Computational Biology and Chemistry 91 (2021) 107456

Contents lists available at ScienceDirect

Computational Biology and Chemistry

journal homepage: www.elsevier.com/locate/cbac

Research Article

Structural protein fold recognition based on secondary structure and evolutionary information using machine learning algorithms

Xinyi Qin [1,2], Min Liu [1,3], Lu Zhang [1,2], Guangzhong Liu [1,3,*]

College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China

ARTICLE INFO

Keywords:
Protein fold recognition
ASTRAL
Secondary structure
Evolutionary information
Feature selection algorithm
IFS

ABSTRACT

Understanding the function of protein is conducive to research in advanced fields such as gene therapy of diseases, the development and design of new drugs, etc. The prerequisite for understanding the function of a protein is to determine its tertiary structure. The realization of protein structure classification is indispensable for this problem and fold recognition is a commonly used method of protein structure classification. Protein sequences of 40% identity in the ASTRAL protein classification database are used for fold recognition research in current work to predict 27 folding types which mostly belong to four protein structural classes: α, β, α/β and $\alpha+\beta$. We extract features from primary structure of protein using methods covering DSSP, PSSM and HMM which are based on secondary structure and evolutionary information to convert protein sequences into feature vectors that can be recognized by machine learning algorithm and utilize the combination of LightGBM feature selection algorithm and incremental feature selection method (IFS) to find the optimal classifiers respectively constructed by machine learning algorithms on the basis of tree structure including Random Forest, XGBoost and LightGBM. Bayesian optimization method is used for hyper-parameter adjustment of machine learning algorithms to make the accuracy of fold recognition reach as high as 93.45% at last. The result obtained by the model we propose is outstanding in the study of protein fold recognition.

1. Introduction

A protein is a polymer formed by linking 20 different amino acids. The amino acids that make up the protein are also called residues. Protein structure refers to the spatial structure of protein molecules composing of chemical elements such as carbon (C), hydrogen (H), oxygen (O) and nitrogen (N). Protein structure can be divided into four levels. Protein primary structure refers to amino acid sequence; protein secondary structure is periodic structural objects; protein tertiary structure is the three-dimensional structure of entire polypeptide chain; protein quaternary structure is a complex formed by several protein molecules. Proteins can only perform their functions when they are correctly folded into a specific structure type through non-covalent interactions including hydrogen bonds, van der Waals forces, hydrophobic interactions, etc. (Sela et al., 1957). The premise of understanding the function of a certain protein is to determine its tertiary structure. Researchers usually use X-ray (Bragg, 1976; Stuart, 1976), nuclear magnetic resonance (Baldwin et al., 1991) and other techniques to determine the structure of protein, but these methods are relatively slow and inefficient. Nowadays, the number of proteins is increasing faster and faster and it is far greater than the update speed of some authoritative protein structure classification databases such as SCOP (Murzin et al., 1995; Andreeva et al., 2020), SCOPe (Chandonia et al., 2019), ASTRAL (Chandonia et al., 2004), etc. Therefore, it is quite meaningful to study a method with more comprehensive coverage and higher prediction accuracy to realize protein structure classification.

Since the research of protein structure classification provides relevant information about protein folding, protein–protein interactions and protein functions, researchers have been actively discussing and studying this problem in recent decades (Cohen and Kuntz, 1987; Liang et al.,

* Corresponding author.
E-mail addresses: 201930310113@stu.shmtu.edu.cn (X. Qin), liumin@shmtu.edu.cn (M. Liu), 201940310007@stu.shmtu.edu.cn (L. Zhang), gzhliu@shmtu.edu.cn (G. Liu).
[1] Principal corresponding author.
[2] This is the first author footnote, but is common to third author as well.
[3] Another author footnote, this is a very long footnote and it should be a really long footnote. But this footnote is not yet sufficiently long enough to make two lines of footnote text.

https://doi.org/10.1016/j.compbiolchem.2021.107456
Received 23 August 2020; Received in revised form 4 January 2021; Accepted 6 February 2021
Available online 12 February 2021
1476-9271/

2015). Researchers usually bring about the classification of protein structure via fold recognition. In recent years, many protein structure classification methods have been proposed and used for protein fold recognition (Wei et al., 2015; Wei and Zou, 2016; Feng et al., 2016; Chen et al., 2016; Ibrahim and Abadeh, 2018; Yan et al., 2019, 2020). Actually, the study of protein structure classification is mainly divided into two aspects: one is from the perspective of feature extraction methods; the other is from the perspective of machine learning algorithms.

Nowadays, most of the datasets coming from some classic protein structure databases such as SCOP (Murzin et al., 1995; Andreeva et al., 2020), ASTRAL (Chandonia et al., 2004), CATH (Orengo et al., 1997), PDB (Berman et al., 2000), etc. have been used to research the problem of protein fold recognition. For a given dataset, we first need to extract features of the protein, whose purpose is to translate the primary structure of protein, namely, amino acid sequence into feature vector. The main methods include sequence information-based methods such as Amino Acid Composition (AAC) (Dubchak et al., 1995; Lin et al., 2013); methods based on physical and chemical properties such as Pseudo Amino Acid Composition (PseAAC) (Chou, 2010); methods based on secondary structure such as DSSP (Chen et al., 2016), ASA (Heffernan et al., 2015); methods on the basis of evolutionary information, for instance, PSSM (Paliwal et al., 2014), HMM (Lyons et al., 2016). In the field of Bioinformatics, using machine learning algorithm to build the classifier to do model training is a classic and effective method to realize fold recognition. In the past few decades, many researchers have used various machine learning algorithms to classify protein structures involving Random Forest (Mehta and Himanshu, 2019), Support Vector Machines (Ding and Dubchak, 2001), k-nearest neighbors (Kavousi et al., 2011), neural network (Ibrahim and Abadeh, 2018) and ensemble learning (Chen et al., 2016), etc.

The target of current work is to classify 27 folding types of the ASTRAL dataset selected from the ASTRAL protein classification database. Most of these 27 folding types belong to four structural classes covering α, β, α/β and $\alpha+\beta$. As far as protein feature extraction is concerned, method based on protein secondary structure, namely, DSSP and methods based on protein evolutionary information, that is, PSSM and HMM are adopted. These methods are combined to generate the final 490-dimensional feature vector which is going to be respectively input into the classifiers constructed by machine learning algorithms for model training. In terms of machine learning algorithms, tree structure-based algorithms including Random Forest, XGBoost and LightGBM are selected. In addition, LightGBM feature selection algorithm is supposed to be used to reduce the dimensionality of the high-dimensional feature vector and incremental feature selection method (IFS) requires to be combined with it to get optimal classification effect after converting protein amino acids sequences into feature vectors using the feature extraction methods. The specific process of the entire fold recognition experiment on the ASTRAL dataset is shown in Fig. 1.

2. Materials and methods

2.1. Dataset

The ASTRAL protein classification database (Chandonia et al., 2004) is a relatively classic protein structure classification database. It is mainly merged and updated by the SCOPe database (Chandonia et al., 2019) which is an expanded version of the SCOP database (Murzin et al., 1995; Andreeva et al., 2020) and belongs to the latest and largest protein database at this stage. The ASTRAL database helps to investigate the relationship between protein structure and its evolutionary process and also provides many tools to help study and analyze the structure of proteins. It is achieved by a subset of proteins that span the classification structure or domains of proteins, which solves the problem of protein bias in nature due to the great similarity between most of the protein sequences in the protein database and other protein sequences. We use ASTRAL protein sequences of 40% identity for SCOPe 2.07 in this study

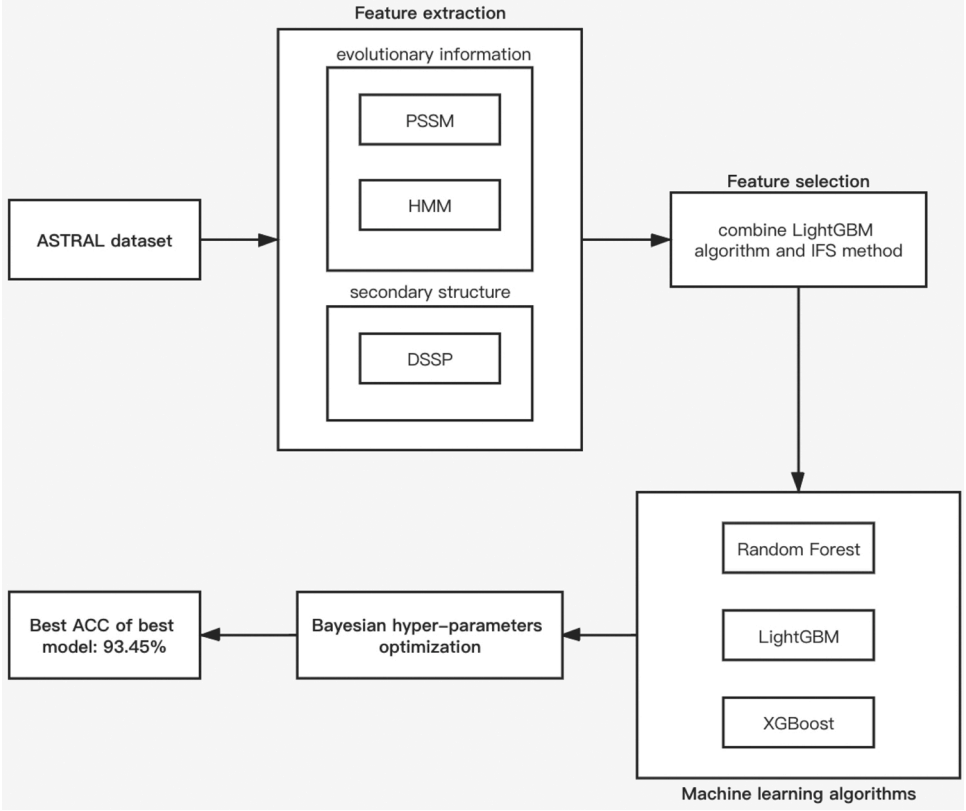

Fig. 1. Flowchart of protein fold recognition on the ASTRAL dataset in this study.

and select samples with the same 27 types of folds which mainly covers four structural classes of α, β, α/β and $\alpha+\beta$ as the DD dataset (Ding and Dubchak, 2001). The selected dataset contains a total of 4778 proteins. The details of protein sequences available for 27 protein folding types in the ASTRAL dataset are given in Table 1.

2.2. Feature extraction

2.2.1. Position specific scoring matrix (PSSM)

Altschul et al. proposed an analysis tool BLAST for aligning similar sequences in protein databases and PSI-BLAST which can be used to align protein sequences (Altschul et al., 1997). BLAST compares similar sequences in protein databases and outputs protein sequence similarity scores. PSI-BLAST can output two PSSM matrices, respectively the conservative value scoring matrix and the probability information matrix, which respectively represent the scoring and probability of an amino acid in a protein sequence being replaced by another amino acid. Each protein sequence must be aligned with the homologous sequence in the selected BLAST local protein database by means of PSI-BLAST tool. For a protein sequence P of length L, the two PSSM matrices obtained are both L rows and 20 columns. The L rows represent the positions of L amino acids in the protein sequence and the ith row represents the ith position in the protein sequence. The 20 columns represent the 20 amino acids composing of the protein and the jth column represents amino acid j.

In order to obtain the PSSM matrix, it is firstly necessary to establish a local protein database for homology comparison of each protein sequence. The SWISS-PROT database (Renaux, 2018) is selected as the BLAST local protein database. The next step needs to utilize PSI-BLAST tool to compare each protein sequence in the ASTRAL dataset with the SWISS-PROT database to obtain the corresponding PSSM matrix. The PSSM feature extraction method used in this study mainly does processing on the conservative value scoring matrix of the first 20 columns. Calculate the average value of each column and finally achieve a 20-dimensional feature vector. Assuming that the element in the matrix is represented by E_{ij} $(1 \le i \le L,\ 1 \le j \le 20)$, PSSM 20-dimensional feature vector are expressed as Eq. (1), where $\frac{\sum_{i=1}^{L} E_{ij}}{L}$ $(1 \le j \le 20)$ represents the average value of each column of the conservative value scoring matrix.

$$F_{\mathrm{PSSM}} = [\frac{\sum_{i=1}^{L} E_{i1}}{L}, \frac{\sum_{i=1}^{L} E_{i2}}{L}, \ldots, \frac{\sum_{i=1}^{L} E_{i20}}{L}] \quad (1)$$

Table 1
The details of protein sequences available for 27 protein folding types in the ASTRAL dataset.

Class	Fold index	Fold	Remark	Number
α	fold1	a.1	Globin-like	58
	fold2	a.3	Cytochrome c	42
	fold3	a.4	DNA/RNA-binding 3-helical bundle	410
	fold4	a.24	Four-helical up-and-down bundle	78
	fold5	a.26	4-Helical cytokines	33
	fold6	a.39	EF hand-like	84
β	fold7	b.1	Immunoglobulin-like beta-sandwich	546
	fold8	b.6	Cupredoxin-like	52
	fold9	b.29	Concanavalin A-like lectins/glucanases	89
	fold10	b.34	SH3-like barrel	173
	fold11	b.40	OB-fold	184
	fold12	b.42	beta-Trefoil	63
	fold13	b.47	Trypsin-like serine proteases	57
	fold14	b.60	Lipocalins	49
	fold15	b.121	Nucleoplasmin-like/VP (viral coat and capsid proteins)	69
α/β	fold16	c.1	TIM beta/alpha-barrel	499
	fold17	c.2	NAD(P)-binding Rossmann-fold domains	332
	fold18	c.3	FAD/NAD(P)-binding domain	74
	fold19	c.23	Flavodoxin-like	226
	fold20	c.37	P-loop containing nucleoside triphosphate hydrolases	328
	fold21	c.47	Hioredoxin fold	204
	fold22	c.55	Ribonuclease H-like motif	167
	fold23	c.69	alpha/beta-Hydrolases	139
	fold24	c.93	Periplasmic binding protein-like I	92
$\alpha+\beta$	fold25	d.15	beta-Grasp (ubiquitin-like)	148
	fold26	d.58	Ferredoxin-like	444
	fold27	g.3	Knottins (small inhibitors, toxins, lectins)	138

2.2.2. Definition of secondary structure of proteins (DSSP)

DSSP is a dictionary of protein secondary structure. It does not predict the secondary structure, but indicates which secondary structure is in each position of a protein whose tertiary structure has been determined according to the definition of the secondary structure. The DSSP program was designed and developed by Wolfgang Kabsch and Chris Sander (Kabsch and Sander, 1983; Touw et al., 2014). It is used to standardize protein secondary structure. It calculates the hydrogen bond energy between all atoms through the atomic position of the PDB file. The best hydrogen bond energy between two atoms determines the most similar secondary structure type of each residue in the protein. The PDB file can be translated into the DSSP file by the DSSP program and the DSSP sequence is able to be extracted from the DSSP file. An example of a DSSP sequence is shown in Fig. 2. There are a total of eight DSSP secondary structure types in the DSSP sequence including H, B, E, G, I, T, S and $-$. These eight secondary structure types can be divided into four groups. The introduction and grouping of the eight DSSP secondary structure types are shown in Table 2.

According to the distribution of eight DSSP secondary structure types (H, B, E, G, I, T, S and $-$) and four groups, 40-dimensional feature vector can be generated from the DSSP sequence to comprehensively show the secondary structure information of the protein, whose detailed description is shown in Table 3, where N is the length of the DSSP sequence, N_t $(1 \le t \le 8)$ is the number of secondary structures belonging to the same type, N_g $(1 \le g \le 4)$ is the number of secondary structures belonging to the same group, N_s is the number of strings of the DSSP sequence and adjacent and identical secondary structures form a string, N_{st} is the number of strings and the secondary structures that make up the strings belong to the same type, N_{sg} is the number of strings and the secondary structures that make up the strings belong to the same group, N_a $(1 \le a \le 4)$ is the number of alternate grouping which means the left and right elements of a secondary structure in a DSSP sequence belong to the same group.

2.2.3. Hidden Markov models (HMM)

Given a protein sequence, use the HHblits tool (Remmert et al., 2012) to extract the HMM profile. The HMM profile will generate an $L \times 30$ matrix, where L is the length of the protein sequence. Use the equation $p = 2^{-N/1000}$ to convert the value generated by HHblits in the HMM matrix into linear probability, where N is the probability number in the matrix. The first 20 columns of the matrix indicate the probabilities of amino acids in the protein sequence being replaced by 20 amino acids and the last 10 columns indicate the information of three states including insertion, deletion and match which are used to represent sequence changes.

Assuming the element in the HMM matrix is represented by h_{ij} $(1 \le i \le L,\ 1 \le j \le 30)$, three sets of features will be extracted for the HMM feature extraction method, which are as follows:

(2) HMM-monogram:

The global sum of the probabilities of amino acids in the protein sequence being replaced by other amino acids. We can calculate the sum of each column of the first 20 columns of the HMM matrix (Eq. (2)) to get this set of 20-dimensional feature vector.

```
-HHHHTTSHHHHHHHHHHHHHHHHT-TTTGGGGTT--HHHHHHHHHHHHHHHHTT-SS---S--HHHHHTTS---HHHHHH
HHHHHHHHHHHTT--HHHHHHHHHHHHTTHHHH---
```

Fig. 2. An example of a DSSP sequence composed of eight types of secondary structure including $H, B, E, G, I, T, S, -$.

Table 2
The introduction and grouping of eight DSSP secondary structure types.

Grouping	Types	Code of types	Description of types
alpha-helix (H)	3_10 helix (G)	G	Helix-3
	alpha-helix (H)	H	Alpha helix
	pi-helix (I)	I	Helix-5
beta-strand (E)	beta-strand (E)	E	Strand
	beta-bridge (B)	B	Beta bridge
coil region (C)	beta-turn (T)	T	Turn
	High curvature loop (S)	S	Bend
Irregular (L)	Irregular (L)	–	Empty, no secondary structure assigned

Table 3
The detailed description of 40-dimensional feature vector extracted by DSSP.

Index	Feature description	Equation	Dimension
1	Percentage of each type	$N_t \div N$	8
2	Percentage of each group	$N_g \div N$	4
3	Number of consecutive types	N_{st}	8
4	Number of consecutive groups	N_{sg}	4
5	Alternate grouping	$N_a \div (N-2)$	4
6	Percentage of consecutive types	$N_{st} \div N_s$	8
7	Percentage of consecutive groups	$N_{sg} \div N_s$	4
–	Total	–	40

$$F_{\mathrm{HMM1}} = \left[\sum_{i=1}^{L} h_{i1}, \sum_{i=1}^{L} h_{i2}, \ldots, \sum_{i=1}^{L} h_{i20}\right] \tag{2}$$

(3) HMM-bigram:

The interaction of adjacent amino acids in the protein sequence is interpreted as the possibility of two consecutive amino acids appearing in the protein sequence. The 400-dimensional feature vector can be obtained based on the first 20 columns of the HMM matrix using Eqs. (3) and (4), where $1 \le i \le 20$, $1 \le j \le 20$.

$$B(i,j) = \sum_{m=1}^{L-1} h_{\mathrm{mi}} h_{(m+1)j} \tag{3}$$

$$F_{\mathrm{HMM2}} = [B(1,1), B(1,2), \ldots, B(20,20)] \tag{4}$$

(4) HMM-state:

The state information is expressed as the synthesis of each column of the last 10 columns of the HMM matrix. We can obtain the last 10-dimensional feature vector by Eq. (5).

$$F_{\mathrm{HMM3}} = [\sum_{i=1}^{L} h_{i21}, \sum_{i=1}^{L} h_{i22}, \ldots, \sum_{i=1}^{L} h_{i30}] \tag{5}$$

2.3. Machine learning algorithms

Current work adopts ensemble learning (Chen et al., 2016) in machine learning algorithms to start the work. Ensemble learning is able to improve the generalization ability and robustness of the learner by combining the prediction results of multiple base learners. According to the generation methods of base learners, the current ensemble learning methods are roughly divided into two categories: serialization methods that have strong dependency between base learners must be generated serially; parallelization methods which have no strong dependency between base learners can be generated at the same time. The former is represented by Boosting including LightGBM and XGBoost used in this study, the representative of the latter is Bagging and Random Forest.

2.3.1. Random Forest algorithm

Random Forest is a classifier composed of multiple decision trees (Breiman, 2001) and its base learner is limited to decision tree. The classification decision tree model is a tree structure that classifies sample instances. Each decision tree consists of nodes and directed edges. The nodes are made up of internal nodes and leaf nodes. Internal node represents one feature or attribute, leaf node represents a class. Classification by decision tree is a process of testing a certain feature of an object from the root node and then assigning the object to its child node according to the result. In this process, each child node corresponds to a value of one feature. The classification decision tree model needs to recursively test and allocate the object until reaching the leaf node. At last, the object can be allocated to the class corresponding to the leaf node. The forest established by Random Forest consists of multiple decision trees and there is no correlation between decision trees. After the forest is established, each decision tree in the forest will classify the object separately when a new input object enters and the most selected category is the class to which the object belongs.

2.3.2. Light gradient boosting machine algorithm (LightGBM)

GBDT (Gradient Tree Boosting; Gradient Boosting Decision Tree) (Friedman, 2001) is an integrated model of decision trees and it is trained in sequence. For a general loss function, GBDT algorithm uses the value of the negative gradient of the loss function in the current model as an approximation of the residual and it has reached the most advanced performance in many machine learning tasks. However, with the emergence of big data, the implementation of GBDT will take longer and longer and it will gradually fail to meet the requirements of researchers. In order to solve the problem of long time-consuming implementation of GBDT algorithm for big data, Guolin Ke et al. developed a novel GBDT algorithm called LightGBM, which contains two novel techniques: Gradient-based One-Side Sampling and Exclusive Feature Bundling to deal with large number of data instances and large number of features respectively (Ke et al., 2017). LightGBM algorithm divides the training data into multiple models, performs local voting to determine the selection of the first k features, and then performs global voting to determine the selection of the first $2k$ features in each iteration. The default training decision tree in the LightGBM algorithm uses histogram algorithm whose principle is to improve training speed and save memory space at the expense of certain segmentation accuracy. Algorithms based on histogram algorithm do not need to find splits on the classified feature values, but store continuous feature values in discrete bins which are going to be used to construct feature histograms during the training process. This can reduce the number of data instances and the number of features, so that there is a shorter training time on big data. Compared with GBDT algorithm, LightGBM algorithm can accelerate the training process up to twenty times while achieving almost the same accuracy.

2.3.3. Extreme gradient boosting algorithm (XGBoost)

XGBoost (Chen and Guestrin, 2016) is a scalable end-to-end machine learning algorithm based on GBDT. One of the key points in tree learning is to find the best split, which requires exact greedy algorithm to enumerate all possible splits, and then find the best split. XGBoost algorithm supports the exact greedy algorithm. GBDT uses the negative gradient of the model on the data as the approximate value of the residual to fit the residual. XGBoost also needs to fit the residual on the data, but it is an approximation of the model loss residual using Taylor expansion. Furthermore, it improves the model's loss function and adds a regular term of model complexity. XGBoost automatically uses CPU for multi-threaded parallel computing and performs Taylor second-order expansion of the loss function. At the same time, it takes the complexity of the tree model as a regular item of the objective function to avoid overfitting. The principle of XGBoost is to obtain accurate classification through iterative calculation of weak classifiers. XGBoost algorithm have been recognized in many machine learning tasks such as store sales prediction, high-energy physics event classification and online text classification, etc. The most important factor for XGBoost's success is attributed to its scalability in all situations. In addition, it employs out-of-core computing and can process hundreds of millions of problems on the desktop. XGBoost can use the least resources to solve real-world problems.

2.4. Feature selection

A typical machine learning task is to predict the value of a sample by the features of it. If the sample has few features, we will consider adding features. For example, Polynomial Regression (Chen, 1986) is a typical algorithm for adding features. In reality, there are often too many features and some features need to be reduced. Feature selection has important practical significance. It can not only avoid overfitting, reduce the number of features and improve the generalization ability of the model, but also make the model better interpretability, enhance the understanding between features and feature values and accelerate the model training speed. In general, it will make the model get better performance. In order to perform effective feature fusion and eliminate redundant and noise information in the initial feature vector, we select LightGBM algorithm (Ke et al., 2017) as the feature selection algorithm to determine the final subset of important features.

2.5. Incremental feature selection method (IFS)

In order to select a significant feature subset from the whole feature matrix of protein sequences to build an optimal classifier, not only the feature selection algorithm is required to obtain the importance score of each feature and ranking of all features, but also the IFS method (Liu and Setiono, 1998; Li et al., 2018; Chen et al., 2019) is essential. IFS is a process of building an increasing number of feature subsets by gradually adding features. When a certain feature subset is used to make the evaluation index of the trained classifier reach the maximum, it means that the optimal feature subset has been found. In this way, the classifier constructed based on the optimal feature subset becomes the optimal classifier. We set the feature subset incremental step and starting feature index to one, namely, we select the first feature in the list of initial feature vector after ranking the features by feature selection algorithm to construct the feature subset in the first place, then IFS makes the number of features add one by one to set up the corresponding feature subset, and so on. Under the circumstances, *N*-dimensional feature vector can obtain *N* feature subsets. We can input a series of feature subsets as training data into the classifiers built by various machine learning algorithms in turn. Finally, we can find the optimal classifier by evaluating the predictability of the classifier on different subsets.

2.6. Performance evaluation and model construction

Current work employs Accuracy (ACC), Matthews correlation coefficient (MCC), macro-Precision and macro-Recall (Powers, 2011) as the evaluation criteria of model, whose detailed calculation methods are given in Eq. (6, 7), (8) and (9), where $1 \le n \le 27$, $P = \frac{TP}{TP+FP}$, $R = \frac{TP}{TP+FN}$, TP is the number of positive classes predicted as positive classes, FN is the number of positive classes predicted as negative classes, FP is the number of negative classes predicted as positive classes and TN is the number of negative classes predicted as negative classes. We select ACC and MCC as the key measurements in this study. ACC is the most intuitive and frequently used performance parameter indicating the correctly classified samples against all samples. MCC is essentially a correlation coefficient describing the relationship between actual classification and predicted classification. Precision represents the ratio of the number of correctly classified positive examples to the number of classified positive examples. Recall represents the ratio of the number of correctly classified positive cases to the actual number of positive cases. Since current work is a multi-classification task, the traditional binary evaluation criteria such as Precision and Recall cannot be used to express the predictability of the model. In this case, Macro Average or Micro Average should be considered to calculate Precision and Recall. Macro calculates the Precision and Recall of each category separately, and then averages the Precision and Recall separately. Micro calculates the total number of TP, FP, TN and FN firstly, and then calculates the Precision and Recall. Eqs. (10) and (11) show the calculation methods of micro-Precision and micro-Recall.

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{6}$$

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}} \tag{7}$$

$$\mathrm{macro} - \mathrm{Precision} = \frac{1}{n}\sum_{i=1}^{n} P_i \tag{8}$$

$$\mathrm{macro} - \mathrm{Recall} = \frac{1}{n}\sum_{i=1}^{n} R_i \tag{9}$$

$$\mathrm{micro} - \mathrm{Precision} = \frac{\overline{\mathrm{TP}}}{\overline{\mathrm{TP}} \times \overline{\mathrm{FP}}} \tag{10}$$

$$\mathrm{micro} - \mathrm{Recall} = \frac{\overline{\mathrm{TP}}}{\overline{\mathrm{TP}} \times \overline{\mathrm{FN}}} \tag{11}$$

In this study, we construct the model to realize protein fold recognition in the ASTRAL database through the following calculational process:

1. Dataset. The 4778 protein samples belonging to 27 folding types from ASTRAL protein classification database are selected as the dataset to carry out protein fold recognition.
2. Feature extraction. We employ DSSP based on secondary structure and PSSM and HMM based on evolutionary information to extract 490 features for each protein sequence. In addition, we make comparisons with the AAC feature extraction method which is related to sequence information and physical chemical properties of protein.
3. Feature selection and IFS. In order to remove redundant features and improve the efficiency of the model, the combination of feature selection algorithm and IFS method is introduced. LightGBM feature selection algorithm is used to rank all features and IFS is employed to feed the sorted features into classifier one by one. PCA and LDA are used to further justify the feature selection scheme we employ in this study.

4. Machine learning algorithms. We choose to use three ensemble tree structure-based machine learning algorithms including Random Forest, XGBoost and LightGBM to classify 27 folding types of protein with 5-fold cross-validation after making comparisons with other machine learning algorithms including KNN, SVM, SimpleRNN and LSTM.
5. Hyper-parameter optimization. The Bayesian hyper-parameter optimization method is used to select the optimal hyper-parameters for the classification algorithms including Random Forest, XGBoost and LightGBM respectively to construct the optimal model for protein fold recognition.

3. Results and discussion

3.1. Performance of feature extraction methods

In this study, we use feature extraction methods based on evolutionary information and secondary structure including PSSM, HMM and DSSP to achieve the final 490-dimensional feature vector for each protein sequence. The AAC feature extraction method containing sequence information and physical chemical properties is also a commonly used approach which can extract a 188-dimensional feature vector from a protein sequence (Chen et al., 2016). When we use the classifier built by the machine learning algorithm LightGBM with 5-fold cross-validation for classification research, Table 4 shows us the results of feature extraction methods including PSSM, HMM, DSSP and AAC on the classification of protein folding types. It can be observed that both PSSM and HMM, which are the feature extraction methods based on protein evolutionary information, can play a great role in this experiment. Especially the method of interaction of adjacent amino acids in the protein sequence of HMM, namely, HMM-Bigram, this set of features can make the ACC value reach 0.9090 and MCC reach 0.9029. However, the AAC method has the lowest contribution to protein folding classification. Although the effect of using DSSP features related to secondary structure for protein folding classification is not as good as PSSM features and HMM features, the final ACC value achieved is also 3% higher than using AAC features related to sequence information and physical chemical properties. Therefore, we do not add the set of features extracted by AAC to the final feature vector in current work.

In addition, we have done a detailed study on the combination of different feature extraction methods to elaborate the fusion effect between different features. As shown in Table 5, the results of combination of secondary structure and sequence information or evolutionary information and sequence information or combination of these three methods are worse than the method based on secondary structure and evolutionary information when the LightGBM machine learning algorithm is selected as the classifier. The ACC value of using feature extraction methods on the basis of secondary structure and evolutionary information can reach 0.9180, which is the maximum among four different combinations, so does MCC value. Although the results of methods using the interaction of adjacent amino acids in the protein sequence of HMM are similar, but the dimension of the feature vector obtained through the combination of evolutionary information and sequence information or combination of all three methods is too large, namely, it may have too many redundant features, which may cause a lot of time for model training. Therefore, for feature extraction methods, using the combination of DSSP, PSSM and HMM based on secondary structure and evolutionary information can get superior result in this study and its training speed is relatively fast.

Table 4
The effect of feature extraction methods including PSSM, HMM, DSSP and AAC on the classification of protein folding types.

Feature	ACC	MCC	Macro-Precision	Macro-Recall
DSSP(40D)	0.5693	0.5378	0.6145	0.5054
PSSM(20D)	0.7097	0.6894	0.7812	0.6335
HMM-Monogram(20D)	0.8564	0.8467	0.8863	0.8229
HMM-Bigram(400D)	0.9090	0.9029	0.9309	0.8794
HMM-State(10D)	0.6545	0.6303	0.6625	0.5712
AAC(188D)	0.5343	0.4986	0.5824	0.3795

Table 5
The details of comparison between the combination of secondary structure and evolutionary information and other combinations.

	ACC	MCC	Macro-Precision	Macro-Recall
Secondary structure and evolutionary information(490D)	0.9180	0.9125	0.9393	0.8850
Secondary structure and sequence information(228D)	0.6789	0.6564	0.7617	0.5827
Evolutionary information and sequence information(638D)	0.9125	0.9067	0.9361	0.8836
Secondary structure and evolutionary information and sequence information(678D)	0.9161	0.9105	0.9385	0.8813

3.2. Performance of feature selection methods

Because of using the combination of LightGBM feature selection algorithm and IFS method, each classifier respectively constructed by machine learning algorithms including Random Forest, XGBoost and LightGBM based on tree structure is able to get 490 sets of model training evaluation results for the 490 features extracted from the dataset. We take number of features which sorted by LightGBM feature selection algorithm as the abscissa, ACC or MCC value obtained by the classifier with 5-fold cross-validation as the ordinate to draw the IFS curve, so that we can visually observe how different feature subsets help improve the prediction effect of classifiers. The following Fig. 3(a) gives the IFS curves with respect to ACC values of three classifiers and Fig. 3 (b) indicates the IFS curves in regard to MCC values of three classifiers. It is observed from Fig. 3 that the three classifiers based on tree structure have excellent classification effects and close results. The ACC value and

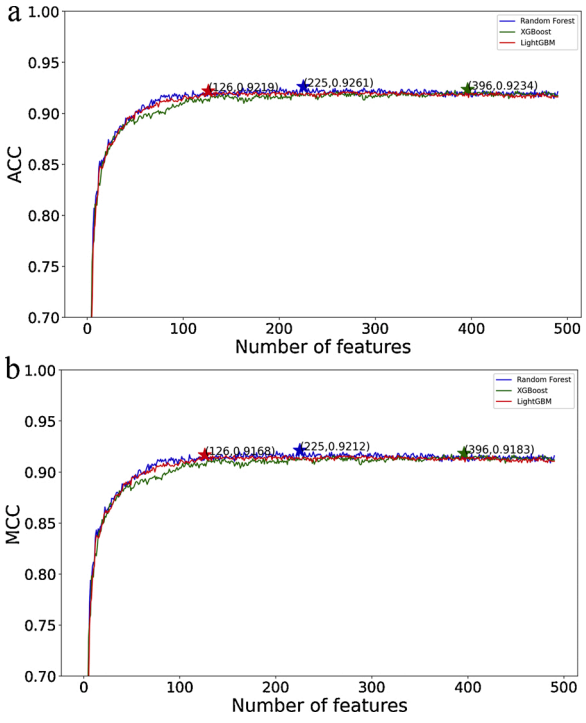

Fig. 3. The IFS curves in regard to ACC and MCC values of classifiers constructed by three machine learning algorithms including Random Forest, XGBoost and LightGBM.

MCC value of Random Forest algorithm achieve the best classification effect in the subset composed of the first 225 features of the sorted feature list. ACC reaches 0.9261 and MCC reaches 0.9212. XGBoost algorithm's ACC value and MCC value achieve the best classification effect in the subset made up of the first 396 features of the sorted feature list. ACC is 0.9234 and MCC is 0.9183. The ACC value and MCC value of LightGBM algorithm reach their own maximum in the subset consisting of the first 126 features of the sorted feature list. ACC value reaches 0.9219 and MCC value reaches 0.9168. Among the three machine learning algorithms, Random Forest can achieve the highest accuracy, but the other two algorithms are not much different from it and the maximum accuracy difference does not exceed 0.005; LightGBM can reach the highest point fastest, while XGBoost is relatively slow to reach the highest point. In addition, the three algorithms are able to keep the prediction effect curve basically the same after number of features reaches a certain number, that is, too many redundant features will not affect the performance of the classifier greatly. The details of best evaluation criteria information including ACC, MCC, macro-Precision and macro-Recall are shown in Table 7. In order to clearly show the superiority and necessity of LightGBM feature selection algorithm and IFS method, we test the performance of initial 490-dimensional feature vector in the same three classifiers, the details are shown in Table 6. It can be seen that the ACC and MCC value of the case without carrying out feature selection algorithm and IFS method are approximately 1% lower than that after doing feature selection.

What is more, Fig. 4 shows the analysis of the selected features after utilizing feature selection method for the three machine learning algorithms. We can see that the selected features are involved in the three feature extraction methods including PSSM, HMM and DSSP, that is, they all contain features based on evolutionary information and features based on secondary structure. Among the 490 features obtained after feature extraction, features based on evolutionary information accounted for most. The proportion of features do not change significantly after carrying out feature selection. Therefore, both evolutionary information-based and secondary structure-based features play a key role in protein folding classification.

To further justify the combination of LightGBM feature selection algorithm and IFS method to do feature selection in this study, we employ some standard dimensionality reduction techniques including PCA (Principal Component Analysis) (Hervé and Williams, 2010) and LDA (Linear Discriminant Analysis) (Riffenburgh and Clunies-Ross, 2013) to make comparisons with the effect of it. Tables 8 and 9respectively gives the specific effect of PCA and LDA. It can be seen that the two dimensionality reduction techniques are very similar in both ACC and MCC values, but the effects of them are not as good as the feature selection method we employ in current work. Hence the combination of LightGBM feature selection algorithm and IFS method is a great choice to carry out feature selection to improve the ACC value of protein folding classification.

3.3. Performance of machine learning algorithms

KNN (K-nearest neighbor) (Keller et al., 1985) and SVM (Support Vector Machines) (Hearst et al., 1998) are also two classic algorithms of machine learning algorithms and so as some Artificial Neural Networks such as SimpleRNN and LSTM (Graves et al., 2013). For the sake of reflecting the superiority of tree structure-based machine learning algorithms on the multi-classification problem clearly, we input the same dataset into the other four algorithms including KNN, SVM, SimpleRNN and LSTM with the same settings. The following Table 10 shows the details of comparison between KNN, SVM, SimpleRNN, LSTM algorithms and Random Forest, XGBoost, LightGBM algorithms. It can be seen from Table 10 that the classification effect of the SVM algorithm is relatively good, the ACC value of it is as high as 0.9031 and the MCC value is 0.8969, but the ACC value is still about 2% lower than that of the

Table 6
The respective performance of Random Forest, XGBoost and LightGBM without doing feature selection.

	ACC	MCC	Macro-Precision	Macro-Recall
Random Forest	0.9144	0.9087	0.9469	0.8854
XGBoost	0.9196	0.9143	0.9329	0.8951
LightGBM	0.9180	0.9125	0.9393	0.8850

Table 7
The respective performance of Random Forest, XGBoost and LightGBM after using the combination of LightGBM feature selection algorithm and IFS method.

	ACC	MCC	Macro-Precision	Macro-Recall
Random Forest	0.9261	0.9212	0.9560	0.8959
XGBoost	0.9234	0.9183	0.9344	0.9001
LightGBM	0.9219	0.9168	0.9456	0.8912

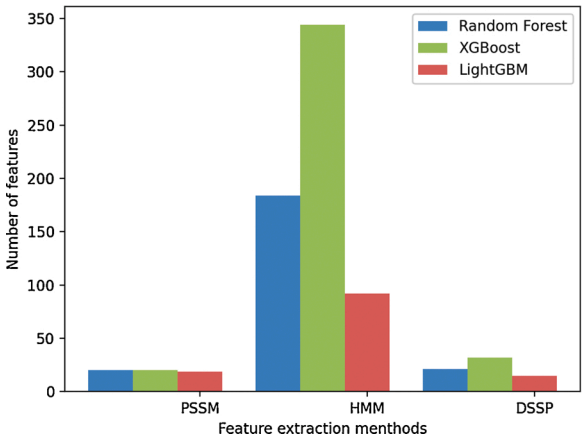

Fig. 4. The number of the selected features after doing feature selection.

Table 8
The respective performance of Random Forest, XGBoost and LightGBM after using PCA.

	ACC	MCC	Macro-Precision	Macro-Recall
Random Forest	0.8826	0.8749	0.9425	0.8445
XGBoost	0.8815	0.8735	0.8993	0.8493
LightGBM	0.8757	0.8675	0.9233	0.8397

Table 9
The respective performance of Random Forest, XGBoost and LightGBM after using LDA.

	ACC	MCC	Macro-Precision	Macro-Recall
Random Forest	0.8876	0.8802	0.9025	0.8589
XGBoost	0.8765	0.8683	0.8864	0.8499
LightGBM	0.8828	0.8750	0.9000	0.8541

Table 10
The details of comparison between KNN, SVM, SimpleRNN, LSTM algorithms and Random Forest, XGBoost, LightGBM algorithms.

	Size	ACC	MCC	Macro-Precision	Macro-Recall
Random Forest	225	0.9261	0.9212	0.9560	0.8959
XGBoost	396	0.9234	0.9183	0.9344	0.9001
LightGBM	126	0.9219	0.9168	0.9456	0.8912
KNN	129	0.8642	0.8551	0.8756	0.8428
SVM	415	0.9031	0.8969	0.9514	0.8708
SimpleRNN	488	0.9182	0.9127	0.9207	0.9066
LSTM	365	0.9215	0.9163	0.9263	0.9123

tree structure-based algorithms. The effect of KNN algorithm is obviously weaker than other methods, the ACC value of which is about 6% lower than that of Random Forest, XGBoost and LightGBM. It can be also observed that SimpleRNN and LSTM algorithms are not as good as tree structure-based machine learning algorithms used in this study. The ACC values obtained by these two algorithms are approximately 92% similar to the ACC values obtained by three tree structure-based algorithms such as LightGBM algorithm, but they still do not exceed the tree structure-based algorithms. Furthermore, it can be known that the time required for these two algorithms to reach the maximum accuracy is relatively long after experiencing IFS. Therefore, the classifiers constructed by tree structure-based machine learning algorithms including Random Forest, XGBoost and LightGBM are selected to classify folding types of the ASTRAL dataset and carry on the next experiment.

3.4. Performance of hyper-parameter optimization

By adjusting the hyper-parameters in the machine learning algorithm, the classification effect can have certain ups and downs, which is an effective way to improve the overall accuracy. In current work, Bayesian optimization is used for hyper-parameter adjustment, which can obtain the best hyper-parameters of a given model. Bayesian optimization for machine learning tuning was proposed by Snoek et al. (2012). The main idea is that given an optimized objective function, update the posterior distribution of the objective function by continuously adding sample points until the posterior distribution basically fits the true distribution. Simply put, it takes the information about the last parameters into account to better adjust the current parameters. For Random Forest, XGBoost and LightGBM algorithms, we all use Bayesian optimization to adjust some main hyper-parameters of these algorithms and perform 50 iterations. In the previous experiment, we know that using the first 126 features after feature selection can make the LightGBM algorithm reach the highest accuracy. On the basis of these 126 features, we use Bayesian optimization method to optimize the hyper-parameters of the LightGBM algorithm. The red polyline in Fig. 5 shows us the ACC values corresponding to 50 adjustments. We can see that the ACC value can reach 0.9345 under the best circumstances after hyper-parameter adjustment and the corresponding best hyper-parameters are given in Table 11. This result is 1.26% higher than before carrying out hyper-parameters optimization. Random Forest algorithm can make the classification effect best when the first 225 features are input after feature selection. In this case, Bayesian optimization method is used for the algorithm to adjust its hyper-parameters. The final ACC value can reach a maximum of 0.9280, which is 0.19% higher than the ACC without optimizing the hyper-parameters. The effect of hyper-parameter optimization of Random Forest algorithm is not so obvious. The specific tuning curve of Random Forest algorithm is shown by the blue polyline in Fig. 5 and the best hyper-parameters is listed in Table 12. The green polyline in Fig. 5 and Table 13 indicate the details of tuning results of XGBoost algorithm, whose features are the first 396 features in the sorted feature list after feature selection. ACC value of this algorithm can highly reach 0.9299. Table 14 shows us the specific evaluation index information of Random Forest, XGBoost and LightGBM algorithms after Bayesian hyper-parameter optimization. It is observed from Table 14 that LightGBM algorithm can reach the highest accuracy when compared with the other two algorithms after Bayesian optimization and ACC value is 93.45%.

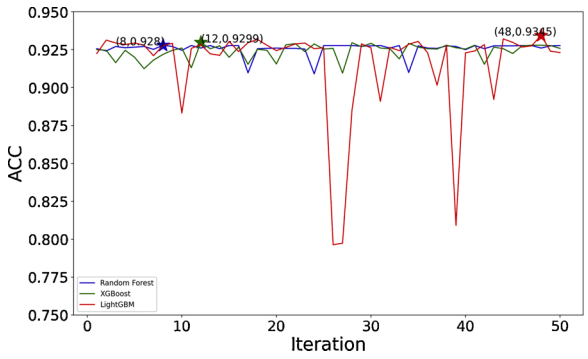

Fig. 5. The hyper-parameter optimization process of three machine learning algorithms including Random Forest, XGBoost and LightGBM using Bayesian optimization.

Table 11
Best hyper-parameters used for LightGBM algorithm for protein fold recognition.

Hyper-parameter	Description	Value
learning_rate	Determine whether the objective function can converge	0.17502795976044572
max_bin	The maximum number of bins that the feature will store	34
max_depth	Maximum depth of the tree	12
n_estimators	Number of subtrees	294
num_leaves	Number of leaves	7

Table 12
Best hyper-parameters used for Random Forest algorithm for protein fold recognition.

Hyper-parameter	Description	Value
max_depth	Maximum depth of the tree	18
n_estimators	Number of subtrees	367
criterion	The function to measure the quality of a split	gini
min_samples_leaf	The minimum number of samples required to be at a leaf node	1
min_samples_split	The minimum number of samples required to split an internal node	2

3.5. Analysis of other methods for protein fold recognition

The study of protein fold recognition is of great significance for understanding protein structure and analyzing its function. So far, many researchers have studied protein fold recognition problem and most of the research on it is based on the SCOP database (DD, TG, EDD). The DD dataset is a public and stringent dataset proposed by Ding and Dubchak (2001). Since the 27 folding types in the ASTRAL protein classification database selected in this experiment are the same as the 27 folding types in the DD dataset, we analysis the results of this experiment and other methods in recent 5 years for protein fold recognition. In order to reflect the superiority of the methods used in current work for the study of protein fold recognition, we list the classification effect of other methods. Table 15 shows us the effect of different methods on fold recognition through the ACC value. We can see that researchers have

Table 13
Best hyper-parameters used for XGBoost algorithm for protein fold recognition.

Hyper-parameter	Description	Value
colsample_bytree	Control the proportion of each randomly sampled column	0.7653748806177161
gamma	The minimum loss function drop required for node splitting	0.1801710329845556
learning_rate	Determine whether the objective function can converge	0.05861333738708681
max_depth	Maximum depth of the tree	10
min_child_weight	The sum of the smallest leaf node weights	3
n_estimators	Number of subtrees	398
subsample	The proportion of samples randomly sampled from each tree	0.6174251420225305

Table 14
The performance of Random Forest, XGBoost and LightGBM algorithms after Bayesian hyper-parameter optimization on the basis of their respective optimal feature matrices.

	ACC	MCC	Macro-Precision	Macro-Recall
Random Forest	0.9280	0.9233	0.9604	0.9006
XGBoost	0.9299	0.9252	0.9426	0.9069
LightGBM	0.9345	0.9302	0.9450	0.9094

Table 15
The respective results of different methods used in protein fold recognition research.

Methods	Overall Accuracy (%)
PFPA (2015) (Wei et al., 2015)	73.6
Feng et al. (2016)	70.8
ProFold (2016) (Chen et al., 2016)	76.2
PHMM-DP (2016) (Lyons et al., 2016)	82.7
MV-fold(2019) (Yan et al., 2019)	83.5
MT-fold(2019) (Yan et al., 2019)	88.2
Current work	93.5

never stopped studying the problem of protein fold recognition in the past five years and the classification effect has also been continuously improved. This experiment is a research based on ASTRAL database and we choose PSSM, HMM and DSSP as the feature extraction methods, select Random Forest, LightGBM and XGBoost as the classification algorithms and combine the feature selection algorithm and IFS method to make the classification results of protein folding types as high as 93.45%. For the same classification problem of 27 protein folding types, the result of this experiment is outstanding.

4. Conclusion

Protein fold recognition is a significant approach to understand the function of protein. For the sake of realizing the classification of 27 protein folding types covering four protein structural classes composed of α, β, α/β and $\alpha+\beta$ in the ASTRAL protein classification database, we first use feature extraction methods including DSSP, PSSM and HMM based on secondary structure and evolutionary information of protein to translate protein sequences into the final 490-dimensional feature vectors after making comparisons with AAC feature extraction method which is related to the sequence information and physical chemical properties of protein. Then with the assist of the combination of LightGBM feature selection algorithm and IFS method, the final feature matrix can be respectively fed into the classifiers constructed by three machine learning algorithms on the basis of tree structure, for instance, Random Forest, XGBoost and LightGBM. In general, the ACC value of using ensemble machine learning algorithms based on tree structure is 6% higher than that of using k-nearest neighbor algorithm and 2% higher than that of using Support Vector Machines algorithm with the same feature extraction methods and other settings. And the efficiency of these ensemble algorithms is higher than that of the SimpleRNN and LSTM algorithms, too. Finally, after performing hyper-parameter adjustment through Bayesian optimization method, the final ACC value of the best classification effect can reach 93.45%. In summary, the combination of feature extraction methods based on secondary structure and evolutionary information and utilizing ensemble machine learning algorithms based on tree structure can make the folding classification effect of the ASTRAL dataset superior and outstanding.

References

Altschul, S.F., Madden, T.L., Schäffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J., 1997. Gapped blast and PSI-blast: a new generation of protein database search programs. Nucleic Acids Res. 25, 3389–3402. https://doi.org/10.1093/nar/25.17.3389.

Andreeva, A., Kulesha, E., Gough, J., Murzin, A.G., 2020. The SCOP database in 2020: expanded classification of representative family and superfamily domains of known protein structures. Nucleic Acids Res. (D1), D1. https://doi.org/10.1093/nar/gkz1064.

Baldwin, E.T., Weber, I.T., Charles, R.S., Xuan, J.C., Appella, E., Yamada, M., Matsushima, K., Edwards, B.F., Clore, G.M., Gronenborn, A.M., 1991. Crystal structure of interleukin 8: symbiosis of NMR and crystallography. Proc. Natl. Acad. Sci. 88 (2), 502–506. https://doi.org/10.1073/pnas.88.2.502.

Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E., 2000. The protein data bank. Int. Tables Crystallogr. 67 (Suppl), 675–684. https://doi.org/10.1107/97809553602060000722.

Bragg, S.L., 1976. The development of X-ray analysis. Contemp. Phys. 17 (1), 103–104. https://doi.org/10.1080/00107517608210844.

Breiman, L., 2001. Random forests. Mach. Learn. 45 (1), 5–32. https://doi.org/10.1023/A:1010933404324.

Chandonia, J.M., Fox, N.K., B.S, E., 2019. Scope: classification of large macromolecular structures in the structural classification of proteins-extended database. Nucleic Acids Res. https://doi.org/10.1093/nar/gky1134.

Chandonia, J.M., Hon, G., Walker, N.S., Lo Conte, L., Koehl, P., Levitt, M., Brenner, S.E., 2004. The ASTRAL compendium in 2004. Nucleic Acids Res. 32, D189–D192. https://doi.org/10.1093/nar/gkh034.

Chen, B., 1986. Polynomial regression. Springer Texts Stat. 235–268. https://doi.org/10.1007/978-94-009-5836-4_7.

Chen, D.Z., Tian, X.Y., Zhou, B., Gao, J., 2016. Profold: Protein fold classification with additional structural features and a novel ensemble classifier. BioMed Res. Int. 2016, 1–10. https://doi.org/10.1155/2016/6802832.

Chen, L., Pan, X.Y., Zhang, Y.H., Liu, M., Huang, T., Cai, Y.D., 2019. Classification of widely and rarely expressed genes with recurrent neural network. Comput. Struct. Biotechnol. J. 17, 49–60. https://doi.org/10.1016/j.csbj.2018.12.002.

Chen, T.Q., Guestrin, C., 2016. Xgboost: A Scalable Tree Boosting System. ACM, pp. 785–794. https://doi.org/10.1145/2939672.2939785.

Chou, K.C., 2010. Prediction of protein cellular attributes using pseudo-amino acid composition. Proteins-Struct. Funct. Bioinf. 43 (3), 246–255. https://doi.org/10.1002/prot.1035.

Cohen, F.E., Kuntz, I.D., 1987. Prediction of the three-dimensional structure of human growth hormone. Proteins Struct. Funct. Bioinf. 22, 162–166. https://doi.org/10.1002/prot.340020209.

Ding, C.H., Dubchak, I., 2001. Multi-class protein fold recognition using support vector machines and neural networks. Bioinformatics 17 (4), 349–358. https://doi.org/10.1093/bioinformatics/17.4.349.

Dubchak, I., Muchnik, I., Holbrook, S.R., Kim, S.H., 1995. Prediction of protein folding class using global description of amino acid sequence. Proc. Natl. Acad. Sci. U. S. A. 92 (19), 8700–8704. https://doi.org/10.1073/pnas.92.19.8700.

Feng, Z.X., Hu, X.Z., Jiang, Z., Song, H.Y., Ashraf, M.A., 2016. The recognition of multi-class protein folds by adding average chemical shifts of secondary structure elements. Saudi J. Biol. Sci. 23 (2), 189–197. https://doi.org/10.1016/j.sjbs.2015.10.008.

Friedman, J.H., 2001. Greedy function approximation: a gradient boosting machine. Ann. Stat. 29 (5), 1189–1232. https://doi.org/10.2307/2699986.

Graves, A., Mohamed, A.R., Hinton, G., 2013. Speech recognition with deep recurrent neural networks. Acoust. Speech Signal Process. https://doi.org/10.1109/ICASSP.2013.6638947.

Hearst, M.A., Dumais, S.T., Osman, E., Platt, J., Scholkopf, B., 1998. Support vector machines. IEEE Intell. Syst. 13 (4), 18–28. https://doi.org/10.1109/5254.708428.

Heffernan, R., Paliwal, K., Lyons, J., Dehzangi, A., Sharma, A., Wang, J.H., Sattar, A., Yang, Y.D., Zhou, Y.Q., 2015. Improving prediction of secondary structure, local backbone angles, and solvent accessible surface area of proteins by iterative deep learning. Reports 5, 11476. https://doi.org/10.1038/srep11476.

Hervé, A., Williams, L.J., 2010. Principal component analysis. Wiley Interdiscip. Rev. Comput. Stat. 2 (4), 433–459. https://doi.org/10.1002/wics.101.

Ibrahim, W., Abadeh, M.S., 2018. Protein fold recognition using deep kernelized extreme learning machine and linear discriminant analysis. Neural Comput. Appl. (4), 1–14. https://doi.org/10.1007/s00521-018-3346-z.

Kabsch, W., Sander, C., 1983. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 22 (12), 2577–2637. https://doi.org/10.1002/bip.360221211.

Kavousi, K., Moshiri, B., Sadeghi, M., Araabi, B.N., Moosavi-Movahedi, A.A., 2011. A protein fold classifier formed by fusing different modes of pseudo amino acid composition via PSSM. Comput. Biol. Chem. 35 (1), 1–9. https://doi.org/10.1016/j.compbiolchem.2010.12.001.

Ke, G.L., Meng, Q., Finley, T., Wang, T.F., Chen, W., Ma, W.D., Ye, Q.W., Liu, T.Y., 2017. Lightgbm: a highly efficient gradient boosting decision tree. Conference on Neural Information Processing Systems.

Keller, J.M., Gray, M.R., Givens, J.A., 1985. A fuzzy k-nearest neighbor algorithm. IEEE Trans. Syst., Man, Cybern. SMC-15, 580–585. https://doi.org/10.1109/TSMC.1985.6313426.

Li, J.R., Lu, L., Zhang, Y.H., Liu, M., Chen, L., Huang, T., Cai, Y.D., 2018. Identification of synthetic lethality based on a functional network by using machine learning algorithms. J. Cell. Biochem. 120, 405–416. https://doi.org/10.1002/jcb.27395.

Liang, Y.Y., Liu, S.Y., Zhang, S.L., 2015. Prediction of protein structural classes for low-similarity sequences based on consensus sequence and segmented PSSM. Comput. Math. Methods Med. 2015, 1–9. https://doi.org/10.1155/2015/370756.

Lin, C., Zou, Y., Qin, J., Liu, X.R., Jiang, Y., Ke, C.H., Zou, Q., 2013. Hierarchical classification of protein folds using a novel ensemble classifier. PLoS One 8 (2), e56499. https://doi.org/10.1371/journal.pone.0056499.

Liu, H., Setiono, R., 1998. Incremental feature selection. Appl. Intell. 9 (3), 217–230. https://doi.org/10.1023/A:1008363719778.
Lyons, J., Paliwal, K.K., Dehzangi, A., Hefferman, R., Tsunoda, T., Sharma, A., 2016. Protein fold recognition using HMM–HMM alignment and dynamic programming. J. Theor. Biol. 393, 67–74. https://doi.org/10.1016/j.jtbi.2015.12.018.
Mehta, A., Himanshu, M., 2019. Predicting structural class for protein sequences of random forest algorithm. Comput. Biol. Chem. 84, 107164. https://doi.org/10.1016/j.compbiolchem.2019.107164.
Murzin, A.G., Brenner, S.E., Hubbard, T., Chothia, C., 1995. SCOP: a structural classification of proteins database for the investigation of sequences and structures. J. Mol. Biol. 247 https://doi.org/10.1006/jmbi.1995.0159.
Orengo, C.A., Michie, A.D., Jones, S., Jones, D.T., Swindells, M.B., Thornton, J.M., 1997. CATH – a hierarchic classification of protein domain structures. Structure 5 (8), 1093–1108. https://doi.org/10.1016/S0969-2126(97)00260-8.
Paliwal, K.K., Sharma, A., Lyons, J., Dehzangi, A., 2014. A tri-gram based feature extraction technique using linear probabilities of position specific scoring matrix for protein fold recognition. NanoBioscience 13 (1), 44–50. https://doi.org/10.1109/TNB.2013.2296050.
Powers, D., 2011. Evaluation: from Precision, Recall and F-measure to ROC, informedness, markedness and correlation. J. Mach. Learn. Technol. 2, 37–63.
Remmert, M., Biegert, A., Hauser, A., Söding, J., 2012. HHblits: lightning-fast iterative protein sequence searching by HMM–HMM alignment. Nat. Methods 9, 173–175. https://doi.org/10.1038/nmeth.1818.
Renaux, A., 2018. UniProt: the universal protein knowledgebase. Nucleic Acids Res. 45 (D1), D158–D169. https://doi.org/10.1093/nar/gky092.
Riffenburgh, R.H., Clunies-Ross, C.W., 2013. Linear discriminant analysis. Chicago 3 (6), 27–33. https://doi.org/10.1007/978-1-4419-9863-7_395.
Sela, M., Anfinsen, C.B., Harrington, W.F., 1957. The correlation of ribonuclease activity with specific aspects of tertiary structure. Biochim. Biophys. Acta 26 (3), 502–512. https://doi.org/10.1016/0006-3002(57)90096-3.
Snoek, J., Larochelle, H., Adams, R.P., 2012. Practical Bayesian optimization of machine learning algorithms. Adv. Neural Inf. Process. Syst. 4.
Stuart, D.I., 1976. Protein Crystallography. Academic Press. https://doi.org/10.1007/978-1-4757-0166-1_6.
Touw, W.G., Baakman, C., Black, J., te Beek, T.A.H., Krieger, E., Joosten, R.P., Vriend, G., 2014. A series of PDB-related databanks for everyday needs. Nucleic Acids Res. 43 (D1) https://doi.org/10.1093/nar/gku1028.
Wei, L.Y., Liao, M.H., Gao, X., Zou, Q., 2015. Enhanced protein fold prediction method through a novel feature extraction technique. IEEE Trans. Nanobiosci. 14 (6), 649–659. https://doi.org/10.1109/TNB.2015.2450233.
Wei, L.Y., Zou, Q., 2016. Recent progress in machine learning-based methods for protein fold recognition. Int. J. Mol. Sci. 17 (12) https://doi.org/10.3390/ijms17122118.
Yan, K., Fang, X.Z., Xu, Y., Liu, B., 2019. Protein fold recognition based on multi-view modeling. Bioinformatics 35 (17), 2982–2990. https://doi.org/10.1093/bioinformatics/btz040.
Yan, K., Wen, J., Liu, J.X., Xu, Y., Liu, B., 2020. Protein fold recognition by combining support vector machines and pairwise sequence similarity scores. IEEE/ACM Trans. Comput. Biol. Bioinf. PP (99) https://doi.org/10.1109/TCBB.2020.2966450, 1-1.