Briefings in Bioinformatics, 2025, **26(4)**, bbaf420
https://doi.org/10.1093/bib/bbaf420
Problem Solving Protocol

MKFGO: integrating multi-source knowledge fusion with pretrained language model for high-accuracy protein function prediction

Yi-Heng Zhu[1], Shuxin Zhu[1], Xuan Yu[2], He Yan[3], Yan Liu[4], Xiaojun Xie[1], Dong-Jun Yu[5,*], Rui Ye[1,*]

[1]College of Artificial Intelligence, Nanjing Agricultural University, 666 Binjiang Avenue, Jiangbei New District, Nanjing, Jiangsu Province, 211800, China
[2]Department of Computer Science, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon Tong, Hong Kong SAR (HKG), 999077, China
[3]College of Information Science and Technology & Artificial Intelligence, Nanjing Forestry University, 159 Longpan Road, Xuanwu District, Nanjing, Jiangsu Province, 210037, China
[4]School of Information Engineering, Yangzhou University, 196 Huayang West Road, Hanjiang District, Yangzhou, Jiangsu Province, 225000, China
[5]School of Computer Science and Engineering, Nanjing University of Science and Technology, 200 Xiaolingwei Street, Xuanwu District, Nanjing, Jiangsu Province, 210094, China
[*]Corresponding authors. Dong-Jun Yu, School of Computer Science and Engineering, Nanjing University of Science and Technology, 200 Xiaolingwei Street, Xuanwu District, Nanjing, Jiangsu Province, 210094, China.
E-mail: njyudj@njust.edu.cn; Rui Ye, College of Artificial Intelligence, Nanjing Agricultural University, 666 Binjiang Avenue, Jiangbei New District, Nanjing, Jiangsu Province, 211800, China. E-mail: yerui@njau.edu.cn

Abstract

Accurately identifying protein functions is essential to understand life mechanisms and thus advance drug discovery. Although biochemical experiments are the gold standard for determining protein functions, they are often time-consuming and labor-intensive. Here, we proposed a novel composite deep-learning method, Multi-source Knowledge Fusion for Gene Ontology prediction (MKFGO), to infer Gene Ontology (GO) attributes through integrating five complementary pipelines built on multi-source biological data. MKFGO was rigorously benchmarked on 1522 nonredundant proteins, demonstrating superior performance over 12 state-of-the-art function prediction methods. Comprehensive data analyses revealed that the major advantage of MKFGO lies in its two deep-learning components, handcrafted feature representation–based GO prediction (HFRGO) and protein large language model (PLM)–based GO prediction (PLMGO), which derive handcrafted features and PLM–based features, respectively, from protein sequences in different biological views, with effective knowledge fusion at the decision-level. HFRGO leverages a long short-term memory (LSTM)–attention network embedded with handcrafted features, in which the triplet loss–based guilt-by-association strategy is designed to enhance the correlation between feature similarity and function similarity. PLMGO employs the PLM to capture feature embeddings with discriminative functional patterns from sequences. Meanwhile, another three components provide complementary insights for further improving prediction accuracy, driven by protein–protein interaction, GO term probability, and protein-coding gene sequence, respectively. The source codes and models of MKFGO are freely available at https://github.com/yiheng-zhu/MKFGO.

Keywords: protein function; deep learning; pretrained language model; LSTM-attention network; multi-source knowledge fusion

Introduction

Proteins play a fundamental role in various biological processes, such as enzymatic activity, gene expression regulation, and supporting cell structures [1, 2]. Accurate identification of protein functions is vital to unravel life mechanisms and guide drug design, with functions categorized into three aspects, i.e. molecular function (MF), biological process (BP), and cellular component (CC), under the widely used Gene Ontology (GO) annotation [3]. While biochemical experiments are the gold standard for determining protein functions, they are often labor-intensive and may yield incomplete results, leaving numerous sequenced proteins without known functional annotations [4]. As of March 2025, the UniProt database [5] housed ∼253 million protein sequences, but fewer than 0.1% were annotated with GO terms supported by experimental evidence. To bridge this gap, there is an urgent need to develop efficient computational methods for protein function prediction [6, 7].

The existing function prediction methods can be divided into three categories: template detection–, statistical machine learning–, and deep learning–based methods. In the early stage, template detection–based methods were predominant in function prediction, focusing on identifying templates with similar sequences or structures to the query for functional inference [8]. For example, GoFDR [9] and Blast2GO [10] utilize BLAST alignments [11] to search sequence templates, whereas FINDSITE [12] and COFACTOR [8] employ TM-align [13] to detect structure templates.

An inherent drawback of template detection–based methods is that their prediction accuracy heavily depends on the availability and quality of functional templates. To eliminate this dependence,

Received: March 26, 2025. **Revised:** June 27, 2025. **Accepted:** July 23, 2025

statistical machine learning algorithms have been employed as an alternative. This could be implemented by extracting handcrafted feature representations (e.g. k-mer sequence encoding [14] and position-specific scoring matrix [15]) from protein sequences, which are then processed with statistical machine learning algorithms (e.g. support vector machine [16] and random forest [17]) to train function prediction models, exemplified by GOPred [18], FFPred [19], and GOLabeler [20]. Although these methods complement template detection methods, their prediction accuracy is still insufficient [21]. The major reason is that the used machine learning models fail to derive the deep-level functional patterns buried in feature representations. To partly address these issues, deep learning techniques have emerged in function prediction [22].

The significant advantage of deep learning methods over statistical machine learning methods is that they can capture the sophisticated functional patterns from handcrafted feature representations through designing complex neural networks, such as convolutional neural network (CNN) [23] and long short-term memory (LSTM) [24], with the classical examples of DeepGO [22], DeepGOCNN [25], TALE [26], DeepGOZero [27], FFPred-GAN [28], and AnnoPRO [29]. Moreover, the protein large language models (PLMs), such as ESM2 [30] and ProtTrans [31], are increasingly demonstrating their potential in the feature representation through encoding the primary sequences as the discriminative feature embeddings [32]. Since the PLMs are pretrained on networks with dozens of layers over hundreds of millions of sequences, they learn abundant evolution knowledge related to function, resulting in the encoded feature embeddings containing distinctive functional patterns. Therefore, several function prediction methods directly employ PLMs to generate feature embeddings instead of traditional handcrafted features, which are then fed to neural networks for implementing prediction models. The typical examples include ATGO [33], GAT-GO [34], SPROF-GO [35], TransFew [36], DeepFRI [37], and DeepGO-SE [38].

Despite the great progress, challenges remain. First, the above-mentioned works have entirely replaced handcrafted features with PLM-based features, potentially leading to the incomplete capture of functional patterns. The underlying reason is that PLM-based features focus solely on the view of protein sequence evolution, whereas handcrafted features can extract function-related knowledge from other complementary views, such as protein secondary structure and family. Thus, the effective fusion of the knowledge from handcrafted and PLM-based features remains a significant challenge. Second, most existing function prediction methods derive knowledge from the sequence alone, overlooking other crucial biological data sources [e.g. protein–protein interaction (PPI) network and protein-coding gene] that contain complementary knowledge. Therefore, another challenge lies in the integration of multiple biological data sources to further improve prediction performance.

In this work, we proposed a composite protein function prediction method, Multi-source Knowledge Fusion for Gene Ontology prediction (MKFGO), through integrating five complementary pipelines built on multi-source biological data. First, we designed two deep learning–based GO prediction pipelines, handcrafted feature representation–based GO prediction (HFRGO) and protein large language model (PLM)–based GO prediction (PLMGO), embedded with handcrafted and PLM-based features, respectively, from amino acid sequences. HFRGO leverages the LSTM-attention architecture with three powerful handcrafted features from the views of sequence conversion, secondary structure, and family domain, respectively. Meanwhile, the triplet loss–based guilt-by-association strategy [33] is employed to enhance the correlation between sequence feature similarity and function similarity. PLMGO employs the ProtTrans transformer [31] to encode the sequences as feature embeddings with functional patterns from the view of evolution diversity, which are then decoded by the fully connected neural network. Second, we implemented another three pipelines, driven by PPI inference, naïve probability, and coding-gene sequence. Finally, a composite model was derived by incorporating the outputs of five complementary pipelines. Computational experiments on 1522 nonredundant test proteins have demonstrated two points. First, HFRGO and PLMGO complement each other, with decision-level knowledge fusion outperforming feature-level fusion. Second, the composite MKFGO exhibits a significant advantage in the accurate prediction of GO terms over the existing state-of-the-art approaches, as each of its five components contributes to the overall performance improvement. The source codes and models of MKFGO are freely available at https://github.com/yiheng-zhu/MKFGO.

Materials and methods

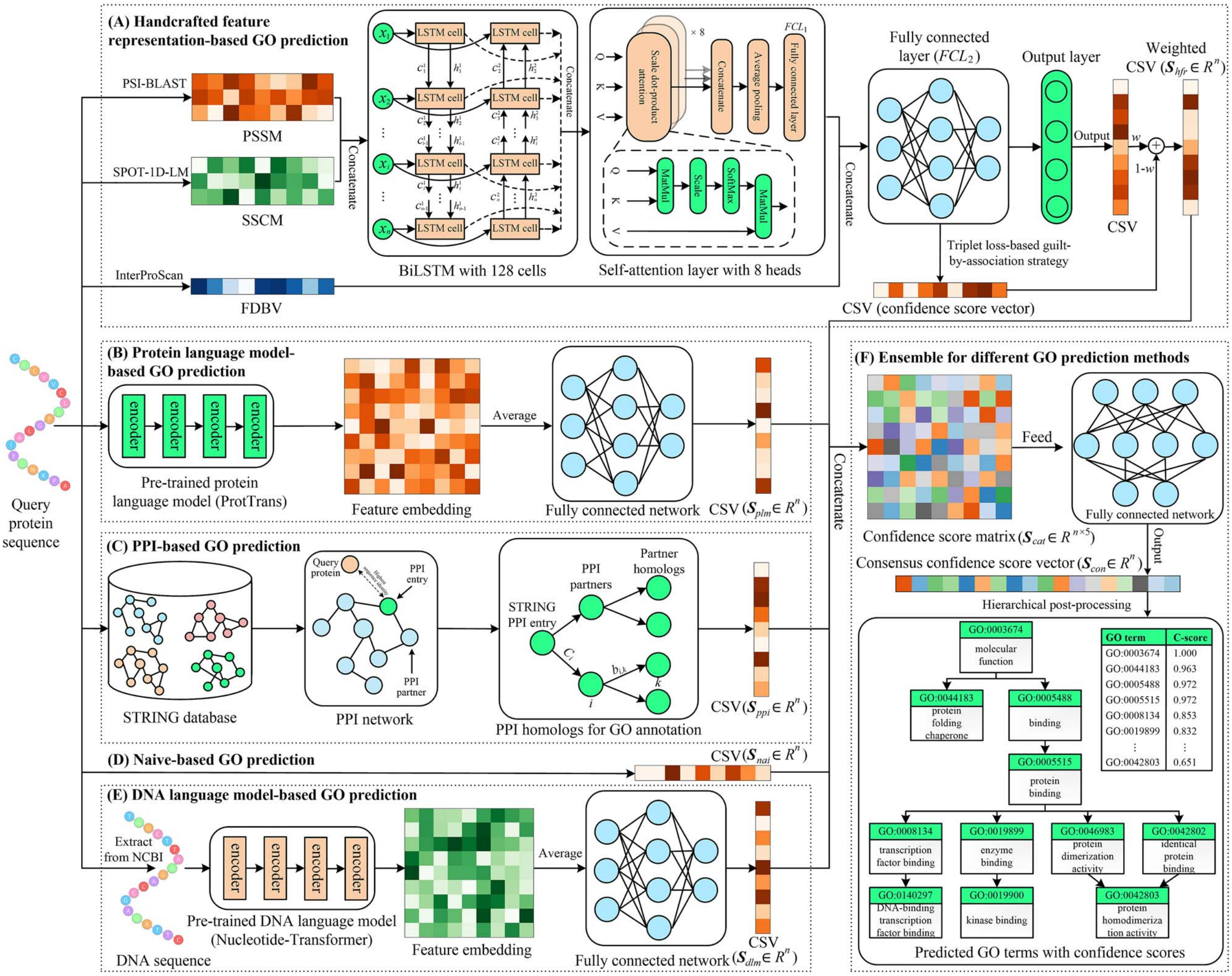
Benchmark datasets

We employed an approach closely mirroring the Critical Assessment of protein Function Annotation (CAFA) experiment to construct benchmark datasets. Specifically, we downloaded all protein sequences from the UniProt database [5] with the corresponding functional annotations from the Gene Ontology Annotation database [39]. Then, we filtered out proteins by only selecting those that have been manually reviewed with the available function annotations by at least one of the eight experimental evidence codes, namely, EXP, IDA, IPI, IMP, IGI, IEP, TAS, and IC [40, 41]. After this, we collected 80 653 high-quality proteins, which could be further split into training, validation, and test datasets. The 1522 proteins were selected as the test datasets, which were released in the UniProt database after 1 July 2021, and the 974 proteins as the validation datasets, released in the UniProt from 1 July 2020 to 30 June 2021. The remaining proteins have been filtered out by removing the redundant proteins aligned with test and validation proteins using CD-HIT [42] software with a sequence identity cut-off of 30%, yielding a training dataset of 70 712 proteins.

The number of entries in each dataset across different GO categories is presented in Table S1 of the Supporting Information (SI). The training, validation, and test datasets were used independently to train models, optimize the models' parameters, and assess the models' performance.

The architecture of Multi-source Knowledge Fusion for Gene Ontology prediction

As depicted in Fig. 1, MKFGO is a composite deep-learning model for protein function prediction, where the input is a protein sequence with UniProt ID, and the output includes the confidence scores of predicted functional terms for three GO aspects. This model consists of five pipelines, i.e. (A) HFRGO, (B) PLMGO, (C) PPI–based GO prediction (PPIGO), (D) naïve-based GO prediction (NAIGO), and (E) DNA language model–based GO prediction (DLMGO), which are driven by the protein sequence (A and B), interaction network (C), GO term probability (D), and coding-gene sequence (E), respectively. The input sequence is independently fed to five pipelines to generate the confidence scores of GO terms, which are further ensembled by the neural network to output the consensus scores.


(A) Handcrafted feature representation-based GO prediction
PSI-BLAST
PSSM
SPOT-1D-LM
SSCM
Concatenate
x_1
x_2
x_i
x_n
LSTM cell
Concatenate
BiLSTM with 128 cells
× 8
Q
K
V
Scale dot-product attention
Concatenate
Average pooling
FCL_1
Fully connected layer
MatMul
Scale
SoftMax
MatMul
Self-attention layer with 8 heads
InterProScan
FDBV
Concatenate
Fully connected layer (FCL_2)
Output layer
Output
w
1-w
CSV
Weighted CSV ($S_{hfr} \in R^n$)
Triplet loss-based guilt-by-association strategy
CSV (confidence score vector)
Query protein sequence
(B) Protein language model-based GO prediction
encoder
Pre-trained protein language model (ProtTrans)
Feature embedding
Average
Fully connected network
CSV ($S_{plm} \in R^n$)
(C) PPI-based GO prediction
STRING database
Query protein
Highest similarity
PPI entry
PPI partner
PPI network
STRING PPI entry
PPI partners
Partner homologs
C_i
$b_{i,k}$
i
k
PPI homologs for GO annotation
CSV ($S_{ppi} \in R^n$)
(D) Naive-based GO prediction
CSV ($S_{nai} \in R^n$)
(E) DNA language model-based GO prediction
Extract from NCBI
DNA sequence
encoder
Pre-trained DNA language model (Nucleotide-Transformer)
Feature embedding
Average
Fully connected network
CSV ($S_{dlm} \in R^n$)
Concatenate
(F) Ensemble for different GO prediction methods
Feed
Fully connected network
Output
Confidence score matrix ($S_{cat} \in R^{n \times 5}$)
Consensus confidence score vector ($S_{con} \in R^n$)
Hierarchical post-processing
GO:0003674 molecular function
GO:0044183 protein folding chaperone
GO:0005488 binding
GO:0005515 protein binding
GO:0008134 transcription factor binding
GO:0019899 enzyme binding
GO:0046983 protein dimerization activity
GO:0042802 identical protein binding
GO:0140297 DNA-binding transcription factor binding
GO:0019900 kinase binding
GO:0042803 protein homodimerization activity
Predicted GO terms with confidence scores

GO term	C-score
GO:0003674	1.000
GO:0044183	0.963
GO:0005488	0.972
GO:0005515	0.972
GO:0008134	0.853
GO:0019899	0.832
⋮	⋮
GO:0042803	0.651

Figure 1. The flowchart of MKFGO.

Handcrafted feature representation-based Gene Ontology prediction

Feature representation

For the query sequence with the length L, we use PSI-BLAST [11], SPOT-1D-LM [43], and InterProScan [44] programs to extract the corresponding feature representations, i.e. position-specific scoring matrix (PSSM), secondary structure coding matrix (SSCM), and family domain–based binary vector (FDBV), with the scale of $L \times 20$, $L \times 8$, and 45899-D, respectively, see details in Text S1 of the SI.

LSTM-attention network-based function prediction

The PSSM and SSCM are concatenated and then fed to an LSTM-attention module consisting of a BiLSTM layer with 128 cells, a self-attention layer with eight heads, an average-pooling layer, and a fully connected layer (FCL_1) with 1024 neurons. The output of this module is concatenated with the FDBV and then processed by another fully connected layer (FCL_2) with 1024 neurons to output a feature embedding vector, as carefully described in Text S2 of the SI.

The feature embedding vector is further fed to the output layer with a Sigmoid activation function to generate a confidence score vector $\mathbf{s}_{sig}$ for predicted GO terms. Meanwhile, the triplet loss–based guilt-by-association (TL-GBA) strategy [33] is performed on this embedding vector to produce another confidence score vector $\mathbf{s}_{gba}$. Finally, two confidence score vectors are weightedly combined to generate the final confidence score vector $\mathbf{s}_{hfr}$ for the HFRGO pipeline.

Triplet loss–based guilt-by-association strategy

For a query protein, we select the top K templates, which have the highest sequence feature similarity with itself, from the training dataset for function annotation:

$$s_{gba}\left(\mathrm{GO}_j\right) = \sum_{k=1}^{K} w_k \bullet \frac{I_k\left(\mathrm{GO}_j\right)}{\sum_{k=1}^{K} w_k}, w_k = 1 - \left(r_k - 1\right)/K \qquad (1)$$

where GO_j is the j-th candidate GO term; $I_k\left(\mathrm{GO}_j\right) = 1$, if the k-th template is associated with GO_j in the experimental annotation; otherwise, $I_k\left(\mathrm{GO}_j\right) = 0$; r_k is the rank of the k-th template in K templates based on the feature similarity with query.

The feature similarity between the template and query is measured by the Euclidean distance of feature embedding vectors outputted by the FCL_2. To improve the quality of selected templates, we employ the triplet loss [45] to enhance the correlation between sequence feature similarity and functional similarity:

$$Loss_t = E_{x \sim X} \max\left(d(x, pos)_{max} + d_m - d(x, neg)_{min},\ 0\right) \qquad (2)$$

where x is a protein sequence in the training dataset X; d_m is a preset margin value; $d(x, pos)_{max}$ ($d(x, neg)_{min}$) is the maximum (minimum) values of distances between x and all positive (negative) partners which have the same (different) function to x. Two proteins are defined as having the same function if their functional similarity is higher than a preset threshold c_f, as carefully described in Text S3. Minimizing this triplet loss helps ensure that the selected templates exhibit both higher feature similarity and functional similarity to the query.

Loss function

Considering that triplet loss is hardly converged in the training stage, we have added the cross-entropy loss to form a composite loss function [46–48]:

$$Loss = \alpha \bullet Loss_t + Loss_c \tag{3}$$

$$Loss_c = -\frac{1}{m} \bullet \frac{1}{n} \bullet \sum_{i=1}^{m}\sum_{j=1}^{n}\Big\{\log s_{sig}(i,j) \bullet I(i,j) + \log\left(1 - s_{sig}(i,j)\right) \bullet \left(1 - I(i,j)\right)\Big\} \tag{4}$$

where α is a balanced parameter, m and n are the numbers of training proteins and GO terms, respectively; $s_{sig}(i,j)$ is the confidence score that the i-th training protein is associated with the j-th GO term predicted by the output layer in LSTM-attention network; and $I(i,j) = 1$ if i-th protein is associated with the j-th term in the experimental annotation. This loss function could be minimized to optimize the hyperparameters of the HFRGO using the Adam optimization algorithm [49]. In addition, the values of d_m, c_f, α, and K for three GO aspects are listed in Table S2 of the SI.

Protein language model–based Gene Ontology prediction

The input sequence is fed to the pretrained protein language model, i.e. ProtTrans [31], to extract the feature embedding matrix, which is then averaged over the full sequence length to generate the embedding vector with 1024 dimensions. This vector is further processed by a fully connected layer with 1024 neurons and an output layer to yield the confidence score vector $\boldsymbol{s}_{plm}$ for GO terms. The cross-entropy loss [see details in Equation (4)] is employed to optimize the hyperparameters of the neural network. Here, we utilize ProtT5-XL-UniRef50, a representative model in the ProtTrans family and pretrained on over 45 million protein sequences from the UniRef50 dataset. This model consists of 24 self-attention layers, each of which is composed of 32 attention heads and a multi-layer perceptron (MLP) with a hidden size of 1024, totaling ∼3 billion parameters. Additional details about ProtTrans can be found in reference [31].

Protein–protein interaction–based Gene Ontology prediction

The Blastp [11] program is utilized to hit a PPI entry P_e, which has the highest sequence identity to the query protein, against the STRING database [50]. For each PPI partner of P_e, the Blastp is employed again with the e-value of 0.1 to search the corresponding homologs from the training sequence dataset. These homology proteins are used as templates to generate the confidence score vector $\boldsymbol{s}_{ppi}$for GO annotations, as carefully described in Text S4.

Naïve-based Gene Ontology prediction

The confidence score that the query is associated with a GO term could be directly assigned by the frequency of this term in the training dataset:

$$s_{nai}\left(GO_j\right) = N\left(GO_j\right)/N_{all} \tag{5}$$

where $N\left(GO_j\right)$ and N_{all} are the number of proteins associated with GO_j and all proteins in the training dataset, respectively.

DNA language model–based Gene Ontology prediction

For the query protein, we download the DNA sequence of its coding gene from the National Center for Biotechnology Information (NCBI) [51] through mapping its UniProt ID to the Entrez ID of the coding gene. This DNA sequence is fed to the pretrained DNA language model, i.e. Nucleotide-Transformer [52], to capture the feature embeddings, then further processed by the fully connected neural network to output the confidence score vector $\boldsymbol{s}_{dlm}$ of predicted GO terms, using the same architecture in the PLMGO pipeline. Here, we employed two model versions of Nucleotide-Transformer, namely, NT-Multispecies (2.5B) and NT-1000G (2.5B), each pretrained on >100 billion nucleotides and capable of encoding the DNA sequence as an embedding vector with 2560-D. These two models share the same architecture, i.e. a transformer encoder with 32 self-attention layers, each comprising 20 attention heads and an MLP with 2560 hidden units, with a total of ∼250 million parameters (see details in reference [52]).

Ensemble for different Gene Ontology prediction methods

The confidence score vectors of five GO prediction pipelines are concatenated as a confidence score matrix $\boldsymbol{S}_{cat} \in R^{n\times 5}$, where n is the number of the candidate GO terms. This matrix is then fed to a fully connected network layer with N_f neurons, followed by an output layer with one neuron to output the consensus confidence score vector $\boldsymbol{s}_{con} \in R^n$. Here, the values of N_f are set to 256, 256, and 32 for MF, BP, and CC aspects, respectively. The implementation details of the ensemble strategy for integrating different GO prediction methods are provided in Text S5, with a schematic diagram shown in Fig. S1. Finally, a hierarchical postprocessing procedure is performed on these confidence scores to ensure that the confidence score of a GO term is larger than or equal to those of all its children, as carefully described in Text S6.

Implementation and settings for training

All MKFGO experiments were performed on the Linux machine, with its three deep-learning components (i.e. HFRGO, PLMGO, and DLMGO) implemented using the TensorFlow framework on an NVIDIA GeForce RTX 4090 GPU. The Adam optimizer [49] with a learning rate of 0.0001 was used to train HFRGO, PLMGO, and DLMGO models with batch sizes of 64, 256, and 256, respectively, over 50, 200, and 200 epochs, with the corresponding training time, inference time, memory usage, and model storage listed in Table S3 of the SI. The time complexity and the number of hyperparameters for three deep learning models are summarized in Table S4.

Evaluation metrics

Following the rules of CAFA competitions, we use three metrics to evaluate our models, i.e. maximum F_1-score (F_{max}), minimum semantic distance (S_{min}), and area under the precision–recall curve (AUPRC) [53, 54]. F_{max} is the highest F-score achieved across all confidence thresholds, offering a single measure of the best trade-off between precision and recall. S_{min} measures the discrepancy between predicted and true GO terms by calculating the semantic distance in the GO hierarchy structure. The AUPRC assesses a model's overall performance in the trade-off between precision and recall over all thresholds. The detailed calculations of these metrics can be found in Text S7.

Table 1. The overall performance of 16 function prediction methods on all 1522 test proteins

	Method	F_{max}			S_{min}			AUPRC			Coverage[f]		
		MF	BP	CC	MF	BP	CC	MF	BP	CC	MF	BP	CC
Single method	Blast-KNN[a,d]	0.642	0.397	0.485	7.77	24.90	8.59	0.346	0.220	0.259	0.832	0.803	0.717
	FunFams[a,d]	0.483	0.311	0.387	9.87	27.24	9.02	0.298	0.141	0.200	0.631	0.599	0.532
	PPIGO[a,d]	0.329	0.273	0.461	11.81	26.74	8.43	0.141	0.126	0.253	0.515	0.558	0.645
	DeepGOCNN[b,e]	0.430	0.296	0.497	11.01	26.67	9.45	0.369	0.204	0.493	1.000	1.000	1.000
	TALE[b,d]	0.457	0.313	0.526	11.19	25.88	8.77	0.397	0.222	0.534	1.000	1.000	1.000
	DeepGOZero [b,d]	0.677	0.396	0.540	7.53	24.86	9.46	0.674	0.319	0.521	1.000	1.000	1.000
	AnnoPRO[b,e]	0.504	0.365	0.535	9.63	25.36	8.67	0.366	0.267	0.504	1.000	1.000	1.000
	HFRGO[b]	0.682	0.412	0.580	7.23	23.91	8.14	0.630	0.340	0.539	1.000	1.000	1.000
	ATGO[c,d]	0.686	0.424	0.607	7.34	23.99	7.87	0.676	0.361	0.625	1.000	1.000	1.000
	DeepGO-SE[c,d]	0.669	0.411	0.573	7.67	24.48	9.44	0.662	0.351	0.600	1.000	1.000	1.000
	DPFunc[c,d]	0.681	0.403	0.583	7.68	24.70	8.08	0.681	0.350	0.585	1.000	1.000	1.000
	PLMGO[c]	0.680	0.424	0.628	7.58	23.95	7.57	0.621	0.355	0.571	1.000	1.000	1.000
Composite method	DeepGOPlus[d]	0.660	0.402	0.574	7.78	24.92	8.56	0.620	0.311	0.517	1.000	1.000	1.000
	TALE+[d]	0.640	0.401	0.581	8.04	24.91	8.37	0.617	0.318	0.550	1.000	1.000	1.000
	ATGO+[d]	0.693	0.430	0.607	7.22	23.88	8.11	0.670	0.371	0.617	1.000	1.000	1.000
	MKFGO	**0.710**	**0.459**	**0.639**	**6.97**	**23.08**	**7.38**	**0.716**	**0.400**	**0.668**	1.000	1.000	1.000

Bold fonts highlight the best performer in each category. [a]Template detection–based methods. [b]Deep learning–based methods with handcrafted feature representations. [c]Deep learning–based methods with PLM-based feature representations. [d]The prediction models are re-trained on our training dataset using the author's source codes. [e]The prediction models are directly downloaded from the author's web platforms. [f]Coverage is the proportion of the number of test proteins with available prediction scores divided by the total number of test proteins.

Results and discussions

Overall performance of Multi-source Knowledge Fusion for Gene Ontology prediction

We benchmarked the proposed methods with 12 state-of-the-art function prediction methods on all 1522 test proteins, including nine single methods (Blast-KNN [20], FunFams [55], DeepGOCNN [25], TALE [26], DeepGOZero [27], ATGO [33], AnnoPRO [29], DeepGO-SE [38], and DPFunc [56]) and three composite methods (DeepGOPlus [25], TALE+ [26], and ATGO+ [33]). These single methods could be categorized into three groups: (i) Blast-KNN and FunFams are template detection–based methods, leveraging sequence homology alignment and protein family search separately; (2) DeepGOCNN, TALE, DeepGOZero, and AnnoPRO are deep learning–based methods with handcrafted feature representations; (3) ATGO, DeepGO-SE, and DPFunc are deep learning methods with PLM-based feature representations. Moreover, DeepGOPlus, TALE+, and ATGO+ are the composite versions for DeepGOCNN, TALE, and ATGO, respectively, through integrating Blast-KNN. Accordingly, our competing methods include the composite MKFGO and its three component methods (i.e. PPIGO, HFRGO, and PLMGO), each corresponding to one of the above-mentioned three groups.

Table 1 summarizes the performance comparison between our methods and 12 existing methods on 1522 test proteins. Overall, the proposed MKFGO achieves the best performance among all function prediction methods. In comparison to the second-best performer, i.e. ATGO+, MKFGO gains 4.8%, 5.3% [= (|6.97–7.22|/7.22 + |23.08–23.88|/23.88 + |7.38–8.11|/8.11)/3 × 100%], and 7.6% average improvement for F_{max}, S_{min}, and AUPRC, respectively, on three GO aspects. Moreover, the composite methods (i.e. ATGO+, TALE+, and DeepGOPlus) all exhibit superior performance compared to their deep-learning counterparts, as BLAST-KNN provides complementary knowledge for function prediction.

Among all single methods, our PLMGO and HFRGO are ranked 4/1/1 and 2/3/4 for MF/BP/CC aspects, respectively. Moreover, HFRGO outperforms all other deep learning methods that use handcrafted feature representations. Taking DeepGOZero as an example, our HFRGO beat it in eight out of nine evaluation metrics, except for the AUPRC in the MF aspect. Importantly, HFRGO consistently outperforms the DeepGO-SE, a PLM-based deep learning method, in terms of F_{max} and S_{min} values across three GO aspects. It is undeniable that ATGO achieves the highest prediction accuracy in MF aspects, likely because it utilizes PLM (i.e. the ESM-1b transformer [57]) to extract feature embeddings from three-level perspectives of sequence evolution, enriching the knowledge related to molecular functions.

Furthermore, we observe that deep learning methods, particularly those employing PLMs, achieve significantly superior performance than template detection–based methods. Part of the reason is that such template detection methods cannot output any prediction results for some test proteins that fail to match available templates, leading to inferior performance in the overall test dataset with low coverage, especially evident in FunFams and PPIGO. Therefore, we conducted an additional benchmark of 16 function prediction methods on a subset of 515 test proteins, for which predictions can be produced by all methods. As illustrated in Table S5, a similar trend is observed, where our methods outperform the control methods by a substantial margin. Meanwhile, BLAST-KNN exhibits noticeably higher prediction accuracy over FunFams and PPIGO in both tests, indicating that sequence homology provides a more reliable basis for protein function inference than PPI and family similarity.

We further assessed the model performance using two additional metrics commonly employed in multi-label prediction tasks, i.e. information content–weighted area under the receiver operating characteristic curve (ICW-AUROC) [58] and Hamming loss [59], with the details in Text S8. The benchmark results of 16 GO prediction methods on 1522 test proteins concerning these two metrics are summarized in Fig. S2. Again, MKFGO achieved the best performance across both metrics, except for the ICW-AUROC value in the MF aspect, where it ranked second with a negligible margin behind the top performer.

Multi-source Knowledge Fusion for Gene Ontology prediction shows great generality to new species and non-homologous proteins

Despite the progress in function prediction, many deep learning methods may exhibit reduced performance on proteins from new

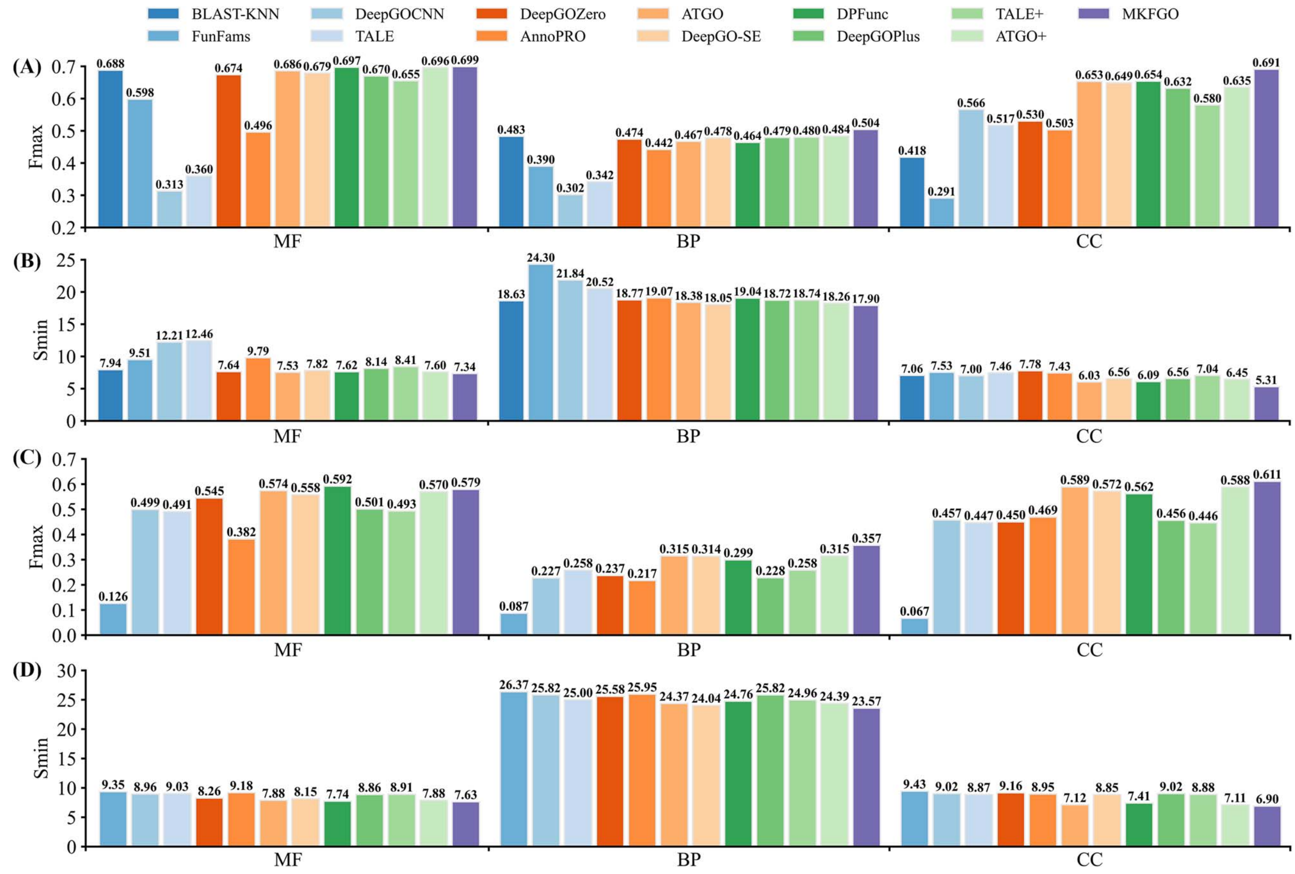


Figure 2. The performance comparison among 13 function prediction methods on the new species and nonhomology proteins across three GO aspects. (A) The $F_{\max}$ values on the 300 test proteins from 158 new species. (B) The $S_{\min}$ values on the 300 new species proteins. (C) The $F_{\max}$ values on the 305 non-homologous test proteins. (D) The $S_{\min}$ values on the 305 non-homologous proteins.

species absent from their training data. To assess the generalizability of MKFGO to new species, we mapped each protein in our dataset to its corresponding species and then gathered 300 test proteins from 158 new species that were never observed in the training dataset.

We further benchmarked the proposed MKFGO with 12 competing function prediction methods on these 300 new species proteins, where the corresponding $F_{\max}$ and $S_{\min}$ values across all GO aspects are shown in Fig. 2A and B. Meanwhile, the AUPRC values of 13 methods are listed in Fig. S3. It could be found that MKFGO achieves the best $F_{\max}$ and $S_{\min}$ values among all methods. Compared to the second-best performer, i.e. DPFunc, MKFGO gains an average improvement of 4.9% in $F_{\max}$ and 7.5% in $S_{\min}$, respectively, on three GO aspects. As for AUPRC, MKFGO is ranked 2/4/1 for the MF/BP/CC aspect. Moreover, the AUPRC gap between MKFGO and the top performer is minimal and almost negligible on the MF/BP aspect. Additionally, the $F_{\max}$, $S_{\min}$, and AUPRC values of MKFGO for these 300 proteins in Fig. 2A and B are largely consistent with those of the entire test dataset in Table 1. These observations demonstrate that MKFGO maintains its strong performance when modeling new species proteins, highlighting the generalizability of its deep-learning approaches.

Since the sequence-based function annotation is heavily dependent on sequence homology, another challenge for deep learning–based methods is the modeling of proteins without sequence homology. In light of this, we further benchmarked MKFGO with 11 competing methods on 305 test proteins that cannot hit any sequence homologies in the training dataset using BLAST search with an e-value of 0.01. Here, BLAST-KNN was excluded because it cannot generate any predictions for these test proteins. Fig. 2C and D summarizes the $F_{\max}$ and $S_{\min}$ values of 12 function prediction methods for three GO aspects on 305 nonhomology test proteins, where the corresponding AUPRC values are illustrated in Fig. S4. Overall, the performance of all prediction methods for these nonhomology proteins in Fig. 2C and D is significantly inferior to that for the whole test dataset in Table 1. This observation further demonstrates the importance of sequence homology in function prediction, both for template detection and deep learning–based methods. However, our MKFGO still outperforms the other 11 methods for all evaluation metrics across three GO aspects, except for the $F_{\max}$ and AUPRC values on the MF aspect. Taking ATGO+ as a reference, MKFGO achieves an improvement of 1.6%, 13.3%, and 3.9% on the $F_{\max}$ and 3.2%, 3.4%, and 3.0% on the $S_{\min}$ for MF, BP, and CC aspects, respectively.

Contribution analysis for different Gene Ontology prediction components

To analyze the contributions of five component methods (i.e. HFRGO, PLMGO, PPIGO, NAIGO, and DLMGO) in MKFGO, we individually remove each component from the MKFGO to generate five reduced-composite methods, including PIND (PLMGO + PPIGO + NAIGO + DLMGO), HIND (HFRGO + PPIGO + NAIGO + DLMGO), HPND (HFRGO + PLMGO + NAIGO + DLMGO), HPID (HFRGO + PLMGO + PPIGO + DLMGO), and HPIN (HFRGO + PLMGO + PPIGO + NAIGO). Here, "+" means that the component

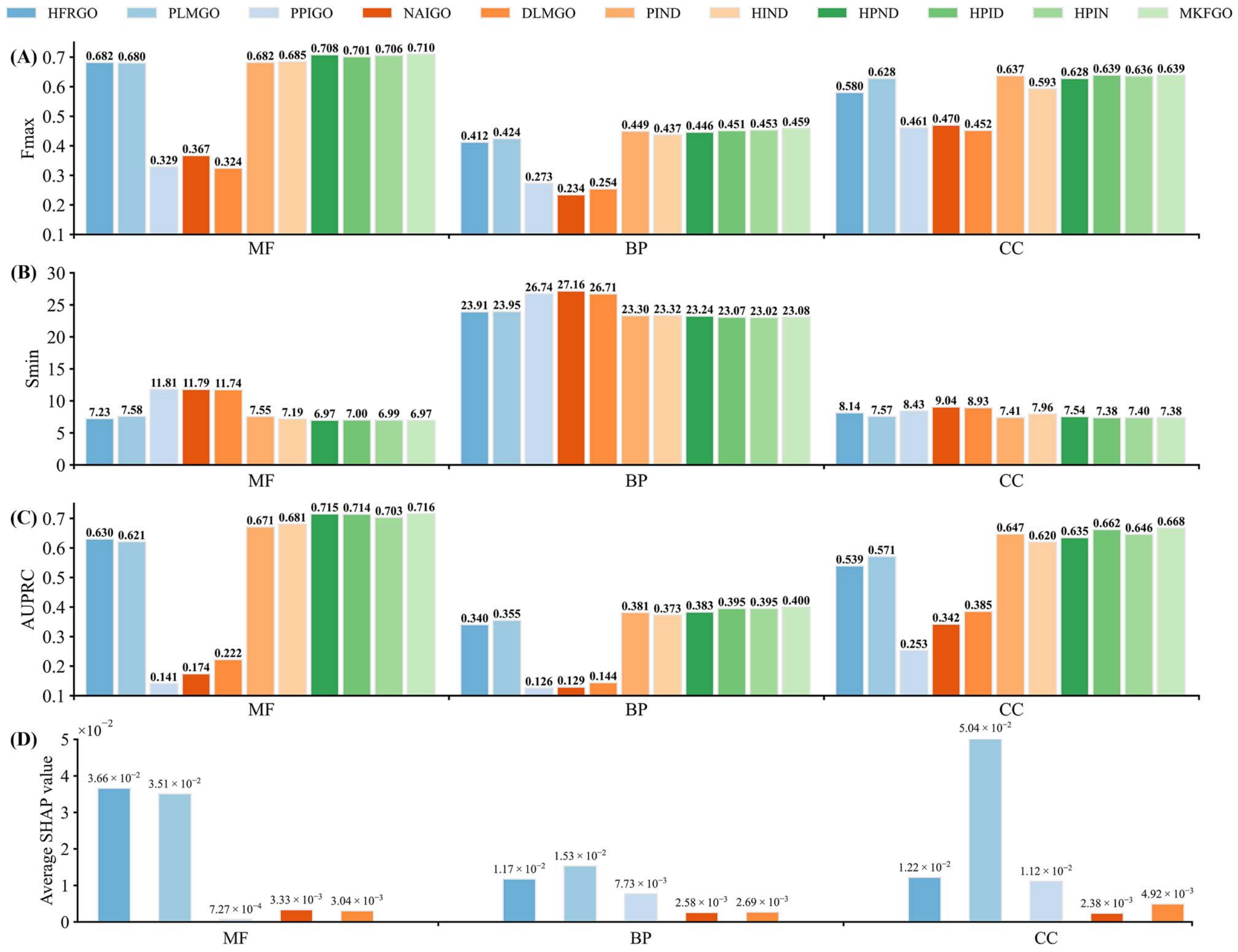


Figure 3. The performance comparison between MKFGO and its 10 component and reduced-composite methods on all 1522 test proteins across three GO aspects. (A) The $F_{\max}$ values. (B) The $S_{\min}$ values. (C) The AUPRC values. (D) The average SHAP values.

methods are ensembled using a fully connected neural network. We further benchmarked MKFGO with its component and reduced-composite methods on all 1522 test proteins, as summarized in Fig. 3A–C.

It could be found that MKFGO yields the best performance across all evaluation metrics among the 11 prediction methods in the three GO aspects, except for the $S_{\min}$ value in the BP aspect. In terms of $F_{\max}$, for example, MKFGO shares an average increase of 2.2%, 5.5%, 1.6%, 1.0%, and 0.8%, respectively, on three GO aspects, in comparison to five reduced-composite methods, i.e. PIND, HIND, HPND, HPID, and HPIN. This observation demonstrates that the five components both help to improve the prediction accuracy of MKFGO, indicating that they could provide complementary knowledge for function prediction.

Among the five individual components, HFRGO and PLMGO are the top two performers, ranking 1/2/2 and 2/1/1 on MF/BP/CC aspects, respectively, with a small margin, on the balance of $F_{\max}$, $S_{\min}$, and AUPRC values. Moreover, these two methods achieve a significant performance advantage over the other three components, primarily because they employ powerful feature representations that extract rich, functionally relevant information directly from amino acid sequences. Since protein sequences are the primary determinants of molecular function, sequence-based models benefit from a more direct and informative signal. In contrast, the other three components rely on indirect sources of functional information (i.e. PPIs, GO term priors, and gene sequences), which, although complementary, tend to be less discriminative and less consistently aligned with functional outcomes. After individually removing five components from MKFGO, the first and second largest performance decreases occur in HIND and PIND, with average decreases of 5.3% and 3.4%, respectively, for three evaluation metrics across all GO aspects. These data show that HFRGO and PLMGO make the most contributions to MKFGO, further demonstrating that the handcrafted and PLM-based features from sequences have nearly equal efficacy for function prediction.

To better understand the contribution of the five components within the MKFGO framework, we employed the Shapley Additive Explanations (SHAP) [60] for model interpretability. Specifically, for each component, the SHAP value, denoted as sv_{ij}, were calculated to estimate its marginal contribution to the final prediction of whether a protein P_i is associated with a GO term GO_j, effectively quantifying how much the inclusion of that component influences the fused output score for the GO term. To evaluate the overall importance of each component across the whole test dataset, we aggregated its SHAP values by calculating the mean absolute SHAP value over all protein–GO term pairs:

$$sv_{all} = \frac{1}{n_t \times n} \sum_{i=1}^{n_t} \sum_{j=1}^{n} |sv_{ij}| \tag{6}$$

where n_t and n are the number of test proteins and candidate GO terms, respectively.

Figure 3D shows the average SHAP values for the five components of MKFGO across three GO aspects. Overall, HFRGO and PLMGO are the top two performers, ranking in 1/2, 2/1, and 2/1 for MF, BP, and CC aspects, respectively. This aligns with the observations in Fig. 3A–C, further confirming that the two sequence-based deep learning methods contribute the most to MKFGO's predictions.

It cannot escape our notice that DLMGO exhibits a limited contribution to MKFGO's prediction, as previously suggested by the SHAP-based interpretability analysis. To statistically assess whether this weak contribution is consistent, we conducted a two-sided Student's t-test [61] to compare the performance of MKFGO and HPIN (i.e. MKFGO excluding DLMGO) on the test dataset, with each model executed five times. The P-values for $F_{\max}$ and AUPRC were all below 3.9×10^{-2} across the three GO aspects, indicating that the observed differences are statistically significant. This reduced contribution is likely attributed to limited functional information embedded in DNA sequences of protein-coding genes. Nevertheless, the DLMGO module shows the great potential in non-coding gene function prediction, a task biologically related to protein function prediction, as discussed in Text S9 of the SI, with the experimental results in Table S6.

Considering that structural information has been widely explored for protein function prediction, we further investigated whether incorporating such information could enhance prediction accuracy within the MKFGO framework. To this end, we designed and evaluated two structure-aware extensions, based on structure alignment and the graph convolutional network [62], respectively. Their details are provided in Text S10, with model architectures and performance comparisons in Figs. S5, S6, and S7 and Table S7. Experimental results demonstrate that neither of these two structure-aware extensions led to any performance gain when integrated into MKFGO, either as additional modules or as replacements for existing components. This is mainly because the functional patterns captured by the current structure-aware extensions can be fully recovered by the five components in MKFGO, which leverage both deep learning techniques and large pretrained language models on multi-source biological data, resulting in redundancy rather than complementarity. Furthermore, since all structural information was obtained from predicted models by AlphaFold2 [63] and ESMFold [30], potential inaccuracies in these structures may introduce noise into structure-based modeling, thereby limiting the effectiveness of these extensions.

Decision-level fusion analysis

Performance comparison between different ensemble techniques

To examine the efficacy of the utilized fully connected neural network (FCNN) for integrating five components of MKFGO, we benchmarked it with three commonly used ensemble techniques, namely, logistic regression (LR), weighted voting (WV), and weighted product (WP), with the details in Text S11 of the SI. Specifically, for each GO term, the corresponding confidence scores of all components of MKFGO could be incorporated as a consensus score using one of the above four ensemble techniques.

Figure 4A illustrates the performance comparison between four ensemble techniques on all 1522 test proteins. Our FCNN achieves the best performance among the four techniques, with an average 1.2% increase in $F_{\max}$ values compared to the second-best performer, i.e. LR. Concerning $S_{\min}$ and AUPRC values, the FCNN demonstrates superior performance to LR at least on two GO aspects. Regarding WV and WP, the FCNN consistently outperforms across all evaluation metrics in all three GO aspects. It is worth noting that WP exhibited the poorest performance, even falling below that of the individual component method in Table 1 on the MF aspect. This finding highlights the important role of the ensemble technique in composite function prediction.

Superiority of decision-level fusion over feature-level fusion

In MKFGO's pipeline, we fused the knowledge buried in handcrafted and PLM-based features at the decision level rather than the feature level. The major reason is that incorporating too many features into a single neural network may lead to a learning bias toward certain features while overlooking other crucial ones. Moreover, such a network may not effectively handle the redundancy among multiple features. To demonstrate this point, we designed three control methods, represented as CM1, CM2, and CM3, as follows:

- **CM1:** The combination of HFRGO and PLMGO at the feature level, where the PLM-based features from PLMGO are incorporated into the HFRGO architecture via feature concatenation (see Fig. S8 for architectures).
- **CM2:** The combination of HFRGO and PLMGO at the decision level. For each GO term, the corresponding confidence scores predicted by HFRGO and PLMGO are ensembled as a consensus score using the neural network, consisting of a fully connected layer with 256 neurons and an output layer with 1 neuron.
- **CM3:** The combination of CM1 and the other three components of MKFGO at the decision level. Specifically, the prediction results of M1, PPIGO, NAIGO, and DLMGO are ensembled using the same fully connected neural network in the CM2.

Figure 4B summarizes the performance comparison between MKFGO and the above three control methods on all 1522 test proteins. It could be observed that CM2 consistently outperforms CM1 for three metrics on all GO aspects. Notably, the performance of CM1 in terms of $F_{\max}$, $S_{\min}$, and AUPRC on the CC aspect in Fig. 4B is even inferior to that of PLMGO alone, as reported in Table 1. Moreover, after integrating M1 with the other three components (i.e. PPIGO, NAIGO, and DLMGO), the CM3 still underperforms MKFGO across all evaluation metrics, except for the $S_{\min}$ value on the BP aspect. These data demonstrate that decision-level fusion provides a more effective strategy than feature-level fusion for integrating knowledge from handcrafted and PLM-based features.

We further conducted a hyperparameter sensitivity analysis of the decision-level fusion module and examined the impact of batch size on loss convergence in the MFKGO framework, as detailed in Texts S12 and S13, with corresponding experimental results presented in Figs. S9 and S10. These data demonstrate that MKFGO exhibits consistently strong performance and reliable convergence behavior across a wide range of fusion parameters and batch size configurations.

Ablation study for handcrafted feature representation–based GO prediction

Contribution analysis of algorithmic modules

We conducted an ablation experiment to analyze the contributions of algorithmic innovations in HFRGO to its enhanced performance. Beginning with the HFRGO model (M0), we gradually remove algorithmic components. First, we remove the TL-GBA module (Module I) from M0 to build the model M1; Then, we

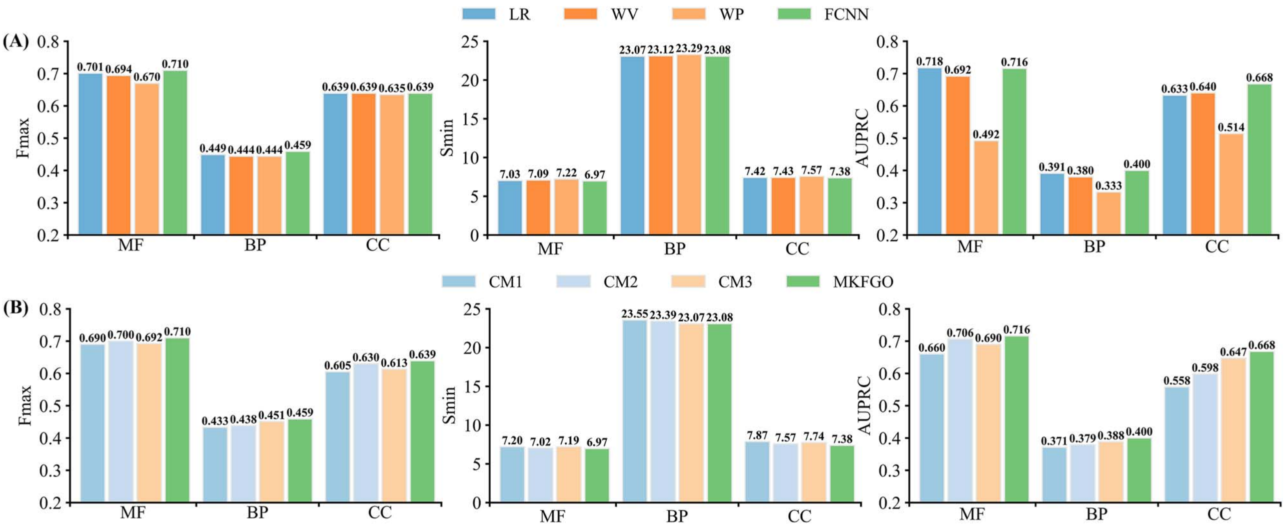


Figure 4. Detailed analysis in decision-level fusion. (A) The performance comparison between four ensemble techniques for incorporating all components of MKFGO on all 1522 test proteins, where FCNN is employed in MKFGO. (B) The performance comparison between four control methods employing either feature-level fusion or decision-level fusion on all 1522 test proteins.

individually exclude the [PSSM + SSCM + LSTM-attention layer] (Module II) and [FDBV + fully connected layer] (Module III) from M1 to develop the other two ablation models (M2 and M3), with the architectures in Fig. S11.

Figure 5A illustrates the performance comparison between four ablation models on all 1522 test proteins. Compared with M0, M1 without Module I exhibits reduced performance, with the average decrease of 1.1%, 1.2%, and 3.5% for $F_{\max}$, $S_{\min}$, and AUPRC, respectively, on three GO aspects. After individually removing Modules II and III from M1, the performance of M2 and M3 continuously drops. Taking M3 as an example, it shows inferior performance across all evaluation metrics except for the AUPRC values on MF and CC aspects, in comparison with M1. These data indicate that each of the three modules helps enhance the overall performance of HFRGO.

Interpreting self-attention via functional domain alignment

To better understand how HFRGO makes predictions, we performed a residue-level attention weight analysis in relation to functional domains. For each protein, attention weights were averaged across all heads to obtain per-residue attention distributions, and the top 20 residues with the highest weights were selected. Functional domains were annotated using InterProScan, and we then calculated the proportion of these top-attention residues falling within domain regions. Based on this, we defined the attention-domain overlap rate as the percentage of proteins in which at least 90% of the top 20 attention residues are located within the annotated functional domains.

For MF, BP, and CC aspects, the overlap rates are 58.4%, 20.1%, and 36.9%, respectively. The higher overlap observed for MF can be attributed to the nature of InterProScan annotations, which primarily capture conserved domains directly related to molecular functions, such as catalytic or binding regions. In contrast, BP terms are often involved in complex regulatory pathways and multi-protein interactions that are less located within specific conserved domains, leading to lower overlap. CC terms, related to protein localization, partially depend on domain-related structural signals, leading to intermediate alignment. These findings suggest that the self-attention mechanism in HFRGO is capable of capturing biologically meaningful regions, particularly those associated with molecular function.

Attention weight visualization for model interpretability

To further enhance the HFRGO's model interpretability, we visualized residue-level attention weight distributions for two representative proteins from the test set, with the UniProt IDs of O24527 and Q9LSC4. For each protein, we plotted attention weight distributions from two ablation models (i.e. M1 and M3) under MF prediction, with each distribution aligned to the functional domain regions annotated by InterProScan, as shown in Fig. 5B. M1 integrates both the self-attention mechanism and InterProScan-derived domain features, while M3 relies solely on self-attention. Additionally, the predicted MF terms for both models on these two proteins are listed in Table S8.

For protein O24527, the M3 model exhibits a prominent attention peak that falls entirely within the InterProScan-annotated domain region (positions 249–503), indicating strong consistency between the deep learning model's learned attention and biologically defined functional regions. This alignment explains why M3 and M2 (which only uses InterProScan-derived domain features) predict exactly the same set of GO terms, both correctly identifying all 10 terms. Furthermore, in the M1 model, which integrates both self-attention and domain features, the attention peak remains within the domain region without significant deviation while maintaining the same prediction accuracy.

For protein Q9LSC4, the M3 model shows one major and one minor attention peak. While the major peak aligns with the InterProScan-annotated domain (positions 129–322), the minor peak falls outside. This misalignment contributes to the inconsistent GO term predictions between M3 and M2. In contrast, the M1 model displays a similar overall peak position; however, the attention weight of the previously minor peak is noticeably amplified. This suggests that integrating self-attention with InterProScan-derived features enables M1 to capture additional functional signals beyond annotated domains. As a result, M1 achieves significantly improved prediction performance, correctly identifying all nine GO terms, where four terms were missed by both M2 and M3.

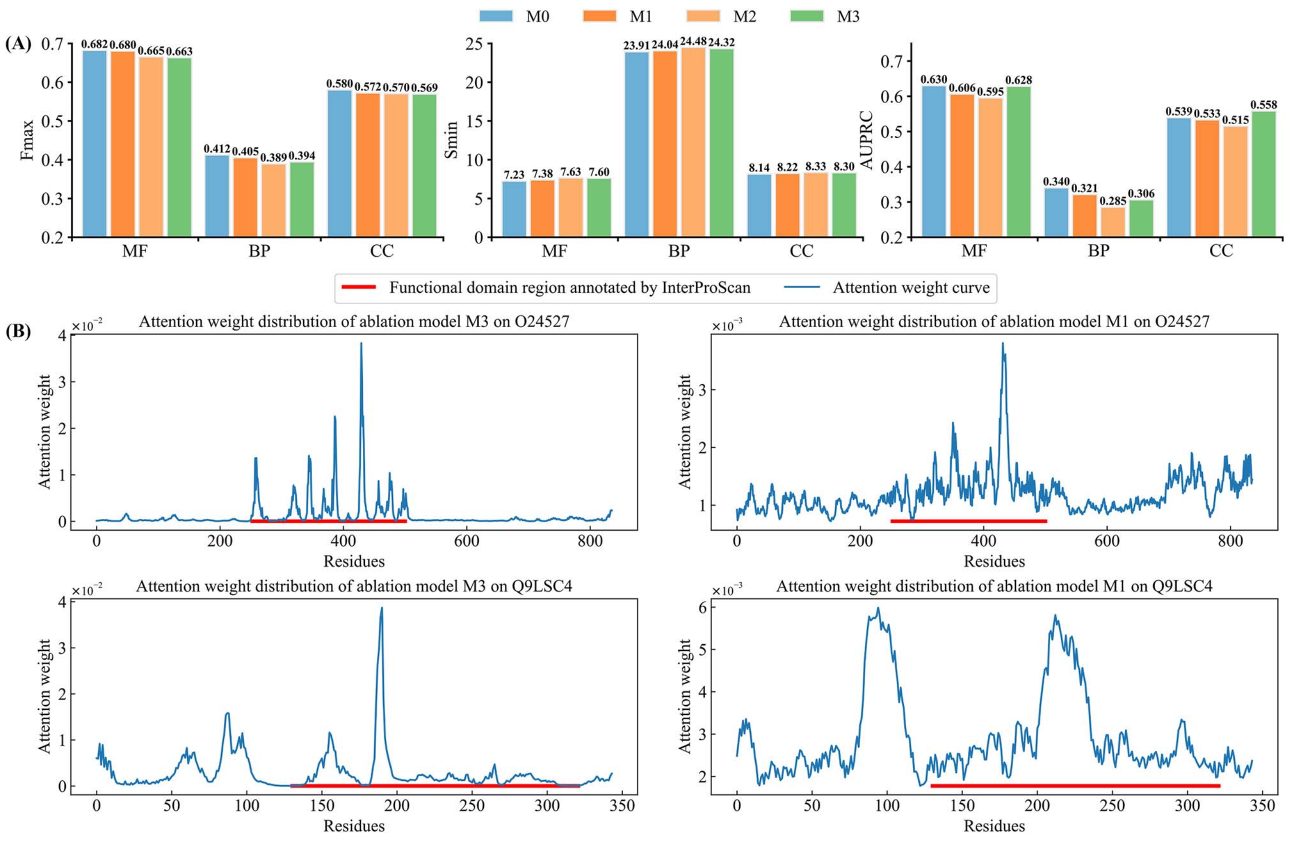


Figure 5. Detailed analysis in the ablation study of HFRGO. (A) The performance comparison between four ablation models on all 1522 test proteins. (B) Attention weight visualization for two representative test proteins.

These results demonstrate that the attention weights learned by the deep learning model are not only consistent with annotated functional domains but also complementary to them. This also helps explain the comparable performance of M2 and M3 in Fig. 5A, as each model captures distinct yet partially overlapping functional signals. By integrating both information sources, the M1 model achieves improved prediction accuracy in GO term annotation.

Case study

To further investigate the effects of different GO prediction methods, three representative proteins from our test dataset were selected for illustration, with the UniProt IDs of A0A2L2DDE6, Q8I2J3, and Q7Q2T8. These proteins are associated with 25, 14, and 7 GO terms, respectively, in the experimental annotation for the BP aspect, excluding the root term (GO:0008150, biological process). Table 2 shows the performance comparison between MKFGO, its five components, and ATGO+ (i.e. the second-best performer in Table 1) on three representative proteins. Meanwhile, the correctly predicted GO terms (i.e. true positives) for these seven methods are visualized as directed acyclic graphs in Fig. 6. Moreover, the mistakenly predicted terms (false positives) for each method are listed in Table S9. It is worth noting that the predicted GO terms for different methods are determined by their respective cut-off setting to maximize the F_1-score.

These data reveal several interesting insights. Overall, MKFGO is the best performer with the highest F_1-score among all seven GO prediction methods across three cases. In A0A2L2DDE6, HFRGO and PLMGO predict nearly the same number of true positives, with 20 and 21 GO terms, respectively, significantly outperforming the other three component methods (PPIGO, NAIGO, and DLMGO). Importantly, among the five component methods, either PLMGO or HFRGO can correctly predict these five GO terms: GO:0051715, GO:0019835, GO:0044179, GO:0009620, and GO:0050832. After incorporating five components, MKFGO successfully inherits all of the 23 true positives. In Q8I2J3, five components gain a total of 9 true positives, where only NAIGO and PPIGO separately correctly identify GO:0044237 and GO:0006508 with confidence scores of 0.208 and 1.000, respectively. As a result, the composite MKFGO yields 8 true positives without false positives. It cannot escape our notice that the GO:0044237 is excluded from the modeling results of MKFGO. The underlying reason is that the low confidence score of 0.208 from NAIGO is further diluted to 0.078, falling below the cut-off value of MKFGO, after decision-level fusion. Occasionally, the main contributors (HFRGO, PLMGO, and PPIGO) cannot provide any true positive GO terms, as observed in the case of Q9SV19, listed in Table S10. Even in this case, MKFGO can inherit part of the predictions from NAIGO and DLMGO, maintaining acceptable performance. These cases demonstrate that the complementary functional knowledge embedded in different component methods can be effectively integrated into MKFGO.

Sometimes, one component method can capture all true positives yielded by other methods. Taking Q7Q2T8 as an example, HFRGO correctly hit all seven GO terms, covering the true positives of the other four components. Other examples include Q9KG76, A0A1D5RMD1, and J9VWW9, in which the PLMGO, PPIGO, and DLMGO could individually encompass all true positives predicted by other components, as listed in Tables S11, S12, and S13. Even in these cases, the final MKFGO can effectively integrate all true positives from the five components with the least false positives, leading to further improved performance.

Table 2. The modeling results of MKFGO in comparison with six competing GO prediction methods on three representative cases in BP prediction

Method	A0A2L2DDE6			Q8I2J3			Q7Q2T8		
	TP	FP	F_1-score	TP	FP	F_1-score	TP	FP	F_1-score
HFRGO	20	5	0.800	7	11	0.438	7	5	0.737
PLMGO	21	0	0.913	5	1	0.500	2	2	0.364
PPIGO	0	0	0.000	7	7	0.500	0	0	0.000
NAIGO	3	33	0.098	7	29	0.280	5	31	0.233
DLMGO	0	0	0.000	5	9	0.357	5	18	0.333
ATGO+	23	9	0.807	6	0	0.600	7	10	0.583
MKFGO	23	1	**0.939**	8	0	**0.727**	7	2	**0.875**

TP, the number of correctly predicted GO terms; FP, the number of mistakenly predicted GO terms. Bold fonts highlight the best performer in each category.

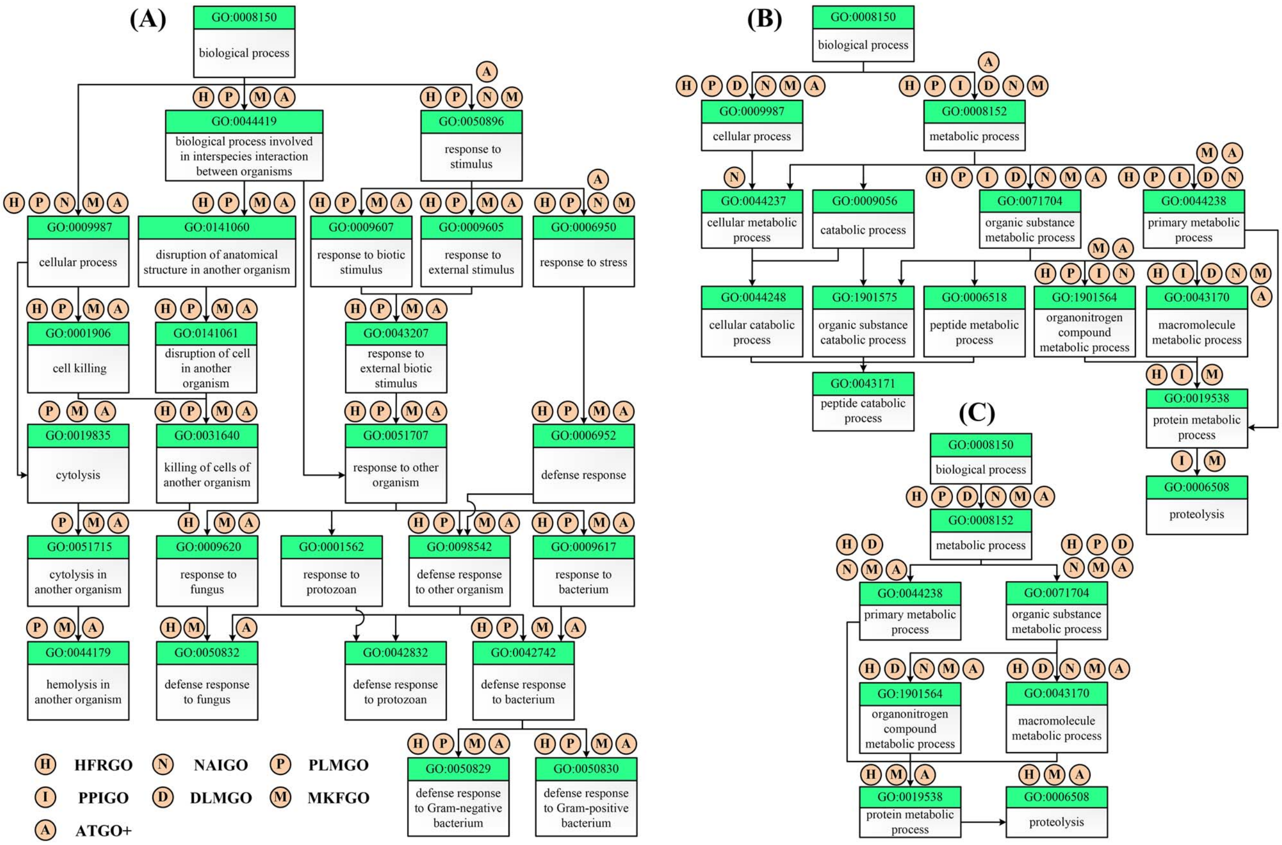

(A)
GO:0008150 biological process
GO:0044419 biological process involved in interspecies interaction between organisms
GO:0050896 response to stimulus
GO:0009987 cellular process
GO:0141060 disruption of anatomical structure in another organism
GO:0009607 response to biotic stimulus
GO:0009605 response to external stimulus
GO:0006950 response to stress
GO:0001906 cell killing
GO:0141061 disruption of cell in another organism
GO:0043207 response to external biotic stimulus
GO:0019835 cytolysis
GO:0031640 killing of cells of another organism
GO:0051707 response to other organism
GO:0006952 defense response
GO:0051715 cytolysis in another organism
GO:0009620 response to fungus
GO:0001562 response to protozoan
GO:0098542 defense response to other organism
GO:0009617 response to bacterium
GO:0044179 hemolysis in another organism
GO:0050832 defense response to fungus
GO:0042832 defense response to protozoan
GO:0042742 defense response to bacterium
GO:0050829 defense response to Gram-negative bacterium
GO:0050830 defense response to Gram-positive bacterium
H HFRGO
N NAIGO
P PLMGO
I PPIGO
D DLMGO
M MKFGO
A ATGO+
(B)
GO:0008150 biological process
GO:0009987 cellular process
GO:0008152 metabolic process
GO:0044237 cellular metabolic process
GO:0009056 catabolic process
GO:0071704 organic substance metabolic process
GO:0044238 primary metabolic process
GO:0044248 cellular catabolic process
GO:1901575 organic substance catabolic process
GO:0006518 peptide metabolic process
GO:1901564 organonitrogen compound metabolic process
GO:0043170 macromolecule metabolic process
GO:0043171 peptide catabolic process
GO:0019538 protein metabolic process
GO:0006508 proteolysis
(C)
GO:0008150 biological process
GO:0008152 metabolic process
GO:0044238 primary metabolic process
GO:0071704 organic substance metabolic process
GO:1901564 organonitrogen compound metabolic process
GO:0043170 macromolecule metabolic process
GO:0019538 protein metabolic process
GO:0006508 proteolysis

Figure 6. The directed acyclic graph of GO terms in the BP aspect for three representative cases. The circles above each GO term represent prediction methods, where a circle filled with "X" on GO term "Y" signifies that method "X" correctly predicts term "Y." (A) A0A2L2DDE6. (B) Q8I2J3. (C) Q7Q2T8.

Conclusion

We developed a novel composite deep learning method, MKFGO, to predict protein functions using the integration of five GO prediction pipelines built on multi-source biological data. Large-scale benchmarking on 1522 nonredundant test proteins demonstrated that MKFGO consistently outperforms 12 existing state-of-the-art methods in GO prediction accuracy. The performance advantage of MKFGO mainly stems from several advancements. First, two deep-learning component methods, HFRGO and PLMGO, could capture the function-related knowledge from protein sequences in different views, with effective knowledge fusion at the decision level. Specifically, HFRGO derived three handcrafted features from the views of sequence conversion, secondary structure, and family domain, which are then associated with function prediction through integrating the designed LSTM-attention network with the TL-GBA strategy. PLMGO employs the ProtTrans transformer to encode the sequences into feature embeddings with evolution diversity, then decoded by the fully connected neural network. Second, another three components, driven by PPI, GO term probability, and coding-gene sequence, provide complementary knowledge for function prediction.

Despite the promising prediction performance, there remains significant potential for further improvements. First, the confidence scores from the five component methods are merged into a consensus score using a simple one-layer fully connected neural network. However, employing a more advanced deep learning approach could further enhance the integration of confidence scores. Second, 3D structural information remains a promising direction for protein function prediction. The GCN-based models explored in this study are relatively simple, and more advanced graph neural network architectures will be investigated to better capture structural patterns. Research in these areas is currently ongoing.

Key Points

- Accurate determination of protein functions is crucial for understanding life mechanisms and advancing drug discovery. This study has developed MKFGO, a novel composite deep learning model, to predict GO terms of proteins by integrating five complementary pipelines built on multi-source biological data.
- Experimental results demonstrate that MKFGO significantly outperforms existing state-of-the-art methods in GO prediction accuracy. The key strength of MKFGO lies in its two deep learning components, HFRGO and PLMGO, which extract functional knowledge from protein sequences in different views, with effective knowledge fusion at the decision level.
- HFRGO leverages an LSTM-attention network embedded with handcrafted features, in which a TL-GBA strategy is designed to strengthen the correlation between feature similarity and function similarity. PLMGO utilizes the ProtTrans transformer to encode the sequences into feature embeddings with evolution diversity, which are then decoded using a fully connected neural network.

Acknowledgements

We thank Prof. Zhiwei Ji, Dr. Fan Fei, and Dr. Xiao Han for the insightful discussions during the lab meeting.

Supplementary data

Supplementary data are available at *Briefings in Bioinformatics* online.

Funding

This work is supported by the National Natural Science Foundation of China (No. 62402227, No. 62306142, and No. 62372234), Fundamental Research Funds for the Central Universities (YDZX2025024), Jiangsu Funding Program for Excellent Postdoctoral Talent (No. 2023ZB224), Natural Science Foundation of the Higher Education Institutions of Jiangsu Province of China under grant 24KJB520041.

Code availability

The source codes and models can be freely downloaded at https://github.com/yiheng-zhu/MKFGO.

References

1. Eisenberg D, Marcotte EM, Xenarios I. *et al.* Protein function in the post-genomic era. *Nature* 2000;**405**:823–6. https://doi.org/10.1038/35015694.
2. Karplus M, Kuriyan J. Molecular dynamics and protein function. *Proc Natl Acad Sci* 2005;**102**:6679–85. https://doi.org/10.1073/pnas.0408930102.
3. G. O. Consortium. The gene ontology resource: 20 years and still GOing strong. *Nucleic Acids Res* 2019;**47**:D330–8. https://doi.org/10.1093/nar/gky1055.
4. Peng J, Xue H, Wei Z. *et al.* Integrating multi-network topology for gene function prediction using deep neural networks. *Brief Bioinform* 2020;**22**:2096–105. https://doi.org/10.1093/bib/bbaa036.
5. U. Consortium. UniProt: A hub for protein information. *Nucleic Acids Res* 2015;**43**:D204–12. https://doi.org/10.1093/nar/gku989.
6. Franz M, Rodriguez H, Lopes C. *et al.* GeneMANIA update 2018. *Nucleic Acids Res* 2018;**46**:W60–4. https://doi.org/10.1093/nar/gky311.
7. Urzúa-Traslaviña CG, Leeuwenburgh VC, Bhattacharya A. *et al.* Improving gene function predictions using independent transcriptional components. *Nat Commun* 2021;**12**:1464. https://doi.org/10.1038/s41467-021-21671-w.
8. Zhang C, Freddolino PL, Zhang Y. COFACTOR: Improved protein function prediction by combining structure, sequence and protein–protein interaction information. *Nucleic Acids Res* 2017;**45**:W291–9. https://doi.org/10.1093/nar/gkx366.
9. Gong Q, Ning W, Tian W. GoFDR: A sequence alignment based method for predicting protein functions. *Methods* 2016;**93**:3–14. https://doi.org/10.1016/j.ymeth.2015.08.009.
10. Conesa A, Götz S, García-Gómez JM. *et al.* Blast2GO: A universal tool for annotation, visualization and analysis in functional genomics research. *Bioinformatics* 2005;**21**:3674–6. https://doi.org/10.1093/bioinformatics/bti610.
11. Altschul SF, Madden TL, Schäffer AA. *et al.* Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Res* 1997;**25**:3389–402. https://doi.org/10.1093/nar/25.17.3389.
12. Skolnick J, Brylinski M. FINDSITE: A combined evolution/structure-based approach to protein function prediction. *Brief Bioinform* 2009;**10**:378–91. https://doi.org/10.1093/bib/bbp017.
13. Zhang Y, Skolnick J. TM-align: A protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res* 2005;**33**:2302–9. https://doi.org/10.1093/nar/gki524.
14. Jing X, Dong Q, Hong D. *et al.* Amino acid encoding methods for protein sequences: A comprehensive review and assessment. *IEEE/ACM Trans Comput Biol Bioinform* 2019;**17**:1918–31. https://doi.org/10.1109/TCBB.2019.2911677.
15. Ahmad S, Sarai A. PSSM-based prediction of DNA binding sites in proteins. *BMC Bioinformatics* 2005;**6**:1–6. https://doi.org/10.1186/1471-2105-6-33.
16. Suthaharan S. Support vector machine. In: *Machine Learning Models and Algorithms for Big Data Classification. Integrated Series in Information Systems*, vol **36**. Springer, Boston, MA, 2016. https://doi.org/10.1007/978-1-4899-7641-3_9.
17. Rigatti SJ. Random forest. *J Insur Med* 2017;**47**:31–9. https://doi.org/10.17849/insm-47-01-31-39.1.
18. Saraç ÖS, Atalay V, Cetin-Atalay R. GOPred: GO molecular function prediction by combined classifiers. *PLoS One* 2010;**5**:e12382. https://doi.org/10.1371/journal.pone.0012382.
19. Cozzetto D, Minneci F, Currant H. *et al.* FFPred3: Feature-based function prediction for all gene ontology domains. *Sci Rep* 2016;**6**:1–11. https://doi.org/10.1038/srep31865.
20. You R, Zhang Z, Xiong Y. *et al.* GOLabeler: Improving sequence-based large-scale protein function prediction by learning to rank. *Bioinformatics* 2018;**34**:2465–73. https://doi.org/10.1093/bioinformatics/bty130.
21. Cerri R, Barros RC, de Carvalho ACPLF. *et al.* Reduction strategies for hierarchical multi-label classification in protein

function prediction. *BMC Bioinformatics* 2016;**17**:373. https://doi.org/10.1186/s12859-016-1232-1.

22. Kulmanov M, Khan MA, Hoehndorf R. DeepGO: Predicting protein functions from sequence and interactions using a deep ontology-aware classifier. *Bioinformatics* 2018;**34**:660–8. https://doi.org/10.1093/bioinformatics/btx624.
23. Yu X, Hu J, Zhang Y. SNN6mA: Improved DNA N6-methyladenine site prediction using siamese network-based feature embedding. *Comput Biol Med* 2023;**166**:107533. https://doi.org/10.1016/j.compbiomed.2023.107533.
24. Zeng W, Yu X, Shang J. *et al.* LBi-DBP, an accurate DNA-binding protein prediction method based lightweight interpretable BiLSTM network. *Expert Syst Appl* 2024;**249**:123525. https://doi.org/10.1016/j.eswa.2024.123525.
25. Kulmanov M, Hoehndorf R. DeepGOPlus: Improved protein function prediction from sequence. *Bioinformatics* 2020;**36**:422–9. https://doi.org/10.1093/bioinformatics/btz595.
26. Cao Y, Shen Y. TALE: Transformer-based protein function annotation with joint sequence–label embedding. *Bioinformatics* 2021;**37**:2825–33. https://doi.org/10.1093/bioinformatics/btab198.
27. Kulmanov M, Hoehndorf R. DeepGOZero: Improving protein function prediction from sequence and zero-shot learning based on ontology axioms. *Bioinformatics* 2022;**38**:i238–45. https://doi.org/10.1093/bioinformatics/btac256.
28. Wan C, Jones DT. Protein function prediction is improved by creating synthetic feature samples with generative adversarial networks. *Nat Mach Intell* 2020;**2**:540–50. https://doi.org/10.1038/s42256-020-0222-1.
29. Zheng L, Shi S, Lu M. *et al.* AnnoPRO: A strategy for protein function annotation based on multi-scale protein representation and a hybrid deep learning of dual-path encoding. *Genome Biol* 2024;**25**:41. https://doi.org/10.1186/s13059-024-03166-1.
30. Lin Z, Akin H, Rao R. *et al.* Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* 2023;**379**:1123–30. https://doi.org/10.1126/science.ade2574.
31. Elnaggar A, Heinzinger M, Dallago C. *et al.* ProtTrans: Toward understanding the language of life through self-supervised learning. *IEEE Trans Pattern Anal Mach Intell* 2021;**44**:7112–27. https://doi.org/10.1109/TPAMI.2021.3095381.
32. Rao B, Yu X, Bai J. *et al.* E2EATP: Fast and high-accuracy protein–ATP binding residue prediction via protein language model embedding. *J Chem Inf Model* 2023;**64**:289–300. https://doi.org/10.1021/acs.jcim.3c01298.
33. Zhu Y-H, Zhang C, Yu D-J. *et al.* Integrating unsupervised language model with triplet neural networks for protein gene ontology prediction. *PLoS Comput Biol* 2022;**18**:e1010793. https://doi.org/10.1371/journal.pcbi.1010793.
34. Lai B, Xu J. Accurate protein function prediction via graph attention networks with predicted structure information. *Brief Bioinform* 2022;**23**:bbab502. https://doi.org/10.1093/bib/bbab502.
35. Yuan Q, Xie J, Xie J. *et al.* Fast and accurate protein function prediction from sequence through pretrained language model and homology-based label diffusion. *Brief Bioinform* 2023;**24**:bbad117. https://doi.org/10.1093/bib/bbad117.
36. Boadu F, Cheng J. Improving protein function prediction by learning and integrating representations of protein sequences and function labels, *bioinformatics. Advances* 2024;**4**:vbae120. https://doi.org/10.1093/bioadv/vbae120.
37. Gligorijević V, Renfrew PD, Kosciolek T. *et al.* Structure-based protein function prediction using graph convolutional networks. *Nat Commun* 2021;**12**:3168. https://doi.org/10.1038/s41467-021-23303-9.
38. Kulmanov M, Guzmán-Vega FJ, Duek Roggli P. *et al.* Protein function prediction as approximate semantic entailment. *Nat Mach Intell* 2024;**6**:220–8.
39. Camon E, Magrane M, Barrell D. *et al.* The gene ontology annotation (Goa) database: Sharing knowledge in uniprot with gene ontology. *Nucleic Acids Res* 2004;**32**:D262–6. https://doi.org/10.1093/nar/gkh021.
40. Radivojac P, Clark WT, Oron TR. *et al.* A large-scale evaluation of computational protein function prediction. *Nat Methods* 2013;**10**:221–7. https://doi.org/10.1038/nmeth.2340.
41. Jiang Y, Oron TR, Clark WT. *et al.* An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biol* 2016;**17**:1–19. https://doi.org/10.1186/s13059-016-1037-6.
42. Li W, Godzik A. Cd-hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics* 2006;**22**:1658–9. https://doi.org/10.1093/bioinformatics/btl158.
43. Singh J, Paliwal K, Litfin T. *et al.* Reaching alignment-profile-based accuracy in predicting protein secondary and tertiary structural properties without alignment. *Sci Rep* 2022;**12**:7607. https://doi.org/10.1038/s41598-022-11684-w.
44. Zdobnov EM, Apweiler R. InterProScan–an integration platform for the signature-recognition methods in InterPro. *Bioinformatics* 2001;**17**:847–8. https://doi.org/10.1093/bioinformatics/17.9.847.
45. Schroff F, Kalenichenko D, Philbin J. Facenet: A unified embedding for face recognition and clustering, *Proceedings of the 28th IEEE Conference on Computer Vision and Pattern Recognition*, Boston: IEEE, 2015, pp. 815–23, https://doi.org/10.1364/OE.539548.
46. Taha A, Chen Y-T, Misu T. *et al.* Boosting standard classification architectures through a ranking regularizer In: *The IEEE/CVF Winter Conference on Applications of Computer Vision*, IEEE, 2020, 758–66.
47. Zhou Q, Zhong B, Lan X. *et al.* Fine-grained spatial alignment model for person re-identification with focal triplet loss. *IEEE Trans Image Process* 2020;**29**:7578–89. https://doi.org/10.1109/TIP.2020.3004267.
48. Memon SA, Khan KA, Naveed H. HECNet: A hierarchical approach to enzyme function classification using a Siamese triplet network. *Bioinformatics* 2020;**36**:4583–9. https://doi.org/10.1093/bioinformatics/btaa536.
49. Kingma DP, Ba J. Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980*, 2014, https://doi.org/10.48550/arXiv.1412.6980.
50. Mering CV, Huynen M, Jaeggi D. *et al.* STRING: A database of predicted functional associations between proteins. *Nucleic Acids Res* 2003;**31**:258–61. https://doi.org/10.1093/nar/gkg034.
51. Wheeler DL, Barrett T, Benson DA. *et al.* Database resources of the national center for biotechnology information. *Nucleic Acids Res* 2007;**36**:D13–21. https://doi.org/10.1093/nar/gkm1000.
52. Dalla-Torre H, Gonzalez L, Mendoza-Revilla J. *et al.* Nucleotide transformer: Building and evaluating robust foundation models for human genomics. *Nat Methods* 2025;**22**:287–97. https://doi.org/10.1038/s41592-024-02523-z.
53. Zhou N, Jiang Y, Bergquist TR. *et al.* The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome Biol* 2019;**20**:1–23.
54. Wang Y, Wang Z, Yu X. *et al.* MORE: A multi-omics data-driven hypergraph integration network for biomedical data

classification and biomarker identification. *Brief Bioinform* 2025;**26**:bbae658. https://doi.org/10.1093/bib/bbae658.

55. Das S, Lee D, Sillitoe I. *et al.* Functional classification of CATH superfamilies: A domain-based approach for protein function annotation. *Bioinformatics* 2015;**31**:3460–7. https://doi.org/10.1093/bioinformatics/btv398.
56. Wang W, Shuai Y, Zeng M. *et al.* DPFunc: Accurately predicting protein function via deep learning with domain-guided structure information. *Nat Commun* 2025;**16**:70. https://doi.org/10.1038/s41467-024-54816-8.
57. Rives A, Meier J, Sercu T. *et al.* Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc Natl Acad Sci* 2021;**118**:1–12. https://doi.org/10.1073/pnas.2016239118.
58. Mandrekar JN. Receiver operating characteristic curve in diagnostic test assessment. *J Thorac Oncol* 2010;**5**:1315–6. https://doi.org/10.1097/JTO.0b013e3181ec173d.
59. Wu G, Zhu J. Multi-label classification: Do hamming loss and subset accuracy really conflict with each other? *Adv Neural Inf Proces Syst* 2020;**33**:3130–40.
60. Antwarg L, Miller RM, Shapira B. *et al.* Explaining anomalies detected by autoencoders using Shapley additive explanations. *Expert Syst Appl* 2021;**186**:115736. https://doi.org/10.1016/j.eswa.2021.115736.
61. Mishra P, Singh U, Pandey CM. *et al.* Application of student's t-test, analysis of variance, and covariance. *Ann Card Anaesth* 2019;**22**:407–11. https://doi.org/10.4103/aca.ACA_94_19.
62. Zhang S, Tong H, Xu J. *et al.* Graph convolutional networks: A comprehensive review. *Comput Soc Netw* 2019;**6**:1–23. https://doi.org/10.1186/s40649-019-0069-y.
63. Jumper J, Evans R, Pritzel A. *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* 2021;**596**:583–9. https://doi.org/10.1038/s41586-021-03819-2.

Briefings in Bioinformatics, 2025, **26(4)**, bbaf420
https://doi.org/10.1093/bib/bbaf420
Problem Solving Protocol