Neurocomputing 428 (2021) 19–29

Contents lists available at ScienceDirect

Neurocomputing

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

DeepPPF: A deep learning framework for predicting protein family

Shehu Mohammed Yusuf, Fuhao Zhang, Min Zeng, Min Li *

School of Computer Science and Engineering, Central South University, Changsha 410083, PR China

ARTICLE INFO

Article history:
Received 12 March 2020
Revised 19 September 2020
Accepted 23 November 2020
Available online 9 December 2020
Communicated by: Zidong Wang

Keywords:
Multi-scale convolutional neural network
Protein functional family
Protein sequence
Deep learning

ABSTRACT

Machine learning pipelines for protein functional family prediction are urgently needed especially now that only 1% of raw protein sequences have been manually annotated. Although existing machine learning algorithms have achieved a decent performance in modeling and predicting the functional families of protein sequences, they still have two drawbacks. First, biological dependencies among nucleotides are not rich enough to describe motifs for these methods. Also, existing algorithms are not accurate enough to predict the functional families of newly discovered proteins. To address the above limitations simultaneously, we propose a novel deep learning framework for predicting protein family, DeepPPF, which employs the word2vec technique in capturing distributional dependencies among nucleotides and discovers rich features from diverse motif lengths to characterize proteins. The novelty of the DeepPPF is in utilizing distributional dependencies among nucleotides. Experimental results on G protein-coupled receptor hierarchical datasets show the effectiveness of DeepPPF in achieving the state-of-the-art performance in items of Mathew's correlation coefficients (MCC) of 97.62%, 88.45% and, 83.09% for family, subfamily and, sub-subfamily hierarchical levels, respectively. Also, DeepPPF outperformed existing methods in terms of prediction accuracy and Mathew's correlation coefficients on the cluster of orthologous groups (COG) and phage of orthologous groups (POG) datasets. Furthermore, we analyzed the ability of DeepPPF framework to discover rich motifs for functional classes with the least sets of protein sequences. The experimental results show that rich motif discovery is key to improving the modeling performance of protein families through deep learning techniques. Finally, we investigated the effect of transferring a low-level functional domain level to a high-level functional domain and results show that the target domain prediction can be improved with transfer learning. Therefore, our proposed deep learning framework can be useful in characterizing protein functional families. The codes and datasets are available at https://github.com/CSUBioGroup/DeepPPF.

1. Introduction

Annotation of proteins based on their family domain functions plays an indispensable role in understanding the theory behind the life cycle at the molecular level [1,2]. Applying a practical and standard approach to figure out the biochemical family of unknown proteins, is one of the primary goals in protein function prediction. Previous studies [3] have shown that computational function annotation methods are much less expensive and less time-consuming to characterize the exponentially increasing unknown protein sequences when compared with experimental methods [4–6]. Despite previous studies have achieved a decent performance, extracting useful sequence information for protein function prediction remains a challenging task in protein bioinformatics [7]. One challenge is identifying combinations of rich motifs that must be present or absent in a sequence for classifiers to reliably assign it to a functional family. Machine learning represents an attractive domain to help fill this gap by detecting informative patterns [8]. These informative patterns can not only help characterize newly discovered protein sequences, but also identify hormones in humans, bioactive ligands and, discovery and development of therapeutic drugs [9,10]. In addition, using machine learning to discover motifs [11,12] for evolutionary relationships has always been a challenging task [13], since it calls for accurate biophysical models of protein sequences. For this requirement, several computational pipelines for predicting evolutional functions have been developed by researchers using protein sequences as input. Such pipelines can broadly be classified into alignment-based and alignment-free models.

Alignment-based protein family modeling methods compare and generate alignment motifs from multiple sequences using a position-specific scoring matrix (PSSM) [14,15]. A couple of such modeling are based on ClustalW [16], MUSCLE [17], Omega [18],

* Corresponding author.
E-mail address: limin@mail.csu.edu.cn (M. Li).

https://doi.org/10.1016/j.neucom.2020.11.062
0925-2312/© 2020 Published by Elsevier B.V.

and PASTA [19] and HHblits [20]. Multiple sequence alignment techniques are capable of providing valuable information on sequence conservation but need to handle insertions and deletion of amino acids [14]. Thus, protein family modeling from multiple sequences requires sophisticated techniques [4]. A more accurate alignment method that can handle these challenges is the profile hidden Markov model (pHMM) [21,22]. Furthermore, multiple sequence alignment is a global alignment algorithm. Thus, much recombination of conserved regions by rearrangement, inversion, transposition or translocation is nearly impossible without information loss.

Alignment-free methods have been successful in protein family modeling. In this type of modeling category, multiple sequences are not aligned to generate a position weight matrix (PWM) [23]. Techniques belonging to this group include *k*-mer based logistic regression [24] and protvec logistic regression (ProtVec LR) [25]. *k*-mer techniques are more accurate than existing alignment-free methods, and can process an arbitrary number of domains and can speed up protein sequence modeling. A critical limitation of modeling with *k*-mers is losing the order of biological information in protein sequences. Thus, *k*-mers are not successful enough to model protein families. Actually, there is no optimal way to determine *k* when using *k*-mers, which can affect the sensitivity and specificity of the modeling task.

Several studies have shown that motif-based function modeling can extract useful information from conserved sub-regions or residues of a protein chain [1,26]. Wang *et al* showed that an automated motif-based protein function classifier could identify combinations of motifs that must be present or absent in a sequence to reliably assign it to a functional family [27]. Studies in [28] and [29] showed that sequence-based techniques could be effective in identifying proteins that incorporate transmembrane proteins. In recent years, deep learning methods have achieved state-of-the-art performance in language and biological modeling [12,30–32]. The study in [30] used a trigram-based global vector (GloVe) embeddings with several neural network architectures to model and predict protein functional family. Although gated recurrent units (GRU) outperformed long short-term memory (LSTM), bidirectional LSTM (biLSTM), and convolutional neural network (CNN) with a fixed filter size, much better results could be achieved with CNN by utilizing convolutional units with the variable size to extract rich motifs. DeepFam [4] was the first attempt to apply an alignment-free deep CNN to the motif-discovery task, by which protein functional family is well characterized. Although DeepFam achieved decent performance in terms of accuracy, the deep learning model accuracy, sensitivity and specificity [33], need to be further improved by utilizing more biological information.

Because of the limitations of existing pipelines, there is a need for a new pipeline for characterizing functional families of protein sequences. This study leverages on the recent success of alignment-free deep learning modeling to develop a novel pipeline, DeepPPF, for modeling and predicting protein families. Our pipeline utilizes word2vec embedding features as inputs to a multi-scale convolutional neural network. The novelty of this work is the use of dense distributional motifs to capture the correlations between nucleotides of protein sequences for protein family prediction. Previous studies on alignment-free protein family predictions have utilized one-hot encoding to represent each nucleotide of a protein. However, one-hot encoding is sparse and cannot capture the relationship between these nucleotides. Therefore, we propose distributional encoding to characterize the relationship between nucleotides of a protein. This distributional representation is the co-occurrence relationships between individual nucleotides instead of that between blocks of nucleotides; as in the case of *k-mer* representations. For instance, if the one-letter code for alanine, a nucleotide among the standard 20 IUPAC amino acids, is ‘A’, then its distributional relationship, obtained using word2vec, is a vector of continuous distribution (...0.003, 0.345, 0.053...), instead of the one-hot vector (0, 0, 0, 1, 0...0); where a single entry is 1 and others are zeroes. Therefore, DeepPPF calculates the dense distribution of proteins using word2vec and feeds these distributions into a stacked convolutional layer, 1-max pooling, addition and, concatenation layers to extract rich conserved regions. Subsequently, hidden units are utilized to detect longer conserved regions. Then, a fully connected neural network is utilized to extract high-order features and generate feature vectors as output. Finally, a softmax function is utilized to infer family probabilities. The computational experiments demonstrate that our proposed deep learning framework improves the performance of function prediction over other methods, and performs particularly well in predicting functional families of protein sequences. Furthermore, we investigate the ability of DeepPPF framework to discover rich motifs for functional classes with the least sets of protein sequences. The findings indicate that rich motif discovery is key to improve the performance of protein family modeling via deep learning. Finally, to explore the best network architecture for hierarchical level modeling and prediction, we transfer the knowledge learned from the lowest hierarchical functional level domain to two target functional levels. Our findings indicate that the transfer learning approach can be used to improve the performance of higher levels.

The developed deep learning framework has several advantages. First, dense vectors are generated directly from raw sequences without requiring multiple sequence alignment. Second, using three different convolutional kernels and merge operations to combine nearby short captured conserved regions, the problem of determining the optimal length of convolutional units for rich motifs can be overlooked. In addition, knowledge learned from our deep learning model can be transferred to target hierarchical level domains. Finally, our framework can automatically extract rich motifs and make prediction simultaneously on particularly challenging low similarity proteins.

2. Materials and methods

In order to improve the accuracy and performance of protein family prediction, an alignment-free deep learning framework is proposed. To validate the performance of the deep learning framework, we selected the G-protein coupled receptor (GPCR) hierarchical level dataset, cluster of orthologous groups (COG) database [4], and phages of orthologous groups (POG).

2.1. G-protein coupled receptor superfamily

GPCR [28], a well-studied protein superfamily consisting of diverse proteins with seven highly conserved transmembrane segments, was used as the benchmark dataset. This dataset contains divergent proteins that are hierarchically annotated. For our experiment GDS, one of the largest GPCR datasets is utilized. The hierarchical evolutional levels of the GDS dataset include 5 families, 40 subfamilies and, 108 sub-sub families associated with 8354 unidentical protein sequences. The authors of [4] utilized 8222 protein sequences belonging to 5 families, 38 subfamilies and 86 sub-subfamilies. These protein sequences were downloaded at http://epigenomics.snu.ac.kr/DeepFam/data.zip.

Using the GPCR dataset for functional family modeling and prediction has several difficulties. First, the distribution of functional families is highly biased. Fig. 1 shows the top 50 sub-subfamily’s distribution of the GPCR protein sequences with the highest frequencies.

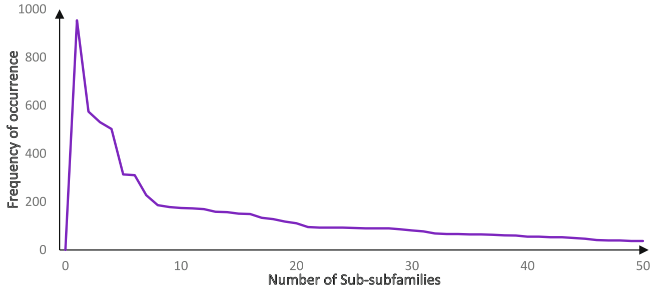

Fig. 1. Distribution of top 50 Sub-subfamilies with the highest frequencies in GPCR dataset.

As seen in Fig. 1, 11 sub-subfamilies with the highest frequencies make up about 50.19% of the entire samples. The bias of the distribution can be understood more qualitatively from Table 1, in which five GPCR sub-subfamilies, ranked as the first, 10th, 20th, 40th and 80th, in terms of frequency, are presented.

The most frequent functional sub-subfamily (Taste) appears in 11.6% of the GPCR dataset, while the 80th frequent functional sub-subfamily (Gastric) appears in 0.16% of the GPCR dataset. This noticeable bias in the distribution usually leads to poor classification performance. Similar bias can be noticed in the family and sub-family hierarchical functional levels, as presented in Table 2 and Table 3, respectively.

Second, there is a large variation in the number of amino acids in each sequence. For instance, a protein sequence may have 400 amino acids, while another may have 900 amino acids. Furthermore, 284 protein sequences have above 1000 amino acids in each. This can lead to a huge computational cost if such proteins are included in a training set.

To overcome the difficulties arising from the characteristics of the GPCR dataset, we filtered 284 sequences with the length longer than 1000 amino acids or ambiguous amino acid codes (B, O, J, U, X, and Z). This resulted in a total of 7938 protein sequences. Then, we employ a deep learning algorithm to improve the performance of GPCR functional prediction. For each hierarchical class, 10-fold cross-validation was used to evaluate the modeling method [34]. Furthermore, the long sequences excluded during training were used for testing.

2.2. Cluster of orthologous groups

COG [35], another well-studied protein family database consisting of diverse proteins of microbial genomes, was used as another benchmark dataset. This dataset has been accessible to the public since 1997 and the most recent update was published in 2014 [4]. Each protein family contains a varying number of sequences (ranging from 1 to 10,632) and sequence lengths (ranging from 21 to 29,202). The authors of [4], in one of their experiments, utilized 1,129,428 protein sequences belonging to 1074 families. In this dataset, the smallest family is that with a threshold of 500 protein sequences. These protein sequences were downloaded at http://epigenomics.snu.ac.kr/DeepFam/data.zip. Fig. 2 shows the top 50 family's distribution of the COG protein sequences with the highest frequencies. As seen in Fig. 2, 10 families with the highest frequencies make up about 35.6% of the entire samples. Therefore, the COG dataset is also a bias distribution.

Table 1
Frequencies of appearances for five selected GPCR sub-subfamilies.

Functional name	Frequency rank	Ratio of proteins with function
Taste	1	0.1160
Adrenoreceptor	10	0.0212
Latrophilin	20	0.0135
Calcitonin	40	0.0067
Gastric	80	0.0016

Table 2
Frequencies of appearances for the five GPCR families.

Functional name	Frequency rank	Ratio of proteins with function
Class A	1	0.6568
Class C	2	0.2642
Class B	3	0.0752
Class E	4	0.0022
Class D	5	0.0016

Table 3
Frequencies of appearances for five selected GPCR subfamilies.

Functional name	Frequency rank	Ratio of proteins with function
Peptide	1	0.3180
CalcSense	4	0.0715
GlutaMeta	8	0.0216
GABA	16	0.0105
cAMP	32	0.0022

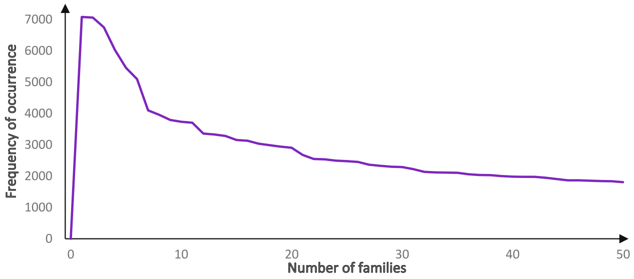

Fig. 2. Distribution of top 50 families with the highest frequencies in COG dataset.

2.3. Phages of orthologous groups

POG database is a collection of conserved orthologs of phage genomes [36]. This dataset has been accessible to the public since 2011 and the most recent update was published in 2013. For this work, POG-07 with 13,086 sequences belonging to 1689 families was utilized. These protein sequences were downloaded at ftp://ftp.ncbi.nlm.nih.gov/pub/kristensen/annotatedPOGs-07/. First, we filtered out sequences of length shorter than 50 and those of length longer than 1000. This resulted in a dataset of 12,677 protein sequences belonging to 1,635 families. Subsequently, we filtered out families with less than 4 protein sequences. This resulted in a POG dataset of 11,411 proteins with 1,201 families. Fig. 3 shows the top 50 family's distribution of the COG protein sequences with the highest frequencies. As seen in Fig. 3, 10 families with the

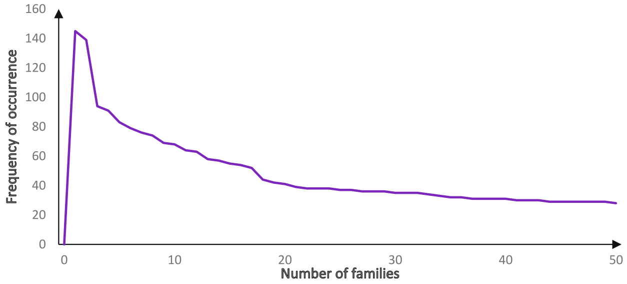

Fig. 3. Distribution of top 50 families with the highest frequencies in POG dataset.

highest frequencies make up about 37.7% of the entire samples. Similarly, the POG dataset is a bias distribution.

3. Methods

In this section, we first introduce an overview of our proposed deep learning architectural framework. Also, the model implementation, baseline model, and evaluation metrics are discussed in Sections 3.2, 3.3 and 3.4, respectively.

3.1. Overview of our proposed deep learning framework

This work proposed DeepPPF, an alignment-free protein family prediction pipeline, that takes raw protein sequences as input, generates dense embedding features, and infer the functional family of proteins as output. First, the framework calculates dense distributional motifs from a center amino acid model trained with the word2vec technique. Each center word is an IUPAC amino acid code notation [4]. Thus, the word2vec encoding vector is a 21-D vector. Second, the existence scores of rich conserved regions are calculated with three different convolutional filters, 1-max pooling, addition, and concatenation layers. Next, hidden units are utilized to detect longer conserved regions that are activated frequently for proteins of the same family. Then, to extract high-order features from the existence of conserved regions and generate feature vectors as output, a fully connected neural network is utilized. Finally, in order to infer the probabilities of being a member of each family from the feature vector, a softmax layer is adopted. The end-to-end framework for our model is shown in Fig. 4.

3.1.1. Sequence encoding

The work of [37] showed that deep learning based on the representation of pretrained distributed embedding like word2vec could achieve remarkable performance. The word2vec technique [38–40] can capture informative semantic features through unsupervised learning [41]. Moreover, this embedding has been utilized a lot in natural language processing [42] and RBP binding site prediction [37]. In this study, word2vec embedding technique is used to map each amino acid to a dense embedding vector of 21 dimensions in a vocabulary. Using this vocabulary, we encode each sequence as a vector of 1000 indices. If the length of a sequence is less than 1000, the vector is padded with zeroes at the end [27]. Also, a protein is ignored if the length of a sequence is more than 1000. Therefore, a protein sequence of the length of 1000 is represented as a 1000 × 21 matrix. One-hot encoding is utilized to encode the true label, Y_t, which is defined as:

$$Y_t = \begin{cases} 1, & if \quad y = i^{th} \ in \ labelset \\ 0, & \qquad otherwise \end{cases} \tag{1}$$

Here, $i \in \{1, \cdots, L\}$ and $t \in \{1, \cdots, N_{label}\}$.

3.1.2. Mult-scale CNN

Convolutional Neural Networks (CNNs) apply convolution filters over inputs to learn multiple local features and provide insight into conserved regions of data. In our work, 1D multi-scale convolution is applied over the encoded protein sequence. If k is the number of convolution filters, and W is a filter kernel with the dimension of $m_k \times 1$, then the mechanism of a multiscale convolution can be represented a follow [4].

$$h_{k,i} = \sigma\left(b_k + \sum_{l=1}^{m_k} X_{i+l-1} * W_{k,l}\right) \tag{2}$$

In this work, we apply filters with 3 different sizes $(k = 3)$, to extract different high-level features [41], from the input sequence matrix. The activation functionσ, is ReLU which is defined as follows.

$$\sigma(x) = max(0, x) \tag{3}$$

To focus on the existence of locally conserved regions, 1-max pooling [4,14,43] is utilized to select the highest activated value among the $l - m_k + 1$ neurons; as follows.

$$h_k^{max} = \max_{1 \le i \le l-m}(h_{k,i}) \tag{4}$$

With the extracted multi-scale features, we add them into a vector as a local context feature. In order to build a much rich local context feature, the output of the added high-level features is concatenated into a fixed-size vector made up of the richest motif

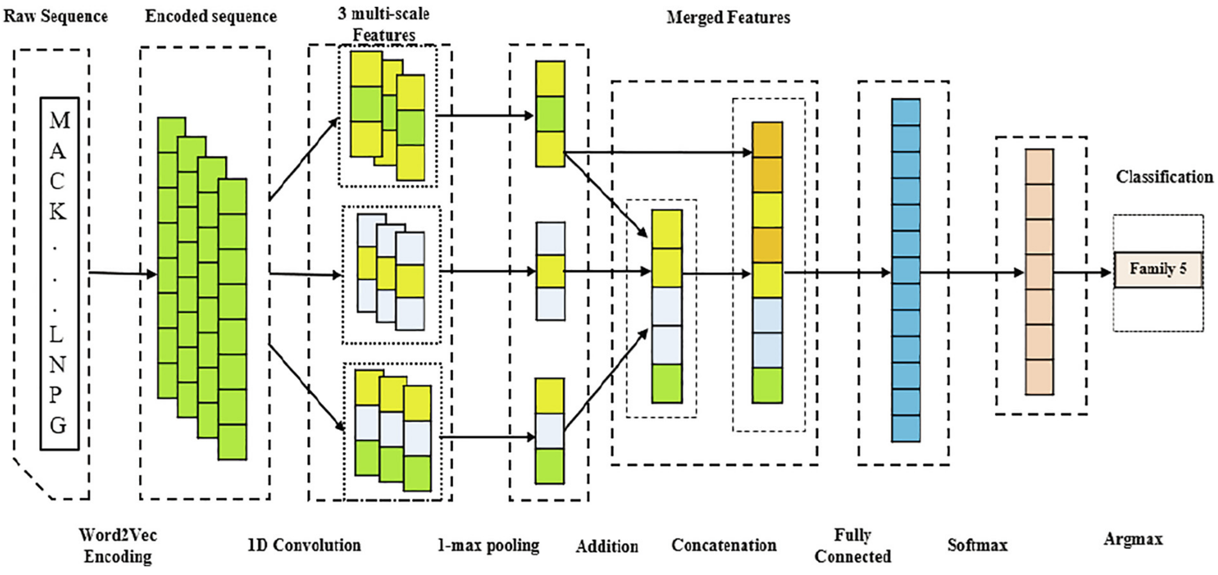

Fig. 4. An overview of DeepPPF framework. It is a multi-scale convolutional neural network whose input are protein sequences. The word2vec technique is utilized to obtain a 21-D vector. The 21-D vector is fed into three 1-D convolutional layers with 3 kernel scales, three 1-max pooling layers and, 2 merge layers to calculate the scores of existing conserved regions. Then, conserved region scores are combined and fed to a fully-connected neural network, with 2000 hidden units to detect complex sites. Finally, a softmax layer is used to infer the probabilities of each functional family.

from the 3 scales. Dropout [34,38,44–46], before and after concatenation is used to prevent overfitting and train robust features. To combine features from sequences and extract high-order features, a dense hidden layer is applied as follows.

$$Z_l = \sigma\left(B_l^{'} + \sum_{k=1}^{N_{flt}} h_k^{max} W_{k,l}^{'}\right) \tag{5}$$

Batch normalization, after the dense hidden layer, is used to control the distribution of the combined feature vectors. Finally, the conditional probability distribution over each functional family is calculated, using a softmax function, as follows.

$$O_t = B_t^{''} + \sum_{l=1}^{N_{hdn}} Z_l * W_{l,t}^{''} \tag{6}$$

$$\widehat{y}_t = \frac{e^{O_t}}{\sum_r e^{O_r}} \tag{7}$$

3.2. Model implementation and optimization

In training DeepPPF, the multi-output categorical cross-entropy loss function was minimized with Adam optimizer as the update rule [47]. The training process was done through the stochastic gradient descent over shuffled mini-batch. Xavier glorot uniform [48] was utilized in initializing weights of our model. The model was fit with 90% of the training set and the remaining 10% for validation. The optimization terminates after 20 epochs [4]. To prevent overfitting [49], dropout layers and $l2/l1$ regularization [50] were used. Keras with Tensorflow [24] as backend was utilized to implement our deep learning model. To accelerate the training process, we used NVIDIA Ge Force RTX 2080 Ti. The training time on the GPCR datasets was less than 10 min and the inference time was less than 1 s. In our work, the hyperparameter setup is presented in Table 4.

These hyperparameters, except for filter sizes and dropout rate, were set to same values utilized in the work of Seo *et al.* [4]. To set 3 different filter sizes, we found an appropriate filter size, in the range of 8 to 32, for the convolutional layer by assuming a single filter size was to be utilized. Findings showed that a filter size of 20 yielded the best performance in terms of prediction accuracy. This was followed by a filter size of 19 and then 8. Filter sizes in the range of 9 to 18 either yielded similar or poor prediction results as compared with filter sizes of 8, 19 and 20. Therefore, the single multi-scale convolutional neural network layer was designed by setting these three filter sizes {8, 19, 20}, to capture useful local distributional features. Then, we manually selected an appropriate dropout, in the set {25%, 30%, 35%, 40%}, for our deep learning framework using the training and validation data set of the GPCR sub-subfamily. Findings showed that a dropout rate of 35% yielded the best performance in terms of accuracy and Mathew's correlation coefficient. Therefore, the dropout was set to be 35% for our model.

Table 4
Hyperparameter setup.

Hyperparameter	Set Value
Filter size, m_k	{8, 19, 20}
Number of convolutional filters, N_{flt}	{250, 250, 250}
Number of hidden units, N_{hdn}	2000
Coefficient of regularization, λ	0.0005
Dropout rate	In range {25% - 40%}
Learning rate	0.001
Batch size	100

3.3. Baseline model for performance comparison

Seo *et al.* [4] demonstrated that DeepFam obtained a decent performance for the functional family prediction. In their experiment, the 2-D convolution of one-hot encoded input was carried out with 8 different filter sizes; {8, 12, 16, 20, 24, 28, 32}. After 1-max pooling of the outputs of each filter, the resulting local feature scores were concatenated and fed to a fully connected hidden layer. Finally, a softmax layer was utilized to calculate the probabilities of a protein sequence belonging to each class. For an extensive comparison of our model, we obtained the prediction results of the top six models on GPCR proteins sequences dataset from [4]. Also, the prediction results of DeepPPF on the COG and POG datasets were obtained. Furthermore, we generated test sets from validation sets such that they contain GPCR protein sequences belonging to training classes with the least numbers of protein sequences. These test sets were utilized to investigate the impact of bias distributions on our model.

3.4. Assessment metrics

We evaluated DeepPPF with five metrics that are commonly used in protein function prediction. These measures include prediction accuracy(Acc), Mathew's correlation coefficient (MCC), average precision ($AvgPr$), average recall ($AvgRc$), and average f1-measure ($Avgf_1$). The formulas for computing these metrics are as follows.

$$Acc = \frac{\#TP + \#TN}{\#TP + \#FP + \#TN + \#FN} \tag{8}$$

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{9}$$

$$pr_i = \frac{\#TP}{\#TP + \#FP} \tag{10}$$

$$rc_i = \frac{\#TP}{\#TP + \#FN} \tag{11}$$

$$AvgPr = \frac{1}{n} \cdot \sum\nolimits_{i=1}^{n} pr_i \tag{12}$$

$$AvgRc = \frac{1}{n} \cdot \sum\nolimits_{i=1}^{n} rc_i \tag{13}$$

$$Avgf_1 = \left\{ \frac{2 \cdot AvgPr \cdot AvgRc}{AvgPr + AvgRc)} \right\} \tag{14}$$

pr_i and rc_i are the precision and recall of a predicted family term i, respectively. n is the number of family terms. #TP and #TN represent the number of the positive and negative terms of predicted proteins which are classified correctly, respectively. #FP and #FN represent the number of the positive and negative terms of proteins which are misclassified, respectively.

4. Results

The performance of DeepPPF, in terms of prediction accuracy, is compared with six existing methods in Section 4.1 on 10-fold cross-validation using the GPCR dataset. In Sections 4.2 and 4.3, we further compared the performance of DeepPFF on the COG and POG dataset respectively. In Section 4.4, we investigate the impact of bias distribution on DeepPPF. Finally, in Section 4.5, we compare the performance of DeepPPF with transfer learning.

4.1. Performance of DeepPPF model in 10-fold cross-validation

The performance of DeepPPF was first evaluated on the GPCR dataset, with the 10-fold cross-validation, using a bottom-up approach following up the hierarchy of GPCR labels [2]. Cross-validation was employed to reduce the impact of data dependency and to improve the reliability of results [51]. Six methods used for comparison were trained with the same data used to train our model. These six methods include DeepFam, profile Hidden Markov Model (pHMM), 3-mer based logistic regression model (3-mer LR), Protvec logistic regression (Protvec LR), Selective top-down model (Std) and Naïve Bayes model (NB). Our model achieved the best accuracy in the family, subfamily, and sub-subfamily predictions, respectively, as shown in Table 5.

From Table 5, we can see the overall accuracies of all the machine learning models decreased as the hierarchical level became deeper; from family to sub-subfamily. This is due to the much-unbalanced classes at a deep level as compared to an immediate level.

To show DeepPPF distinguishing capability in predicting functional family over the state-of-the-art alignment-free deep learning method, our deep learning framework was further evaluated with the baseline model, DeepFam, using *MCC* and *bAcc* as metrics. The hyperparameter settings for DeepFam is similar to that in [4]. To capture more features, we increased the number of filters in the DeepFam model to 250. Fig. 5 shows the plot for functional family prediction. We can see that the performance of our model is better over *Acc*, *bAcc* and *MCC* measures with the 10-fold cross-validation. Similarly, Figs. 6 and 7 presents the results at the sub-family and sub-subfamily hierarchical levels, respectively. However, it can be seen that at the sub-family level, DeepFam is slightly higher in bAcc measure.

4.2. Performance of DeepPPF model on COG dataset

Furthermore, we evaluated DeepPPF on the COG dataset using 3-fold cross-validation [2]. Then, results obtained were compared with four existing methods. These four methods include DeepFam, profile Hidden Markov Model (pHMM), 3-mer based logistic regression model (3-mer LR), Protvec logistic regression. Our model achieved lower accuracy, for the COG dataset than DeepFam, as shown in Table 6.

However, our model slightly outperformed DeepFam, in terms of *bAcc* and *MCC* measures as shown in Fig. 8.

4.3. Performance of DeepPPF model on POG dataset

To further evaluate the performance of DeepPPF, we utilized the POG dataset for training and testing. We split the total samples in such that 75% were training set and 25% for the testing set. Our model achieved a slightly better predictive power in terms of $Acc, AvgRc, Avgf1$ and *MCC*, for the POG dataset than DeepFam, as shown in Table 7.

Table 5
Prediction accuracy (%) at each hierarchical level.

Method	Family	Sub-family	Sub-subfamily
DeepPPF	**98.89**	**90.31**	**84.38**
DeepFam*	97.17	86.82	81.17
pHMM*	95.77	85.39	78.50
3-mer LR*	95.59	83.51	77.06
Protvec LR*	88.58	74.98	67.32
Std**	95.87	8.77	69.98
NB**	77.29	52.60	36.66

Note: Results marked with * are extended from [4]. Results marked ** are extended from [28]. Bold indicates the best performance for each data set.

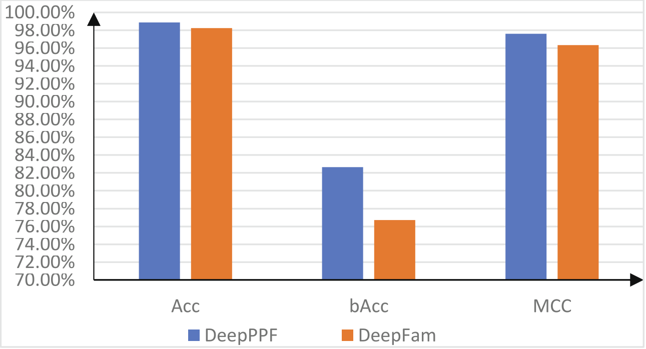

Fig. 5. Comparison of family prediction performance between DeepPPF and DeepFam.

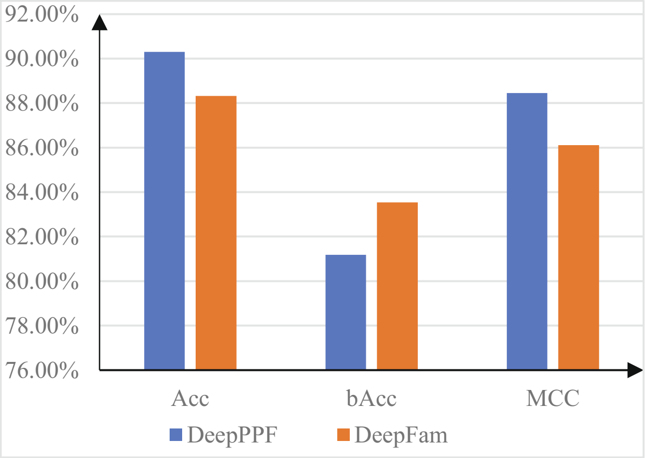

Fig. 6. Comparison of sub-family prediction performance between DeepPPF and DeepFam.

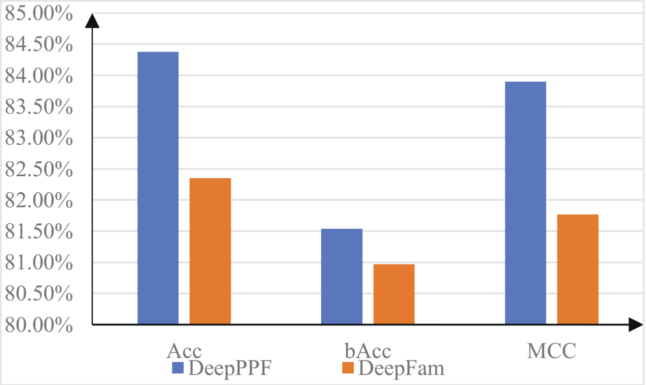

Fig. 7. Comparison of sub-subfamily prediction performance between DeepPPF and DeepFam.

4.4. Impact of bias distribution on our model

Furthermore, we investigated the impact of bias distribution in modeling classes with few protein sequences. The experiment was done as follows. We computed the frequency distribution of classes at each hierarchical functional level. From the distributions,

Table 6
Prediction accuracy (%) on COG dataset.

Method	Accuracy
DeepPPF	91.83
DeepFam*	**95.40**
pHMM*	91.75
3-mer LR*	85.59
Protvec LR*	47.34

Note: Results marked with * are extended from [4]. Bold indicates the best performance for the data set.

we inferred 25, 25 and 3 functional classes with the least frequencies at the sub-subfamily, sub-family and family levels, respectively. Then, we created test sets from the validation sets such that they contain only protein sequences belonging to these inferred classes. For instance, from Table 3, family's 'B', 'E' and 'D' contains the least number of GPCR proteins in the training data set. Therefore, only protein sequences belonging to these three classes are extracted from the validation set to form the test set for functional family prediction. Also, the 'BrainSpec' protein family has the least number of sequences in the sub-subfamily training set. Therefore, protein sequences in the validation set belonging to 'BrainSpec' family were among the sequences extracted to create the sub-subfamily test set. Finally, these test sets were predicted using our pretrained models and DeepFam models for each fold. Finally, the contribution of our model in discovering rich motifs for these proteins sequences is determined by comparing its $Acc, AvgPr, AvgRc, Avgf1$ and MCC with those of the baseline model. Tables 8–10 show the results of the contributions to family, sub-family and, sub-subfamily predictions, respectively. Comparing these metrics can determine the impact of bias on DeepPPF using the baseline model as the reference.

Results in Table 8 indicated that our model has a better predictive performance than DeepFam in all metrics. This signifies the ability of our model to capture rich motifs for functional family prediction. Thus, our model is more robust in handling bias distribution. Similarly, with a slightly better performance indicated in Table 9, our model can compete with DeepFam in capturing rich motifs for sub-family predictions. Additionally, the proposed model performed better than DeepFam in modeling sub-subfamily functions of the GPCR dataset; as indicated in Table 10. Therefore, the bias distribution of the GPCR dataset affected the baseline model more than our model.

4.5. Comparing the performance of deep learning method with transfer learning

Furthermore, we investigated the possibility of improving the predictive performance of DeepPPF by transfer learning. As shown in Fig. 9, the transfer learning framework consists of two steps. Firstly, we trained our proposed deep learning framework for a low hierarchical family level prediction (source domain task) using the GPCR dataset. For this step, we selected one of the cross-validation models utilized in Section 4.1. For instance, we selected the seventh cross-validation model of the sub-subfamily classification task and the fifth cross-validation model of the sub-family classification task. The second step was to fix the shared network parameters and then, fine-tune the hidden dense layer and retrain the weights of the output layer for the upper family-level classification using the GPCR dataset. The shared network layers include multi-scale convolutional networks, addition layer and concatenate layer.

Two source domain models were utilized for transfer learning. First, the selected sub-subfamily (source domain) model was utilized to fix parameters for sub-family (target domain) and, family (target domain) predictions. Also, the selected sub-family model (source domain) was utilized to fix the parameters for family (target domain) prediction. Finally, our proposed deep learning framework with and without transfer learning was evaluated, with the 10-fold cross-validation, by comparing their $Acc, AvgPr, AvgRc, Avgf1$ and MCC. Fig. 10 shows the results of family prediction with and without transfer learning.

It is obvious, from Fig. 10, that the prediction accuracy and Mathew's correlation coefficient obtained by transfer learning outperforms our proposed model without transfer learning. Our model, when the sub-subfamily model was the source domain task, obtained the values of Acc, macro$AvgPr$, macro$AvgRc$, macro$Avgf1$ and MCC, in percentages, being 99.35, 83.42, 81.07, 81.91 and 98.63, respectively. The corresponding values, in percentages, obtained when subfamily-based source model was transferred, are 99.37, 83.61, 83.07, 83.28, and 98.66, respectively. The results above show that the subfamily model is more suitable than sub-subfamily, as a source domain task, for the family target domain. However, DeepPPF model without transfer learning, outperformed the transfer learning models, in terms

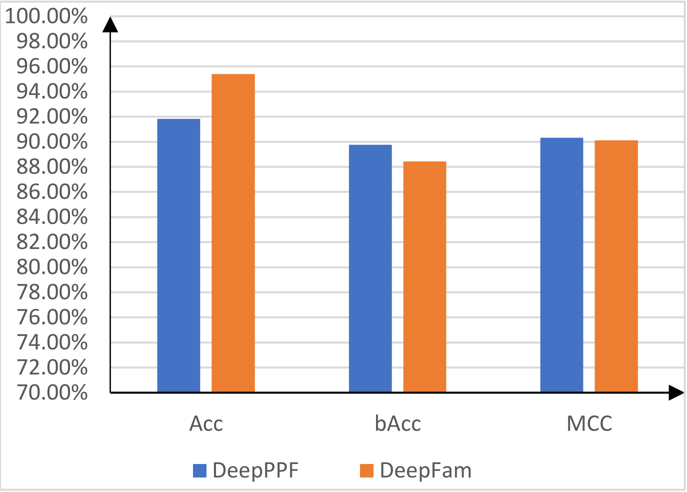

Fig. 8. DeepPPF vs. DeepFam on 3-fold cross-validation.

Table 7
Prediction results on POG dataset.

Method	*Acc*	*AvgPr*	*AvgRc*	*Avgf*1	*MCC*
DeepPPF	**78.54**	71.57	**72.41**	**70.35**	**78.20**
DeepFam	78.23	**72.18**	71.95	70.18	78.04

Note: Bold indicates the best performance.

Table 8
Impact of bias distribution on family modeling.

Method	*Acc*	*AvgPr*	*AvgRc*	*Avgf*1	*MCC*
DeepPPF	**92.34**	**56.00**	**52.57**	**53.92**	**62.36**
DeepFam	91.04	51.50	47.90	49.35	58.87

Note: Bold indicates the model with less impact from bias distribution.

Table 9
Impact of bias distribution on sub-family modeling.

Method	*Acc*	*AvgPr*	*AvgRc*	*Avgf*1	*MCC*
DeepPPF	**83.64**	**80.20**	74.75	76.40	**83.29**
DeepFam	79.23	80.12	**74.78**	**76.41**	78.97

Note: Bold indicates the model with less impact from bias distribution.

Table 10
Impact of bias distribution on sub-subfamily modeling.

Method	*Acc*	*AvgPr*	*AvgRc*	*Avgf*1	*MCC*
DeepPPF	**83.53**	**76.72**	**73.09**	**74.24**	**83.55**
DeepFam	80.83	73.46	69.30	70.59	80.83

Note: Bold indicates the model with less impact from bias distribution.

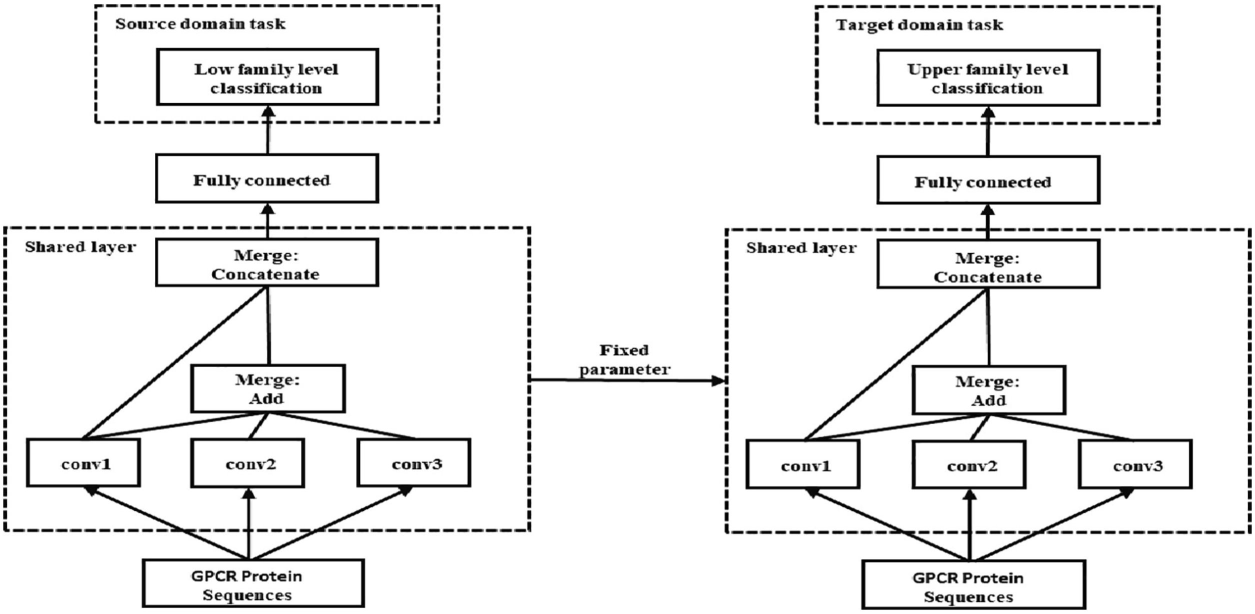

Fig. 9. DeepPPF framework with transfer learning.

ofmacro*AvgPr*, macro*AvgRc*, andmacro*Avgf*1 (86.40, 86.19, and 86.06, respectively), the values obtained, in percentages, for *Acc*and *MCC* (98.89 and 97.62, respectively) are the least. Thus, transfer learning can improve protein functional family prediction.

Similarly, Fig. 11 presents the comparison of our model, with and without transfer learning, for protein sub-family prediction. From Fig. 11, DeepPPF with transfer learning performs better than DeepPPF without transfer learning in GPCR protein sub-family prediction. Without transfer learning, *Acc*, macro*AvgPr*, macro*AvgRc*, macro*Avgf*1 and *MCC*, in percentages, increased from 90.31, 84.72, 81.67, 81.90, and 88.45 to 91.21, 85.17, 84.15, 83.61, and 89.54, respectively. In summary, sub-subfamily (source) domain task can improve the performances of DeepPPF model sub-family (target domain) of GPCR proteins.

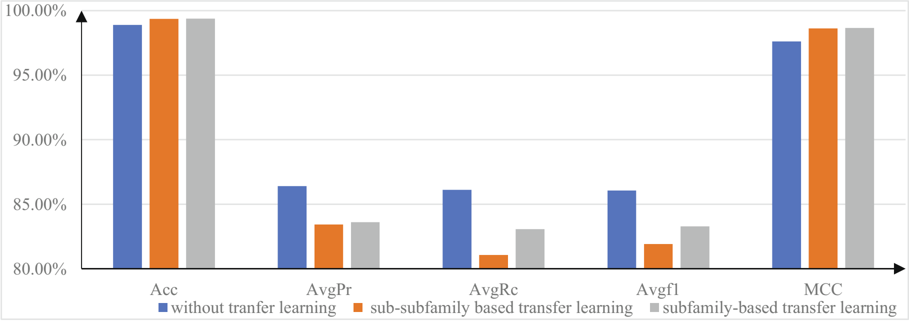

Fig. 10. Performances of DeepPPF model with and without hierarchical transfer learning during family prediction.

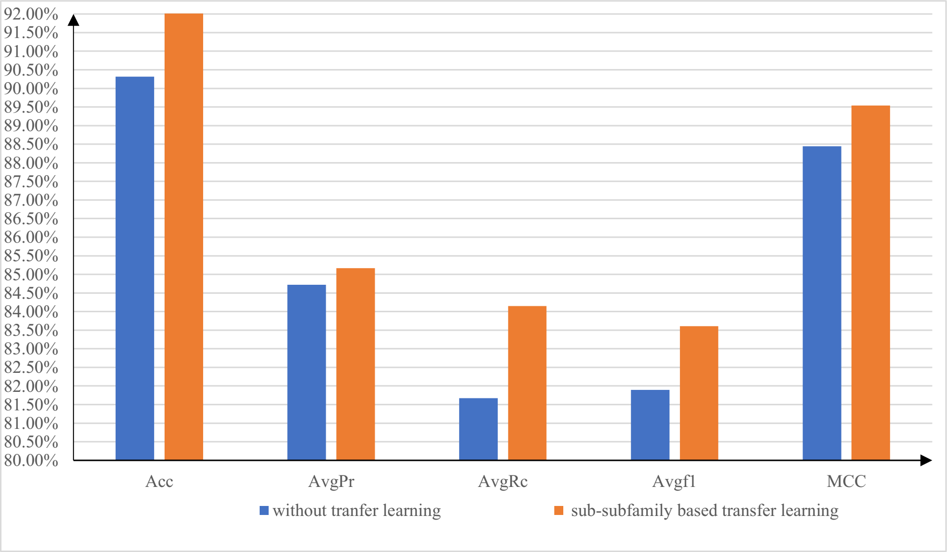

Fig. 11. Performances of DeepPPF model with and without hierarchical transfer learning during sub-family prediction.

5. Conclusion

This work developed and tested DeepPPF, a new deep learning-based alignment-free framework for functional family prediction of proteins. Also, this work has shown the advantage of distributional representation of each nucleotide over one-hot encoding. Results presented show that DeepPPF model, which utilizes distributional representation, has a better predictive performance than the state-of-the-art machine learning methods which utilize one-hot encoding in capturing motifs. DeepPPF achieved the best predictive performance in terms of Mathew's correlation coefficients (97.62%, 88.45% and, 83.09%) and prediction accuracy (98.89%, 90.31%, 84.38%) on the GPCR family, sub-family and sub-subfamily datasets, respectively. DeepPPF slightly outperformed other existing methods, in terms of Mathew's correlation coefficients (90.31), on the COG dataset. Also, DeepPPF performed better than DeepFam, in terms of Mathew's correlation coefficients (78.20%) and prediction accuracy (78.54%), on the POG dataset. The main contribution of our work is related to the use of distributional representation and deep learning techniques to effectively capture rich motifs from protein sequences. This makes DeepPPF have some desirable advantages. First, the experimental tests show that our model is more accurate than existing machine learning pipelines in modeling and predicting functional families of protein sequences. Second, DeepPPF is another alignment-free technique based on multi-scale CNN, which is scalable, in terms of the number of convolutional kernels, without sacrificing the modeling power. In addition, we showed that using distributional motifs as inputs, our model can discover rich motifs for modeling functional families with very few proteins. Since a huge number of protein sequences generated can belong to highly unbalanced functional families created from manually annotated sequences, a more accurate model with high specificity and sensitivity is very important. Thus, DeepPPF can be useful in this regard. Although some proteins may belong to more than one functional family, the DeepPPF model is limited to a multi-class protein functional family problem. Therefore, DeepPPF can predict only a functional family for a protein. Informed by the performance of DeepPPF in predicting the COG dataset, another limitation of DeepPPF is its decreasing predictive accuracy and precision when utilized to model datasets with proteins of short sequence lengths. Also, DeepPPF is affected by data imbalance. Few protein functional families have a lot of protein sequences while many protein families have few numbers of protein sequences. This can cause poor mining of protein family features. Thus, resulting in misclassification of some proteins. Also, the thresholds set for selecting protein sequence and functional family is another limitation. Therefore, leading to information loss.

As part of future work, there is a need to further improve the predictive power of our model by exploring additional sources of sequence-derived information. For instance, quantitative biophysical properties can be a potential inclusion. Also, there is a need to further capture rich motifs for our model by increasing its complexity without the explosion of its parameters. For example, adopting max-pooling strategies and combining other competitive deep learning methods in natural language processing with our model are options. Furthermore, due to recent successes of deep transfer learning, there is a need for a comprehensive study of transfer learning frameworks for protein family prediction. Finally, there is a need to extend our work to a multi-label problem because some proteins have multiple functions. DeepPPF can be applied, in the field of neuroscience, to improve proteome characterization and pattern recognition. Additionally, DeepPPF can be applied practically in predicting GPCR's families, thereby facilitating drug discovery.

CRediT authorship contribution statement

Shehu Mohammed Yusuf: Conceptualization, Methodology, Software, Validation, Writing - original draft, Writing - review & editing. **Fuhao Zhang:** Methodology, Data curation, Software. **Min Zeng:** Methodology, Writing - original draft, Writing - review & editing. **Min Li:** Supervision, Writing - original draft, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work is supported in part by the National Natural Science Foundation of China under Grants Nos. 61832019, 61772552, the 111 Project (No. B18059), and the Hunan Provincial Science and Technology Program (2018WK4001).

References

[1] F. Zhang, H. Song, M. Zeng, Y. Li, L. Kurgan, M. Li, DeepFunc: a deep learning framework for accurate prediction of protein functions from protein sequences and interactions, Proteomics 19 (2019). https://doi:10.1002/pmic.201900019.

[2] M. Li, W. Li, F.-X. Wu, Y. Pan, J. Wang, Identifying essential proteins based on sub-network partition and prioritization by integrating subcellular localization information, J. Theor. Biol. 447 (2018) 65–73.

[3] K.K. Yang, Z. Wu, F.H. Arnold, Machine-learning-guided directed evolution for protein engineering, Nat. Methods 16 (2019) 687–694.

[4] S. Seo, M. Oh, Y. Park, et al., DeepFam: deep learning-based alignment-free method for protein family modeling and prediction", Bioinformatics 34 (2018) i254–i262.

[5] M. Zeng, M. Li, Z. Fei, et al., A deep learning framework for identifying essential proteins by integrating multiple types of biological information, IEEE/ACM Trans. Comput. Biol. Bioinf. (2019), https://doi.org/10.1109/TCBB.2019.2897679.

[6] M. Kulmanov, M.A. Khan, R. Hoehndorf, DeepGO: predicting protein functions from sequence and interactions using a deep ontology-aware classifier", Bioinformatics 34 (4) (2018) 660–668.

[7] H. Chen, D. Kihara, Effect of using suboptimal alignments in template-based protein structure prediction, Proteins Struct. Funct. Bioinforma. 79 (1) (2011) 315–334.

[8] R. Fa, D. Cozzetto, C. Wan, D.T. Jones, Predicting human protein function with multitask deep neural networks, PLoS ONE 13 (6) (2018) 1–16.

[9] W. Thomsen, J. Frazer, D. Unett, Functional assays for screening GPCR targets, Curr. Opin. Biotechnol. (2005).

[10] D.A. Sykes, L.A. Stoddart, L.E. Kilpatrick, S.J. Hill, Binding kinetics of ligands acting at GPCRs", Mol. Cell. Endocrinol. 485 (2019) 9–19.

[11] B. Alipanahi, A. Delong, M.T. Weirauch, et al., Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning, Nat. Biotechnol. 33 (8) (2015) 831–838.

[12] H. Zeng, M.D. Edwards, G. Liu, D.K. Gifford, Convolutional neural network architectures for predicting DNA-protein binding, Bioinformatics 32 (12) (2016) i121–i127.

[13] Q. Zhang, L. Zhu, D.S. Huang, High-Order Convolutional Neural Network Architecture for Predicting DNA-Protein Binding Sites, IEEE/ACM Trans. Comput. Biol. Bioinform. 16 (4) (2019) 1184–1192.

[14] G. Aoki, Y. Sakakibara, Convolutional neural networks for classification of alignments of non-coding RNA sequences, Bioinformatics 34 (13) (2018) i237–i244.

[15] A. Zielezinski, S. Vinga, J. Almeida, et al., Alignment-free sequence comparison: benefits, applications, and tools, Genome Biol. 18 (2017) 186.

[16] J.D. Thompson, D.G. Higgins, T.J. Gibson, CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice, Nucleic Acids Res. 22 (22) (1994) 4673–4680.

[17] R.C. Edgar, MUSCLE: multiple sequence alignment with high accuracy and high throughput, Nucleic Acids Res. 32 (5) (2004) 1792–1797.

[18] F. Sievers, A. Wilm, D. Dineen et al., Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega, Molecular systems biology, 7(1), pp. 1-6.

[19] S. Mirarab, N. Nguyen, S. Guo, et al., PASTA: ultra-large multiple sequence alignment for nucleotide and amino-acid sequences, J. Comput. Biol. 22 (5) (2015) 377–386.

[20] M. Remmert, A. Biegert, A. Hauser, et al., HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment, Nat. Methods 9 (2) (2012) 173.

[21] S. R. Eddy, Profile hidden Markov models, Bioinformatics (Oxford, England), vol. 14(9) (1998) pp. 755-763.

[22] A. Bateman, L. Coin, R. Durbin et al., The Pfam protein families database, Nucleic acids research, 32, (1) (2004) pp. D138-D141.

[23] G.D. Stormo, DNA binding sites: representation and discovery", Bioinformatics 16 (1) (2000) 16–23.

[24] M. Abadi, P. Barham, J. Chen et al., Tensorflow: A system for large-scale machine learning, In: Proceedings of the12th USENIX Symposium on Operating Systems Design and implementation (OSDI 16), (2016) pp. 265-283.

[25] E. Asgari, M.R.K. Mofrad, Continuous distributed representation of biological sequences for deep proteomics and genomics, PLoS ONE 10 (11) (2015) e0141287.

[26] A. Jabeen, S. Ranganathan, Applications of machine learning in GPCR bioactive ligand discovery, Curr. Opin. Struct. Biol. 55 (2019) 66–76.

[27] X. Wang, D. Schroeder, D. Dobbs, V. Honavar, Automated data-driven discovery of motif-based protein function classifiers, Inf. Sci. (Ny) 155 (1–2) (2003) 1–18.

[28] M.N. Davies, A. Secker, A.A. Freitas, M. Mendao, J. Timmis, D.R. Flower, On the hierarchical classification of G protein-coupled receptors, Bioinformatics 23 (23) (2007) 3113–3118.

[29] Ö.S. Saraç, V. Atalay, R. Cetin-Atalay, GOPred: GO Molecular Function Prediction by Combined Classifiers, PLoS ONE 5 (8) (2010) e12382.

[30] T.K. Lee, T. Nguyen, Protein family classification with neural networks, Stanford University (2016) 1–9.

[31] J. Zhou, O.G. Troyanskaya, Predicting effects of noncoding variants with deep learning-based sequence model, Nat. Methods 12 (10) (2015) 931–934.

[32] D. R. Kelley, J. Snoek, and J. L. Rinn, Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks, Genome research, 26(7), pp. 990-999.

[33] J. Xi, A. Li, M. Wang, HetRCNA: a novel method to identify recurrent copy number alternations from heterogeneous tumor samples based on matrix decomposition framework, IEEE/ACM Trans. Comput. Biol. Bioinforma. 17 (2) (2020) 422–434, https://doi.org/10.1109/TCBB.2018.2846599.

[34] Q. Zhang, L. Zhu, D.S. Huang, High-order Convolutional neural network architecture for predicting DNA-protein binding sites", IEEE/ACM Trans. Comput. Biol. Bioinforma. 16 (4) (2018) 1184–1192.

[35] M.Y. Galperin, K.S. Makarova, Y.I. Wolf, E.V. Koonin, Expanded microbial genome coverage and improved protein family annotation in the COG database, Nucleic Acids Res. 43 (D1) (2014) D261–D269, https://doi.org/10.1093/nar/gku1223.

[36] D.M. Kristensen, X. Cai, A. Mushegian, Evolutionarily conserved orthologous families in phages are relatively rare in their prokaryotic hosts, J. Bacteriol. 193 (8) (2011) 1806–1814, https://doi.org/10.1128/JB.01311-10.

[37] X. Pan, H.-B. Shen, Learning distributed representations of RNA sequences and its application for predicting RNA-protein binding sites with a convolutional neural network, Neurocomputing 305 (2018) 51–58.

[38] E. Altszyler, M. Sigman, S. Ribeiro, D.F. Slezak, Comparative study of LSA vs Word2vec embeddings in small corpora: a case study in dreams database, Conscious. Cogn. 56 (2016) 178–187.

[39] B. Zhang, X. Xu, X. Li, X. Chen, Y. Ye, Z. Wang, Sentiment analysis through critic learning for optimizing convolutional neural networks with rules, Neurocomputing 356 (2019) 21–30.

[40] Y. Guo, D. Zhou, R. Nie, X. Ruan, W. Li, DeepANF: a deep attentive neural framework with distributed representation for chromatin accessibility prediction, Neurocomputing (2019), https://doi.org/10.1016/j.neucom.2019.10.091.

[41] G. Liu, J. Guo, Bidirectional LSTM with attention mechanism and convolutional layer for text classification", Neurocomputing 337 (2019) 325–338.

[42] J. Xu et al., Incorporating context-relevant knowledge into convolutional neural networks for short text classification, Neurocomputing (2019), https://doi.org/10.1016/j.neucom.2019.08.080.

[43] J. Han, C. Moraga, The influence of the sigmoid function parameters on the speed of backpropagation learning, in Proceedings of International Workshop on Artificial Neural Networks, Springer, Berlin, Heidelberg, 1995, pp. 195–201.

[44] D. Veltri, U. Kamath, A. Shehu, Deep learning improves antimicrobial peptide recognition, Bioinformatics 34 (16) (2018) 2740–2747.

[45] X. Pan, P. Rijnbeek, J. Yan, H. Bin Shen, Prediction of RNA-protein sequence and structure binding preferences using deep convolutional and recurrent neural networks, BMC Genomics 19 (1) (2018) 1–11.

[46] M. Zeng, M. Li, Z. Fei, Y. Yu, Y. Pan, J. Wang, Automatic ICD-9 coding via deep transfer learning, Neurocomputing 324 (2019) 43–50.

[47] D. P. Kingma and J. L. Ba, Adam: A method for stochastic optimization, 3rd Int. Conf. Learn. Represent. (ICLR 2015) - Conf. Track Proc., (2015) pp. 1–15, arXiv preprint arXiv: 1412.6980.

[48] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, J. Mach. Learn. Res. 9 (2010) 249–256.

[49] J. Xi, X. Yuan, M. Wang, A. Li, X. Li, Q. Huang, Inferring subgroup-specific driver genes from heterogeneous cancer samples via subspace learning with subgroup indication, Bioinformatics 36 (6) (2020) 1855–1863, https://doi.org/10.1093/bioinformatics/btz793.

[50] S. Wang, J. Peng, W. Liu, An ℓ2/ℓ1 regularization framework for diverse learning tasks", Signal Process. 109 (2015) 206–211.

[51] L. Zhang, G. Yu, D. Xia, J. Wang, Protein–protein interactions prediction based on ensemble deep neural networks, Neurocomputing 324 (2019) 10–19.

Shehu Mohammed Yusuf received the B.Eng. and M.Sc. degrees from Ahmadu Bello University, Zaria, Nigeria, in 2008 and 2015, respectively. He is currently working towards the PhD degree in the School of Computer Science and Engineering, Central South University, China. His research interests include bioinformatics, pattern recognition, machine learning, deep learning and signal processing.

Fuhao Zhang received his BSc degrees in Chongqing University of Posts and Telecommunications, China in 2014. He is currently a postgraduate student in Bioinformatics at Central South University. His currently research interests include bioinformatics, network representation learning and deep learning.

Min Zeng received the B.S. degree from Lanzhou University in 2013, and the M.S. degree from Central South University in 2016. He is currently working toward the PhD degree in the School of Computer Science and Engineering, Central South University, China. His research interests include bioinformatics, machine learning and deep learning.

Min Li received the PhD degree in Computer Science from Central South University, China, in 2008. She is currently a Professor at the School of Computer Science and Engineering, Central South University, Changsha, Hunan, P.R. China. Her research interests include computational biology, systems biology and bioinformatics. She has published more than 80 technical papers in refereed journals such as Bioinformatics, IEEE/ACM Transactions on Computational Biology and Bioinformatics, and Proteomics.