Check for updates

scientific reports

OPEN



Yu Mao^{1,2}, WenHui Xu^{1,2}, Yue Shun¹, LongXin Chai¹, Lei Xue¹, Yong Yang^{1 (2)} & Mei Li^{1 (2)}

Protein function, which is determined by sequence, structure, and other characteristics, plays a crucial role in an organism's performance. Existing protein function prediction methods mainly rely on sequence data and often ignore structural properties that are crucial for accurate prediction. Protein structure provides richer spatial and functional insights, which can significantly improve prediction accuracy. In this work, we propose a multi-modal protein function prediction model (MMPFP) that integrates protein sequence and structure information through the use of GCN, CNN, and Transformer models. We validate the model using the PDBest dataset, demonstrating that MMPFP outperforms traditional single-modal models in the molecular function (MF), biological process (BP), and cellular component (CC) prediction tasks. Specifically, MMPFP achieved AUPR scores of 0.693, 0.355, and 0.478; F_{max} scores of 0.752, 0.629, and 0.691; and S_{min} scores of 0.336, 0.488, and 0.459, showing a 3–5% improvement over single-modal models. Additionally, ablation studies confirm the effectiveness of the Transformer module within the GCN branch, further validating MMPFP's superior performance over existing methods. This multi-modal approach offers a more accurate and comprehensive framework for protein function prediction, addressing key limitations of current models.

Keywords Protein function prediction, Multi-modal models, GCN, CNN, Transformer, Protein sequence, Protein structure

The importance of proteins in biological systems is self-evident, as they play crucial roles in the life processes of organisms. Accurately identifying the function of proteins not only contributes to a deeper understanding of biological processes but also promotes advancements in fields such as drug discovery, crop breeding, and biofuel development. Therefore, developing more efficient technologies and methods to improve the accuracy of protein function prediction is of paramount importance.

In recent years, protein function prediction methods have primarily relied on manual feature extraction^{1,2} and machine learning or deep learning algorithms^{3,4}. Deep learning has become a core tool in contemporary scientific research. In 2024, Geoffrey Hinton, the "father" of deep learning, was awarded the Nobel Prize in Physics for his outstanding contributions to the field⁵. In the same year, the Nobel Prize in Chemistry was awarded to David Baker and others for their groundbreaking work in developing the AlphaFold2⁶ model, which revolutionized protein structure prediction. This model has solved a problem that has puzzled the scientific community for over 50 years, enabling the prediction of the structures of approximately 200 million proteins and has been used by over 2 million users.

Currently, the three most commonly used methods for protein function prediction are: prediction based on protein amino acid sequences, prediction based on protein three-dimensional structures, and prediction based on protein-protein interaction networks⁷. These methods have collectively advanced the field of protein function prediction. The earliest protein function prediction methods were based on homology, such as BLAST (Basic Local Alignment Search Tool)⁸. However, these methods have several limitations, such as the fact that proteins with similar sequences do not necessarily have similar functions, and vice versa. Even functionally similar proteins may have different sequences^{9,10}. These methods fail to fully account for the complexity of protein attributes and their actual functions when calculating similarity, leading to deficiencies in considering related variables. In contrast, machine learning and deep learning-based methods offer advantages in time complexity and higher prediction accuracy because they do not require a pairwise comparison of query sequences with each training sequence.

The amino acid sequence of a protein can be viewed as a set of word vectors, a characteristic that closely resembles tasks in natural language processing. The method proposed by Asgari et al.¹¹ has made significant contributions in this field. Due to the varying lengths of protein sequences, various methods have been developed to encode protein sequences and input them into neural networks for training. For example, Ko et al.¹² used convolutional neural networks for feature extraction, while Ranjan's ProVecGen¹³ method improved the prediction accuracy of long protein

¹State Key Laboratory of Biocatalysis and Enzyme Engineering, School of Life Sciences, Hubei University, Wuhan 430062, Hubei, China. ²Yu Mao and WenHui Xu contributed equally to this work. [⊠]email: yangyong@hubu.edu.cn; meili@hubu.edu.cn

sequences. However, despite these advances, it was shown by Ranjan et al.¹⁴. that relying on a single mechanism or input for prediction is not sufficient to achieve optimal results. The protein sequence alone may fail to capture important structural and functional nuances, which is why a more comprehensive approach is necessary. Therefore, integrating both protein sequence and structure information has the potential to improve prediction accuracy significantly. This multi-modal approach could better capture the intricate relationships between sequence and structure, leading to more robust and precise protein function predictions.

Based on the current research landscape, we propose a multi-modal model for protein function prediction (MMPFP) that takes protein amino acid sequences and structures as fundamental inputs and integrates deep learning methods and artificial neural networks.

Materials and methods Overview of MMPFP

The architecture of our model is shown in Fig. 1, consisting of two modalities and three main modules: the protein sequence encoding module, the multilayer graph convolutional network protein representation module, and the protein convolution module. Each module processes inputs from both the protein sequence and protein structure modalities.

First, the input from the protein sequence modality undergoes encoding through two different embedding methods. The resulting features are then fused with the encoded features from the protein structure modality. Through the complementary information from both modalities and the deep interaction of the three modules, we construct our multi-modal protein function prediction model. After encoding each modality, the inputs are jointly trained within the multi-modal model to fully exploit the complex protein functions.

Next, we will provide a detailed explanation of the input requirements, encoding methods, and structural design for each modality and module in the model.

Protein sequence modality encoding

The left portion of Fig. 1 illustrates two embedding methods for protein sequence data: amino acid embedding and positional embedding, followed by further processing through the Transformer decoder. First, each amino acid in the protein sequence is converted into a dense vector through amino acid embedding. These embedding



Fig. 1. The architecture of the MMPFP model consists of two modalities and three distinct modules: (**A**) represents the protein sequence modality, (**B**) represents the protein structure modality, and (**C**) represents the transformer-based feature fusion block, which includes GCN, CNN, and transformer modules.

vectors effectively capture the fundamental characteristics of amino acids, providing an initial representation of the sequence for subsequent processing.

Next, to preserve the positional information of amino acids within the sequence, we use positional encoding. Positional encoding is generated using sine and cosine functions, which assign each amino acid its relative position within the sequence. This helps the model understand the order and structure of the sequence.

The two embedded sequences are then input into the Transformer encoder (shown as the Decoder block in Fig. 1). The Transformer model, utilizing self-attention, learns the global dependencies between amino acids within the sequence and automatically captures the interactions between different positions in the sequence. The output of the encoder generates a high-dimensional representation of the protein sequence, capturing the complex features of the sequence. These representations can then be used for protein function prediction tasks, providing rich contextual information and sequence features, thereby enhancing the accuracy and robustness of predictions.

Amino acid embedding

Amino acid embedding^{15,16} maps each amino acid (typically represented by an integer index) to a dense vector space using a lookup table. This dense vector contains the feature information of the amino acid:

$$e_{aai} = W_{aa}[aa_i] \tag{1}$$

Let the embedding vector of the i - th amino acid, and $W_{aa}[a_{a_i}]$ denote the amino acid embedding lookup table. The size of $W_{aa}[a_{a_i}]$ is $V_{aa} \times d$, where V_{aa} represents the size of the amino acid dictionary and d is the dimension of the embedding vector.

Positional encoding

Positional encoding^{17,18} is used to capture the positional information of amino acids within the sequence. A commonly used approach is to calculate the positional encoding using sine and cosine functions based on the position of each amino acid. The computation is given by the following formula:

$$PE(i,2k) = \sin\left(\frac{i}{10000^{2k/d}}\right) \tag{2}$$

$$PE(i, 2k+1) = \cos\left(\frac{i}{10000^{2k/d}}\right)$$
 (3)

PE(i, 2k) and PE(i, 2k + 1) represent the embedding values of position *i* in dimension *k*, where *i* denotes the position of the amino acid in the sequence, *k* refers to the dimensional index of the positional embedding, and *d* represents the dimensionality of the embedding vector. Subsequently, we need to add the amino acid embeddings to the positional embeddings to provide each amino acid with a representation that incorporates both its features and position, preparing the feature for input into the decoder:

$$\mathbf{e}_{\text{input}_i} = \mathbf{e}_{\text{aa}_i} + PE(i) \tag{4}$$

 \mathbf{e}_{input_i} represents the final embedding representation of the i - th amino acid, which contains both the feature and positional information of the amino acid, \mathbf{e}_{aa_i} represents the embedding vector of the i - th amino acid, and PE(i) represents the positional information of the i - th amino acid. After combining all the amino acid embeddings and positional embeddings, the representation of the entire protein sequence can be passed as input to the Transformer encoder. Suppose there is a protein sequence of length L, and each amino acid has an embedding dimension of d, then the input to the Transformer can be represented as:

$$L_1 = -\sum_{c=1}^C y_c \log(\hat{y}_c) \tag{5}$$

Here, *E* is a matrix of shape $L \times d$, representing all amino acids in the protein sequence and their corresponding positional information. Finally, we obtain the loss function as follows:

$$\mathbf{E} = [\mathbf{e}_{\text{input}_1}, \mathbf{e}_{\text{input}_2}, \dots, \mathbf{e}_{\text{input}_L}] \in {}^{L \times d}$$
(6)

Overview of the protein structure modality

The protein structure modality consists of two submodules: the GCN module and the CNN module. The raw input to this modality is the three-dimensional structure of the protein. Given a protein structure, we construct an amino acid contact map as an auxiliary input, which represents the distances between all pairs of amino acid residues within the protein structure. The amino acid contact map and the protein's amino acid sequence are then fed into the GCN and CNN modules, respectively.

The protein structure information is divided into two components: the first component is the amino acid sequence information, and the second component is the protein contact map. After encoding the amino acid sequence information, it is passed to the CNN module for processing, while the protein contact map, which contains richer spatial structural information, is input into the GCN module. By performing a weighted fusion of the outputs from these two components, we obtain the final protein structure modality output.



Fig. 2. The workflow within the CNN module is illustrated. In this case, the RepVGG¹⁹ module is employed, which demonstrates improved performance during training while maintaining lower computational overhead during inference. Compared to traditional deep convolutional neural networks such as VGG, RepVGG addresses challenges like overfitting and long training and inference times, offering enhanced scalability.



Fig. 3. The workflow of the GCN module in the protein structure modality is shown in the figure, with the lower section illustrating the calculation process of single-head attention. Compared to multi-head attention, the single-head attention process is more streamlined and practical, making it especially suitable for deployment in environments with limited computational resources.

This multi-module input and fusion strategy effectively combines the spatial structural features of the protein with the relationships between amino acids, thus enhancing the model's performance in protein function prediction tasks.

Multi-layer deep convolutional networks in the protein structure modality

Next, we describe the input and processing steps within the CNN module of this modality. The amino acid sequence information is also encoded. The features of the sequence are composed of two parts: sequence embeddings and label embeddings. Let the set *A* contain 20 standard amino acids and 5 non-standard amino acids. For a protein sequence $s \in A^n$ of length *n*, we combine a trainable sequence matrix with positional information, using this data to embed each subsequence (patch) composed of feature characters into a *h*-dimensional space. Additionally, two types of encoding are applied to the sequence: one-hot encoding and Esm encoding. Each amino acid is encoded as a specific number. To further enrich the feature representation, we introduce EMS-1b encoding on top of the one-hot encoding. These two encoding schemes significantly enhance the feature representation of the protein structure modality. The computation flow of the CNN is shown in Fig. 2.

Graph convolutional network module in protein structural modality

After describing the input and related processes of the Convolutional Neural Network (CNN), we now turn our attention to the processing of protein spatial information such as the protein contact map. Protein contact maps and other structural information carry rich spatial and positional features. In the GCN module, the input first undergoes preliminary feature extraction through a Transformer. The embedding matrix forms theref basis of the Transformer encoder component. Compared to traditional CNNs, Transformers have a distinct advantage in terms of interpretability and capturing long-range dependencies between sequences. In contrast to LSTM, Transformers are more easily parallelized, and the training process is more efficient. Moreover, Transformer-based architectures represent some of the most advanced techniques in the field of deep learning. As shown in the Fig. 3, our GCN component consists of both the Transformer module and the GCN block. Due to experimental limitations, the traditional self-attention mechanism typically uses multi-head attention. However, considering

machine performance constraints, we have employed a lightweight single-head self-attention mechanism in this project, as shown in Fig. 3 below.

After passing through the encoder, the output matrix is denoted as $P \in {}^{nh}$, representing the hidden layer dimension. We treat the GO term as the label embedding for the Transformer and embed it into a *c*-dimensional binary vector $\gamma_i \in \{0, 1\}^c$, which *c* represents the total number of GO terms (labels). Next, we need to encode this, similarly to how sequence embeddings are handled, by calculating the dot product between the label matrix W_{label} and γ_i as follows:

$$Q_i = \gamma_i \cdot W_{label} \tag{7}$$

We then calculate the dot product between P and Q, and pass the result through a softmax layer to compute the similarity M:

$$M = \operatorname{softmax}(P \cdot Q^T) \tag{8}$$

After the dot product calculation between the sequence embedding and label embedding to compute the score, the label embedding will be processed by a 1D convolutional feature extraction module. Following pooling, the result is denoted as a. Subsequently, the sequence embedding branch is again subjected to a dot product operation with a to obtain:

$$e = P^T \cdot a \tag{9}$$

This step forms a residual-like structure, and eventually, they will pass through a fully connected layer to output the probability values of the GO terms. Finally, we define the loss of this module using the binary cross-entropy loss function:

$$L_2 = \left[-\frac{1}{c} \sum_{i=1}^{c} y_i y_i^* + (1 - y_i)(1 - y_i^*) \right] \lambda_a$$
(10)

Where y_i^* represents the model's output, y_i denotes the label values, and λ_a is a learnable hyper parameter used to adjust the contribution of each modality during the model training. The amino acid contact map and the amino acid one-hot encoding are fed into the network for feature extraction. Here, we reuse the one-hot encoding mentioned earlier. Ultimately, these are combined with the outputs from the three submodules of the GCN, and the final predicted score is obtained (Fig. 3). Taking a single input as an example, our input consists of an adjacency matrix *E* representing the edges in the protein structure graph, along with a degree matrix *D* and a weight matrix *W*. The output, *H* is computed as:

$$H = D^{-0.5} A D^{-0.5} \cdot W \tag{11}$$

This is the output of one layer. Then, similar to ViT(Vision transformer), it is fed into the softmax layer for output calculation using Q(query matrix), K(Key matrix), and V(Value matrix):

$$\Psi_{i} = \operatorname{softmax} \frac{Q_{i}K_{i}^{T}}{\sqrt{D}}V_{i}$$
(12)

Several Ψ_i form an output layer:

$$A = [\Psi_1, ..., \Psi_N] \tag{13}$$

The final output is obtained by multiplying ESM and one-hot encoding by a coefficient related to α .

$$A_{final} = \alpha \cdot A_{esm} + (1 - \alpha) \cdot A_{one-hot} \tag{14}$$

Finally, after passing through activation functions and basic operations like dropout, the output A_{final} is fed into a fully connected layer to obtain the prediction score. Once the prediction score is obtained, we can define the loss function as follows:

$$L_3 = -\left[\sum_{m=1}^{M} \sum_{n=1}^{C} \left(y_{mn} \log(\hat{y}_{mn}) + (1 - y_{mn}) \log(1 - \hat{y}_{mn})\right] \lambda_b$$
(15)

Here, *M* represents the number of sequences, *C* represents the number of GO terms, and λ_b is also a constant coefficient. At this point, we can combine the loss functions of the two modalities and three main modules into the final loss function:

$$L = L_1 + L_2 + L_3 \tag{16}$$

	AUPR			F _{max}			S_{min}		
Method	MF	BP	CC	MF	BP	CC	MF	BP	CC
TAWFN ²¹	0.718	0.385	0.488	0.762	0.628	0.693	0.326	0.483	0.454
DeepGO ²²	0.391	0.189	0.258	0.576	0.500	0.589	0.475	0.578	0.553
DeepFRI ²³	0.495	0.265	0.274	0.627	0.546	0.617	0.432	0.543	0.530
HEAL ²⁴	0.661	0.339	0.435	0.733	0.613	0.673	0.357	0.499	0.475
BLAST ⁸	0.136	0.067	0.096	0.326	0.336	0.443	0.643	0.662	0.632
ATGO ²⁵	0.708	0.249	0.306	0.76	0.318	0.703	0.336	0.600	0.539
DeepGOPlus ²⁶	0.422	0.199	0.266	0.595	0.555	0.611	0.453	0.566	0.621
GAT-GO ²⁷	0.660	0.381	0.479	0.633	0.492	0.547	0.437	0.521	0.466
MultiPreGO ²⁸	0.363	0.326	0.568	0.367	0.328	0.536	Nan	Nan	Nan
MMPFP	0.721	0.401	0.495	0.769	0.632	0.695	0.320	0.480	0.448

Table 1. Performance of Eight Baseline Models on F_{max} , S_{min} and AUPR Scores.

	F_{max}			AUPRC			
Method	MF	BP	CC	MF	BP	CC	
Sequence only	0.423	0.211	0.398	0.433	0.295	0.302	
Structure only	0.461	0.299	0.455	0.479	0.288	0.352	
GCN with LSTM	0.685	0.578	0.634	0.579	0.316	0.491	
GCN with transformer	0.769	0.632	0.695	0.721	0.401	0.495	

Table 2. Performance of Models with Two Independent Modalities and Transformer Replaced by LSTM.

Experimental results Model training and evaluation

The datasets used in our experiments are from the PDB database (PDBset) and the AlphaFold protein structure database (AFset). PDBset contains 36,629 protein structures, while AFset includes 42,994 protein structures with GO term annotations. Each protein structure in PDBset includes at least one functional annotation and provides high-resolution PDB chains. We divided both PDBset and AFset datasets into training and testing sets with a 7:3 ratio for model training and evaluation. Specifically, 70% of the data from each dataset was used for training the model, while the remaining 30% was set aside for testing. We ensured a strict separation between the training and testing sets, ensuring no overlap between the two and effectively preventing data leakage. The training set is then fed into the MMPFP multi-modal model, as shown in Fig. 1, for model training and evaluation. We selected these datasets because they are publicly available, easily accessible, and widely used by researchers, making them an ideal choice for benchmarking and evaluating our model.

It is important to emphasize that the testing datasets used are solely from the same division and category, and no proprietary or external datasets were introduced. Additionally, we selected a subset of 10,000 data points from the CAFA dataset for comparison with other baseline models using the MMPFP model. The purpose of this comparison was to mitigate the risk of overfitting on a single or unit dataset, thereby demonstrating the robustness of the model. The detailed comparison results can be found in Table S1 in the Appendix.

Additionally, GO term annotations, including Molecular Function (MF), Biological Process (BP), and Cellular Component (CC), were extracted from studies such as SIFTS²⁰. Furthermore, we constructed amino acid contact maps, which served as foundational data for this research.

We used several commonly adopted metrics in the academic community to evaluate the performance of our model, including F_{max} , S_{min} , and AUPR. F_{max} represents the maximum value among all computed predictions, S_{min} indicates the semantic distance between the true and predicted values, and AUPR is used to assess the model's performance across different prediction thresholds. In our task, higher values of F_{max} and AUPR indicate better model performance, while a smaller value of S_{min} signifies better model performance. We compared our model with several baseline models using these metrics for performance evaluation.

Our model achieved AUPR scores of 0.721, 0.401, and 0.495 for the MF, BP, and CC tasks, respectively. The F_{max} scores were 0.769, 0.632, and 0.695, while the S_{min} scores were 0.320, 0.480, and 0.448. These results, which can be found in Table 1, outperform the current state-of-the-art methods based on single-modal GCN and CNN approaches, demonstrating that our proposed model can more comprehensively learn protein features, including structural information. Additionally, the Transformer module in the GCN effectively captures features of the protein graph through self-attention mechanisms, showing a clear advantage over LSTM-based approaches. These results are presented in Table 2, Figs. 4, and 5. These factors together contribute to the outstanding performance of our model in protein function prediction.

Ablation study

Our model architecture integrates two types of modality inputs and combines CNN, GCN, and Transformer modules. The performance scores of this complete architecture are shown in the table above. To validate the



Fig. 4. Comparison of the MMPFP model and the model without using the Transformer across different GO frequency ranges in the Test set. Panels A, B, and C display the distribution of different GO terms, with the subplots showing the relationship between Log(GO frequency) and frequency.





feasibility and effectiveness of the multi-modal model, we designed ablation experiments to evaluate the impact of different modality inputs and the three main modules on the model's performance. First, we conducted experiments using only a single modality for protein function prediction. Then, we performed ablation studies to evaluate the effectiveness of the Transformer module within the GCN branch. Specifically, in the protein structure modality, we replaced the Transformer component within the GCN module with LSTM as part of the ablation experiment. The choice of LSTM for the ablation module stems from the fact that LSTM is a classical model in deep learning for handling sequential data, and our input can be viewed as a sequence. Consequently, we further conducted ablation experiments with these two modules.

We conducted experiments on the AFset test set, using protein structures predicted by AlphaFold2 for protein function prediction. The experimental results, as shown in Table 2, indicate that models using either the structural modality or the sequence modality alone perform worse in protein function prediction compared to the multi-modal model. This result suggests that the multi-modal protein prediction model is capable of learning a broader range of protein features and better integrating both sequence and structural information, thereby significantly improving the accuracy of function prediction.

In the MMPFP model, when the Transformer module is used, the model performs better than the one using an LSTM-based encoder (Table 2). However, the performance of the Transformer component under different GO frequencies and sequence identities is also a key focus of our investigation. As shown in Figs. 4 and 5, applying the Transformer component to process structural sequences is not only effective but also essential and practically feasible.

Discussion and conclusion

The importance of protein function prediction and limitations of existing methods

Protein function prediction is crucial in bioinformatics, as it helps reveal the biological roles and functions of proteins. However, existing methods primarily rely on unimodal protein representations (such as sequences or structures)⁷, which have limitations when dealing with the complexity of protein function prediction tasks. To

address these challenges, our study proposes the MMPFP model, a multi-modal approach that integrates both protein sequence and structural information. This method effectively overcomes the limitations of unimodal methods, significantly enhancing the accuracy and comprehensiveness of protein function prediction.

Existing approaches and innovations in our method

Traditional protein function prediction methods mainly use unimodal representations, such as one-hot encoding of protein sequences or convolutional neural networks (CNNs) for feature extraction. These methods fail to capture the full spectrum of protein features and thus limit prediction performance. Although advanced single-modality models, such as those employing Transformer architectures^{29,30}, have shown improvements, they still struggle to outperform multi-modal models. This limitation arises because even with sophisticated sequence modeling, they lack the ability to integrate additional contextual or structural data, which are essential for a more accurate prediction. In contrast, our MMPFP model builds upon previous approaches by integrating protein sequence, structure, and other multi-modal features. The inclusion of a Transformer module within the model enables efficient capture of complex relationships within protein graphs through self-attention mechanisms, providing a significant advantage over LSTM-based models. Experimental results show that MMPFP outperforms traditional unimodal models by 3%-5% in metrics such as F_{max} , AUPR, and S_{min} across several public datasets.

Applicability, prospects, and future directions of the new method

The MMPFP model demonstrates strong performance in protein function prediction, particularly in handling complex multi-modal data. Looking ahead, we plan to introduce additional learnable features and explore the fusion of further modalities, such as incorporating protein-protein interaction networks as new modalities within the multi-modal framework, alongside advanced deep learning models. Additionally, we aim to extend the model's functionality beyond protein function prediction to multitask learning. For example, the model could also be applied to protein structure prediction, creating a unified multitask, multi-modal model. As real-world problems often involve multiple attributes with nonlinear relationships, the development of multi-modal models represents a natural and forward-looking direction for future research. Although multi-modal protein prediction models have been explored by other researchers, the results are not always superior. For instance, experiments in the work by^{28,31} suggest that certain multi-modal approaches may even underperform compared to unimodal or feature-fusion models. Therefore, while the integration of multi-modal data in protein prediction is essential, equal attention must be given to the selection and adaptation of advanced modules within these models to maximize their effectiveness.

Data availability

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Received: 1 January 2025; Accepted: 14 March 2025 Published online: 26 March 2025

References

- 1. Jing, X., Dong, Q., Hong, D. & Lu, R. Amino acid encoding methods for protein sequences: A comprehensive review and assessment. *IEEE/ACM Trans. Comput. Biol. Bioinf.* 17, 1918–1931 (2019).
- Yu, N., Li, Z. & Yu, Z. Survey on encoding schemes for genomic data representation and feature learning-from signal processing to machine learning. *Big Data Min. Anal.* 1, 191–210 (2018).
- 3. Lv, Z., Ao, C. & Zou, Q. Protein function prediction: From traditional classifier to deep learning. *Proteomics* 19, 1900119 (2019).
- 4. Bonetta, R. & Valentino, G. Machine learning techniques for protein function prediction. *Proteins Struct. Funct. Bioinform.* 88, 397–413 (2020).
- 5. https://www.nobelprize.org/prizes/physics/2024/summary (2024).
- 6. Jumper, J. et al. Highly accurate protein structure prediction with Alphafold. Nature 596, 583-589 (2021).
- Dhanuka, R., Singh, J. P. & Tripathi, A. A comprehensive survey of deep learning techniques in protein function prediction. *IEEE/* ACM Trans. Comput. Biol. Bioinf. 20, 2291–2301 (2023).
- Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. Basic local alignment search tool. *J. Mol. Biol.* 215, 403–410 (1990).
 Gerlt, J. A. & Babbitt, P. C. Can sequence determine function?. *Genome Biol.* 1, 1–10 (2000).
- 10. Friedberg, I. Automated protein function prediction-the genomic challenge. Brief. Bioinform. 7, 225-242 (2006).
- Asgari, E. & Mofrad, M. R. Continuous distributed representation of biological sequences for deep proteomics and genomics. PLoS One 10, e0141287 (2015).
- 12. Ko, C. W., Huh, J. & Park, J.-W. Deep learning program to predict protein functions based on sequence information. *MethodsX* 9, 101622 (2022).
- Ranjan, A., Fahad, M. S., Fernández-Baca, D., Deepak, A. & Tripathi, S. Deep robust framework for protein function prediction using variable-length protein sequences. *IEEE/ACM Trans. Comput. Biol. Bioinf.* 17, 1648–1659 (2019).
- Ranjan, A., Tiwari, A. & Deepak, A. A sub-sequence based approach to protein function prediction via multi-attention based multi-aspect network. *IEEE/ACM Trans. Comput. Biol. Bioinf.* 20, 94–105 (2021).
- 15. Mikolov, T. Efficient estimation of word representations in vector space 3781. arXiv:1301.3781 (2013).
- Cui, F., Zhang, Z. & Zou, Q. Sequence representation approaches for sequence-based protein prediction tasks that use deep learning. *Brief. Funct. Genomics* 20, 61–73 (2021).
- 17. Vaswani, A. Attention is all you need. In Advances in Neural Information Processing Systems (2017).
- Kenton, J. D. M.-W. C. & Toutanova, L. K. Bert: Pre-training of deep bidirectional transformers for language understanding. In aProceedings of naacL-HLT, vol. 1, 2 (Minneapolis, Minnesota, 2019).
- Ding, X. et al. Repvgg: Making vgg-style convnets great again. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 13733–13742 (2021).
- Dana, J. M. et al. Sifts: Updated structure integration with function, taxonomy and sequences resource allows 40-fold increase in coverage of structure-based annotations for proteins. *Nucleic Acids Res.* 47, D482–D489 (2019).

- 21. Meng, L. & Wang, X. Tawfn: A deep learning framework for protein function prediction. Bioinformatics 40, btae571 (2024).
- Kulmanov, M., Khan, M. A. & Hoehndorf, R. Deepgo: Predicting protein functions from sequence and interactions using a deep ontology-aware classifier. *Bioinformatics* 34, 660–668 (2018).
- 23. Gligorijević, V. et al. Structure-based protein function prediction using graph convolutional networks. *Nat. Commun.* **12**, 3168 (2021).
- 24. Gu, Z., Luo, X., Chen, J., Deng, M. & Lai, L. Hierarchical graph transformer with contrastive learning for protein function prediction. *Bioinformatics* 39, btad410 (2023).
- Zhu, Y.-H., Zhang, C., Yu, D.-J. & Zhang, Y. Integrating unsupervised language model with triplet neural networks for protein gene ontology prediction. *PLoS Comput. Biol.* 18, e1010793 (2022).
- 26. Kulmanov, M. & Hoehndorf, R. Deepgoplus: Improved protein function prediction from sequence. *Bioinformatics* **36**, 422–429 (2020).
- 27. Lai, B. & Xu, J. Accurate protein function prediction via graph attention networks with predicted structure information. *Briefings Bioinform.* 23, bbab502 (2022).
- Giri, S. J., Dutta, P., Halani, P. & Saha, S. Multipredgo: deep multi-modal protein function prediction by amalgamating protein structure, sequence, and interaction information. *IEEE J. Biomed. Health Inform.* 25, 1832–1838 (2020).
- 29. Cao, Y. & Shen, Y. Tale: Transformer-based protein function annotation with joint sequence-label embedding. *Bioinformatics* 37, 2825–2833 (2021).
- 30. Qiu, X.-Y., Wu, H. & Shao, J. Tale-cmap: Protein function prediction based on a tale-based architecture and the structure information from contact map. *Comput. Biol. Med.* 149, 105938 (2022).
- 31. Fa, R., Cozzetto, D., Wan, C. & Jones, D. T. Predicting human protein function with multi-task deep neural networks. *PLoS One* 13, e0198216 (2018).

Author contributions

Y.M. conceived the research and conducted the experiments. Y.S. and L.X.C. analyzed the data. W.H.X. contributed to writing parts of the manuscript. L.X. was responsible for conducting part of the experiments, and Y.Y. reviewed the manuscript. M.L. also reviewed the manuscript. All authors approved the final version.

Funding

This work was supported by Hubei Provincial Talent Project (1070202403147).

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/1 0.1038/s41598-025-94612-y.

Correspondence and requests for materials should be addressed to Y.Y. or M.L.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

© The Author(s) 2025