

1 **StackPDB: predicting DNA-binding proteins based on XGB-RFE**

2 **feature optimization and stacked ensemble classifier**

3 Qingmei Zhang ^{a,b,1}, Peishun Liu ^{c,1}, Yu Han ^{a,b}, Yaqun Zhang ^{a,b}, Xue Wang ^{a,b}, Bin Yu ^{a,b,d,*}

4 ^a *College of Mathematics and Physics, Qingdao University of Science and Technology, Qingdao*
5 *266061, China*

6 ^b *Artificial Intelligence and Biomedical Big Data Research Center, Qingdao University of Science*
7 *and Technology, Qingdao 266061, China*

8 ^c *College of Information Science and Engineering, Ocean University of China, Qingdao 266100,*
9 *China*

10 ^d *School of Life Sciences, University of Science and Technology of China, Hefei 230027, China*

11

12 **ABSTRACT**

13 DNA binding proteins (DBPs) not only play an important role in all aspects of genetic activities
14 such as DNA replication, recombination, repair, and modification but also are used as key
15 components of antibiotics, steroids, and anticancer drugs in the field of drug discovery. Identifying
16 DBPs becomes one of the most challenging problems in the domain of proteomics research.
17 Considering the high-priced and inefficient of the experimental method, constructing a detailed
18 DBPs prediction model becomes an urgent problem for researchers. In this paper, we propose a
19 stacked ensemble classifier based method for predicting DBPs called StackPDB. Firstly, pseudo
20 amino acid composition (PseAAC), pseudo position-specific scoring matrix (PsePSSM),
21 position-specific scoring matrix-transition probability composition (PSSM-TPC), evolutionary
22 distance transformation (EDT), and residue probing transformation (RPT) are applied to extract
23 protein sequence features. Secondly, extreme gradient boosting-recursive feature elimination
24 (XGB-RFE) is employed to gain an excellent feature subset. Finally, the best features are applied
25 to the stacked ensemble classifier composed of XGBoost, LightGBM, and SVM to construct
26 StackPDB. After applying leave-one-out cross-validation (LOOCV), StackPDB obtains high ACC
27 and MCC on PDB1075, 93.44% and 0.8687, respectively. Besides, the ACC of the independent
28 test datasets PDB186 and PDB180 are 84.41% and 90.00%, respectively. The MCC of the
29 independent test datasets PDB186 and PDB180 are 0.6882 and 0.7997, respectively. The results
30 on the training dataset and the independent test dataset show that StackPDB has a great predictive

* Corresponding author.

E-mail address: yubin@qust.edu.cn (B. Yu).

¹ These authors contributed equally to this work.

31 ability to predict DBPs.

32 *Keywords:* DNA binding proteins; Position-specific scoring matrix; Extreme gradient
33 boosting-recursive feature elimination; Multi-information fusion; Stacked ensemble classifier.

34 **1. Introduction**

35 DNA binding proteins (DBPs) are proteins that can bind and interact with DNA and
36 participate in many biological processes [1]. For example, transcription factors participate in the
37 DNA transcription process while nucleases can cut DNA molecules. Besides, histones are related
38 to the packaging of chromatin in the nucleus [2]. DBPs are essential components of anticancer
39 drugs, antibiotics, and steroids in the research of anticancer drugs and the treatment of genetic
40 diseases. Meanwhile, DBPs have an irreplaceable role in the biophysical, biochemical, and
41 biological research of DNA [3]. Early identification of DBPs generally used experimental methods
42 such as filter combining analysis [4], genetic analysis [5], chromatin immunoprecipitation [6], and
43 X-ray crystallography [7]. With the deep research of high-throughput sequencing technology,
44 protein sequences continue to emerge. However, traditional biological experiment methods are
45 time-consuming and expensive. Identifying DBPs based on experimental methods that are far
46 from meeting the research needs [8]. Therefore, computational methods are used as powerful tools
47 to predict DBPs.

48 Researchers have developed numerous calculation methods to identify DBPs. The important
49 step of predicting DBPs is to extract features from protein sequences. Feature extraction methods
50 can dig four types of protein sequence information which are sequence information,
51 physicochemical properties, structural information, and evolutionary information. Rahman et al. [9]
52 used amino acid composition (AAC), dipeptides composition (DC), tripeptides composition (TC),
53 n-gapped-dipeptides (nGDip), and position-specific n-grams (PSN) to obtain protein sequence
54 information. Zhang et al. [10] used 14 kinds of physicochemical property, protein secondary
55 structural information, and evolutionary information to predict DBPs. Chowdhury et al. [11] used
56 PSI-BLAST to obtain the PSSM, which indicated the evolutionary information. SPIDER2 was
57 used to extract the secondary structural information of the protein sequences. Nanni et al. [12]
58 used AAC and quasi residue couple (QRC) to extract protein sequence information. Meanwhile,
59 physicochemical properties were extracted by the autocovariance approach (AC). In addition,
60 pseudo-position specific scoring matrix (PsePSSM), N-gram features (NGR) and texture
61 descriptors (TD) extracted evolutionary information. Sang et al. [13] obtained the HMM matrix
62 according to the hidden Markov model (HMM) for each sequence. AAC, autocovariance
63 transformation (ACT), and cross-covariance transformation (CCT) were used to convert the HMM
64 matrix into feature vectors of the same length. Then DBPs prediction was performed after fusing

65 multiple features.

66 Although the fusion of multiple features can fully represent the information contained in the
67 protein sequence, it may also bring redundancy and noise that will reduce the efficiency of the
68 model. Therefore, choosing an appropriate dimension reduction method is also an important step
69 in the process of DBPs identification. Hu et al. [14] fused four feature extraction methods of AAC,
70 pseudo predicted relative solvent accessibility (PsePRSA), PsePSSM, and pseudo predicted
71 probabilities of DNA-binding sites (PsePPDBS). Support vector machine recursive feature
72 elimination and correlation bias reduction (SVM-RFE+CBR) [15] was used to convert the
73 nonlinear learning issue in the original feature space to a linear learning issue in the high
74 dimension feature space. The optimal feature subset containing 131-dimension vectors was
75 obtained by SVM-RFE+CBR. Zhou et al. [16] used dipeptide deviation from the expected mean
76 (DDE), normalized Moreau-broto autocorrelation (NMBAC), PSSM-distance-bigram
77 transformation (PSSM-DBT), and PSSM-discrete wavelet transformation (PSSM-DWT) to extract
78 features. After fusing the obtained features, SVM-RFE+CBR was used for dimension reduction to
79 obtain a feature subspace containing 424-dimension vectors. Ali et al. [17] performed feature
80 extraction based on PSSM, PSSM-DWT, and split amino acid composition (SAAC). Then they
81 used maximum relevance and minimum redundancy (mRMR) to decrease the number of fused
82 features. mRMR sorted each feature in the feature space according to the maximum relevance and
83 minimum redundancy with the target class, and finally obtained the optimal subset containing
84 264-dimension features. Ji et al. [18] adopted AAC, DC, chaos game representation (CGR), fractal
85 dimension (FD), composition transition and distribution (CTD), Moreau-Broto (MB), PseAAC,
86 sequence order (SO) and PSSM to extract features of the training dataset. Multi-class MSVM-RFE
87 was used for dimension reduction. MSVM-RFE converted the multi-objective optimization issue
88 to a single-objective optimization issue. The redundant features are gradually removed according
89 to the sorting criteria, and the optimal subset containing 100-dimension features is obtained.

90 In addition to choosing appropriate feature extraction and feature selection algorithms,
91 another key factor for the success of DBPs prediction is the choice of classification algorithms.
92 Appropriate classification algorithms can efficiently shorten the running time and learn the
93 relationship between tags and categories. Some machine learning methods are commonly used,
94 such as K Nearest Neighbor (KNN) [19], Neural Network [20], Naïve Bayes [21], Hidden Markov
95 Model [22], Gradient Boosting Decision Tree (GBDT) [23], Support Vector Machine (SVM) [24]
96 and (RF) [25] and etc. Ali et al. [26] proposed the DP-BINDER model. According to the feature
97 selection method SVM-RFE+CBR, 84-dimension features were input into RF and SVM for
98 prediction. Based on the LOOCV, the prediction accuracy of the training dataset PDB1075
99 reached 92.46% and 91.72%, respectively. Kumar et al. [27] used amino acid and dipeptide

100 composition, PSSM-400, four-part amino acid composition for feature extraction. Additionally,
101 SVM was used for prediction. The ACC of the model reached 74.22%. Wei et al. [28] proposed
102 the Local-DPP model, which used Local PsePSSM to get the local protection information. Taking
103 the obtained 120-dimension feature vectors as the input of RF, the ACC of the Local-DPP model
104 over the LOOCV reached 79.2%. Chauhan et al. [29] added 0 vectors to the PSSM to generate a
105 fixed-length padded matrix (pPSSM) and then used deep convolutional neural networks (CNNs)
106 to predict DBPs. Liu et al. [30] proposed the MFSBinder method, which used Local-DPP, 188D,
107 PSSM-DWT, and AC-struct to extract evolutionary information, sequence information,
108 physicochemical properties, and structural information, respectively. Finally, a stacked ensemble
109 classifier was used to predict DBPs. Xu et al. [31] extracted physicochemical property, amino acid
110 composition and distribution information. Then the features were used to predict DBPs based on
111 unbalanced-AdaBoost. Liu et al. [32] proposed the iDNA-KACC model which combined
112 contour-based protein expression, self-crossing covariance transformation, and Kmer composition
113 features. The features were fed to an ensemble classifier composed of 4 SVMs for prediction. The
114 ACC of the iDNA-KACC model was 75.16% based on LOOCV.

115 Although the existing methods can effectively predict DBPs, the running speed and accuracy
116 of the methods need to be improved. First, the influence of protein sequence features on DBPs
117 prediction has not been fully elucidated. It still has to be improved in DBPs prediction by
118 extracting features based on protein sequences. Second, feature fusion brings redundancy and
119 noise. Choosing a suitable dimension reduction method can reduce the feature dimension while
120 retaining effective information. Finally, since the number of protein sequences continuously
121 increase, choosing an effective classifier is also a major challenge for researchers.

122 Hence, we proposed a new DBPs prediction model, called StackPDB. Firstly, the training
123 dataset PDB1075 was encoded into EDT, RPT, PseAAC, PsePSSM, and PSSM-TPC. Compared
124 with the individual feature, the fusion feature can obtain more comprehensive protein information.
125 Secondly, we applied XGB_RFE to the DBPs prediction field for the first time. XGB_RFE can
126 speed up the process of the StackPDB model and choose the best features while deleting irrelevant
127 features and reducing the feature dimension. Finally, the stacked ensemble classifier was used as
128 the final classifier. In the first stage, two XGBoost and two LightGBM were used for the first time.
129 Then the output probability of the base-classifier was input into the meta-classifier SVM for DBPs
130 prediction. The ACC of StackPDB on the training dataset PDB1075 reached 93.44% over the
131 LOOCV test. Using the independent test datasets PDB186 and PDB180 to test the generalization
132 ability of the StackPDB model, StackPDB obtained an ACC value of 84.40% and 90.00%,
133 respectively. Compared with other competitive methods, StackPDB has higher stability and can
134 significantly improve the recognition ability of DBPs.

135 2. Materials and methods

136 2.1. Datasets

137 Choosing the appropriate data set is a key step to build a model. In this article, we chose the
138 dataset PDB1075 as the training dataset. Xu et al. [33] established the training dataset PDB1075
139 which contains 525 DBPs and 550 non-DBPs. The dataset construction process met the following
140 criteria: (1) Searching from the updated protein database (PDB) to acquire DBPs sequences; (2)
141 Protein sequences that less than 50 in length or contained the character "X" were removed; and (3)
142 Sequences with sequence similarity greater than 25% in the same dataset were removed by the
143 software PISCES. During the experiment in this article, we found 8 abnormal sequences in the
144 training dataset: (1) 1AOIL, (2) 4FCYC, (3) 4JJNJ, (4) 4JJNI, (5) 3THWD, (6) 4GNXL, (7)
145 4GNXZ, (8) 2RAUA, where the first four were DNA sequences, and the PSSM matrix of the last
146 four sequences were not available in the PSI-BLAST [34] program. After deleting abnormal
147 sequences, the training dataset consists of 518 DBPs and 549 non-DBPs were used in this article.

148 To test our model, we chose PDB186 and PDB180 as independent test datasets. The
149 independent test dataset PDB186 was collected by Lou et al. [35] which contains 93 DBPs and 93
150 non-DBPs. The independent test dataset PDB180 was proposed by Xu et al. [36] which contains
151 81 DBPs and 99 non-DBPs. The two independent test sets used the same processing method in the
152 construction process. During the construction of two independent test sets, length of protein
153 sequences less than 60 or the character "X" were removed. BLASTCLUST software was used to
154 remove sequences with a sequence similarity greater than 25% in the same dataset.

155 2.2. Feature extraction

156 2.2.1. Pseudo amino acid composition (PseAAC)

157 Chou [37] proposed PseAAC, which extracted protein sequence and physicochemical
158 information. PseAAC has been applied in many fields, e.g., the subcellular location of apoptosis
159 proteins [38], protein structural prediction [39], protein post-translational modification site
160 prediction [40], protein submitochondrial localization prediction [41] and etc.

161 The feature vector is obtained by PseAAC as follows:

$$162 X = [x_1, x_2, \dots, x_u, \dots, x_{20+\lambda}]^T \quad (\lambda < L) \quad (1)$$

163 The calculation method x_u is shown in formula (2)

$$164 \quad x_u = \begin{cases} \frac{f_u}{\sum_{u=1}^{20} f_u + \omega \sum_{m=1}^{\lambda} \theta_m}, (1 \leq u \leq 20) \\ \frac{\omega \theta_{u-20}}{\sum_{u=1}^{20} f_u + \omega \sum_{m=1}^{\lambda} \theta_m}, (20+1 \leq u \leq 20+\lambda) \end{cases} \quad (2)$$

165 where L represents the length of the protein sequence while f_u is the frequency of the u -th
 166 amino acid in the protein sequence S . θ_m is the m -layer sequence correlation factor. ω is the
 167 weighting factor where $\omega = 0.05$. PseAAC extracts $20 + \lambda$ -dimension feature vectors. The first
 168 20-dimension vectors represent amino acid sequence information, and the latter λ -dimension
 169 represents amino acid sequence order information and physicochemical properties.

170 2.2.2. Position-specific scoring matrix (PSSM)

171 Evolutionary information is vital information in protein function annotation. It has been
 172 widely used in many fields, such as protein-protein interaction prediction [42], RNA-protein
 173 interaction prediction [43], DNA binding proteins prediction [44] and etc. In this paper, PsePSSM,
 174 PSSM-TPC, EDT, and RPT are used to extract evolutionary information. The four feature
 175 extraction methods are based on the PSSM, so PSSM is initially introduced. Jones et al. [45]
 176 firstly proposed PSSM, using the PSI-BLAST [34] program to perform three iterative searches in
 177 the Swiss-Prot database, and the E value threshold was set as 0.001. By performing multiple
 178 sequences comparisons on protein sequences, a $L \times 20$ PSSM is generated, as shown in formula
 179 (3).

$$180 \quad PSSM = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,20} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,20} \\ \cdots & \cdots & \cdots & \cdots \\ p_{L,1} & p_{L,2} & \cdots & p_{L,20} \end{bmatrix}_{L \times 20} \quad (3)$$

181 where $p_{i,j}$ represents the score of the i -th amino acid mutates into the j -th standard amino
 182 acid during the evolution process. L represents the length of the protein sequence. To eliminate
 183 the dimensional error, the PSSM is standardized according to formula (4):

$$184 \quad p'_{i,j} = \frac{p_{i,j} - \frac{1}{20} \sum_{k=1}^{20} p_{i,k}}{\sqrt{\frac{1}{20} \sum_{l=1}^{20} \left(p_{i,l} - \frac{1}{20} \sum_{k=1}^{20} p_{i,k} \right)^2}}, (i = 1, 2, \dots, L; j = 1, 2, \dots, 20) \quad (4)$$

185 where $p'_{i,j}$ represents the PSSM element after standardization. PSSM is changed to a vector with
 186 equal length by formula (5-6).

$$187 \quad P_{PSSM} = [p_1, p_2, \dots, p_{20}]^T \quad (5)$$

$$188 \quad p_j = \frac{1}{L} \sum_{i=1}^L p'_{i,j}, \quad (j = 1, 2, \dots, 20) \quad (6)$$

189 where P_{PSSM} represents a feature vector of length 20 and p_j represents a vector element.

190 2.2.3. Pseudo-position specific scoring matrix (PsePSSM)

191 Although P_{PSSM} contains evolutionary information, it ignores the sequence order
 192 information. At present, PsePSSM [46] has been applied to human protein subcellular localization
 193 identification [47], protein submitochondrial localization [48], drug-target interaction prediction
 194 [49], membrane protein recognition [50] and etc. PsePSSM is shown in equation (7).

$$195 \quad P_{PsePSSM} = [p_1, p_2, \dots, p_{20}, p_1^\xi, p_2^\xi, \dots, p_{20}^\xi]^T, (\xi = 0, 1, \dots, L-1) \quad (7)$$

196

$$197 \quad p_j^\xi = \frac{1}{L-\xi} \sum_{i=1}^{L-\xi} (p'_{i,j} - p'_{i+\xi,j})^2, (j = 1, 2, \dots, 20) \quad (8)$$

198 where p_j^ξ represents the correlation of the PSSM score between two amino acids separated by ξ .

199 The accuracy of prediction is changed by adjusting ξ . We took ξ from 1 to 9 with 1 as the
 200 interval and determined the optimal ξ value of 2. According to PsePSSM, $20+20 \times \xi = 60$
 201 -dimension feature vectors can be obtained for each protein sequence.

202 2.2.4. Position-specific scoring matrix-transition probability composition (PSSM-TPC)

203 To reduce the loss of sequence information in the evolution process, transition probability
 204 composition (TPC) is applied to PSSM. The procedure given in [51] is used to calculated TPC by
 205 the transition probability matrix (TPM). The PSSM-TPC vector can be expressed by formula (9):

$$206 \quad P_{PSSM_TPC} [\bar{P}'_{1,1}, \dots, \bar{P}'_{1,20}, \dots, \bar{P}'_{i,1}, \dots, \bar{P}'_{i,20}, \dots, \bar{P}'_{20,1}, \dots, \bar{P}'_{20,20}] \quad (9)$$

$$207 \quad \bar{P}'_{m,n} = \frac{\sum_{k=1}^{L-1} Y_{k,i} \times Y_{k+1,j}}{\sum_{j=1}^{20} \sum_{k=1}^{L-1} Y_{k+1,j} \times Y_{k,i}}, 1 \leq i, j \leq 20 \quad (10)$$

208 where $\bar{P}'_{m,n}$ represents the transition probability from the m -th amino acid to the n -th amino
 209 acid. $Y_{i,j}$ which satisfies $\sum_{j=1}^{20} Y_{i,j} = 1, (i = 1, 2, \dots, L)$ represents the relative probability of the j -th
 210 amino acid appearing at the i -th position.

211 2.2.5. Evolutionary distance transformation (EDT)

212 EDT was proposed by Zhang et al. [52] which calculated the non-co-occurrence probability
 213 of two amino acids. The amino acids are separated by d ($d = 1, 2, \dots, L_{\min} - 1$). EDT can be
 214 calculated by the formula (11):

$$215 \quad P_{EDT} = [f(P_1, P_1), f(P_1, P_2), \dots, f(P_1, P_{20}), \dots, f(P_x, P_y), \dots, f(P_{20}, P_{20})] \quad (11)$$

216 The non-co-occurrence probability $f(P_x, P_y)$ of two amino acids separated by d can be

217 calculated by the formula (12):

$$218 \quad f(P_x, P_y) = \sum_{d=1}^{L_{\min}-1} \frac{1}{L-d} \sum_{i=1}^{L-d} (P_{i,x}, P_{i+d,y}), (x, y = 1, 2, \dots, 20) \quad (12)$$

219 where L_{\min} represents the minimum sequence length and P_x, P_y represents 20 different standard
 220 amino acids. $P_{i,x}$ and $P_{i+d,y}$ are both elements in PSSM. Hence, EDT extracts 400-dimension
 221 features representing non-collinear probability information.

222 2.2.6. Residue probing transformation (RPT)

223 RPT was proposed by Jeong et al. [53], grouping the evolution scores in the PSSM to
 224 emphasize domains with similar conservation. The rows of the same amino acid in the PSSM are
 225 divided into one group. Thus, a total of 20 groups are obtained. For each group, the sum of the
 226 elements in each column is calculated. In this way, each protein sequence can get an 20×20 RPT
 227 matrix, as shown in equation (13):

$$228 \quad RPT = \begin{bmatrix} R_{1,1} & R_{1,2} & \cdots & R_{1,20} \\ R_{2,1} & R_{2,2} & \cdots & R_{2,20} \\ \cdots & \cdots & \cdots & \cdots \\ R_{20,1} & R_{20,2} & \cdots & R_{20,20} \end{bmatrix} \quad (13)$$

229 A 400-dimension row vector is obtained by expanding the RPT matrix, as shown in formula
 230 (14):

$$231 \quad P_{RPT} = [v_{R_{1,1}}, v_{R_{1,2}}, \dots, v_{R_{i,j}}, \dots, v_{R_{20,20}}] \quad (14)$$

$$232 \quad v_{R_{i,j}} = \frac{R_{i,j}}{L}, (i, j = 1, 2, \dots, 20) \quad (15)$$

233 where $R_{i,j}$ represents the RPT element. L is the sequence length, and P_{RPT} represents the
 234 400-dimension feature vector obtained by RPT.

235 2.3. Extreme gradient boosting-recursive feature elimination (XGB-RFE)

236 The XGBoost algorithm was proposed by Yu et al. [54], which sorted the input features
 237 according to their importance. First, the algorithm uses XGBoost to obtain significance mark of
 238 every feature, and assign weights to the features. Then, the weighted sum of the scores of each
 239 feature in all boost trees is used to obtain the final importance score. Then the features are sorted
 240 according to the final score. In this paper, XGBoost and recursive feature elimination algorithm
 241 (RFE) [55] are combined for the first time in the field of DBPs prediction.

242 Given a set $D = \{(x_{i,1}, y_i), (x_{i,2}, y_i), \dots, (x_{i,m}, y_i)\}$, the element $(x_{i,m}, y_i) = (x_{i,1}, x_{i,2}, \dots, x_{i,m})$

243 indicates that the label of m -th feature vector is y_i .

244
$$\hat{y}_i = \sum_{k=1}^K f_k(x_i) \quad (16)$$

245 where $f_k(x_i)$ represents the importance score of i -th feature vector on k -th tree.

246 Then the objective function can be expressed as formula (17):

247
$$L(\phi) = \sum_i l(y_i, \hat{y}_i) + \sum_k \Omega(f_k) \quad (17)$$

248 where $l(\hat{y}_i, y_i)$ represents the loss between the true value and the predicted value.

249 $\Omega(f) = \gamma T + \frac{1}{2} \lambda \omega^2$ controls the complexity of the model.

250 Assuming that each iteration can generate a tree, the objective function becomes as follows.

251

252
$$L(\phi)^{(t)} = \sum_i l(y_i, \hat{y}_i^{(t)}) + \sum_k \Omega(f_k) \quad (18)$$

253 where $\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i)$ represents the predicted value of t -th iteration. Supposing the $k-1$

254 -th tree is known while generating the k -th tree.

255
$$L^{(t)} = \sum_{i=1}^n \left[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} b_i f_t^2(x_i) \right] + \Omega(f_t) \quad (19)$$

256 where $L^{(t)}$ is the objective function. $g_i = \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$ and $b_i = \partial_{\hat{y}_i^{(t-1)}}^2 l(y_i, \hat{y}_i^{(t-1)})$ represent

257 the first-order and second-order statistics of the loss function, respectively.

258 After getting the importance ranking of features, RFE is used to delete the least important
259 features from the current feature space. The process repeats N times until the required number of
260 features is obtained.

261 2.4. Stacked ensemble classifier

262 The stacked ensemble classifier is an integrated method proposed by Wolpert et al. [56]. The
263 prediction results of multiple ordinary learners are used as new features for retraining. By doing
264 this, the stacked ensemble classifier can achieve the purpose of minimizing the error rate of the
265 prediction model. At present, this method has been applied to predict ncRNA-protein interactions
266 [57], Bacterial Type IV Secreted Effectors [58], anticancer drug response [59], MicroRNA
267 automatic classification [60] and etc. In this paper, a stacked ensemble classifier which including
268 two stages of learning is used to predict DBPs. In the first stage, the features are input into the
269 base-classifier to output the binding probability and non-binding probability of DBPs. In order to
270 enrich the features that are input into the meta-classifier, we chose base-classifier from 9
271 classifiers, e.g., k-nearest neighbor (KNN) [61], support vector machines (SVM) [62], random
272 forest (RF) [63], gradient boosting decision tree (GBDT) [64], Naïve Bayes classifier (NB) [65],
273 logistic regression (LR) [66], light gradient boosting machine (LightGBM) [67], extreme gradient
274 boosting (XGBoost) [54], and adaptive boosting (AdaBoost) [68]. Finally, XGBoost and
275 LightGBM are selected as the best combination of base-classifier. Then the output results of the

276 first stage input into the meta-classifier. To make full use of the features from the first stage, we
277 chose the best meta-classifier among 9 classifiers, e.g., NB, XGBoost, AdaBoost, LightGBM,
278 KNN, RF, GBDT, LR, and SVM. The prediction results show that the StackPDB model
279 constructed by the meta-classifier SVM and the base-classifier XGBoost and LightGBM is the
280 best. Finally, two XGBoost and two LightGBM are used as the base-classifier, and SVM is our
281 meta-classifier. Algorithm 1 represents the pseudo code of the stacked ensemble classifier.

Algorithm 1 Stacked ensemble classifier

Input: training data $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$;

Base-classifier $\zeta_1, \zeta_2, \dots, \zeta_T$;

Meta-classifier ζ .

Output: ensemble classifier H

1: *Step 1: learn base-classifiers*

2: **for** $t = 1, 2, \dots, T$ **do**

3: $h_t = \zeta_t(D)$;

4: **end for**

5: $D' = \emptyset$;

6: *Step 2: construct new dataset of predictions*

7: **for** $i = 1, 2, \dots, m$ **do**

8: **for** $t = 1, 2, \dots, T$ **do**

9: $z_{it} = h_t(x_i)$;

10: **end for**

11: $D' = D' \cup ((z_{i1}, z_{i2}, \dots, z_{iT}), y_i)$;

12: *Step 3: learn a meta-classifiers*

13: $h' = \zeta(D')$;

14: $H = h'(h_1(x), h_2(x), \dots, h_T(x))$;

15: **return** H

282 2.5. Model construction and evaluation

283 In this study, we propose a novel model for predicting DBPs, called StackPDB, and the
284 flowchart is shown in Fig. 1. All experiments are performed on Windows Server 2012r 2 Intel (R)
285 Xeon (TM) CPU E5-2650@2.30GHz 2.30GHz, 32.0GB memory, MATLAB2014a, and Python
286 3.6 programming. The specific algorithm flow is as follows:

287 1) Data preparation. The training dataset PDB1075 and the independent test datasets PDB186
288 and PDB180 were obtained from the protein database. The protein sequences and their

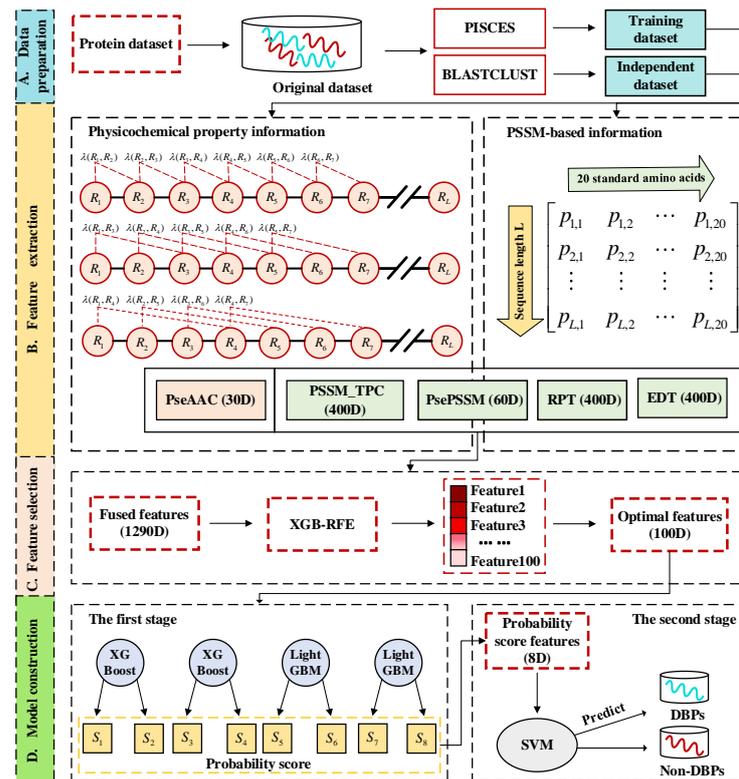
289 corresponding DBPs labels were entered into StackPDB.

290 2) Feature extraction. 400-dimension feature vectors were obtained from EDT, RPT, and
 291 PSSM-TPC, respectively. 30-dimension and 60-dimension feature vectors were obtained from
 292 PseAAC and PsePSSM, respectively. After fusing the five features, an initial feature space that
 293 contained 1290-dimension vectors was obtained.

294 3) Feature selection. The feature selection algorithm XGB-RFE was used to remove the
 295 redundancy and noise of the initial feature space in 2). Then 100-dimension optimal feature
 296 vectors were obtained.

297 4) Model construction. The optimal feature vector was input into the base-classifier XGBoost
 298 and LightGBM to output the binding probability and non-binding probability of DBPs. The output
 299 probability of the base-classifier was input into the meta-classifier SVM to construct the
 300 StackPDB.

301 5) Model verification and evaluation. The effectiveness of StackPDB was tested on the
 302 independent test datasets PDB186 and PDB180.



303
 304 **Fig. 1.** Flow chart of StackPDB. StackPDB firstly collects datasets (A) and then uses five methods
 305 to extract protein features (B). StackPDB reduces the dimension of the fusion features (C). Finally
 306 stacked ensemble classifier predicts whether the sequence is DBPs or non-DBPs (D).

307 The LOOCV [69], K-fold cross-validation method, and an independent test method are
 308 commonly used methods to evaluate the performance of the model. The LOOCV method is
 309 chosen as the validation method. In the verification process, LOOCV selects N-1 samples as the

310 training set and one sample as the test set. LOOCV trains N times on the data set to ensure that
311 each sequence is tested. LOOCV can calculate the accuracy of the prediction model objectively
312 and rigorously and test the generalization ability of the model. It has been widely used in
313 proteomics research [70].

314 Five evaluation indicators are used to evaluate the quality of the model: Accuracy (ACC),
315 Sensitivity (SN), Matthew's Correlation Coefficient (MCC), and Specificity (SP) [71].

$$316 \quad ACC = \frac{TN + TP}{TN + TP + FN + FP} \quad (20)$$

$$317 \quad SN = \frac{TP}{TP + FN} \quad (21)$$

$$318 \quad SP = \frac{TN}{TN + FP} \quad (22)$$

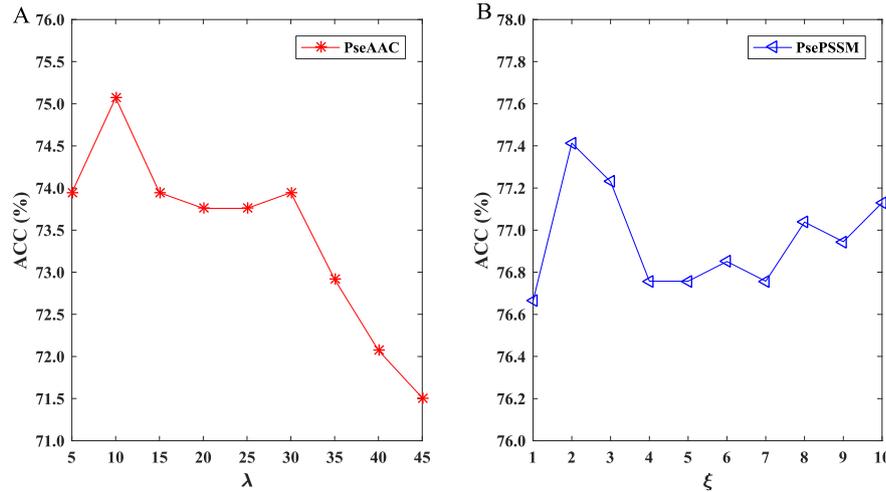
$$319 \quad MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FN) \times (TN + FP) \times (TP + FP) \times (TN + FN)}} \quad (23)$$

320 where FN represents the number of DBPs predicted as non-DBPs, FP represents the number of
321 non-DBPs predicted as DBPs, TN represents the number of non-DBPs predicted correctly, and TP
322 represents the number of correct DBPs predicted. Besides, the area under the ROC curve (AUC)
323 and the area under the PR curve (AUPR) are also used as important indicators for evaluating the
324 quality of the model [72, 73].

325 **3. Results and discussion**

326 *3.1. Selection of feature extraction parameters λ and ξ*

327 It is essential to select the excellent parameters when constructing StackPDB model. If the
328 parameter is set too small, the information will be insufficiently extracted. If the parameter is too
329 large, redundant features will be produced. When selecting the best parameter λ in PseAAC, the
330 value λ is set to 5~45 with an interval of 5. Similarly, the parameter ξ in PsePSSM is set to
331 1~10 with an interval of 1. The features with different parameters are used as the input of the
332 stacked ensemble classifier. The prediction results verified by the LOOCV are shown in
333 Supplementary Table S1 and Table S2. The influence of different λ of PseAAC and ξ of
334 PsePSSM on ACC is shown in Fig. 2.



335

336 **Fig. 2.** The effect of choosing different λ (A) and ξ (B) values on the training dataset PDB1075.

337 In Fig. 2 (A), the performance of PseAAC changes when λ gradually increases. The ACC
338 value of PseAAC is the largest when $\lambda = 10$. As λ increasing, the ACC value of StackPDB
339 decreases. As we can see from Fig. 2 (B), the performance of PsePSSM changes when ξ
340 increases. The ACC of PsePSSM reaches the maximum value when $\xi = 2$, and then it gradually
341 decreases. When $\lambda = 10$ the ACC value of PseAAC reaches a maximum of 75.07%. When
342 $\xi = 2$ the ACC of PsePSSM reaches the maximum value of 77.41%, which can fully express
343 protein information. We choose $\lambda = 10$ as the best parameter of PseAAC so that the PseAAC
344 features can be fully extracted. Finally $20 + \lambda = 30$ -dimension feature vectors can be obtained by
345 PseAAC. Similarly, we choose $\xi = 2$ as the best parameter of PsePSSM, so that the PsePSSM
346 features can be fully extracted. Finally $20 + 20 \times \xi = 60$ -dimension feature vectors can be obtained
347 by PsePSSM.

348 3.2. Comparison of different feature extraction methods

349 After determining the best parameters of PseAAC and PsePSSM, EDT, RPT, PseAAC,
350 PsePSSM, and PSSM-TPC are fused to obtain more comprehensive information. To measure the
351 differences between EDT, RPT, PseAAC, PsePSSM, and PSSM-TPC, the 5 individual features
352 and the fusion feature (Fusion) are fed to the stacked ensemble classifier. The results based on the
353 LOOCV are shown in Table 1.

354 **Table 1**

355 Performance of 5 feature extraction methods on the training dataset PDB1075.

Algorithm	ACC (%)	SN (%)	SP (%)	MCC
EDT	76.10	80.50	71.95	0.5256
RPT	76.76	83.59	70.31	0.5425
PseAAC	75.07	74.71	73.22	0.4791
PsePSSM	77.41	81.85	73.22	0.5519

PSSM-TPC	82.48	78.98	85.79	0.6521
Fusion	89.50	87.07	91.80	0.7903

356 It can be seen from Table 1 that PSSM-TPC performs best among 5 features with an ACC
357 value of 82.48% and an MCC value of 0.6521. The ACC values of EDT, RPT, PseAAC and
358 PsePSSM are 76.10%, 76.76%, 75.07% and 77.41%, respectively, and the MCC values are 0.5256,
359 0.5425, 0.4791, 0.5519, respectively. For the Fusion features, the value of each evaluation index is
360 improved based on the LOOCV. The MCC and ACC of Fusion are 0.7903 and 89.50%,
361 respectively, which are 13.82% and 7.02% higher than the best single feature PSSM-TPC. Besides,
362 we draw the ROC and PR curves between the single feature extraction method and Fusion as
363 shown in Supplementary Figure S1. The results show that an individual feature can only capture a
364 single aspect of the protein sequence. The Fusion features can obtain more comprehensive
365 information so that it improves the prediction accuracy of DBPs. Nevertheless, multi-information
366 fusion will inevitably bring redundant information.

367 3.3. Comparison of different dimension reduction methods

368 The dimension reduction method can delete the redundancy while reducing the feature
369 dimension and selecting the optimal feature. After applying fusion of EDT, RPT, PseAAC,
370 PsePSSM, and PSSM-TPC, 1290-dimension feature vectors are obtained. In this paper, 7 feature
371 selection methods are tested on training dataset PDB1075, namely LASSO [47], Elastic net [74],
372 SVM-RFE [26], LinearSVC [75], locally linear embedding (LLE) [76], singular value
373 decomposition (SVD) [77] and XGB_RFE [54]. The parameters are set as follows, (1) The penalty
374 parameter of LASSO is 0.01, thus 197-dimension features are selected; (2) L1_ratio of Elastic net
375 is set to 0.4; (3) SVM-RFE selects the linear kernel function; (4) The penalty of LinearSVC is set
376 to L1; and (5) The optimal features of LLE, SVD and XGB_RFE are set to 100. The final number
377 of features retained by LASSO, Elastic net, SVM-RFE, and LinearSVC are 197, 144, 100, and 386
378 respectively. The optimal feature subsets obtained by different dimension reduction methods are
379 classified by stacked ensemble classifier. The prediction results are shown in Table 2.

380 **Table 2**

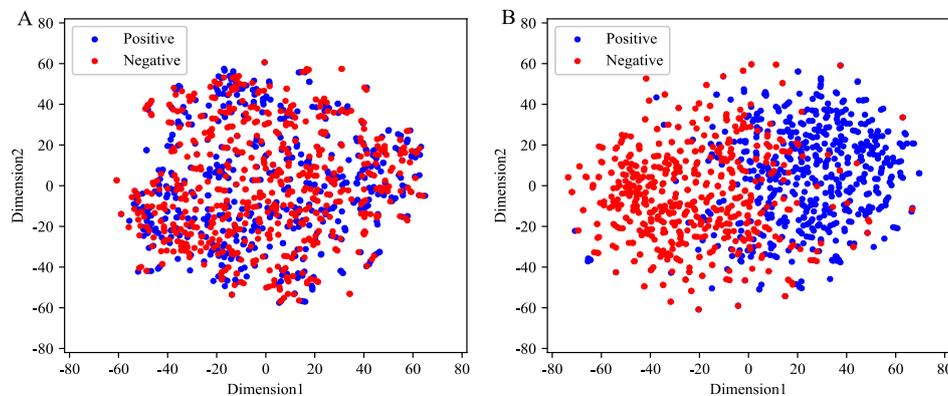
381 Performance of 7 dimension reduction methods on training dataset PDB1075.

Algorithm	ACC (%)	SN (%)	SP (%)	MCC
LLE	78.26	82.82	73.95	0.5690
SVD	82.10	83.59	80.69	0.6426
SVM-RFE	90.82	88.22	93.26	0.8167
LASSO	91.75	89.38	93.99	0.8354
Elastic net	92.60	91.12	93.99	0.8519
LinearSVC	92.03	91.70	92.35	0.8405
XGB-RFE	93.44	93.44	93.44	0.8687

382 It can be seen from Table 2 that XGB_RFE has the best performance among the 7 dimension
383 reduction methods. The values of ACC and MCC both reach the highest which are 93.44% and
384 0.8687 respectively. The ACC value of XGB_RFE is 15.18%, 11.34%, 2.62%, 1.69%, 0.84% and
385 1.41% higher than LLE, SVD, SVM-RFE, LASSO, Elastic net and LinearSVC respectively. The
386 MCC value of XGB_RFE is 29.97%, 22.61%, 5.20%, 3.33%, 1.68% and 2.82% higher than LLE,
387 SVD, SVM-RFE, LASSO, Elastic net and LinearSVC respectively. ROC and PR curves can more
388 intuitively compare the performance of 7 different feature selection methods in Supplementary
389 Figure S2. From the above analysis, it shows that XGB-RFE can reduce model complexity while
390 eliminating redundant and irrelevant features. It can also improve model accuracy and shorten
391 model running time. Therefore, we choose XGB-RFE as the dimension reduction method and
392 finally get the 100-dimension optimal feature.

393 3.4. Feature visualization

394 The distribution of the Fusion feature and the optimal feature (Fusion (XGB-RFE)) are
395 shown in the feature space to explain that XGB-RFE can improve prediction accuracy. For
396 comparison, the original feature space and the optimal feature space are converted to a
397 two-dimension space by T-distributed Stochastic Neighbor Embedding (t-SNE) [78]. The t-SNE
398 visualization is shown in Fig. 3.



399
400 **Fig. 3.** The t-SNE visualization of the Fusion feature (A) and Fusion (XGB-RFE) features (B) in
401 two-dimension space.

402 It can be seen from Fig. 3 (A) that the positive and negative examples of the Fusion feature
403 are mixed in a two-dimension space. There is no obvious distinction between the positive
404 examples and negative examples, which brings greater challenges to the prediction of DBPs.
405 Compared with the distribution of Fusion features, the distribution of positive and negative
406 samples in Fusion (XGB-RFE) features is more obvious from Fig. 3 (B). The positive and
407 negative examples are gathered in different areas in the two-dimension space, which can capture
408 the difference between the positive and negative samples. Also, XGB-RFE is effective in

409 transforming features from high-dimension space to low-dimension space, which can shorten
410 training time. It can provide more effective information for the identification of DBPs and
411 improve the prediction accuracy of the model.

412 3.5. Selection of base-classifier

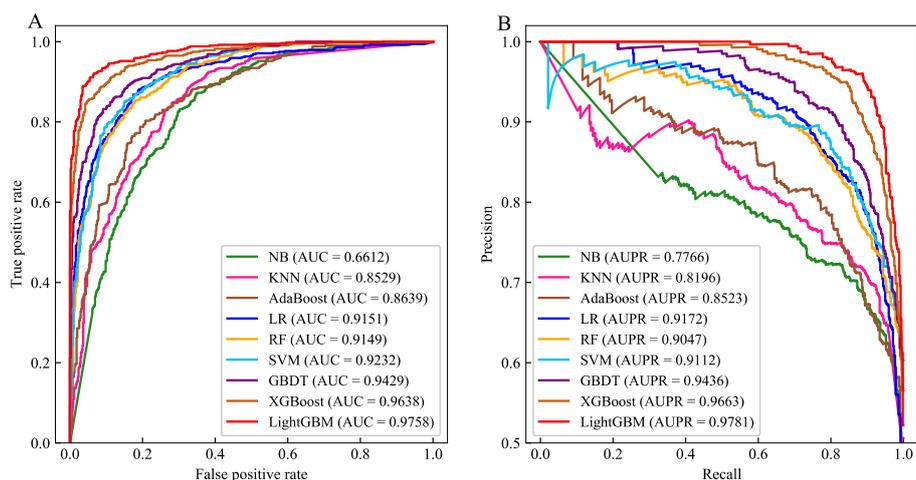
413 To determine the most suitable classifier, 9 machine learning classifiers are tested. The
414 parameters of 9 machine learning classifiers are as follows, i.e., (1) The closest neighbor of KNN
415 is 5; (2) SVM uses the RBF kernel function; (3) RF sets the number of base decision trees to 500
416 and the maximum learning depth to 10; (4) The number of GBDT iterations is 500; (5) The
417 number of iterations of XGBoost is 500; (6) AdaBoost sets the number of base decision trees to
418 500; (7) The number of iterations of LightGBM is 500; and (8) NB and LR use default parameters.
419 The prediction results of 9 classifiers on the training dataset PDB1075 are as Table 3.

420 **Table 3**

421 Performance of 9 base-classifiers on the training dataset PDB1075.

Model	ACC (%)	SN (%)	SP (%)	MCC
NB	65.60	36.68	92.90	0.3600
KNN	75.54	66.02	84.52	0.5156
RF	83.51	84.92	82.15	0.6707
LR	83.88	81.47	86.16	0.6775
SVM	84.72	85.33	84.15	0.6945
AdaBoost	86.41	84.56	88.16	0.7280
GBDT	86.69	84.17	89.07	0.7339
XGBoost	90.07	88.42	91.62	0.8013
LightGBM	92.59	89.59	95.45	0.8528

422 In Table 3, the ACC of NB, KNN, RF, LR, SVM, AdaBoost, GBDT, XGBoost and
423 LightGBM are 65.60%, 75.54%, 83.51%, 83.88%, 84.72%, 86.41%, 86.69%, 90.07%, and
424 92.59%, respectively. The ACC of LightGBM is 26.99% and 17.05% higher than that of NB and
425 KNN. The ACC values of LightGBM and XGBoost classifiers both exceed 90%. XGBoost is only
426 2.52% lower than LightGBM. The MCC of LightGBM and XGBoost are 0.8528 and 0.8013,
427 respectively. LightGBM is 0.4928 higher than NB on MCC, and XGBoost is 0.4413 higher than
428 NB on MCC.



429

430 **Fig. 4.** ROC and PR curve of different base-classifiers on the training dataset PDB1075.

431 The ROC and PR curves can more vividly represent the performance of 9 different classifiers,
 432 as shown in Fig. 4. In Fig. 4, the AUC of LightGBM is 0.9758, which is the highest among 9
 433 base-classifiers. The area covered by ROC curve of XGBoost is second-largest with an AUC value
 434 of 0.9638. From Fig. 4 (B), the AUPR value of LightGBM is largest which is 0.9781. The AUPR
 435 value of XGBoost is second-largest which is 0.9663. Considering the performance of 9
 436 base-classifiers, XGBoost and LightGBM have high accuracy and stability. Thus, XGBoost and
 437 LightGBM are selected as the best combination of base-classifier.

438 3.6. Selection of meta-classifier

439 After the training on the first stage, the binding probability and non-binding probability of
 440 each protein sequence are obtained from LightGBM and XGBoost. The output probability is input
 441 into the meta-classifier for training again. Therefore, the choice of meta-classifier also plays a
 442 significant role in the model establishment. The specific parameters of 9 classifiers are as follows,
 443 (1) the number of XGBoost iterations is 500; (2) The base-classifier of AdaBoost and GBDT both
 444 select decision trees (500); (3) LightGBM iterates 500 times; (4) The number of KNN neighbors is
 445 5; (5) SVM uses the RBF kernel function; (6) The base decision trees number of RF is 500 and the
 446 maximum learning depth as 10; and (7) NB and LR use default parameters. The performance of
 447 9 meta-classifiers is shown in Table 4.

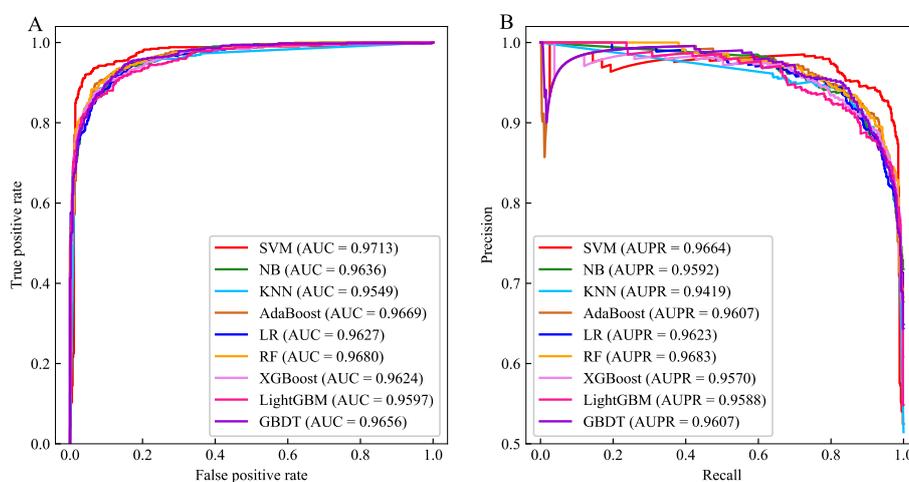
448 **Table 4**

449 The performance of 9 meta-classifiers on the training dataset PDB1075.

Model	ACC (%)	SN (%)	SP (%)	MCC
NB	89.50	89.58	89.44	0.7900
XGBoost	88.38	91.70	85.25	0.7698
AdaBoost	89.03	93.05	85.25	0.7838
LightGBM	88.75	91.12	86.52	0.7763
KNN	89.03	92.66	85.61	0.7833

LR	89.41	88.42	90.35	0.7880
GBDT	89.60	93.24	86.16	0.7946
RF	90.07	93.44	86.89	0.8036
SVM	93.44	93.44	93.44	0.8687

450 In Table 4, SVM outperforms 9 classifiers. SVM has 93.44% ACC, which is 4.41%, 4.03%,
 451 3.84%, and 3.37% higher than KNN, LR, GBDT, and RF respectively. The MCC of SVM is
 452 0.8687, which is 7.87%, 9.89%, 8.49% and 9.24% higher than NB, XGBoost, AdaBoost and
 453 LightGBM respectively. The combination of SVM, XGBoost, and LightGBM increases the
 454 diversity of the stacked ensemble classifier and obtains better prediction results. We further
 455 evaluate the performance of the 9 meta-classifiers through ROC and PR curves, as shown in Fig.
 456 5.



457
 458 **Fig. 5.** The ROC and PR curves of 9 meta-classifiers on the training dataset PDB1075.

459 In Fig. 5., the area covered by ROC curve of SVM is maximal with an AUC value of 0.9713.
 460 The AUC value of SVM is 0.33%-1.64% higher than NB, KNN, AdaBoost, LR, RF, XGBoost,
 461 LightGBM, and GBDT (0.9731 vs. 0.9636, 0.9549, 0.9669, 0.9627, 0.9680, 0.9624, 0.9597,
 462 0.9656). The area covered under the PR curve of the SVM is 0.9664, which is 0.0019 lower than
 463 the AUPR value of RF. The AUPR value of SVM is 0.41%-2.45% higher than NB, KNN,
 464 AdaBoost, LR, RF, XGBoost, LightGBM, and GBDT (0.9664 vs. 0.9592, 0.9419, 0.9607,
 465 0.9623, 0.9683, 0.9570, 0.9588, 0.9607). Comparing with other classifiers, SVM shows strong
 466 predictive ability. SVM realizes the mapping from low-dimension space to high-dimension space
 467 by RBF function. The optimal hyperplane is found in the high-dimension space to distinguish
 468 between DBPs and non-DBPs. Thus, SVM is selected as a meta-classifier.

469 3.7. Comparison with other state-of-the-art methods

470 To verify the effectiveness of StackPDB, StackPDB is compared with PSSM-DT [33],
 471 HMMBinder [79], iDNAPro-PseAAC [80], DBPPred-PDSD [17], iDNAProt-ES [11], HMPred
 472 [13], Local-DPP [28], DP-BINDER [26]. PSSM-DT [33] proposed a new feature extraction

473 method PSSM distance transformation (PSSM-DT) and combined with SVM to predict DBPs.
474 HMMBinder [79] used monogram features and bigram features for feature extraction which
475 converted HMM matrix into the same length vectors. Then the feature vectors were input into
476 SVM to construct the HMMBinder model. iDNAPro-PseAAC [80] extracted protein sequence
477 features based on physicochemical properties and evolutionary information and used SVM to
478 construct iDNAPro-PseAAC. Table 5 shows the comparison of StackPDB and other published
479 methods.

480 **Table 5**

481 Comparison of StackPDB with other DBPs prediction methods on the training set PDB1075 based
482 on the LOOCV.

Methods	ACC (%)	SN (%)	SP (%)	MCC
iDNAPro-PseAAC [80]	76.56	75.62	77.45	0.5300
Local-DPP [28]	79.20	84.00	74.50	0.5900
PSSM-DT [33]	79.96	81.91	78.00	0.6220
HMMPred [13]	83.90	83.98	83.82	0.6800
HMMBinder [79]	86.33	87.07	85.55	0.7200
DBPPred-PDSD [17]	89.02	89.14	88.88	0.7800
iDNAProt-ES [11]	90.18	90.38	90.00	0.8000
DP-BINDER [26]	92.46	91.80	93.07	0.8400
StackPDB	93.44	93.44	93.44	0.8687

483 In Table 5, the ACC of StackPDB reaches 93.44%, which is 16.88%, 14.24%, 13.48%, 9.54%,
484 7.11%, 4.42%, 3.26% and 0.98% higher than the ACC values of iDNAPro-PseAAC, Local-DPP,
485 PSSM-DT, HMMPred, HMMBinder, DBPPred-PDSD, iDNAProt-ES and DP-BINDER,
486 respectively. The MCC of StackPDB is 0.8687, which exceeds the MCC values of
487 iDNAPro-PseAAC, Local-DPP, PSSM-DT and HMMPred by 33.87%, 27.87%, 24.67%, and
488 18.87% respectively. The histogram of StackPDB compared with other DBPs prediction methods
489 is shown in Supplementary Figure S3. Compared with other 8 published methods, StackPDB
490 performs the best.

491 To evaluate the predictive ability of StackPDB more fairly and objectively, PDB186 and
492 PDB180 are applied to verify our StackPDB. Then the test results are compared with several
493 published methods. The feature extraction parameters, dimension reduction method, and classifier
494 parameters of the independent test datasets are consistent with the training set, which can make the
495 test results more rigorous and reliable. Considering the validity of the comparison results, the test
496 results of the independent test set PDB186 are compared with those already published methods
497 HMMPred [13], HMMBinder [79], DBPPred [35], Local-DPP [28], PSSM-DT [33], MSFBinder
498 [30] and iDNAProt-ES [11]. Compared the test results of the independent test set PDB180 with
499 competitive DNAbinder [27], DNA-Prot [81], iDNA-Prot [82] and Top-2-gram-SVM [36].

500 DBPPred [35] extracted features based on sequence information, solvent accessibility, secondary
501 structural information, and evolutionary information. RF was used to feature selection. Finally,
502 Gaussian Naïve Bayes (GNB) was used to predict DBPs. Top-2-gram-SVM [36] combined
503 PseAAC and top-n-grams to extract evolutionary information and physicochemical properties.
504 Finally, the classifier SVM was used to predict DBPs. DNA-Prot [81] extracted the
505 physicochemical properties and secondary structural information of protein sequences and used
506 RF to predict DBPs. iDNA-Prot [82] was proposed by Lin et al., using grey system theory to
507 improve PseAAC and choosing RF for DBPs prediction. The comparison results are shown in
508 Table 6 and Table 7.

509 **Table 6**

510 Comparison of the independent test dataset PDB186 with other state-of-art methods under the
511 verification of the LOOCV method.

Methods	ACC (%)	SN (%)	SP (%)	MCC
HMMBinder [79]	69.02	61.53	76.34	0.3900
DBPPred [35]	76.90	79.60	74.20	0.5380
Local-DPP [28]	79.00	92.50	65.60	0.6250
PSSM-DT [33]	80.00	87.09	72.83	0.6470
MSFBinder [30]	80.11	92.47	67.74	0.6200
HMMPred [13]	81.18	94.62	67.74	0.6480
iDNAProt-ES [11]	80.64	81.31	80.00	0.6100
StackPDB	84.40	83.87	84.95	0.6882

512 **Table 7**

513 Comparison of the independent test dataset PDB180 with other state-of-art methods under the
514 verification of the LOOCV method.

Methods	ACC (%)	SN (%)	SP (%)	MCC
DNAbinder [27]	78.89	54.32	98.98	0.6100
DNA-Prot [81]	76.67	66.67	84.85	0.5300
iDNA-Prot [82]	81.11	72.84	87.88	0.6200
Top-2-gram-SVM [36]	85.56	82.72	87.88	0.7100
StackPDB	90.00	91.36	88.89	0.7997

515 In Table 6, the ACC value of StackPDB on PDB186 exceeds other prediction methods. The
516 ACC of StackPDB is 84.40%, which is 3.22%-15.38% higher than the ACC of HMMBinder,
517 DBPPred, Local-DPP, PSSM-DT, MSFBinder, HMMPred, and iDNAProt-ES (84.40 vs. 69.02,
518 76.90, 79.00, 80.00, 80.11, 81.18, 80.64). From the perspective of model stability, the MCC of
519 StackPDB is 0.6882, which is 29.82%-4.02% higher than the MCC of HMMBinder, DBPPred,
520 Local-DPP, PSSM-DT, MSFBinder, HMMPred, and iDNAProt-ES (0.6882 vs. 0.39, 0.5380,
521 0.6250, 0.647, 0.62, 0.648, 0.61). It can be seen that the StackPDB model also has high stability.
522 As we can see from Table 7, the prediction results of the StackPDB are better than other methods.

523 The ACC value of the StackPDB model reached 90.00%, which is 11.11%, 13.33%, 8.89% and
524 4.44% higher than DNAbinder, DNA-Prot, iDNA-Prot and Top-2-gram-SVM respectively. The
525 MCC value reaches 0.7997, which is 18.97%, 26.97%, 17.97% and 8.97% higher than DNAbinder,
526 DNA-Prot, iDNA-Prot and Top-2-gram-SVM respectively. Supplementary Figure S4 and Figure
527 S5 shows the histograms of the independent test datasets PDB186 and PDB180 compared with
528 other DBPs prediction methods. The performance of StackPDB on the independent test datasets
529 PDB186 and PDB180 show that the StackPDB model not only has the high predictive ability but
530 also shows great potential in the generalization ability and stability. Hence, StackPDB is a
531 competitive predictor of DBPs.

532 **4. Conclusion**

533 DBPs not only play a significant role in human life activities but also guide the development
534 of disease treatment and drug research and development. With the rapid growth of DBPs, the
535 development of DBPs prediction models has become a central issue in bioinformatics. We propose
536 a new method, called StackPDB. First, five feature extraction methods extract the information,
537 where PsePSSM, EDT, RPT, and PSSM-TPC extract evolutionary information. Especially,
538 PSSM-TPC extracts the evolutionary information. PseAAC can effectively obtain the
539 physicochemical properties information. Fusion of five features can obtain different aspects of
540 protein sequence information. Second, we use XGB-RFE to decrease the feature dimension.
541 XGB-RFE combines the gradient boosting and recursive feature elimination, which can fully learn
542 the importance score of each feature. It can also eliminate redundant and irrelevant features
543 without losing important features and reduce the complexity of the model. The final predictor of
544 DBPs is stacked ensemble classifier which composed of XGBoost, LightGBM and SVM. Stacked
545 ensemble classifier can take advantage of multiple classifiers, reduce generalization errors, and
546 have stronger predictive ability than ordinary machine learning classifiers. StackPDB has achieved
547 good prediction results on the training dataset PDB1075 based on LOOCV. Compared with other
548 state-of-art methods, StackPDB shows strong predictive ability on the independent test set
549 PDB186 and PDB180. In future work, deep learning methods are considered to predict DNA
550 binding proteins. Deep learning has powerful fitting capabilities and can approximate any complex
551 function. In particular, it has a great advantage in processing data with a large sample size, which
552 can make better accuracy of DBPs prediction.

553 **Declaration of competing interest**

554 No author associated with this paper has disclosed any potential or pertinent conflicts which
555 may be perceived to have impending conflict with this work.

556 **Acknowledgments**

557 This work was supported by the National Nature Science Foundation of China (No.
558 61863010), the Key Research and Development Program of Shandong Province of China (No.
559 2019GGX101001), and the Natural Science Foundation of Shandong Province of China (No.
560 ZR2018MC007, ZR2019MEE066).

561 **References**

- 562 [1] Y.F. Dai, C.L. Wang, L.Y. Chiu, K. Abbasi, B.S. Tolbert, G. Sauv é Y. Fen, C.C. Liu, Application of
563 bioconjugation chemistry on biosensor fabrication for detection of TAR-DNA binding Protein 43,
564 Biosens. Bioelectron. 117 (2018) 60-67.
- 565 [2] B. Ren, F. Robert, J.J. Wyrick, O. Aparici, E.G. Jennings, I. Simon, J. Zeitlinger, J. Schreiber, N.
566 Hannett, E. Kanin, T.L. Volkert, C.J. Wilson, S.P. Bell, R.A. Young, Genome-Wide location and
567 function of DNA binding proteins, Science 290 (2000) 2306-2309.
- 568 [3] R. Sabarinathan, L. Mularoni, J.D. Pons, A.G. Perez, N.L. Bigas, Nucleotide excision repair is
569 impaired by binding of transcription factors to DNA, Nature 532 (2016) 264-267.
- 570 [4] R. Helwa, J.D. Hoheisel, Analysis of DNA-protein interactions: from nitrocellulose filter binding
571 assays to microarray studies, Anal. Bioanal. Chem. 398 (2010) 2551-2561.
- 572 [5] K. Freeman, M. Gwadz, D. Shore, Molecular and genetic analysis of the toxic effect of RAP1 over
573 expression in yeast, Genetics 141 (1995) 1253-1262.
- 574 [6] M.J. Buck, J.D. Lieb, ChIP-chip: considerations for the design, analysis, and application of
575 genome-wide chromatin immunoprecipitation experiments, Genomics 83 (2004) 349-360.
- 576 [7] C.C. Chou, T.W. Lin, C.Y. Chen, A.H.J. Wang, Crystal structure of the hyperthermophilic archaeal
577 DNA-binding protein Sso10b2 at a resolution of 1.85 angstroms, J. Bacteriol. 185 (2003)
578 4066-4073.
- 579 [8] S. Ambardar, R. Gupta, D. Trakroo, R. Lal, J. Vakhlu, High throughput sequencing: an overview of
580 sequencing chemistry, Indian. J. Microbiol. 56 (2016) 394-404.
- 581 [9] M.S. Rahman, S. Shatabda, S. Saha, M. Kaykobad, M.S. Rahman, DPP-PseAAC: a DNA-binding

- 582 protein prediction model using Chou's general PseAAC, *J. Theor. Biol.* 452 (2018) 22-34.
- 583 [10] J. Zhang, B. Gao, H.T. Chai, Z.Q. Ma, G.F. Yang, Identification of DNA-binding proteins
584 using multi-features fusion and binary firefly optimization algorithm, *BMC Bioinf.* 17 (2016)
585 323.
- 586 [11] S.Y. Chowdhury, S. Shatabda, A. Dehzangi, iDNAProt-ES: Identification of DNA-binding
587 proteins using evolutionary and structural features, *Sci. Rep.* 7 (2017) 14938.
- 588 [12] L. Nanni, S. Brahnam, Set of approaches based on 3D structure and position specificscoring
589 matrix for predicting DNA-binding proteins, *Bioinformatics* 35 (2019) 1844-1851.
- 590 [13] X.Z. Sang, W.Y. Xiao, H.W. Zheng, Y. Yang, T.G. Liu, HMMPred: accurate prediction of
591 DNA-binding proteins based on HMM profiles and XGBoost feature selection, *Comput.*
592 *Math. Method. M.* 2020 (2020) 1-10.
- 593 [14] J. Hu, X.G. Zhou, Y.H. Zhu, D.J. Yu, G.J. Zhang, TargetDBP_accurate DNA-binding protein
594 prediction via sequence-based multi-view feature learning, *IEEE ACM. T. Comput. Bi.* 17
595 (2019) 1419-1429.
- 596 [15] K. Yan, D. Zhang, Feature selection and analysis on correlatedgas sensor data with recursive
597 feature elimination, *Sensor. Actuat. B-Chem.* 212 (2015) 353-363.
- 598 [16] L.L. Zhou, X.N. Song, D.J. Yu, J. Sun, Sequence-based detection of DNA-binding proteins
599 using multiple-view features allied with feature selection, *Mol. Inform.* 39 (2020)
600 doi:10.1002/minf.202000006.
- 601 [17] F. Ali, M. Kabir, M. Arif, Z.N.K. Swati, Z.U. Khan, M. Ullah, D.J. Yu, DBPPred-PDSD:
602 machine learning approach for prediction of DNA-binding proteins using discrete wavelet
603 transform and optimized integrated features space, *Chemometr. Intell. Lab. Syst.* 182 (2018)
604 21-30.
- 605 [18] G.L. Ji, Y. Lin, Q.M. Lin, G.Z. Huang, W.B. Zhu, W.J. You, Predicting DNA-binding
606 proteins using feature fusion and MSVM-RFE, *International Conference on*
607 *Anti-counterfeiting, Security, and Identification (ASID)*, 2016, pp. 109-112.
- 608 [19] X.N. Bui, P. Jaroonpattanapong, H. Nguyen, Q. Tran, N.Q. Long, A novel hybrid model for
609 predicting blast-induced ground vibration based on k-nearest neighbors and particle swarm
610 optimization, *Sci. Rep.* 9 (2019) 1-14.

- 611 [20] C.X. Ma, W. Hao, F.Q. Pan, W. Xiang, Road screening and distribution route multi-objective
612 robust optimization for hazardous materials based on neural network and genetic algorithm,
613 PLoS One, 13 (2018) e0198931.
- 614 [21] L.Y. Bai, H. Dai, Q. Xu, M. Junaid, S.L. Peng, X.L. Zhu, Y. Xiong, D.Q. Wei, Prediction of
615 effective drug combinations by an improved Naïve Bayesian algorithm, Int. J. Mol. Sci. 19
616 (2018) 467.
- 617 [22] I.A. Tamposis, K.D. Tsigros, M.C. Theodoropoulou, P.I. Kontou, P.G. Bagos,
618 Semi-supervised learning of hidden markov models for biological sequence analysis,
619 Bioinformatics 35 (2019) 2208-2215.
- 620 [23] C. Zhou, H. Yu, Y.J. Ding, F. Guo, X.J. Gong, Multi-scale encoding of amino acid sequences
621 for predicting protein interactions using gradient boosting decision tree, PLoS One, 12 (2017)
622 e0181426.
- 623 [24] B. Manavalan, and J. Lee, SVMQA: support-vector-machine-based protein single-model
624 quality assessment, Bioinformatics 33 (2017) 2496-2503.
- 625 [25] G. Taherzadeh, Y.Q. Zhou, A.W. Liew, Y.D. Yang, Structure-based prediction of
626 protein-peptide binding regions using random forest, Bioinformatics 34 (2017) 477-484.
- 627 [26] F. Ali, S. Ahmed, Z.N.K. Swati, S. Akbar, DP-BINDER: machine learning model for
628 prediction of DNA-binding proteins by fusing evolutionary and physicochemical information,
629 J. Comput. Aid. Mol. Des. 33 (2019) 645-658.
- 630 [27] M. Kumar, M.M. Gromiha, G.P.S. Raghava, Identification of DNA-binding proteins using
631 support vector machines and evolutionary profiles, BMC Bioinf. 8 (2007) 463.
- 632 [28] L.Y. Wei, J.J. Tang, Q. Zou, Local-DPP: An improved DNA-binding protein prediction
633 method by exploring local evolutionary information, Inform. Sciences. 384 (2017) 135-144.
- 634 [29] S. Chauhan, S. Ahmad, Enabling full-length evolutionary profiles based deep convolutional
635 neural network for predicting DNA-binding proteins from sequence, Proteins, 88 (2020)
636 15-30.
- 637 [30] X.J. Liu, X.J. Gong, H. Yu, J.H. Xu, A model stacking framework for identifying DNA
638 binding proteins by orchestrating multi-view features and classifiers, Genes 9 (2018) 394.
- 639 [31] R.F. Xu, J.Y. Zhou, B. Liu, L. Yao, Y.L. He, Q. Zou, X.L. Wang, enDNA-Prot: identification

- 640 of DNA-binding proteins by applying ensemble learning, *Biomed Res. Int.* 2014 (2014)
641 294279.
- 642 [32] B. Liu, S.Y. Wang, Q.W. Dong, S.M. Li, X. Liu, Identification of DNA-binding proteins by
643 combining auto-cross covariance transformation and ensemble learning, *IEEE. T. Nanobiosci.*
644 15 (2016) 328-334.
- 645 [33] R.F. Xu, J.Y. Zhou, H.P. Wang, Y.L. He, X.L. Wang, B. Liu, Identifying DNA-binding
646 proteins by combining support vector machine and PSSM distance transformation, *BMC Syst.*
647 *Biol.* 9 (2015) S10.
- 648 [34] S.F. Altschul, T.L. Madden, A.A. Schaffer, J.H. Zhang, Z. Zhang, W. Miller, D.J. Lipman,
649 Gapped BLAST and PSI-BLAST: a new generation of protein database search programs,
650 *Nucleic Acids Res.* 25 (1997) 3389-3402.
- 651 [35] W.C. Lou, X.Q. Wang, F. Chen, Y.X. Chen, B. Jiang, H. Zhang, Sequence based prediction
652 of DNA-binding proteins based on hybrid feature selection using random forest and Gaussian
653 Naïve Bayes, *PLoS One*, 9 (2014) e86703.
- 654 [36] R.F. Xu, J.Y. Zhou, B. Liu, Y.L. He, Q. Zou, X.L. Wang, K.C. Chou, Identification of
655 DNA-binding proteins by incorporating evolutionary information into pseudo amino acid
656 composition via the top-n-gram approach, *J. Biomol. Struct. Dyn.* 33 (2015) 1720-1730.
- 657 [37] K.C. Chou, Some remarks on protein attribute prediction and pseudo amino acid composition,
658 *J. Theor. Biol.* 273 (2011) 236-247.
- 659 [38] B. Yu, S. Li, W.Y. Qiu, M.H. Wang, J.W. Du, Y.S. Zhang, X. Chen, Prediction of subcellular
660 location of apoptosis proteins by incorporating PsePSSM and DCCA coefficient based on
661 LFDA dimensionality reduction, *BMC Genomics* 19 (2018) 478.
- 662 [39] E. Contrerasstorres, Predicting structural classes of proteins by incorporating their global and
663 local physicochemical and conformational properties into general Chou's PseAAC, *J. Theor.*
664 *Biol.* 454 (2018) 139-145.
- 665 [40] W. Hussain, Y.D. Khan, N. Rasool, S.A. Khan, K.C. Chou, SPrenylC-PseAAC: a
666 sequence-based model developed via Chou's 5-steps rule and general PseAAC for identifying
667 S-prenylation sites in proteins, *J. Theor. Biol.* 468 (2019) 1-11.

- 668 [41] B. Yu, W.Y. Qiu, C. Chen, A.J. Ma, J. Jiang, H.Y. Zhou, Q. Ma, SubMito-XGBoost:
669 predicting protein submitochondrial localization by fusing multiple feature information and
670 eXtreme gradient boosting, *Bioinformatics* 36 (2020) 1074-1081.
- 671 [42] S. Hashemifar, B. Neyshabur, A.A. Khan, J.B. Xu, Predicting protein-protein interactions
672 through sequence-based deep learning, *Bioinformatics* 34 (2018) i802-i810.
- 673 [43] H.C. Yi, Z.H. You, D.S. Huang, X. Li, T.H. Jiang, L.P. Li, A deep learning framework for
674 robust and accurate prediction of ncRNA-protein interactions using evolutionary information,
675 *Mol. Ther-Nucl. Acids*.11 (2018) 337-344.
- 676 [44] M. Waris, K. Ahmad, M. Kabir, M. Hayat, Identification of DNA binding proteins using
677 evolutionary profiles position specific scoring matrix, *Neurocomputing* 199 (2016) 154-162.
- 678 [45] D.T. Jones, Protein secondary structure prediction based on position-specific scoring matrices,
679 *J. Mol. Biol.* 292 (1999) 195-202.
- 680 [46] H.B. Shen, K.C. Zhou, Nuc-PLoc: a new web-server for predicting protein subnuclear
681 localization by fusing PseAA composition and PsePSSM, *Protein Eng. Des. Sel.* 20 (2007)
682 561-567.
- 683 [47] Y.J. Ding, J.J. Tang, F. Guo, Human protein subcellular localization identification via fuzzy
684 model on kernelized neighborhood representation, *Appl. Soft Comput.* 96 (2020) 106596.
- 685 [48] W.Y. Qiu, S. Li, X.W. Cui, Z.M. Yu, M.H. Wang, J.W. Du, Y.J. Peng, B. Yu, Predicting
686 protein submitochondrial locations by incorporating the pseudo-position specific scoring
687 matrix into the general Chou's pseudo-amino acid composition, *J. Theor. Biol.* 450 (2018)
688 86-103.
- 689 [49] H. Shi, S.M. Liu, J.Q. Chen, X. Li, Q. Ma, B. Yu, Predicting drug-target interactions using
690 Lasso with random forest based on evolutionary information and chemical structure,
691 *Genomics* 111 (2019) 1839-1852.
- 692 [50] H. Wang, Y.J. Ding, J.J. Tang, F. Guo, Identification of membrane protein types via
693 multivariate information fusion with Hilber-Schmidt independence criterion,
694 *Neurocomputing* 383 (2020) 257-269.

- 695 [51] S.L. Zhang, F. Ye, X.G. Yuan, Using principal component analysis and support vector
696 machine to predict protein structural class for low-similarity sequences via PSSM, *J. Biomol.*
697 *Struct. Dyn.* 29 (2012) 634-642.
- 698 [52] L.C. Zhang, X.Q. Zhao, L. Kong, Predict protein structural class for low-similarity sequences
699 by evolutionary difference information into the general form of Chou's pseudo amino acid
700 composition, *J. Theor. Biol.* 355 (2014) 105-110.
- 701 [53] J.C. Jeong, X.T. Lin, X.W. Chen, On position-specific scoring matrix for protein function
702 prediction, *IEEE/ACM Trans. Comput. Biol. Bioinform.* 8 (2011) 308-315.
- 703 [54] J.L. Yu, S.P. Shi, F. Zhang, G.D. Chen, M. Cao, PredGly: predicting lysine glycation sites for
704 *Homo sapiens* based on XGboost feature optimization, *Bioinformatics* 35 (2018) 2749-2756.
- 705 [55] X.Z. Fu, W. Zhu, B. Liao, L.J. Cai, L.H. Peng, J.L. Yang, Improved DNA-binding protein
706 identification by incorporating evolutionary information into the Chou's PseAAC, *IEEE*
707 *Access* 6 (2018) 66545-66556.
- 708 [56] D.H. Wolpert, Stacked generalization, *Neural networks* 5 (1992) 241-259.
- 709 [57] H.C. Yi, Z.H. You, M.N. Wang, Z.H. Guo, Y.B. Wang, J.R. Zhou, RPI-SE: a stacking
710 ensemble learning framework for ncRNA-protein interactions prediction using sequence
711 information, *BMC Bioinf.* 21 (2020) 60.
- 712 [58] Y. Xiong, Q.K. Wang, J.C. Yang, X.L. Zhu, D.Q. Wei, PredT4SE-stack: prediction of
713 bacterial type IV secreted effectors from protein sequences using a stacked ensemble method,
714 *Front. Microbiol.* 9 (2018) 2571.
- 715 [59] R. Su, X.Y. Liu, G.B. Xiao, L.Y. Wei, Meta-GDBP: a high-level stacked regression model to
716 improve anticancer drug response prediction, *Brief. Bioinform.* 21 (2020) 996-1005.
- 717 [60] S. Saha, S. Mitra, R.K. Yadav, A stack-based ensemble framework for detecting cancer
718 microRNA biomarkers, *Genom. Proteom. Bioinf.* 15 (2017) 381-388.
- 719 [61] N.S. Altman, An introduction to kernel and nearest neighbor nonparametric regression, *Am.*
720 *Stat.* 46 (1992) 175-185.
- 721 [62] V.N. Vapnik, An overview of statistical learning theory, *IEEE Trans. Neural Netw.* 10 (1999)
722 988-999.

- 723 [63] J.Y. Zhou, Q. Lu, R.F. Xu, Y.L. He, H.P. Wang, EL_PSSM-RT: DNA-binding residue
724 prediction by integrating ensemble learning with PSSM relation transformation, *BMC Bioinf.*
725 18 (2017) 379.
- 726 [64] Y. Shi, J. Li, Z.Z. Li, Gradient boosting with piece-wise linear regression trees, *Proceedings*
727 *of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2018, pp.
728 3432-3438.
- 729 [65] X.T. Lin, X.W. Chen, Heterogeneous data integration by tree-augmented Naïve Bayes
730 for protein-protein interactions prediction, *Proteomics* 13 (2013) 261-268.
- 731 [66] S.B. Wan, M.W. Mak, S.Y. Kung, mPLR-Loc: An adaptive decision multi-label classifier
732 based on penalized logistic regression for protein subcellular localization prediction, *Anal.*
733 *Biochem.* 473 (2015) 14-27.
- 734 [67] C. Chen, Q.M. Zhang, Q. Ma, B. Yu, LightGBM-PPI: predicting protein-protein interactions
735 through LightGBM with multi-information fusion, *Chemometr. Intell. Lab. 191* (2019)
736 54-64.
- 737 [68] A. Peerlinck, J.W. Sheppard, J. Senecal, AdaBoost with neural networks for yield and protein
738 prediction in precision agriculture, 2019 International Joint Conference on Neural Networks
739 (IJCNN), 2019, doi:10.1109/ijcnn.2019.8851976.
- 740 [69] S. Sartipi, H. Kalbkhani, P. Ghasemzadeh, M.G. Shayesteh, Stockwell transform of
741 time-series of fMRI data for diagnoses of attention deficit hyperactive disorder, *Appl. Soft*
742 *Comput.* 86 (2020) 105905.
- 743 [70] X.Y. Wang, B. Yu, A.J. Ma, C. Chen, B. Liu, Q. Ma, Protein-protein interaction
744 sites prediction by ensemble random forests with synthetic minority oversampling technique,
745 *Bioinformatics* 35 (2019) 2395-2402.
- 746 [71] F.L. Luo, M.H. Wang, Y. Liu, X.M. Zhao, A. Li, DeepPhos: prediction of protein
747 phosphorylation sites with deep learning, *Bioinformatics*, 35 (2019) 2766-2773.
- 748 [72] C.Z. Kang, Y.H. Huo, L.H. Xin, B.G. Tian, B. Yu, Feature selection and tumor classification
749 for microarray data using relaxed Lasso and generalized multi-class support vector machine,
750 *J. Theor. Biol.* 463 (2019) 77-91.

- 751 [73] K. Zheng, Z.H. You, J.Q. Li, L. Wang, Z.H. Guo, Y.A. Huang, iCDA-CGR: identification of
752 circRNA-disease associations based on chaos game representation, *PLoS Comput. Biol.* 16
753 (2020) e1007872.
- 754 [74] X.M. Sun, T.Y. Jin, C. Chen, X.W. Cui, Q. Ma, B. Yu, RBPro-RF: use Chou's 5-steps rule to
755 predict RNA-binding proteins via random forest with elastic net, *Chemometr. Intell. Lab.*
756 *Syst.* 197 (2020) 103919.
- 757 [75] R.F. Fan, K.W. Chang, C.J. Hsieh, X.R. Wang, C. Lin, LIBLINEAR: alibrary for large linear
758 classification, *J. Mach. Learn. Res.* 9 (2008) 1871-1874.
- 759 [76] Y.W. Zhang, Y.J. Fu, Z.B. Wang, L. Feng, Fault detection based on modified kernel
760 semi-supervised socally sinear smbedding, *IEEE Access* 6 (2017) 479-487.
- 761 [77] A. Franceschini, J.Y. Lin, C.V. Mering, L.J. Jensen, SVD-phy: improved prediction of
762 protein functional associations through singular value decomposition of phylogenetic profiles,
763 *Bioinformatics* 32 (2015) 1085-1087.
- 764 [78] L.V.D. Maaten, G.E. Hinton, Visualizing data using t-SNE, *J. Mach. Learn. Res.* 9 (2008)
765 2579-2605.
- 766 [79] R. Zaman, S.Y. Chowdhury, M.A. Rashid, A. Sharma, A. Dehzangi, S. Shatabda,
767 HMMBinder: DNA-binding protein prediction using HMM Profile based features, *Biomed.*
768 *Res. Int.* 2017 (2017) 4590609.
- 769 [80] B. Liu, S.Y. Wang, X.L. Wang, DNA binding protein identification by combining pseudo
770 amino acid composition and profile-based protein representation, *Sci. Rep.* 5 (2015) 15479.
- 771 [81] K.K. Kumar, G. Pugalenth, P.N. Suganthan, DNA-Prot: Identification of DNA binding
772 proteins from protein sequence information using random forest, *J. Biomol. Struct. Dyn.* 26
773 (2009) 679-686.
- 774 [82] W.Z. Lin, J.A. Fang, X. Xiao, K.C. Chou, iDNA-Prot: identification of DNA binding proteins
775 using random forest with grey model, *PLoS One*, 6 (2011) e24756.