Bioinformatics, 33(23), 2017, 3749–3757
doi: 10.1093/bioinformatics/btx514
Advance Access Publication Date: 16 August 2017
Original Paper

OXFORD

Downloaded from https://academic.oup.com/bioinformatics/article/33/23/3749/4083263 by Shanghai Jiao Tong University School of Medicine user on 30 November 2025

Structural bioinformatics

Improving protein fold recognition by extracting fold-specific features from predicted residue–residue contacts

Jianwei Zhu[1,2], Haicang Zhang[1], Shuai Cheng Li[3], Chao Wang[1], Lupeng Kong[1,2], Shiwei Sun[1], Wei-Mou Zheng[4,*] and Dongbo Bu[1,*]

[1]Key Lab of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China, [2]University of Chinese Academy of Sciences, Beijing 100049, China, [3]Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong and [4]Institute of Theoretical Physics, Chinese Academy of Sciences, Beijing 100190, China

*To whom correspondence should be addressed.
Associate Editor: Alfonso Valencia

Received on January 23, 2017; revised on July 11, 2017; editorial decision on August 8, 2017; accepted on August 9, 2017

Abstract

Motivation: Accurate recognition of protein fold types is a key step for template-based prediction of protein structures. The existing approaches to fold recognition mainly exploit the features derived from alignments of query protein against templates. These approaches have been shown to be successful for fold recognition at family level, but usually failed at superfamily/fold levels. To overcome this limitation, one of the key points is to explore more structurally informative features of proteins. Although residue–residue contacts carry abundant structural information, how to thoroughly exploit these information for fold recognition still remains a challenge.

Results: In this study, we present an approach (called DeepFR) to improve fold recognition at superfamily/fold levels. The basic idea of our approach is to extract fold-specific features from predicted residue–residue contacts of proteins using deep convolutional neural network (DCNN) technique. Based on these fold-specific features, we calculated similarity between query protein and templates, and then assigned query protein with fold type of the most similar template. DCNN has showed excellent performance in image feature extraction and image recognition; the rational underlying the application of DCNN for fold recognition is that contact likelihood maps are essentially analogy to images, as they both display compositional hierarchy. Experimental results on the LINDAHL dataset suggest that even using the extracted fold-specific features alone, our approach achieved success rate comparable to the state-of-the-art approaches. When further combining these features with traditional alignment-related features, the success rate of our approach increased to 92.3%, 82.5% and 78.8% at family, superfamily and fold levels, respectively, which is about 18% higher than the state-of-the-art approach at fold level, 6% higher at superfamily level and 1% higher at family level. An independent assessment on SCOP_TEST dataset showed consistent performance improvement, indicating robustness of our approach. Furthermore, bi-clustering results of the extracted features are compatible with fold hierarchy of proteins, implying that these features are fold-specific. Together, these results suggest that the features extracted from predicted contacts are orthogonal to alignment-related features, and the combination of them could greatly facilitate fold recognition at superfamily/fold levels and template-based prediction of protein structures.

Availability and implementation: Source code of DeepFR is freely available through https://github.com/zhujianwei31415/deepfr, and a web server is available through http://protein.ict.ac.cn/deepfr.

© The Author 2017. Published by Oxford University Press. All rights reserved. For Permissions, please e-mail: journals.permissions@oup.com

Contact: zheng@itp.ac.cn or dbu@ict.ac.cn
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Proteins are large molecules consisting of one or more chains of amino acid residues, and play crucial functions in a wide range of biological processes. The functional properties of proteins are largely determined by their three dimensional structures (called *tertiary structures*), making it vitally important to deduce or predict protein structures from amino acid sequences (Branden *et al.*, 1999). The widely used experimental technologies to deduce protein structures, such as X-ray crystallography, NMR spectroscopy and electron microscopy, have achieved great success; however, these technologies usually cost considerable time and thus cannot keep up with the fast process to acquire protein sequences (Berman *et al.*, 2000; Bairoch *et al.*, 2005). An alternative strategy is to predict protein structures from amino acid sequence, which can be divided into template-based (Ma *et al.*, 2012, 2014; Roy *et al.*, 2010; Wang *et al.*, 2016; Yang *et al.*, 2011) and *ab initio* prediction approaches (Li *et al.*, 2008; Simons *et al.*, 1997; Xu and Zhang, 2012).

During the whole procedure of protein structure prediction, recognizing the templates with similar structure to query protein, known as *fold recognition*, is an important and the first step (Jones *et al.*, 1992; Lindahl and Elofsson, 2000). The key point of fold recognition is to define and calculate the similarity between query protein and templates with known structures. The widely used strategies for fold recognition include: (i) calculating alignment between query protein and templates based on sequence-sequence similarity (Altschul *et al.*, 1990; Eddy, 1998; Pearson and Lipman, 1988; Söding, 2005), or sequence-structure compatibility (also known as *threading*) (Peng and Xu, 2009; Shi *et al.*, 2001; Xu *et al.*, 2003); (ii) designing a binary classifier to decide whether query protein is similar to templates based on specially designed features (Cheng and Baldi, 2006; Cheung *et al.*, 2016; Dong *et al.*, 2009; Jo and Cheng, 2014; Jo *et al.*, 2015); and (iii) adopting the consensus strategy to integrate multiple recognition approaches (Fischer, 2003; Lundström *et al.*, 2001; Xia *et al.*, 2016). These strategies mainly rely on sequence-related features and thus show excellent performance for fold recognition at family level. However, proteins in the same superfamily or fold usually show weak sequence similarities, making fold recognition at superfamily/fold levels a challenging task. This challenge emphasized the importance to exploit more structurally informative features for fold recognition.

Residue–residue contacts present essential information of protein tertiary structures. Specifically, protein structures can be treated as result of the combination of local interactions and non-local interactions among residues, where local interactions lead to the formation of local structural motifs (e.g. secondary structural elements), and non-local interactions guide the arrangement of these motifs (Li *et al.*, 2008; Simons *et al.*, 1997; Zheng, 2014). According to the SCOP hierarchy, proteins in the same fold share similar arrangement or topological connections of secondary structural elements, and protein in the same superfamily share similar tertiary structures (Murzin *et al.*, 1995). Although the sequence similarity among these proteins might be relatively weak, the patterns of residue–residue contacts provide possibility to distinguish different superfamily or folds (Andreani and Söding, 2015; Skwark *et al.*, 2014).

This study presents such an effort to extract fold-specific features from predicted residue–residue contacts to improve fold recognition at superfamily/fold levels. Some ideas of this work come from two lines of research: predicting residue–residue contacts using evolutionary coupling analysis, and deep learning technique to extract fold-specific features and calculate similarity of contact likelihood matrices.

First, the multiple sequence alignment (MSA) of a query protein records its evolutionary history, where co-evolutionary events might occur among certain columns of MSA due mainly to contacts between corresponding residues. These co-evolutionary events, in turn, have been exploited to infer residue–residue contacts, generating contact likelihood maps as results (Göbel *et al.*, 1994; Olmea and Valencia, 1997). As transitive effects among interacting residues lead to considerable false-positive inference of contacts, direct evolutionary coupling (EC) analysis has been developed to reduce these transitive effects and improve contact prediction (Ekeberg *et al.*, 2013; Jones *et al.*, 2012; Kamisetty *et al.*, 2013; Morcos *et al.*, 2011; Skolnick *et al.*, 1997) and protein structure prediction (Adhikari *et al.*, 2015; Kim *et al.*, 2014; Wang *et al.*, 2017).

Second, deep convolutional neural network (DCNN) has proven to be successful for image retrieval (Krizhevsky *et al.*, 2012; LeCun *et al.*, 1998), and face recognition (Schroff *et al.*, 2015). One of the advantages of DCNN lies at its ability to automatically extract compositionally hierarchical features from images. For example, in images of certain objects, local features such as edges form motifs, motifs construct parts and parts assemble into objects (LeCun *et al.*, 2015). Similar hierarchy also exists in protein structures and residue–residue contacts: under local interactions, residues form local structural motifs, and these local structural motifs are packed to form full structure under long-range interactions (Haspel *et al.*, 2003; Simons *et al.*, 1997; Taylor, 2016). This similarity implies the analog of contact maps of proteins to images of certain objects. Thus, the problem of inferring fold type for a protein from its contact map turns into the problem of image retrieval, making it reasonable to apply the DCNN technique for fold recognition. Recently, Wang *et al.* (2017) successfully applied DCNN to remove noises from EC contact matrices, which also implies the possibility to apply DCNN for fold recognition.

It is worth noting that in previous studies, contact information has been explored for fold recognition in the form of contact order, contact number and average contact probabilities of certain residue pairs in query protein (Cheng and Baldi, 2006). However, these approaches cannot thoroughly extract fold-specific information from contacts, which was confirmed by experiments to be shown in subsequent sections. Thus, a more effective way to explore fold-specific information from residue contacts was expected.

Combining these two lines of research, we proposed an approach called DeepFR that applies deep learning technique to extract fold-specific features from predicted contacts for fold recognition. The evaluation on two benchmark sets suggested that by using these extracted features, DeepFR outperformed the state-of-the-art approaches at superfamily/fold levels. Bi-clustering analysis validated that the features are fold-specific. The similarity of predicted contact maps between proteins are orthogonal to traditional features derived from alignments, and the combination of them could greatly facilitate accurate fold recognition as well as protein structure prediction.

Downloaded from https://academic.oup.com/bioinformatics/article/33/23/3749/4083263 by Shanghai Jiao Tong University School of Medicine user on 30 November 2025

2 Materials and methods

The paradigm of our approach DeepFR is shown in Figure 1, which consists of the following three steps: (i) *Contact prediction*: For a query protein, we first predicted its contact map through direct EC analysis of its MSA, yielding contact likelihood matrix as result; (ii) Fold-specific *features extraction*: As contact likelihood matrix usually contains a large amount of noises, we fed it into a pre-trained DCNN to extract fold-specific features, forming a EC-feature vector with fixed-size; (iii) *Fold recognition*: Finally, the EC-feature vector of query protein was compared against those of templates with known fold type using a binary classifier, and the generated similarity scores were used to assign fold types for query protein. The details of these three steps are described as follows.

2.1 Predicting residue–residue contacts for query protein

To predict residue–residue contacts for a query protein, we first collected homologs to the query protein. These similar proteins were collected through running HHblits (Remmert *et al.*, 2012) to search against sequence database `uniprot20_2016_02`, and were then organized as a MSA of the query protein. Next, co-evolutionary events among residues were analyzed by running CCMpred (Seemayer *et al.*, 2014) over the MSA without any filtering, assigning each residue–residue pair with a contact likelihood. The predicted contact likelihood was represented in the form of EC matrix. Currently a variety of methods, such as mfDCA (Marks *et al.*, 2011), PSICOV (Jones *et al.*, 2012) and plmDCA (Ekeberg *et al.*, 2014), are available for evolutionary coupling analysis. Here we chose CCMpred due to its high accuracy and its suitability for GPU platform.

2.2 Extracting fold-specific features from EC matrices

The acquired EC matrix, however, cannot be directly used to infer fold type of query protein due to the following reasons: (i) The relationship between residue–residue contacts and fold types are not well understood. (ii) While EC matrices contain necessary contact information, they also contain a large amount of noises (Dunn *et al.*, 2008; Zhang *et al.*, 2016). (iii) EC matrices of different proteins usually have different size, making it difficult to directly compare them. Thus, how to extract the information of fold types from EC matrices and represent this information as fixed-length feature vectors still remain challenges to fold recognition.

To achieve this goal, we extracted fold-specific features from EC matrices using deep convolutional neural network. Specifically, the DCNN takes predicted EC matrix of a protein as input, and outputs fold type of the protein. We trained a DCNN over a collection of training samples, each sample consisting of EC matrix of a protein together with its fold type as label. The training process aims to maximize the likelihood of the correct labels, which was accomplished through minimizing the cross-entropy loss for each training sample.

Figure 2 shows the architecture of the DCNN used in this study, which includes five convolutional layers, three max-pooling layers and three fully connected layers. The layers are described in more details as follows.

- *Convolutional layer:* A convolutional layer contains multiple *feature maps*, which were designed to capture local patterns of residue–residue contacts. Each element of a feature map is connected with local patches of the previous layer through weighting and non-linear transformation. All elements in a feature map share the same weights; thus, a local pattern, even if appearing at different locations of EC matrices, can also be detected.
- *Max-pooling layer:* In a certain local pattern, the relative position of its components might vary somewhat. To coarse-grain the position of these components, we introduced three max-pooling layers, and thus increased the reliability of detecting local patterns.
- *Fully connected layer:* The fully connected layers were introduced to learn non-linear combination of local patterns reported by convolutional layers or max-pooling layers. Unlike the convolutional layers, each neuron in a fully connected layer is connected to all neurons in previous layer. Here, the three fully connected layers in our DCNN, denoted as `fc6`, `fc7` and `fc8`, have 2048, 1024 and 1221 neurons, respectively. The final fully connected layer `fc8` was designed as a 1221-way softmax function, and each way corresponds to one of the total of 1221 fold types. This way, when feeding the DCNN with contact maps predicted for a protein, the output of `fc8` gives the probability for this protein to have specific fold types.

Our DCNN used two additional techniques to increase accuracy and efficiency. Specifically, batch normalization (Ioffe and Szegedy, 2015) was employed to accelerate training by reducing internal covariate shift, and the dropout technique (Srivastava *et al.*, 2014) was used to improve performance by reducing overfitting.

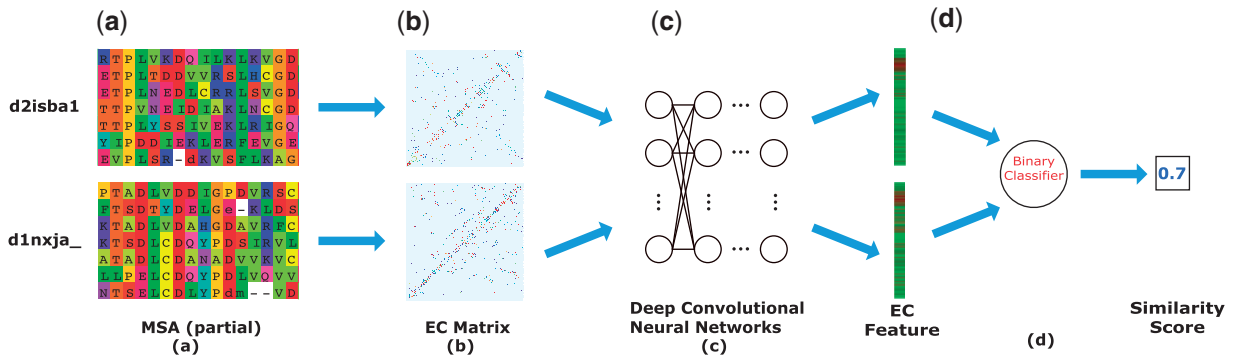

Fig. 1. Paradigm of our fold recognition approaches DeepFR and DeepFRpro. Here, two proteins `d2isba1` and `d1nxja_` are used as examples to explain the procedure to calculate similarity of their fold types. (**a**) Initially, multiple sequence alignment was built for each protein. (**b**) Residue–residue contacts were predicted through directly evolutionary coupling analysis of these alignments. (**c**) From these contact likelihood matrices, fold-specific features were extracted by using a deep convolutional neural network. (**d**) The extracted fold-specific feature vectors were fed into a binary classifier (random forest here) to calculate a score to infer structural similarity between the two proteins

Downloaded from https://academic.oup.com/bioinformatics/article/33/23/3749/4083263 by Shanghai Jiao Tong University School of Medicine user on 30 November 2025

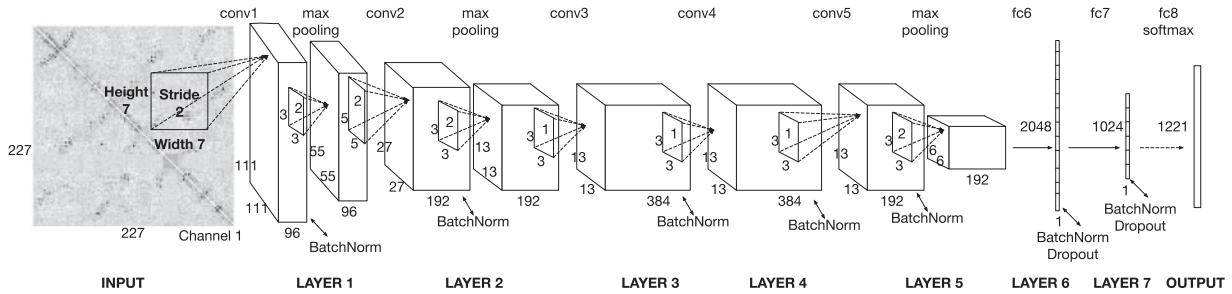

Fig. 2. Architecture of the deep convolutional neural network used to extract fold-specific features from contact likelihood matrix. Here, a 227 × 227 crop of a contact likelihood matrix (with size 256 × 256 and one channel) is presented as the input. This is convolved with 96 different 1st layer filters, each of size 7 × 7, using a stride of 2 in both *x* and *y* axis. Next, the resulting feature maps (i) passed through a rectified linear function (ReLU, not shown), (ii) pooled (max within 3 × 3 regions, using stride 2) and (iii) batch normalized across feature maps to give 96 different 55 × 55 element feature maps. Similar operations are repeated in layers 2 to 5, but the 3rd and 4th layer are not pooled. The next two layers are fully connected, taking features from the top convolutional layer as input in vector form ($6 \times 6 \times 192 = 6912$ dimensions) with batch normalized and dropout applied. The final layer is a 1221-way softmax function, 1221 being the number of fold types. Here, all filters and feature maps are shown as squares

After training the DCNN over a training set, the output of the second fully connected layer `fc7` were used as fold-specific features (denoted as *EC features*) of the input protein for subsequent fold recognition. The underlying rational of this strategy are: (i) In DCNN, higher-level features are combinations of lower-level ones, making it suitable to describe compositionally hierarchical signals. Thus, the high-level layer, say `fc7` in this study, is able to describe correlations among local patterns captured in distant parts of the matrix (Taigman *et al.*, 2014). (ii) More importantly, when the final layer `fc8` shows significantly high training and validation accuracy (Supplementary Table S4), it would be reasonable to deduce that `fc7`, the input of `fc8`, are perfect representations of fold-specific features. Experience in the field of image retrieval suggested that compared with handcrafted features, this off-the-shelf feature extractor often yields better results (Sharif Razavian *et al.*, 2014).

The DCNN used in this study takes 256 × 256 matrix as input and thus has a fixed-size input length; however, different proteins usually have different length of amino acid residues. To make the DCNN with fixed-size input applicable to other proteins with larger or smaller sizes, we adopted sampling or padding operations accordingly, which has proven to be successful in the field of image retrieval (Taigman *et al.*, 2014; Zeiler and Fergus, 2014) (further details in Supplementary Text). The performance of DeepFR/DeepFRpro using different sampling/padding strategies was shown in Supplementary Tables S1, S2 and S3.

2.3 Assigning fold type to query protein

To assign fold type for a query protein, we first compared its fold-specific features (represented as EC-feature vector) against those of templates with known fold types, and then assigned it with fold type of the most similar template. Two similarity measures were examined in this study as follows.

- *DeepFR score*: Given two EC-feature vectors f_1 and f_2, DeepFR score is defined as cosine value of them, i.e.

$$\text{DeepFR score} = \frac{f_1 \cdot f_2}{||f_1||||f_2||}$$

- *DeepFRpro score*: We further combined DeepFR score with a total of 84 similarity measures used in RF-fold, including secondary structure preference, sequence similarity and alignment-related features (Jo and Cheng, 2014). These measures were transformed into a similarity score (called DeepFRpro score) using the random forest technique (Breiman *et al.*, 1984; Breiman, 2001; Pedregosa *et al.*, 2011). The random forest used in this study contains 500 pre-trained decision trees, and each decision tree is a binary classifier trained to determine whether the query protein has the same fold type to the template. Next, the decision trees with positive output were counted, giving the DeepFRpro score.

The whole algorithm was implemented using `caffe` (Jia *et al.*, 2014) running on a GPU card (NVIDIA GTX 1070, 8G). The DCNN was trained using the stochastic gradient descent technique (Bottou, 2012) together with backpropagation method (LeCun *et al.*, 1989; Rumelhart *et al.*, 1988).

2.4 Datasets and evaluation strategies

2.4.1 Datasets

- *Training dataset:* In this study, the latest SCOP dataset (version 2.06, Fox *et al.*, 2014; Murzin *et al.*, 1995) was used to train the DCNN. To guarantee the independence between training dataset and test dataset, the training set was cleaned to remove the proteins that have significant sequence similarity with proteins in test set.

  We reduced the sequence redundancy in the training set by running the combination of tools for calculating sequence similarity, including CD-HIT(Li and Godzik, 2006), BLAST(Li and Godzik, 2006) and MMseqs2 (Hauser *et al.*, 2016; Steinegger and Söding, 2016). Specifically, we executed MMseqs2 (sequence identity: 25%, e-value: 10^{-4}), CD-HIT (sequence identity: 40%) and BLAST (e-value: 10^{-4}) to identify and thereafter remove the proteins in training set similar to those in test set. In this way, no protein in the generated training set shows significant sequence similarity to those in test set (e-value $< 10^{-4}$, Pearson, 2013).

  Furthermore, redundant proteins within training set were also removed to avoid potential over-fitting (filtering criteria: CD-HIT sequence identity $> 95\%$). After these filtering operations, we obtained a training dataset consists of 16133 proteins covering 1154 folds, 1862 superfamilies and 4344 families (further details in Supplementary Text). We also examined other criteria to reduce sequence redundancy and generated different training sets. The performance using these training sets was presented in Supplementary Tables S2 and S3.
- *Test dataset:* We evaluated our approach on two test datasets, namely LINDAHL (Lindahl and Elofsson, 2000) and SCOP_TEST (Jo *et al.*, 2015). The LINDAHL dataset consists of 976 proteins extracted from SCOP (version 1.37) with pairwise

Downloaded from https://academic.oup.com/bioinformatics/article/33/23/3749/4083263 by Shanghai Jiao Tong University School of Medicine user on 30 November 2025

sequence identity less than 40%. In this dataset, 555, 434 and 321 proteins have at least one match at family, superfamily and fold levels, respectively.

The SCOP_TEST dataset consists of 124 protein domains extracted from SCOP (version 1.75), and 106, 56 and 36 of them have at least one match at the family, superfamily and fold levels, respectively. It should be pointed out that the sequence identity between proteins in these two test sets are less than 40%, which enables independent and thorough evaluation of fold recognition approaches.

All of the datasets used in this study, together with the scripts used to generate the training sets, are available from our website http://protein.ict.ac.cn/deepfr/evaluation_data/.

2.4.2 Evaluation strategies

We used two different criteria to evaluate the performance of fold recognition. First, we examined the fraction of true hits in top 1 or top 5 predictions reported by fold recognition approaches. Specifically, we used each protein in test set as query protein, compared it with all other proteins (treated as templates), and ranked them according to the score reported by fold recognition approaches. The sensitivity for top 1 (top 5) predictions, also called success rate of fold recognition, is defined as the percentage of target proteins having at least one correct template ranked as top 1 (top 5). Second, the specificity-sensitivity plots (also called precision-recall plots) were drawn to show more details as performed in previous studies (Lindahl and Elofsson, 2000; Jo *et al.*, 2015; Shi *et al.*, 2001).

To evaluate DeepFRpro, we performed 10-fold cross-validation since DeepFRpro relies on a random forest classifier. Specifically, the LINDAHL dataset was partitioned into 10 subsets, 9 out of which were used to train the random forest classifier, and the remaining subset was used for validation. When evaluating DeepFRpro on the SCOP_TEST dataset, we adopted the strategy used by Cheng and Baldi (2006), i.e. the random forest classifier was trained using the whole LINDAHL dataset.

3 Results and discussion

We first evaluated fold recognition approaches in terms of success rate and specificity-sensitivity plots. Next, we analyzed the source of distinguishing power of the extracted EC-features. Finally, we investigated whether the extracted EC-features could be used to improve threading results.

3.1 Evaluating success rate of fold recognition

The success rate (sensitivity) of our approaches DeepFR and DeepFRpro, together with other widely used approaches, are shown in Table 1, where the results of the other approaches are taken from corresponding papers. As different MSAs were used in these approaches, it might be unfair for the approaches using older MSAs. To make a fair comparison, we run five fold-recognition tools locally with identical MSAs to our approaches, among which MRFalign (Ma *et al.*, 2014) and map_align (Ovchinnikov *et al.*, 2017) also exploit EC information of proteins. The comparison of these approaches on identical MSAs is shown in Table 2.

We first investigated whether DeepFR score alone can be used for fold recognition. As shown in Tables 1 and 2, DeepFR significantly outperformed all the other methods at the fold level. Specifically, the sensitivity of DeepFR for top 1 and top 5 predictions are 56.1% and 70.1%, about 11% and 3% higher than SPARKS-X, which is the best method at fold level before, respectively. At the superfamily level, the sensitivity of DeepFR for the top 1 and top 5 predictions are 51.4% and 67.1%, which are comparable to the state-of-the-art methods. These results suggested that EC-features can be used to facilitate fold recognition. However, we also noted that at the family level, DeepFR alone cannot provide

Table 1. Success rate (sensitivity) of fold recognition evaluated using Top 1 and Top 5 predictions at family, superfamily and fold levels

Method	Family		Superfamily		Fold	
	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5
PSI-Blast	71.2	72.3	27.4	27.9	4.0	4.7
HMMER	67.7	73.5	20.7	31.3	4.4	14.6
SAM-T98	70.1	75.4	28.3	38.9	3.4	18.7
BLASTLINK	74.6	78.9	29.3	40.6	6.9	16.5
SSERCH	68.6	75.5	20.7	32.5	5.6	15.6
SSHMM	63.1	71.7	18.4	31.6	6.9	24.0
THREADER	49.2	58.9	10.8	24.7	14.6	37.7
Fugue	82.2	85.8	41.9	53.2	12.5	26.8
SPARKS	81.6	88.1	52.5	69.1	28.7	47.7
SP3	81.6	86.8	55.3	67.7	30.8	47.4
HHpred	82.9	87.1	58.0	70.0	25.2	39.4
SP4	80.9	86.3	57.8	57.8	30.8	53.6
SP5	82.4	87.6	59.8	70.0	37.9	58.7
RAPTOR	**86.6**	89.3	56.3	69.0	38.2	58.7
SPARKS-X	84.1	90.3	59.0	76.3	45.2	67.0
BoostThreader	86.5	90.5	66.1	76.4	42.6	57.4
FOLDpro	85.0	89.9	55.0	70.0	26.5	48.3
RF-Fold	84.5	91.5	63.4	79.3	40.8	58.3
DN-Fold	84.5	91.2	61.5	76.5	33.6	60.7
RFDN-Fold	84.7	91.5	65.7	78.8	37.7	61.7
DN-FoldS	84.1	91.2	62.7	76.7	33.3	57.9
DN-FoldR	82.3	88.3	56.0	71.0	27.4	56.7
DeepFR	65.4	83.4	51.4	67.1	56.1	70.1
DeepFRpro	83.1	**92.3**	**69.6**	**82.5**	**66.0**	**78.8**

Test set: LINDAHL dataset.

Note: The bold font denotes the highest sensitivity in their respective categories. The results of PSI-Blast, HMMER, SAM-T98, BLASTLINK, SSERCH, SSHMM and THREADER are taken from Lindahl and Elofsson (2000). The results of Fugue, HHpred, RAPTOR and BoostThreader are from Peng and Xu (2009). The results of SPARKS, SP3, SP4, SP5 and SPARKS-X are from Yang *et al.* (2011). The of FOLDpro, RF-Fold and DN-Fold series are from Jo *et al.* (2015).

Table 2. Success rate (sensitivity) of fold recognition at family, superfamily and fold levels with identical identical multiple-sequence alignments as input

Method	Family		Superfamily		Fold	
	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5
HMMER	81.1	89.7	58.5	68.0	17.8	38.3
HHpred	83.6	91.0	71.4	79.3	27.1	44.2
CNFpred	80.5	88.5	59.7	68.0	37.1	52.6
MRFalign	**85.2**	90.8	**72.4**	80.9	38.6	56.7
map_align	53.3	65.0	43.8	54.6	25.2	39.9
DeepFR	65.4	83.4	51.4	67.1	56.1	70.1
DeepFRpro	83.1	**92.3**	69.6	**82.5**	**66.0**	**78.8**

Test set: LINDAHL dataset.

Note: The bold font denotes the highest sensitivity in their respective categories.

Downloaded from https://academic.oup.com/bioinformatics/article/33/23/3749/4083263 by Shanghai Jiao Tong University School of Medicine user on 30 November 2025

Table 3. Success rate (sensitivity) of fold recognition evaluated using top 1 and top 5 predictions at family, superfamily and fold levels

Method	Family		Superfamily		Fold	
	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5
RFDN-Fold	93.4	97.2	82.1	91.1	61.1	86.1
RF-Fold	93.4	**98.1**	83.9	91.1	55.6	72.2
DN-Fold	**94.3**	97.2	82.1	91.1	61.1	86.1
DN-FoldS	**94.3**	**98.1**	82.1	91.1	61.1	86.1
DN-FoldR	93.4	97.2	80.4	89.3	69.4	83.3
map_align	52.8	68.9	46.4	62.5	44.4	63.9
MRFalign	91.5	**98.1**	85.7	**94.6**	63.9	72.2
DeepFR	75.5	93.4	78.6	83.9	**86.1**	**88.9**
DeepFRpro	**94.3**	96.2	**87.5**	**94.6**	83.3	**88.9**

Test set: SCOP_TEST dataset.

Note: The bold font denotes the highest sensitivity in their respective categories of prediction. The results of RFDN-Fold, RF-Fold and DN-Fold series are taken from Jo et al. (2015).

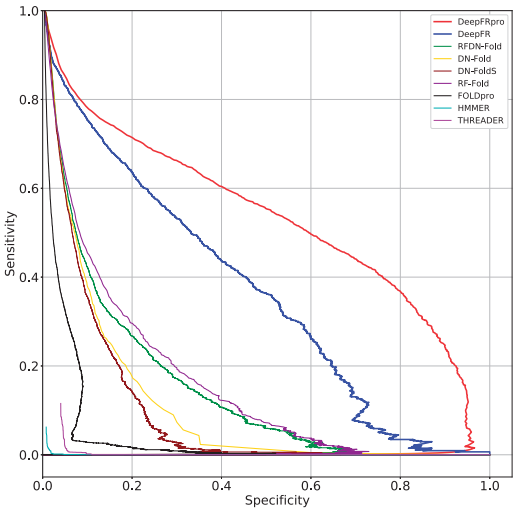

Fig. 3. The specificity-sensitivity plot at fold level. Data other than our approaches were kindly provided by J. Cheng and A. Elofsson

reliable fold recognition, implying the importance to incorporate traditional sequence and structural features.

Next, we examined the performance of DeepFRpro, which enhances DeepFR by incorporating traditional sequence and structural features. Tables 1 and 2 suggested that this strategy leads to a significant performance improvement. Specifically, at the fold level, the sensitivity of DeepFRpro for top 1 and top 5 predictions are 66.0% and 78.8%, about 21% and 12% higher than the best method SPARKS-X, respectively. At the superfamily level, DeepFRpro showed 69.6% and 82.5% sensitivity for top 1 and top 5 predictions, respectively, which is about 3% lower than MRFalign in top 1, but 2% higher in top 5 predictions. More importantly, at the family level, DeepFRpro showed sensitivity of 83.1% and 92.3% for top 1 and top 5 predictions, respectively, which is 3.5% lower in top 1, but 3.0% higher in top 5 predictions than the best approach RAPTOR. These results suggested that DeepFRpro could accurately recognize fold types, especially at fold/superfamily levels.

To fully understand the effect of EC features for fold recognition, we compared DeepFRpro with RF-Fold, since these two methods differs only in an extra EC feature in DeepFRpro. As shown in Table 1, DeepFRpro outperforms RF-Fold except for top 1 prediction at family level. These results clearly suggest the importance of EC features for fold recognition.

To determine whether our approach DeepFRpro is prone to over-fitting, we further performed an independent assessment as in Jo *et al.* (2015) on SCOP_TEST dataset, which is independent from LINDAHL dataset. Table 3 suggests that on SCOP_TEST dataset, DeepFRpro also performed better than other approaches, implying the robustness of DeepFRpro.

Together, these results suggested that EC features alone can be used for fold recognition at fold level, and the combination of EC features and traditional sequence and structural features could significantly improve fold recognition at all levels.

3.2 Evaluating fold recognition using specificity-sensitivity plots

The examination of the fraction of true hits in top 1 and top 5 predictions is very intuitive; however, this evaluation approach tells little about the reliability of hits, as a hit with low score still might be ranked top as long as all other hits have even lower scores. As a more thorough examination, specificity-sensitivity plots were drawn as performed in Lindahl and Elofsson (2000). In these plots, specificity (precision) represents the percentage of predicted positives that are true positives, and sensitivity (recall) represents the percentage of true positives that are predicted as positives. The advantage of specificity-sensitivity plots is that it can show the ability of a fold recognition approach to reliably find all pairwise matches against all templates.

We compared DeepFR and DeepFRpro with 7 fold recognition methods (RFDN-Fold, DN-Fold, DN-FoldS, RF-Fold, FOLDpro, HMMER and THREADER) on LINDAHL dataset at fold, superfamily and family levels (Fig. 3, and Supplementary Figs S3, S4). The specificity-sensitivity results of HMMER and THREADER were provided by A. Elofsson (http://archive.bioinfo.se/protein-id/), and the other five results were provided by J. Cheng (http://cactus.rnet.missouri.edu/dnfold/data/).

At the family level (Supplementary Fig. S3), DeepFRpro consistently outperformed all other methods by >10% for almost all specificity values, and DeepFR performed comparably with the start-of-the-art methods at low specificity region but a little worse at high specificity region. At both superfamily level (Supplementary Fig. S4) and fold level (Fig. 3), DeepFRpro and DeepFR performed significantly better than all other methods. For example, at fold level, the sensitivity of DeepFRpro and DeepFR are nearly twice that of the second best approach at almost all specificity region. These results show that DeepFR and DeepFRpro significantly outperformed a variety of different methods in all categories at almost all specificity regions, indicating that the extracted fold-specific features help improve robustness of fold recognition.

We noticed that the performance of our approaches drop dramatically when specificity is close to 1 (Fig. 3). Similar phenomena have also been reported for RF-Fold and DN-Fold, which was attributed to the existence of several false positives with high scores (Cheng and Baldi, 2006). In-depth examination of these false positives supported this explanation, i.e. some of these false-positives turn out to be true positive under new definition of folds. For example, the pair (`1rnl-d1rnl_1, 1lea-d1lea`) was classified into different fold (`1.31.1.2, 1.4.3.2`) in the LINDAHL dataset based on the old SCOP 1.37 release; however, it has been renamed to (`d1rnla1, d1leaa_`) with identical fold type (`a.4.6.2, a.4.5.2`) in the SCOP 2.06 release. In this study, these false-positives were not corrected to ensure consistency with previous evaluations as conducted in Cheng and Baldi (2006).

Downloaded from https://academic.oup.com/bioinformatics/article/33/23/3749/4083263 by Shanghai Jiao Tong University School of Medicine user on 30 November 2025

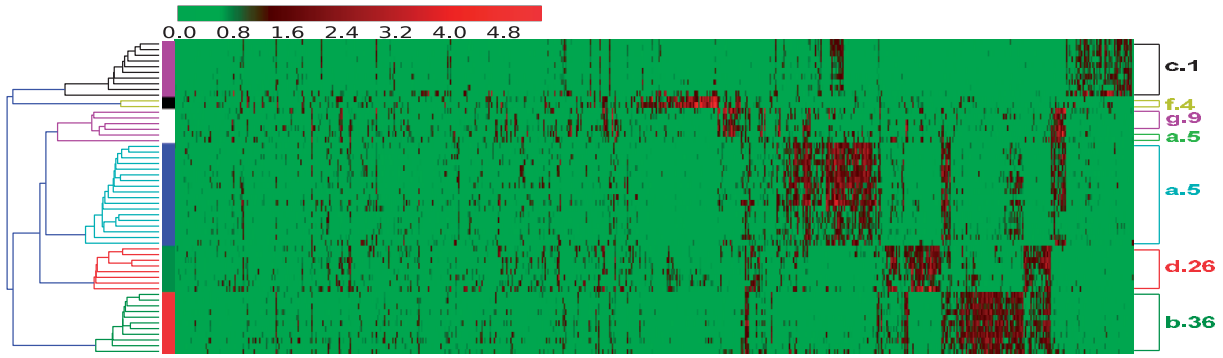

Fig. 4. Analysis of the extracted EC-features by running bi-clustering over 55 proteins (covering 6 folds) selected from SCOP_TEST dataset. The clustering tree are perfectly consistent with fold hierarchy. In addition, the features with high value (shown in red) form blocks. These two observations suggest that the extracted EC-features are fold-specific. A detailed version of this figure with protein name, together with a larger figure showing all proteins in SCOP_TEST, are available in the Supplementary Figures S1, S2

3.3 Analysis of the source of distinguishing power of EC features

An ideal feature set for fold recognition should be fold-specific, i.e. proteins with identical fold type share similar features, while proteins with different fold types have distinct features. To investigate whether the extracted EC-features have such ideal properties, we run bi-clustering on the EC features of proteins in SCOP_TEST dataset (Fig. 4, and Supplementary Figs S1, S2).

As shown in Figure 4, the features with high values (shown in red) form several blocks, suggesting that proteins in the same fold commonly present similar pattern in their EC-features. For example, the up-right block in red (from column 961 to 1022) suggest that proteins in fold `c.1` commonly have high values for corresponding EC-features (such as feature 203, 76, etc.). More importantly, the constructed clusters are in perfect agreement with the fold hierarchy. For example, the proteins in folds `c.1, f.4, g.9, a.5, d.26` and `b.36` form individual sub-trees, indicating that these EC-features are fold specific.

We also observed certain inconsistence between the clustering tree and fold hierarchy. For example, protein `d1pgya_` has fold type of `a.5`; however, it was assigned with fold type `g.9` by DeepFR due to the significant similarity between its EC-feature vector and these of proteins in fold `g.9` (Supplementary Fig. S1). This represents typical errors when applying DeepFR score alone for fold recognition. However, when combining DeepFR score with traditional sequence and structural features, DeepFRpro corrected the errors on these two proteins. Together, these investigation suggested that the extracted EC features are fold-specific, providing foundation for accurate fold recognition.

3.4 Using fold-specific features to improve threading results

We further investigated whether the extracted fold-specific features can be used to improve threading results. Specifically, for each protein in SCOP_TEST dataset, we first used it as query protein and aligned it to templates by running threading tools. The threading tools themselves provided a ranking of the generated alignments according to a specific scoring schema (denoted as *threading score*), and built the final predicted model based on the ranked 1st alignment. Here, we re-ranked the alignments according to the linear combination of the original threading score and DeepFR score, and generated the predicted model accordingly. The predicted models under the two scoring schema were compared to reveal the effect of DeepFR score on improving threading results.

We examined the enhancement of DeepFR on two threading tools, namely, MRFalign (Ma *et al.*, 2014) and hhalign (Söding, 2005). We observed that compared with the original MRFalign score, the combination of MRFalign score and DeepFR score generated better models on 9 query proteins in SCOP_TEST (Supplementary Table S3). Similarly, the combination of hhalign score and DeepFR score outperformed the original hhalign score on 3 proteins. Take protein `d1jo0a_` as an example (Supplementary Fig. S5), MRFalign ranked `d1kida_` as the best template and generated the predicted model accordingly (GDT_TS: 0.27, TMscore: 0.29). In contrast, the template `1whra_` was ranked first when using the combination of MRFalign score and DeepFR score, and a high-quality model was generated as final prediction result (GDT_TS: 0.38, TMscore: 0.45). These results clearly suggested that the integration of fold-specific features could effectively improve threading results.

4 Conclusion

In this study, we employed the deep convolution networks technique to extract fold-specific features from predicted residue–residue contacts. In-depth examination suggests that the extracted features are fold-specific. Using these extracted features alone, our approach DeepFR achieved better performance than state-of-the-art approaches. Moreover, the extracted features are orthogonal to traditional sequence and structural features derived from alignments, and the combination of them can improve success rate of fold recognition at superfamily/fold levels and improve threading results as well.

Acknowledgements

We greatly appreciate Jianlin Cheng, Arne Elofsson, Jinbo Xu and Sergey Ovchinnikov for providing data and program.

Funding

We would like to thank the National Basic Research Program of China (2013CB910104), General Research Fund Grant No. 9041901 (CityU 118413), and the National Natural Science Foundation of China (31671369, 31270834, 61272318, 11175224, 11121403, and 31270909, 31770775) for providing financial supports.

Conflict of Interest: none declared.

Downloaded from https://academic.oup.com/bioinformatics/article/33/23/3749/4083263 by Shanghai Jiao Tong University School of Medicine user on 30 November 2025

References

Adhikari,B. *et al*. (2015) CONFOLD: residue–residue contact-guided ab initio protein folding. *Proteins Struct. Funct. Bioinf*., **83**, 1436–1449.

Altschul,S.F. *et al*. (1990) Basic local alignment search tool. *J. Mol. Biol*., **215**, 403–410.

Andreani,J. and Söding,J. (2015) bbcontacts: prediction of β-strand pairing from direct coupling patterns. *Bioinformatics*, btv041.

Bairoch,A.M. *et al*. (2005) The universal protein resource (UniProt). *Nucleic Acids Res*., **33**, D154–D159.

Berman,H.M. *et al*. (2000) The protein data bank. *Nucleic Acids Res*., **28**, 235–242.

Bottou,L. (2012) Stochastic gradient descent tricks. In: Montavon,G., *et al*. (eds), *Neural Networks: Tricks of the Trade*. Springer, Berlin, Heidelberg, pp. 421–436.

Branden,C.I. *et al*. (1999) *Introduction to Protein Structure*. Garland Science, New York, NY.

Breiman,L. (2001) Random forests. *Mach. Learn*., **45**, 5–32.

Breiman,L. *et al*. (1984) *Classification and Regression Trees*. CRC press, Raton, FL, USA.

Cheng,J. and Baldi,P. (2006) A machine learning information retrieval approach to protein fold recognition. *Bioinformatics*, **22**, 1456–1463.

Cheung,N.J. *et al*. (2016) Protein folds recognized by an intelligent predictor based-on evolutionary and structural information. *J. Comput. Chem*., **37**, 426–436.

Dong,Q. *et al*. (2009) A new taxonomy-based protein fold recognition approach based on autocross-covariance transformation. *Bioinformatics*, **25**, 2655–2662.

Dunn,S.D. *et al*. (2008) Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction. *Bioinformatics*, **24**, 333–340.

Eddy,S.R. (1998) Profile hidden Markov models. *Bioinformatics*, **14**, 755–763.

Ekeberg,M. *et al*. (2013) Improved contact prediction in proteins: using pseudolikelihoods to infer potts models. *Phys. Rev. E*, **87**, 012707.

Ekeberg,M. *et al*. (2014) Fast pseudolikelihood maximization for direct-coupling analysis of protein structure from many homologous amino-acid sequences. *J. Comput. Phys*., **276**, 341–356.

Fischer,D. (2003) 3D-SHOTGUN: a novel, cooperative, fold-recognition meta-predictor. *Proteins Struct. Funct. Bioinf*., **51**, 434–441.

Fox,N.K. *et al*. (2014) SCOPe: Structural classification of proteins-extended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic Acids Res*., **42**, D304–D309.

Göbel,U. *et al*. (1994) Correlated mutations and residue contacts in proteins. *Proteins Struct. Funct. Bioinf*., **18**, 309–317.

Haspel,N. *et al*. (2003) Hierarchical protein folding pathways: a computational study of protein fragments. *Proteins Struct. Funct. Bioinf*., **51**, 203–215.

Hauser,M. *et al*. (2016) Mmseqs software suite for fast and deep clustering and searching of large protein sequence sets. *Bioinformatics*, **32**, 1323–1330.

Ioffe,S. and Szegedy,C. (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *Proceedings of the 32nd International Conference on Machine Learning*, pp. 448–456.

Jia,Y. *et al*. (2014) Caffe: Convolutional architecture for fast feature embedding. *arXiv Preprint arXiv*, 1408.5093.

Jo,T. and Cheng,J. (2014) Improving protein fold recognition by random forest. *BMC Bioinformatics*, **15**, 1.

Jo,T. *et al*. (2015) Improving protein fold recognition by deep learning networks. *Sci. Rep*., **5**, 17573.

Jones,D.T. *et al*. (1992) A new approach to protein fold recognition. *Nature*, **358**, 86–89.

Jones,D.T. *et al*. (2012) PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics*, **28**, 184–190.

Kamisetty,H. *et al*. (2013) Assessing the utility of coevolution-based residue–residue contact predictions in a sequence-and structure-rich era. *Proc. Natl. Acad. Sci. USA*, **110**, 15674–15679.

Kim,D.E. *et al*. (2014) One contact for every twelve residues allows robust and accurate topology-level protein structure modeling. *Proteins Struct. Funct. Bioinf*., **82**, 208–218.

Krizhevsky,A. *et al*. (2012) Imagenet classification with deep convolutional neural networks. In: Pereira F. *et al*. (eds.) *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., Red Hook, NY, pp. 1097–1105.

LeCun,Y. *et al*. (1989) Backpropagation applied to handwritten zip code recognition. *Neural Comput*., **1**, 541–551.

LeCun,Y. *et al*. (1998) Gradient-based learning applied to document recognition. *Proc. IEEE*, **86**, 2278–2324.

LeCun,Y. *et al*. (2015) Deep learning. *Nature*, **521**, 436–444.

Li,S.C. *et al*. (2008) Fragment-HMM: A new approach to protein structure prediction. *Protein Sci*., **17**, 1925–1934.

Li,W. and Godzik,A. (2006) Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, **22**, 1658–1659.

Lindahl,E. and Elofsson,A. (2000) Identification of related proteins on family, superfamily and fold level. *J. Mol. Biol*., **295**, 613–625.

Lundström,J. *et al*. (2001) Pcons: A neural-network–based consensus predictor that improves fold recognition. *Protein Sci*., **10**, 2354–2362.

Ma,J. *et al*. (2012) A conditional neural fields model for protein threading. *Bioinformatics*, **28**, i59–i66.

Ma,J. *et al*. (2014) MRFalign: protein homology detection through alignment of markov random fields. *PLoS Comput. Biol*., **10**, e1003500.

Marks,D.S. *et al*. (2011) Protein 3D structure computed from evolutionary sequence variation. *PloS One*, **6**, e28766.

Morcos,F. *et al*. (2011) Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proc. Natl. Acad. Sci. USA*, **108**, E1293–E1301.

Murzin,A.G. *et al*. (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol*., **247**, 536–540.

Olmea,O. and Valencia,A. (1997) Improving contact predictions by the combination of correlated mutations and other sources of sequence information. *Fold. Des*., **2**, S25–S32.

Ovchinnikov,S. *et al*. (2017) Protein structure determination using metagenome sequence data. *Science*, **355**, 294–298.

Pearson,W.R. (2013) An introduction to sequence similarity ("homology") searching. *Curr. Protoc. Bioinf*., **42**, 3.1.1–3.1.8.

Pearson,W.R. and Lipman,D.J. (1988) Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. USA*, **85**, 2444–2448.

Pedregosa,F. *et al*. (2011) Scikit-learn: Machine learning in python. *J. Mach. Learn. Res*., **12**, 2825–2830.

Peng,J. and Xu,J. (2009) Boosting protein threading accuracy. *Res. Comput. Mol. Biol*., **5541**, 31–45.

Remmert,M. *et al*. (2011) HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nat. Methods*, **9**, 173–175.

Roy,A. *et al*. (2010) I-TASSER: a unified platform for automated protein structure and function prediction. *Nat. Protoc*., **5**, 725–738.

Rumelhart,D.E. *et al*. (1988) Learning representations by back-propagating errors. *Cogn. Model*., **5**, 1.

Schroff,F. *et al*. (2015) Facenet: A unified embedding for face recognition and clustering. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815–823.

Seemayer,S. *et al*. (2014) CCMpred–fast and precise prediction of protein residue–residue contacts from correlated mutations. *Bioinformatics*, **30**, 3128–3130.

Sharif Razavian,A. *et al*. (2014) CNN features off-the-shelf: an astounding baseline for recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 806–813.

Shi,J. *et al*. (2001) FUGUE: sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties. *J. Mol. Biol*., **310**, 243–257.

Simons,K.T. *et al*. (1997) Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and bayesian scoring functions. *J. Mol. Biol*., **268**, 209–225.

Downloaded from https://academic.oup.com/bioinformatics/article/33/23/3749/4083263 by Shanghai Jiao Tong University School of Medicine user on 30 November 2025

Skolnick,J. *et al*. (1997) MONSSTER: a method for folding globular proteins with a small number of distance restraints. *J. Mol. Biol*., **265**, 217–241.

Skwark,M.J. *et al*. (2014) Improved contact predictions using the recognition of protein like contact patterns. *PLoS Comput. Biol*., **10**, e1003889.

Söding,J. (2005) Protein homology detection by HMM–HMM comparison. *Bioinformatics*, **21**, 951–960.

Srivastava,N. *et al*. (2014) Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res*., **15**, 1929–1958.

Steinegger,M. and Söding,J. (2016) Sensitive protein sequence searching for the analysis of massive data sets. *bioRxiv*, 079681.

Taigman,Y. *et al*. (2014) Deepface: Closing the gap to human-level performance in face verification. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1701–1708.

Taylor,W.R. (2016) An algorithm to parse segment packing in predicted protein contact maps. *Algorithms Mol. Biol*., **11**, 17.

Wang,C. *et al*. (2016) FALCON@home: a high-throughput protein structure prediction server based on remote homologue recognition. *Bioinformatics*, **32**, 462–464.

Wang,S. *et al*. (2017) Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLOS Comput. Biol*., **13**, 34.

Xia,J. *et al*. (2016) An ensemble approach to protein fold classification by integration of template-based assignment and support vector machine classifier. *Bioinformatics*, btw768.

Xu,D. and Zhang,Y. (2012) Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field. *Proteins Struct. Funct. Bioinf*., **80**, 1715–1735.

Xu,J. *et al*. (2003) RAPTOR: optimal protein threading by linear programming. *J. Bioinf. Comput. Biol*., **1**, 95–117.

Yang,Y. *et al*. (2011) Improving protein fold recognition and template-based modeling by employing probabilistic-based matching between predicted one-dimensional structural properties of query and corresponding native properties of templates. *Bioinformatics*, **27**, 2076–2082.

Zeiler,M.D. and Fergus,R. (2014) Visualizing and Understanding Convolutional Networks. In: Fleet D., *et al*. (eds) Computer Vision ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8689. Springer, Cham.

Zhang,H. *et al*. (2016) Improving residue–residue contact prediction via low-rank and sparse decomposition of residue correlation matrix. *Biochem. Biophys. Res. Commun*., **472**, 217–222.

Zheng,W.-M. (2014) Proteins: From sequence to structure. *Chin. Phys. B*, **23**, 078705.

Downloaded from https://academic.oup.com/bioinformatics/article/33/23/3749/4083263 by Shanghai Jiao Tong University School of Medicine user on 30 November 2025