Protein Fold Recognition Based on Auto-Weighted Multi-view Graph Embedding Learning Model

Ke Yan, Jie Wen, Yong Xu*, Bin Liu*

Abstract—Protein fold recognition is critical for studies of the protein structure prediction and drug design. Several methods have been proposed to obtain discriminative features from the protein sequences for fold recognition. However, the ensemble methods that combine the various features to improve predictive performance remain the challenge problems. In this study, we proposed two novel algorithms: AWMG and EMfold. AWMG used a novel predictor based on the multi-view learning framework for fold recognition. Each view was treated as the intermediate representation of the corresponding data source of proteins, including the evolutionary information and the retrieval information. AWMG calculated the auto-weight for each view respectively and constructed the latent subspace which contains the common information shared by different views. The marginalized constraint was employed to enlarge the margins between different folds, improving the predictive performance of AWMG. Furthermore, we proposed a novel ensemble method called EMfold, which combines two complementary methods AWMG and DeepSS. The later method was a template-based algorithm using the SPARKS-X and DeepFR programs. EMfold integrated the advantages of template-based assignment and machine learning classifier. Experimental results on the two widely datasets (LE and YK) showed that the proposed methods outperformed some state-of-the-art methods, indicating that AWMG and EMfold are useful tools for protein fold recognition.

Index Terms — Protein Fold Recognition; Template-based method; Multi-view learning method; Laplacian matrix

1 INTRODUCTION

Protein tertiary structure identification is of great significance in understanding the functional properties of the protein, protein functions denotation, protein-protein interactions, etc. Protein fold recognition is to predict the protein three-dimensional structures based on their sequences [1-3]. Proteins in the same fold usually have similar major secondary structures and functions [4]. Therefore, protein fold recognition is critical for protein structure identification and protein function annotation.

Protein fold recognition is a typical classification problem, which recognizes the known templates to the query protein with a similar structure. Protein fold recognition can be divided into two categories, including machine learning methods and template-based methods [5, 6].

The template-based methods calculate the similarity scores between the target sequences and the templates, which are related of proteins with known structure. The quality of the alignment results mainly depends on the template proteins selection. There are different alignment strategies to search the templates, such as the sequence-to-profile methods, sequence-to-structure methods, etc. When sequence identity is higher than 30% [7, 8], the evolutionary relationships can be evaluated using pairwise sequence comparison methods [9-11]. The profiles encoded from the multiple sequences alignment (MSA) [12] are used to measure the evolutionary information [13-15]. A profile can be represented as the position-specific scoring matrix (PSSM) calculated from PSI-BLAST [9] or Hidden Markov Models (HMM) obtained from HHblits [10]. Besides those methods are based on the protein sequence information, some authors proposed the threading methods using the structural information. The alignment scores access the similarity between the query sequences and different known structures using the specific scoring functions. Because of the proteins probably share the structural similarity, the fitness scores are detected from the structure information without using the homologous relationship with the target sequences. For example, Yang et al. proposed a threading tool SPARKS-X [16] by evaluating the similarity between the target sequences and templates based on the Protein Data Bank (PDB) by using structural properties, such as secondary structure, backbone torsion, etc.

For machine learning methods, protein fold recognition methods contain two phases, including feature extraction and designing discriminative classifiers. For the first stage, several features have been extracted from protein sequences. The commonly used descriptors represent the different properties of protein sequences [17], such as the Amino Acid Component (AAC) [18], Pseudo AAC [19], Physicochemical feature [20], Autocross-
covariance (ACC) based on the PSSM [21], etc. For the second stage, some taxonomy algorithms are developed to improve the prediction performance based on some hyper-parameters configurations and the hyper-parameters are selected to fit the validation data appropriately. Some classification algorithms are commonly used to train the model, such as Support Vector Machines (SVM) [20, 22], Random Forest (RF) [23], Naïve Bayes (NB) [24]. Multi-view Learning model [5], etc. For example, Shivashankar et al. proposed a multi-view weighted method to combine multiple representation to predict the protein structures [25]. Liu et al. proposed a DeepSVM-fold which extracts the fold specific features based on the Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (BLSTM) and feds the pairwise features into the SVM for protein fold recognition [22]. Polat et al. proposed the GAL predictor based on a new neural network for protein fold recognition [26].

The aforementioned methods contribute to the development of the protein fold recognition task. However, there are some questions should be addressed in this field: (1) most of the machine learning methods utilize the compound feature which combines the different features to improve the protein fold recognition performance, but the performance of fusing features displays some redundant elements or irrelevant elements in the comprehensive features [27]. The compound feature may cause the curse of dimensionality problem. (2) Some methods ignore the diversity of the special features and others learn a weight for each feature with an additional parameter [28]. As a result, some weak features will decrease the protein fold recognition performance. It is critical to utilize the dependency information of different features reasonably and learn a set of weights with respect to the different features automatically.

Multi-view learning model utilizes multiple data sources from the protein sequences and integrates the various data sources to improve the predictive performance. Each data source contains some features from the protein sequence. For example, the evolutionary features and the retrieval features based on the threading tools are different data sources. In this work, we utilize the multi-view learning model based on subspace learning to predict the protein fold types. Assuming that multiple views from the protein sequences are generated from the same latent subspace, multi-view learning methods obtain the latent subspace using the subspace learning algorithm [29]. Because of the dimension of the latent subspace is lower than any original views, multi-view learning model is able to avoid the curse of dimensionality [29]. Yan et al. proposed a MV-fold method to predict the protein fold by using the multi-view learning model [5].

Inspired by the multi-view learning model [5] and auto-weighted multiple graph learning framework [30], we propose a method for fold recognition based on the auto-weighted multi-view graph embedding learning model called AWMG. The proposed method utilizes multiple views of protein sequences to train the model. For each view, we learn the special weight automatically and capture the nearest neighbor relationship of each sequence using the Laplacian matrix [30]. Then the latent subspace is constituted using the common information of various data sources and is used to predict the protein fold types precisely. The proposed method utilizes the special marginalized constraint to enlarge the margins between the different fold types to improve the discriminative performance. Compared with the combined feature methods, the proposed method constitutes a common latent feature to represent the features from different views. Compared with traditional multi-view learning methods, the proposed method utilizes the auto-weight for different views and a special strategy to enlarge the boundary of different folds. As a result, the AWMG utilizes the different views from the protein sequences and has the discriminative ability for protein fold recognition. Furthermore, an ensemble method called EMfold is proposed by combining the AWMG and two template-based methods: SPARKS-X [10] and DeepFR [31].

2 MATERIALS AND METHODS

2.1 Benchmark dataset

In this study, LE dataset and YK dataset were used to evaluate the performance. The LE dataset proposed by Lindahl and Elofsson [32] was obtained from the Structural Classification of Protein (SCOP) version 1.37. The sequence identity between any two sequences is less than 40%. The dataset contains 321 sequences with 38 folds and is split into two subsets at the fold level, including 159, 162 sequences in training subset and test subset, respectively. These two subsets share no proteins from the same superfamily or family. In order to rigorously simulate the protein fold recognition task, the 2-fold cross-validation was used to evaluate the performance of a predictor on the LE dataset.

The YK dataset derived from SCOP 2.07 was proposed by Yan et al. [5]. The pairwise sequence identity is less than 40%. The dataset contains 4843 sequences with 82 folds. In order to remove the homologous sequence redundancy, the dataset is divided into three subsets according to the SCOPe, including 1536, 1628, 1679 sequences in training, validation, and test subsets, respectively. Any two subsets share no proteins from the same superfamily or family. In other words, no proteins from the same superfamily will be included in three subsets. As a result, the 3-fold cross-validation was used to evaluate the performance of a predictor on the YK dataset so as to rigorously simulate the protein fold recognition task.

2.2 Framework of AWMG

2.2.1 Problem Formulation and Learning Model

The benchmark dataset contains \( n \) sequences \( \{ x_i, y_i \}_{i=1}^n \) of \( c \) fold types from SCOP, where \( x_i \in \mathbb{R}^m \) is the feature...
vector of the $i$-th protein sequence, and $y_i \in \mathbb{R}^c$ represents the corresponding protein fold type. For a clear description of different fold types, $y_i$ is the strict binary vector. If the $i$-th protein sequence belongs to the $j$-th class ($j \in [1, \ldots, c]$), the $j$-th element of $y_i$ is 1 and the others elements are 0. Given a set of $n$ benchmark sequences $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{n \times c}$ and $r$ query sequences $\mathbf{X}' = [\mathbf{x}'_1, \ldots, \mathbf{x}'_r] \in \mathbb{R}^{r \times c}$, where $d \in [1, \ldots, D]$ represents the $d$-th view and $m_i$ denotes the feature dimension of the $d$-th view. $Y = [y_{ij}, \ldots, y_{ij}] \in \mathbb{R}^{n \times c}$ is the strict binary matrix corresponding to different fold types.

Inspired by the multi-view learning model [5] and auto-weighted multiple graph learning framework [30], we embedded the protein sequences from $D$ views with corresponding weights to the following framework,

$$
\min_{\mathbf{W}, \mathbf{\lambda}} \sum_{i,j} a_{ij}(d) \left( \mathbf{P}^{(d)}\mathbf{X}^{(d)} - \mathbf{V}^d \right)^2 + \lambda_r \left( \mathbf{P}^{(d)}\mathbf{X}^{(d)}\mathbf{L}^{(d)}\mathbf{X}^{(d)}\mathbf{P}^{(d)} + \lambda_2 \mathbf{P}^{(d)} \right)
$$

where $a_{ij}(d)$ is used to automatically weighted with the $d$-th view and is defined as:

$$
a_{ij}(d) = \frac{1}{\mathcal{N}^{(d)}_i} - \lambda_r \left( \mathbf{P}^{(d)}\mathbf{X}^{(d)}\mathbf{L}^{(d)}\mathbf{X}^{(d)}\mathbf{P}^{(d)} + \lambda_2 \mathbf{P}^{(d)} \right), \quad r(r > 1)
$$

is a trade-off parameter. $\mathbf{P}^{(d)} \in \mathbb{R}^{n \times n}$ is the transformation matrix corresponding to the $d$-th view and $\mathbf{W} \in \mathbb{R}^{n \times n}$ is the learned projection matrix related to the label matrix. $\mathbf{V} \in \mathbb{R}^{n \times c}$ is the latent subspace which encodes the common information from different views. $\mathbf{L}^{(d)} \in \mathbb{R}^{n \times n}$ is the Laplacian matrix which enables the method to select the few nearest neighbour protein sequences adaptively [30]. $\mathbf{G} \in \mathbb{R}^{n \times n}$ is the learned regression target matrix which represents the categories of fold types of the benchmark dataset. $\lambda_1, \lambda_2, \lambda_3, \lambda_4, r$ and $\varepsilon$ are parameters.

Multi-view learning method obtains a latent subspace shared by multiple views [29]. The proposed method utilizes $D$ view data sources from protein sequences and learns the latent subspace $\mathbf{V}$ which represents the common representation of the multiple views consistently. Due to the parameter $r$ is used to avoid a trivial solution that only considers one view and adjusts the complementarity of multiple views [33], the parameter $a'_{ij}(d)$ assigns the suitable weights to different views and adjusts the complementarity of multiple views. Therefore, the proposed method constructs the common representation of multiple views.

The neighbour graph is widely used in pattern recognition and clustering owing to its success in revealing the geometric structure of data [30, 34]. The proposed method utilizes the neighbour graph regularization based manifold learning method to capture the intrinsic locality structure of data. We obtain the nearest neighbour graph $\mathbf{W}$ to learn the local relationship between the samples and guide the projection matrix learning by the following problem,

$$
\min_{\mathbf{W}} \sum_{k,j} \left| \mathbf{w}_{kj}^{(d)} \right| \left( \mathbf{P}^{(d)}\mathbf{X}^{(d)} - \mathbf{P}^{(d)}\mathbf{X}^{(d)} \right)^2 \mathbf{w}_{kj}^{(d)} \quad (2)
$$

where the element $w_{kj}^{(d)}$ of graph $\mathbf{W}$ can be defined as follows,

$$
w_{kj}^{(d)} = \begin{cases} 
\frac{\mathbf{1} - \mathbf{1}}{2} & \text{if } x_i \in N_k(x_j) \text{ or } x_j \in N_k(x_i) \\
0 & \text{otherwise}
\end{cases}
\quad (3)
$$

where $N_k(x_i)$ is the set of $k$ nearest neighbor samples of sample $x_i$. $w_{kj}$ denotes that the samples $x_j$ is the nearest neighbour of $x_i$ and indicates that these two samples have the similar data distribution. Thus, graph $\mathbf{W}$ captures the local neighbor information of protein sequences. According to the previous study [35], Eq. 2 can be transformed into the following formula,

$$
\mathbf{L}^{(d)}\mathbf{X}^{(d)}\mathbf{L}^{(d)}\mathbf{X}^{(d)} + \lambda_2 \mathbf{P}^{(d)}
$$

where $\mathbf{L}^{(d)}$ is the Laplacian matrix. $\mathbf{L}^{(d)} = \mathbf{D}^{(d)} - \mathbf{W}^{(d)}$, where $\mathbf{D}^{(d)}$ is a diagonal matrix and its entries are column sums of $\mathbf{W}^{(d)}$. As a result, the proposed method captures the local data information of protein sequences.

Because the traditional strict binary regression matrix target $\mathbf{Y}$ has weak separability, a marginalized constraint enforces the regression target of the different protein fold types by enlarging the margins between the observe (true) class and the other (false) classes to the greatest extent possible. Inspired by the previous study [36], we relax the strict binary matrix $\mathbf{Y}$ into a slack learned target matrix $\mathbf{G}$ by adding a marginalized constraint. $\mathbf{G} = [g_1, \ldots, g_c] \in \mathbb{R}^{n \times c}$ is the learned regression target matrix. Now we give a brief explanation to show how a strict binary label matrix into a slack learned target matrix. Let $x_{i,j}, y_{i,j}$ denote the features of three training sequences from three different folds, and the corresponding strict binary matrix is defined as $\mathbf{Y} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ (where the first, second, third columns represent the first, third, and second folds, respectively). According to the previous study [5], the distance between any two samples from different folds is $\sqrt{2}$. However, the sequences from different folds may have the specific property and the distances between two samples from two categories are different. We utilize a slack learned label matrix $\mathbf{G}$ instead of $\mathbf{Y}$. The assumption that the $j$-th sequence $x_j$ is from the $r$-th protein fold (i.e. $y_j = r$), the value of the $r$-th element of the learned target vector $g_j$ should be bigger than the rest elements by a constant $\varepsilon$. Specifically, the slack learned matrix $\mathbf{G}$ is represented as
**Fig. 1.** The flowchart of AWMG. AWMG method comprises three phases: feature extraction, training phase, and test phase. First, the protein sequences are embedded into feature matrices, which are constructed from various views, such as the HHblits, the SPARKS-X, and the DeepFR. Second, they are fed into the AWMG model to train the model. The Laplacian matrices (L) corresponding to each view are constructed based on Eq. 2 and 3. The weight of the query protein is predicted by the Eq. 5. Therefore, AWMG utilizes features from different views in the supervised framework.

### 2.2.2 Solution of the AWMG model

In this section, we describe the method used to solve the optimize AWMG model. An iterative method is presented here.

Step 1 (Update P): Assuming that the data sources from different views are not related, we consider the d-th view training samples $X_d^{(t)} \in \mathbb{R}^{m \times n}$. Fix the other variables $q_d, W, V, G$ and update P by solving the following problem,

$$
P^{(d)} = \arg \min_{P \in \mathbb{R}^{m \times n}} \left( \sum_{d=1}^{D} a_d^{(d)} P^{(d)} x_d^{(d)} - V^{(d)} \right) + \lambda \| W V - G \|_F^2
$$

where the rest term irrelevant to $P^{(d)}$ in Eq. 1 is viewed as the constant and ignore the loss since they make no differences in this particular procedure[36]. The problem (6) is a typical regularized least square problem, and the optimal $P^{(d)}$ can be obtained as,

$$
P^{(d)} = \left( VX_d^{(d)T} \right) \left( X_d^{(d)T} X_d^{(d)} + \lambda X_d^{(d)T} L X_d^{(d)} + \lambda I \right)^{-1}
$$

Step 2 (Update V): Fix the variable $a_d^{(d)}, W, V, G$ and update V by solving the following problem,

$$
V^+ = \arg \min_{V} \left( \sum_{d=1}^{D} a_d^{(d)} \| P^{(d)} x_d^{(d)} - V \|_F^2 + \lambda \| W V - G \|_F^2 \right)
$$

Similarity, the optimal $V$ can be obtained as,

$$
V^+ = \left( \sum_{d=1}^{D} a_d^{(d)} I + \lambda W^T W \right)^{-1} \left( \sum_{d=1}^{D} a_d^{(d)} P^{(d)} x_d^{(d)} \right) + \lambda W^T G
$$

Step 3 (Update W): Fix the variable $q_d, P^{(d)}, V, G$ and update W by solving the following problem,

$$
W^+ = \arg \min_{W} \frac{\lambda}{2} \| W V - G \|_F^2 + \lambda \| W \|_F^2
$$

Similarity, the optimal W can be obtained as,

$$
W^+ = \left( \lambda V V^T \right) \left( \lambda V V^T + \lambda I \right)^{-1}
$$

Step 4 (Update G): Fix the variable $a_d^{(d)}, P^{(d)}, W, V$ and let $Q = W V$. Update G by solving the following problem,

$$
G^+ = \arg \min_{G} \frac{\lambda}{2} \| Q - G \|_F^2 + \max_{i \neq j} \delta v_{ij} \geq 1
$$

According to [36], the optimal solution of G is obtained using the equation,

$$
g_{ij} = \begin{cases} q_{ij} + \delta, & \text{if } i = j, \\ q_{ij} + \min(\delta - \varphi_i), & \text{otherwise} \end{cases}
$$

In summary, the proposed method calculates the special auto-weight $a_d^{(d)}$ and the different transform matrix $P^{(d)}$ based on each view. $P^{(d)}$ is powerful for using the local nearest neighbor information of data. Finally, the query sample is predicted by $\arg \max_d \{ P^{(d)} x_d^{(d)} \}$ with $x_d^{(d)}$ is the feature vector of the query sample of the $d$-th view of the query sequence to predict the fold type. The scores correspond to different folds are calculated by accumulating the results from D views. Larger score denotes that the query sequence has a high likelihood to predict the corresponding fold type.

In the predictive phase, the predicted fold $y_j^{(d)}$ of the query sample $x_d^{(d)}$ with D views is calculated as,

$$
y_j^{(d)} = \max_d \{ P^{(d)} x_d^{(d)} \}
$$

where $P^{(d)} = W \left( \sum_{d=1}^{D} a_d^{(d)} P^{(d)} x_d^{(d)} \right) \in \mathbb{R}^r$, $x_d^{(d)}$ is the feature vector of the query sample of the $d$-th view $(d \in [1,...,D])$. The proposed method utilizes the information of different views of the query sequence to predict the fold type. The slack variable matrix provides more freedom to fit the discriminative regression target. Therefore, AWMG utilizes features from different views to construct the common latent subspace. The slack variable matrix provides more freedom to fit the regression target.
where \( \varphi \in \mathbb{R}^c \) is an auxiliary variable and for \( i \)-th element, \( \varphi = q_{ij} + 1 - q_{ij}. \) \( \delta \) is the learning factor [36].

Step 5 (Update \( a(d) \)): Fixing the variable \( P(d), W, V, G \) and update \( a(d) \) by solving following problem,

\[
a^*_d = \arg \min_{a_d} \mathbb{E}_{X \sim \mathbb{D}(d)} \left[ \sum_{i=1}^{l} \mathbf{P}(d_i) X_i(d_i)^2 \sum_{j=1}^{n} \mathbf{X}(d_i) X_i(d_i)X_j(d_i)X_j(d_i) \right] - 2 \mathbf{P}(d)
\]

Finally, we choose an iterative strategy to solve the primal problem. The pseudo codes of the AWMG methods are shown in Algorithm 1.

**Algorithm 1**

**The Pseudocode of the AWMG Algorithm**

**Input:** There are \( D \) views training data \( \{X_0^{(d)}, \ldots, X_n^{(d)}\} \), and each view has \( n \) data points, such as the points in the \( d \)-th view \( \{X_i^{(d)}\}_{i=1}^{l} \). The corresponding labels \( \{y_i\}_{i=1}^{l} \), A test sample of \( D \) views \( X_0 \), parameter \( \lambda_1, \lambda_2, \lambda_1, \ldots, \) and \( r \) in Eq. 1, maximum number of iterations \( N \).

**Output:** \( j \) - the predicted label.

1: Initialization: construct \( L, P^{(d)}, V, W, a(d) \).
2: Let \( d = 1 \)
3: while not converge do
4: while \( d \leq D \) do
5: Update \( P(d) \) by solving problem Eq. 7
6: end while
7: Update \( V \) by solving problem Eq. 9
8: Update \( W \) by solving problem Eq. 11
9: Update \( G \) by solving problem Eq. 13
10: Update \( a(d) \) by solving problem Eq. 14
11: end while
12: Output: Pojection fold type \( j \)

### 2.3 Multi-view feature representation

Protein fold recognition is a typical classification problem. AWMG method utilizes multiple views of protein sequences to predict fold types. In this section, we introduce three categories of data sources, which include the evolutionary information based on the HHblits, the pairwise alignment information based on the DeepFR and SPARKS-X, respectively.

#### 2.3.1 Representation based on the HHblits

ACC feature proposed by Dong et al. [21] is used to convert the HHblits profile to a fixed-length feature vector, whose the dimension is \( 400 \times LG \) (\( LG \) denotes the distance between two amino acids in the profile). HHblits is a HMM-HMM alignment algorithm which improves the alignmnet quality and the sensitivity/specificity of detection [10]. The profiles are searched against the UniProt20_2013_06 by HHblits with the parameter ‘-n 4’. According to the HHblits manual [10], the frequencies \( p_{ij} \) is calculated via the formula,

\[
p'_{ij} = 2^{-1000 p_{ij}}
\]

where \( p_{ij} \) is the element in the first 20 columns of profile. In this study, ACC is treated as the view of the HHblits.

#### 2.3.2 Representation based on the DeepFR

DeepFR extracts the fold-specific features from the contact map using the deep convolution neural network (DCNN). The contact map calculates the residue-residue likelihood matrix which is predicted by CCMpred [38]. The DeepFR score measures the similarity between the query sequence and benchmark sequence based on the fold-specific features [31]. We use the DeepFR score as the feature to measure the similarity between the query sequence and individual template sequence. In the training phase, the score is set to 0 when the template sequence aligns to itself. In this study, the pairwise DeepFR score similarity between the query-template represents the view of DeepFR.

#### 2.3.3 Representation based on the SPARKS-X

SPARKS-X searches the query sequence against the templates structural profiles based on the Protein Data Bank (PDB) and calculates the similarity probability using some structural properties, such as secondary structure, backbone torsion angle, etc. [16]. We calculate the pairwise similarity scores between the query sequence and training sequences. Instead of using the tool to search the template sequences, we utilize the Z-score associated with query-template fitness as the global feature. The function for running SPARKS-X program is main.intel [16]. In the training phase, the similarity score is set to 0 when the template sequence aligns to itself. In this study, the pairwise Z-score similarity between the query-template represents the view of SPARKS-X.

#### 2.4 An ensemble approach EMfold for protein fold recognition

The Ensemble method integrates multiple predictors to improve the predictive performance [19, 39-42]. In this section, we propose an ensemble learning method called EMfold that integrates AWMG and a template-based method called DeepSS, which linearly combines the probability generated by SPARKS-X and DeepFR. Because the SPARKS-X and DeepFR have different properties, the probability hits are different in the top hits in the training set. The Z-score scores in the SPARKS-X and DeepFR scores in the DeepFR measure the similarity between the query and the templates. That is, a higher score indicates the templates are more homologous to the query sequence. Assuming there are \( c \) possible fold types in the training set and the total number of templates is \( N \). The probability \( p_i \in \mathbb{R}^c \) denotes that the query sequence belongs to the \( i \)-th fold and is defined as follow,

\[
p_i = \frac{\sum_{j=1}^{n_i} s_j}{\sum_{j=1}^{n_i} \delta_j}, i \in [1, \ldots, c]
\]

where the \( s_j \) is the score for the \( j \)-th template in the \( i \)-th fold. \( n_i \) is the number of the templates belong to the \( i \)-th (\( l \)-th) fold. The probability \( P_i \in \mathbb{R}^c \) of the
DeepSS is the linear combination of SPARKS-X and DeepFR and is defined as follows,

\[ p_y = (1 - \omega) p_y^\text{SPARKS-X} + \omega p_y^\text{DeepFR} \]  

(17)

where \( p_y^\text{SPARKS-X} \) and \( p_y^\text{DeepFR} \) denote the probabilities of different folds types in the top \( N \) templates and are calculated by Eq. 16.

In order to make use of the advantages of the multi-view learning method and template-based method, the ensemble method EMfold is proposed here. AWMG calculates the predictive scores \( y'_{(i)} \in \mathbb{R}^v \) of the query sequence using the Eq. 5 and sorts the scores in the descending order, which are denoted as \( \{z_1, \ldots, z_v\} \). Then we calculate the distance \( z = z_i - z_1 \) to evaluate the discriminative ability of the predictive model. The higher \( z \) value is, the more precise the AWMG predict the query sequence. The EMfold method utilizes \( z \) score to decide the predictors and is defined as follows,

\[ \text{EMfold} = \begin{cases} \text{AWMG, if } z > T \\ \text{DeepSS, otherwise} \end{cases} \]  

(18)

when \( T \) is a cutoff threshold. Fig. 2 illustrates the flowchart of the EMfold model.

![Flowchart of EMfold](image)

2.5 Evaluation indices

The accuracy was used to evaluate the performance of various methods in the protein fold recognition. Accuracy is defined as the ratio of the number of the predicted protein correctly to the number of total query sequence using the following equation [21],

\[ \text{Accuracy} = \frac{CM}{M} \times 100\% \]  

(19)

where \( CM \) is the number of the query protein sequences which are predicted correctly, and \( M \) is the number of total query protein sequences.

3 RESULTS AND DISCUSSION

3.1 Determination of parameter and cross-validation

Two widely used benchmark datasets were used to evaluate the performance of different methods, including LE dataset and YK dataset. 2-fold and 3-fold cross-validation were used for LE dataset and YK dataset [5], respectively. The parameters were optimized in the validation dataset, which is independent with the test set.

There are two kinds of parameters in the AWMG method: the parameters associated with the multi-view data sources, and the parameters associated with the auto-weighted multi-view graph embedding learning model. In this study, the parameter \( LG \) associated with the HHblits profile and the parameters \( \lambda_2, \lambda_4, r, t \) associated with the predictor model were optimized on the LE dataset via the 2-fold cross-validation. The optimized values of some parameters are shown in Table 1, which were used in two benchmarks to reduce the risk of the over-fitting. The candidate values for parameters \( \lambda_1 \) and \( \lambda_3 \) were \( 10^{-1}, 10^{-4}, 10^{-3}, 10^{-2}, 5 \times 10^{-2}, 10^{-1}, 5 \times 10^{-1} \). The initial latent subspace \( V \) is obtained by calculating the average of initial subspaces of different views using PCA strategy. The value \( k \) of \( V \) chooses the minimum dimensionality of different initial subspaces.

Three additional parameters in the EMfold method were optimized on the LE dataset: the parameters \( N \) and \( \omega \) are associated with DeepSS and the parameter \( T \) for combining the AWMG and DeepSS. The optimized values are shown in Table 1. The impact of the parameters on the DeepSS and EMfold is shown in Fig. 3. The values of parameters are used for all benchmark dataset to avoid the over-fitting.

![Accuracy with different parameters](image)

![Values of parameters](image)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>( LG )</th>
<th>( \lambda_2 )</th>
<th>( \lambda_4 )</th>
<th>( r )</th>
<th>( t )</th>
<th>( N )</th>
<th>( w )</th>
<th>( T )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values</td>
<td>8</td>
<td>0.01</td>
<td>0.01</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>0.6</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Fig. 3. The accuracy with different parameters on the LE dataset.

(a) The accuracy of EMfold with different \( T \) thresholds. (b) The accuracy of DeepSS with different values of \( N \) and \( w \) on the LE dataset.
3.2 The performance and properties of AWMG

The AWMG utilizes three views to construct the predictor, including ACC, the pairwise similarity scores between the query-template sequences based on SPARKS-X and DeepFR. AWMG was compared with some basic classifiers (LibSVM with RBF kernel, KNN (K = 1), Random Forest (RF), and Collaborative Representation Classification (CRC)) to show the performance of the AWMG model. We combine the features of three views with the same parameters linearly, and then fed the combination feature into the traditional predictors. The performance of different methods is shown in Table 2.

**Table 2**

<table>
<thead>
<tr>
<th>Methods</th>
<th>AWMG</th>
<th>RF</th>
<th>KNN(δ=1)</th>
<th>LibSVM</th>
<th>CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>LE</td>
<td>67%</td>
<td>51.30%</td>
<td>64.76%</td>
<td>65.75%</td>
<td>65.40%</td>
</tr>
<tr>
<td>YK</td>
<td>50.44%</td>
<td>50.19%</td>
<td>45.10%</td>
<td>50.31%</td>
<td>40.31%</td>
</tr>
</tbody>
</table>

* The predictor model of DeepFR is trained based on the Scop 2.06 dataset. We cannot guarantee that the similarity identity between YK dataset and training dataset from DeepFR is lower than 40%. Therefore, AWMG utilizes two views information to predict the protein fold, including the ACC of HHblits profiles and the similarity score of SPARKS-X.

The results in Table 2 show that the proposed method AWMG outperforms other predictors, indicating that the AWMG is effective for protein fold recognition. Compared with the traditional predictors using the combination features, AWMG calculates the auto-weight corresponding to the three views and constitutes the latent common subspace to represent the information of three views. The common information is critical for predictive performance. Compared with the CRC method which utilizes the linear regression model, AWMG relaxes the regression target of different categories by enlarging the margin distance of different folds. Therefore, AWMG outperforms some traditional predictors.

Because the protein sequences from different folds have the special property, we choose the nearest neighbour sequences to construct the Laplacian matrix \( L^{(d)} \) of each data source. Then the \( L^{(d)} \) obtains the local neighbour information of different folds. Fig. 4 shows the \( L^{(d)} \) of the special view captured by the proposed method AWMG. We select some sequences from six folds on the LE dataset and calculate the \( L^{(d)} \) based on the DeepFR. The values in the \( L^{(d)} \) denote the relationship between different sequences in the training set. Large scores show that the corresponding pairwise sequences have a high probability in the nearest neighbors. The red square denotes the pairwise sequences in the training dataset belong to the same fold. The most nonzero elements in the \( L^{(d)} \) in Fig. 4 are from the same fold. This indicates that using the local nearest neighbour as prior knowledge is beneficial to obtain a discriminative performance for protein fold recognition.

The latent subspace \( V \) represents the common information shared by three data sources. Then the query sequence is predicted by the information of latent subspace \( V \) substituted for three data sources. To investigate the performance of different data sources, we run d3heatmap on the features of 27 sequences in the LE dataset (cover 5 different folds). The cluster results from the latent subspace and three original views are shown in the Fig. 5 and Supplementary Fig. S1. The features in Fig. 5 (a) form several blocks, suggesting that proteins in the same fold commonly have similar fold specific features in the latent subspace. The constructed cluster trees are consistent with the fold type hierarchy. For example, the left blocks suggest that proteins in the same fold ‘1_4’ have high values for the corresponding features, such as features 2, 9. As shown in Fig. 5 (b-d), there are some sequences with error cluster fold categories which are denoted as the black cross symbol in the clustering tree. Therefore, latent subspace \( V \) utilizes more effective information for protein fold recognition.

In our experiments, the proposed method utilizes the local nearest neighbour information by the term Laplacian matrix and obtains the latent subspace shared by three different data sources. Then the proposed method achieves the more discriminative regression target by the marginalized constraint. The results show that AWMG method is effective for protein fold recognition.

![Fig. 4. The Laplacian matrix of the special view on the LE dataset.](image)

![Fig. 5. Analysis of the latent common subspace feature and the three different view features by running d3heatmap with R package over 27 proteins (cover 5 folds) selected from LE dataset. (a) The feature represents the latent subspace information. (b) The feature represents the ACC evolutionary information extracted from HHblits profile view. (c) The feature represents the Zscore information extracted from SPARKS-X view. (d) The feature represents the DeepFR score information extracted from DeepFR view. In the (a), the clustering trees are consistent with fold hierarchy. In the (b-d), the sequences with error clustering fold categories are represented.](image)
in the black cross symbol in the clustering tree.

3.3 The performance of EMfold

As discussed above, AWMG method improves the performance of the protein fold recognition. The template-based DeepSS method combines SPARKS-X and DeepFR. The scores from the DeepSS are used to measure the homologous between the templates and query sequence. We further investigated if these two methods are complementary or not. The pairwise comparisons between these two algorithms are shown in Fig. 6. It can be shown that these two methods are complementary. The reason is that DeepSS is sensitive to detect the homologous template protein sequences, while the homology template sequence has little impact on the AWMG model. Furthermore, AWMG constructs the latent subspace shared by three different views features sourced from the protein sequences rather than on known structural homologous between the templates and query sequence. Therefore, we propose an ensemble model to combine those two methods. The performance of the protein fold recognition. The template-based DeepSS method combines SPARKS-X and DeepFR. The scores from the DeepSS are used to measure the homologous between the templates and query sequence. Therefore, we propose an ensemble model to combine those two methods. The results of EMfold on the two datasets were shown in Table 3, from which we can see that EMfold obviously outperforms AWMG and DeepSS method on the two benchmark datasets.

![Fig. 6. Pairwise comparison between AWMG and DeepSS in terms of Accuracy (cf. Eq. 19) of each fold on the LE and YK dataset.](image)

**Table 3**

<table>
<thead>
<tr>
<th>Methods</th>
<th>LE</th>
<th>YK</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWMG</td>
<td>67%</td>
<td>50.44%*</td>
</tr>
<tr>
<td>DeepSS</td>
<td>72%</td>
<td>48.48%*</td>
</tr>
<tr>
<td>EMfold</td>
<td>73.9%</td>
<td>53.03%*</td>
</tr>
</tbody>
</table>

*: The predictor model of DeepFR is trained based on the Scop 2.06 dataset. We cannot guarantee that the similarity identity between YK dataset and training dataset from DeepFR is lower than 40%. Therefore, AWMG method utilizes two views information to predict the protein fold, including the ACC of HHMbits profiles and the similarity score of SPARKS-X. EMfold method sets $\omega=0$ on the YK dataset.

3.4 Comparison with other state-of-the-art methods

To demonstrate the effectiveness of AWMG and EMfold, its performance was directly compared with other state-of-the-art methods for fold recognition, including the HMMFold [43], ACC-fold [21], MV-fold [5], DN-Fold [44], RNDN-Fold [44], RF-fold [45], DeepFR [31], Taxfold [46], FOLDpro [47], dRHP-PseRA [48], HHpred [49], FFAS-3D [50], SPARKS-X [16], HH-fold [51], DeepFRpro [31], TA-fold [51], MT-fold [5], and DeepSVM-fold [22]. The performance of these methods was listed in the Table 4, Table 5, and Fig. 7.

The experimental results show that EMfold outperforms all the other state-of-the-art methods. AWMG exhibits better performance than the MV-fold method, which utilizes the multi-view learning framework based on the latent subspace. The proposed method utilizes the Laplacian matrix to obtain the local nearest neighbor information of different folds. The EMfold method combines the advantage of the AWMG and DeepSS. Therefore, EMfold is better than AWMG. The results confirm that AWMG and EMfold are very useful predictors for protein fold recognition.

![Fig. 7. The Accuracy (cf. Eq. 19) of SVM-fold, HMMfold, MV-fold, ACC-fold and AWMG of each fold on the LE dataset.](image)

**Table 4**

<table>
<thead>
<tr>
<th>Methods</th>
<th>Accuracy</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMMFold</td>
<td>45.5%</td>
<td>[43]</td>
</tr>
<tr>
<td>ACC-fold</td>
<td>29.9%</td>
<td>[21]</td>
</tr>
<tr>
<td>MV-fold</td>
<td>46.6%</td>
<td>[5]</td>
</tr>
<tr>
<td>DN-Fold</td>
<td>33.6%</td>
<td>[44]</td>
</tr>
<tr>
<td>RNDN-Fold</td>
<td>37.7%</td>
<td>[44]</td>
</tr>
<tr>
<td>RF-fold</td>
<td>40.8%</td>
<td>[45]</td>
</tr>
<tr>
<td>DeepFR</td>
<td>56.1%</td>
<td>[31]</td>
</tr>
<tr>
<td>Taxfold</td>
<td>40.6%</td>
<td>[46]</td>
</tr>
<tr>
<td>FOLDpro</td>
<td>26.5%</td>
<td>[47]</td>
</tr>
<tr>
<td>dRHP-PseRA</td>
<td>34.9%</td>
<td>[48]</td>
</tr>
<tr>
<td>HHpred</td>
<td>25.2%</td>
<td>[49]</td>
</tr>
<tr>
<td>FFAS-3D</td>
<td>33.8%</td>
<td>[50]</td>
</tr>
<tr>
<td>SPARKS-X</td>
<td>45.2%</td>
<td>[16]</td>
</tr>
<tr>
<td>HH-fold</td>
<td>42.1%</td>
<td>[51]</td>
</tr>
<tr>
<td>DeepFRpro</td>
<td>66%</td>
<td>[31]</td>
</tr>
<tr>
<td>TA-fold</td>
<td>53.9%</td>
<td>[51]</td>
</tr>
<tr>
<td>MT-fold</td>
<td>54.1%</td>
<td>[5]</td>
</tr>
<tr>
<td>DeepSVM-fold</td>
<td>67.3%</td>
<td>[22]</td>
</tr>
<tr>
<td>AWMG</td>
<td>67%</td>
<td>This study</td>
</tr>
</tbody>
</table>
4. CONCLUSION

Protein fold recognition is essential for understanding protein structures and protein-protein interactions. In this study, we introduce two novel predictors: AWMG and EMfold. AWMG utilizes the linear discriminative regression framework based on the auto-weighted multi-view learning model. EMfold combines AWMG and DeepSS methods. AWMG calculates the weight for each view automatically and constructs the latent subspace shared by three multi-view representations. Then the query sequences are obtained from the latent subspace and their protein folds are predicted by the transformation matrix. Unlike the convention multi-view learning method, AWMG utilizes the Laplacian matrix to obtain the local nearest neighbour information for different folds and applies the marginalized constraint by enlarging the margins between different folds. As an ensemble method, EMfold outperforms AWMG. Experimental results showed that the methods improve the performance of protein fold recognition.

As shown in previous study [52], remote homology detection is important task in protein sequence analysis. Because the AWMG is able to extract several features from protein sequences for binary or multiple classification tasks in bioinformatics, protein remote homology detection would be a potential application area of AWMG algorithm. Our future study will focus on exploring discriminative features and advanced predictors for protein fold recognition.

ACKNOWLEDGMENT

The authors are very much indebted to the three anonymous reviewers, whose constructive comments are very helpful in strengthening the presentation of this article. This work was supported by the National Natural Science Foundation of China (No. 61822306, 61772184), the Beijing Natural Science Foundation (No. JQ19019), Fok Ying-Tung Education Foundation for Young Teachers in the Higher Education Institutions of China (161063), Scientific Research Foundation in Shenzhen (Grant No. JCYJ20150621104225228, JCYJ20170307152201596, JCYJ20180306172207179).

References

Combining PageRank and Hyperlink-Induced Topic Search," *Briefings in Bioinformatics*, vol. DOI: 10.1093/bib/bby104.


Ke Yan received the B.S. degree in electronic information scientific and information from QuFu Normal University, China, in 2009, the M.S. degree in computer science and technology from the Shenzhen Graduate School, Harbin Institute of Technology University, China, in 2013. He is currently pursuing the Ph.D. degree in computer science and technology with the Harbin Institute of Technology, Shenzhen, China. His research interests include pattern recognition, machine learning and bioinformatics.

Jie Wen received the M.S. degree at Harbin Engineering University, China in 2015. He is currently pursuing the Ph.D. degree in computer science and technology, Harbin Institute of Technology, Shenzhen, China. His research interests include, image and video enhancement, pattern recognition and machine learning.

Yong Xu was born in Sichuan, China, in 1972. He received the B.S. and M.S. degree in 1994 and 1997, respectively, and the Ph.D. degree in pattern recognition and intelligence system with Nanjing University of Science and Technology, Nanjing, China, in 2005. He is currently with the Harbin Institute of Technology, Shenzhen, Shenzhen, China. His current interests include pattern recognition, biometrics, machine learning, and video analysis.

Bin Liu received Ph.D. degree from Harbin Institute of Technology, China in 2010. From 2010 to 2012, he was a post-doctoral researcher at The Ohio State University, USA. He worked at Harbin Institute of Technology Shenzhen as a Professor from 2012 to 2019. Now, he is working at Beijing Institute of Technology as a Professor since May 2019. His research interesting includes bioinformatics, machine learning, natural language processing, etc. Now he is putting the focus on exploring the language models of biological sequences, and proposing computational predictors for some important tasks in bioinformatics based on natural language processing techniques. He has published more than 60 SCI papers, including Bioinformatics, Briefings in Bioinformatics, Nucleic Acids Research, BMC Bioinformatics, etc.