves of different feature selection methods

Table 2 Comparison of iDbP with existing methods on dataset PDB186

Method	SN	SP	ACC	MCC
iDbP	0.894	0.722	0.809	0.625
DBPPred	0.796	0.742	0.769	0.538
iDNA-Prot	0.677	0.667	0.672	0.344
nDNA-Prot	0.710	0.623	0.667	0.335
enDNA-Prot	0.602	0.699	0.651	0.303
DNA-Prot	0.699	0.538	0.618	0.240
DNAbinder	0.570	0.645	0.608	0.216
DBD-Threader	0.237	0.957	0.597	0.279

Similarly, the training dataset DNAdset from Zou's method was adopted to train iDbP and the independent testing dataset DNAiset was used to evaluate iDbP and compare with previous studies. As the services of DBPPred and DBDThreader were not availiable. The comparison on Zou's benchmark dataset was performed among iDNA-Prot, DNAbinder, enDNA-Prot, nDNA-Prot, Zou's method and our iDbP. As shown in Table 3, the iDbP yielded the best performance with the SN of 0.908, SP of 0.911, ACC of 0.910 and MCC of 0.803.

Theoretically, protein structures could provide more information than primary sequences. However, our experiments showed that the sequence-based method could produce approximate or even better results. In general, the sequence-based methods are significant supplements for the structure-based methods, especially when the high-resolution 3D structures or the homology templates of the query proteins are hard to obtain.

Comparison with other predictors on DBP189 dataset

To demonstrate the generalization ability of our iDbP, we performed further comparisons with previous methods on DBP189. Three DBP prediction tools, namely DNA-Prot, iDNA-Prot and DNAbinder, still provided online or local prediction services. The prediction results (shown in Table 4) on the DBP189 dataset indicated that our method still characterized by good predictive performance on imbalanced testing dataset. Among these methods, our iDbP achieved the highest

Table 3 Comparison of iDbP with existing methods on dataset DNAiset

Method	SN	SP	ACC	MCC
iDbP	0.908	0.911	0.910	0.803
Zou's method	0.890	0.828	0.900	0.753
iDNA-Prot	0.875	0.798	0.837	0.709
nDNA-Prot	0.779	0.887	0.851	0.664
enDNA-Prot	0.760	0.868	0.832	0.623
DNAbinder	0.717	0.642	0.863	0.473

Table 4 Comparison of predictive quality on the DBP189 dataset

Method	SN	SP	ACC	MCC
iDbP	0.7619	0.9162	0.8989	0.5996
DNA-Prot	0.7143	0.9042	0.8830	0.5415
iDNA-Prot	0.6190	0.8563	0.8298	0.3960
DNAbinder	0.5714	0.8263	0.7979	0.3234

MCC of 0.5996, which was about 5 % more than the second highest method DNA-Prot.

Application to large-scale DBP prediction

In real-life application, computational tools are often used to identify possible candidate proteins in large scale. To simulate this scenario, we collected 15,413 DBPs from five most popular organisms (human, A. thaliana, mouse, S. cerevisiae and fruit fly) in UniProt database. After removing incomplete segments and unannotated proteins, we finally obtained a large-scale testing dataset with 2859 DBPs (Provided in Additional file 2). As shown in Table 5, by using our iDbP, nearly 59 % of human proteins, 53 % of A. thaliana proteins, 54 % of Mouse proteins, 61 % of S. cerevisiae proteins, and 59 % of Fruit fly proteins were successfully predicted as DBPs. In summary, about 56 % proteins were successfully recognized. The results showed that iDbP could be a reliable tool in large-scale applications.

Conclusion

In this work, we proposed a new method, named iDbP, to predict DBPs from primary sequence. Multiple informative features, which derived from evolutionary conservation profile, secondary structure motifs and physiochemical properties, were used to discriminate DBPs from non-binding proteins. Next, a novel improved BFA was forged to perform feature selection and parameter optimization. The experimental results of our predictor on two benchmark datasets outperformed many state-of-the-art predictors, which revealed the effectiveness of our method. Moreover, the promising performance on an independent testing dataset and large-scale proteins from UniProt database proved the good generalization ability of our method. In addition, the novel improved BFA would be of a powerful algorithm which could find widely applications in discrete optimization problems. The webserver is available for academic research at http://59.73.198.144:8080/iDbP/.

Table 5 Number of annotated and recognized DBPs in UniProt database

Category	Number of proteins	Proteins with complete DNA binding annotations	Number of predicted DBPs	SN
Human	6,813	1,049	613	58 %
A. thaliana	3,378	929	489	53 %
Mouse	2,514	424	232	54 %
S. cerevisiae	1,545	314	191	61 %
Fruit fly	1,163	143	84	59 %
Summary	15,413	2,859	1609	56 %

Additional files

Additional file 1: The main dataset and DBP189 used in this study. (PDF 751 kb)

Additional file 2: The large-scale testing dataset compiled from UniProt. (PDF 477 kb)

Acknowledgments

We thank Fundamental Research Funds for the Central Universities (Northeast Normal University) for proving the funding for this work.

Funding

This work was supported by the Fundamental Research Funds for the Central Universities (Grant No. 14ZZ2240).

Availability of data and material

All the datasets used in this research can be found in Additional files associated with this paper.

Authors' contributions

JZ conceived the idea and was in charge of the iDbP implementation. BG, HTC and ZQM optimized the algorithm and participated in the development and validation of the Web server. BG and GFY drafted the first version of the manuscript. JZ and BG designed experiments, gathered test data, and were in charge of the experiments. ZQM and GFY supervised the progress of the whole project and critically checked the first draft. GFY was in charge of the whole process of final revision. All authors have read and approved the final manuscript.

Authors' information

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Received: 21 January 2016 Accepted: 24 August 2016
Published online: 26 August 2016

References

1. Langlois RE, Lu H. Boosting the prediction and understanding of DNA-binding domains from sequence. Nucleic Acids Res. 2010;15:gkq061.
2. Sarai A, Kono H. Protein-DNA recognition patterns and predictions. Annu Rev Biophys Biomol Struct. 2005;34:379–98.
3. Parola M, Bellomo G, Robino G, Barrera G, Dianzani MU. 4-Hydroxynonenal as a biological signal: molecular basis and pathophysiological implications. Antioxid Redox Signal. 1999;1(3):255–84.
4. Chou CC, Lin TW, Chen CY, Wang AH. Crystal structure of the hyperthermophilic archaeal DNA-binding protein Sso10b2 at a resolution of 1.85 Angstroms. J Bacteriol. 2003;185(14):4066–73.
5. Freeman K, Gwadz M, Shore D. Molecular and genetic analysis of the toxic effect of RAP1 overexpression in yeast. Genetics. 1995;141(4):1253–62.
6. Gao M, Skolnick J. DBD-Hunter: a knowledge-based method for the prediction of DNA–protein interactions. Nucleic Acids Res. 2008;36(12):3978–92.

7. Zhao H, Yang Y, Zhou Y. Structure-based prediction of DNA-binding proteins by structural alignment and a volume-fraction corrected DFIRE-based energy function. Bioinformatics. 2010;26(15):1857–63.
8. Nimrod G, Szilágyi A, Leslie C, Ben-Tal N. Identification of DNA-binding proteins using structural, electrostatic and evolutionary features. J Mol Biol. 2009;387(4):1040–53.
9. Kumar M, Gromiha MM, Raghava GP. Identification of DNA-binding proteins using support vector machines and evolutionary profiles. BMC Bioinformatics. 2007;8(1):1.
10. Kumar KK, Pugalenthi G, Suganthan PN. DNA-Prot: identification of DNA binding proteins from protein sequence information using random forest. J Biomol Struct Dynam. 2009;26(6):679–86.
11. Lin WZ, Fang JA, Xiao X, Chou KC. iDNA-Prot: identification of DNA binding proteins using random forest with grey model. PLoS One. 2011;6(9):e24756.
12. Song L, Li D, Zeng X, Wu Y, Guo L, Zou Q. nDNA-prot: identification of DNA-binding proteins based on unbalanced classification. BMC Bioinformatics. 2014;15(1):1.
13. Xu R, Zhou J, Liu B, Yao L, He Y, Zou Q, Wang X. enDNA-Prot: identification of DNA-binding proteins by applying ensemble learning. BioMed Res Int. 2014.
14. Zou C, Gong J, Li H. An improved sequence based prediction protocol for DNA-binding proteins using SVM and comprehensive feature analysis. BMC Bioinformatics. 2013;14(1):1.
15. Lou W, Wang X, Chen F, Chen Y, Jiang B, Zhang H. Sequence based prediction of DNA-binding proteins based on hybrid feature selection using random forest and Gaussian naive Bayes. PLoS One. 2014;9(1):e86703.
16. Rohs R, Jin X, West SM, Joshi R, Honig B, Mann RS. Origins of specificity in protein-DNA recognition. Annu Rev Biochem. 2010;79:233.
17. Chou KC, Shen HB. Recent progress in protein subcellular location prediction. Anal Biochem. 2007;370(1):1–16.
18. Wang G, Dunbrack RL. PISCES: recent improvements to a PDB sequence culling server. Nucleic Acids Res. 2005;33 suppl 2:W94–8.
19. Zhang J, Chen W, Sun P, Zhao X, Ma Z. Prediction of protein solvent accessibility using PSO-SVR with multiple sequence-derived features and weighted sliding window scheme. BioData Min. 2015;8(1):1–15.
20. Zhang J, Zhao X, Sun P, Gao B, Ma Z. Conformational B-cell epitopes prediction from sequences using cost-sensitive ensemble classifiers and spatial clustering. BioMed Res Int. 2014.
21. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 1997;25:3389–402.
22. Bochman ML, Paeschke K, Zakian VA. DNA secondary structures: stability and function of G-quadruplex structures. Nat Rev Genet. 2012;13(11):770–80.
23. Greive SJ, Fung HK, Chechik M, Jenkins HT, Weitzel SE, Aguiar PM, Brentnall AS, Glousieau M, Gladyshev GV, Potts JR, Antson AA. DNA recognition for virus assembly through multiple sequence-independent interactions with a helix-turn-helix motif. Nucleic Acids Res. 2016;44(2):776-789.
24. Jones DT. Protein secondary structure prediction based on position-specific scoring matrices. J Mol Biol. 1999;292(2):195–202.
25. Liu ZP, Wu LY, Wang Y, Zhang XS, Chen L. Prediction of protein–RNA binding sites by a random forest method with combined features. Bioinformatics. 2010;26(13):1616–22.
26. Bordner AJ, Abagyan R. Statistical analysis and prediction of protein–protein interfaces. Proteins Struct Funct Bioinf. 2005;60(3):353–66.
27. Jayaram B, McConnell KJ, Dixit SB, Beveridge DL. Free energy analysis of protein–DNA binding: the EcoRI endonuclease–DNA complex. J Comput Phys. 1999;151(1):333–57.
28. Chaires JB, Satyanarayana S, Suh D, Fokt I, Przewloka T, Priebe W. Parsing the free energy of anthracycline antibiotic binding to DNA. Biochemistry. 1996;35(7):2047–53.
29. Liu S, Liu S, Zhu X, Liang H, Cao A, Chang Z, Lai L. Nonnatural protein–protein interaction-pair design by key residues grafting. Proc Natl Acad Sci. 2007;104(13):5330–5.
30. Ahmad S, Sarai A. Moment-based prediction of DNA-binding proteins. J Mol Biol. 2004;341(1):65–71.
31. Landschulz WH, Johnson PF, McKnight SL. The leucine zipper: a hypothetical structure common to a new class of DNA binding proteins. Science. 1988;240(4860):1759–64.
32. Ip YT, Kraut R, Levine M, Rushlow CA. The dorsal morphogen is a sequence-specific DNA-binding protein that interacts with a long-range repression element in Drosophila. Cell. 1991;64(2):439–46.
33. Yu X, Cao J, Cai Y, Shi T, Li Y. Predicting rRNA-, RNA-, and DNA-binding proteins from primary structure with support vector machines. J Theor Biol. 2016;240(2):175–84.
34. Slattery M, Riley T, Liu P, Abe N, Gomez-Alcala P, Dror I, Zhou T, Rohs R, Honig B, Bussemaker HJ, Mann RS. Cofactor binding evokes latent differences in DNA binding specificity between Hox proteins. Cell. 2011; 147(6):1270–82.
35. PSORT I. PSORT: a program for detecting sorting signals in proteins and predicting their subcellular localization. J Mol Biol. 1997;266:594–600.
36. Mukherjee S, Berger MF, Jona G, Wang XS, Muzzey D, Snyder M, Young RA, Bulyk ML. Rapid analysis of the DNA-binding specificities of transcription factors with DNA microarrays. Nat Genet. 2004;36(12):1331–9.
37. Ahmad S, Gromiha MM, Sarai A. Analysis and prediction of DNA-binding proteins and their binding residues based on composition, sequence and structural information. Bioinformatics. 2004;20(4):477–86.
38. Goodsell D, Dickerson RE. Isohelical analysis of DNA groove-binding drugs. J Med Chem. 2004;29(5):727–33.
39. Chaires JB. A thermodynamic signature for drug–DNA binding mode. Arch Biochem Biophys. 2006;453(1):26–31.
40. Nowak MW, Kearney PC, Saks ME, Labarca CG, Silverman SK, Zhong W, Thorson J, Abelson JN, Davidson N. Nicotinic receptor binding site probed with unnatural amino acid incorporation in intact cells. Science. 1995; 268(5209):439–42.
41. Zhang J, Sun P, Zhao X, Ma Z. PECM: Prediction of extracellular matrix proteins using the concept of Chou's pseudo amino acid composition. J Theor Biol. 2014;363:412–8.
42. Vapnik V. The nature of statistical learning theory, Springer science & business media. 2013.
43. Chang CC, Lin CJ. LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol. 2011;2(3):27.
44. Yang XS. Firefly algorithms for multimodal optimization. In Stochastic algorithms: foundations and applications. Springer Berlin Heidelberg; 2009: 169–178.
45. Hashmi A, Goel N, Goel S, Gupta D. Firefly algorithm for unconstrained optimization. IOSR J Comput Eng. 2013;11(1):75–8.
46. Yang XS, He X. Firefly algorithm: recent advances and applications. Int J Swarm Intell. 2013;1(1):36–50.
47. Palit S, Sinha SN, Molla MA, Khanra A, Kule M. A cryptanalytic attack on the knapsack cryptosystem using binary firefly algorithm. In Int Conf Comput Commun Technol (ICCCT). 2011;2:428–32.
48. Sayadi MK, Hafezalkotob A, Naini SGJ. Firefly-inspired algorithm for discrete optimization problems: an application to manufacturing cell formation. J Manuf Syst. 2013;32(1):78–84.
49. Poursalehi N, Zolfaghari A, Minuchehr A. A novel optimization method, Effective Discrete Firefly Algorithm, for fuel reload design of nuclear reactors. Ann Nuclear Energy. 2015;81:263–75.
50. Chuang LY, Chang HW, Tu CJ, Yang CH. Improved binary PSO for feature selection using gene expression data. Comput Biol Chem. 2008;32(1):29–38.
51. Chandrashekar G, Sahin F. A survey on feature selection methods. Comput Electr Eng. 2014;40(1):16–28.
52. Ding C, Peng H. Minimum redundancy feature selection from microarray gene expression data. J Bioinform Comput Biol. 2005;3(02):185–205.
53 Gao M, Skolnick J. A threading-based method for the prediction of DNA-binding proteins with application to the human genome. PLoS Comput Biol. 2009;5(11):e1000567.