Received: 26 January 2024 | Revised: 15 June 2024 | Accepted: 18 June 2024

DOI: 10.1002/pmic.202300471

Proteomics
Proteomics and Systems Biology

REVIEW

Deep learning methods for protein function prediction

Frimpong Boadu | **Ahhyun Lee** | **Jianlin Cheng**

Department of Electrical Engineering and Computer Science, University of Missouri, Columbia, Missouri, USA

Correspondence
Jianlin Cheng, Department of Electrical Engineering and Computer Science, University of Missouri, Columbia, 65211, MO, USA.
Email: chengji@missouri.edu

Frimpong Boadu and Ahhyun Lee contributed equally this study.

Funding information
Division of Biological Infrastructure, Grant/Award Number: 2308699; National Science Foundation, Grant/Award Numbers: DBI2308699, CCF2343612; National Institutes of Health, Grant/Award Numbers: R01GM093123, R01GM146340

Abstract

Predicting protein function from protein sequence, structure, interaction, and other relevant information is important for generating hypotheses for biological experiments and studying biological systems, and therefore has been a major challenge in protein bioinformatics. Numerous computational methods had been developed to advance protein function prediction gradually in the last two decades. Particularly, in the recent years, leveraging the revolutionary advances in artificial intelligence (AI), more and more deep learning methods have been developed to improve protein function prediction at a faster pace. Here, we provide an in-depth review of the recent developments of deep learning methods for protein function prediction. We summarize the significant advances in the field, identify several remaining major challenges to be tackled, and suggest some potential directions to explore. The data sources and evaluation metrics widely used in protein function prediction are also discussed to assist the machine learning, AI, and bioinformatics communities to develop more cutting-edge methods to advance protein function prediction.

KEYWORDS
artificial intelligence, deep learning, gene ontology, protein function prediction

1 | INTRODUCTION

Proteins are essential molecules in all living organisms. Their role encompasses structural support, biochemical catalysis, gene regulation, enzymatic activities, and signal transduction [1, 2]. Determining the functions of proteins is a key step to understand biological systems and modulate BPs, which plays an important role in biomedical research and biotechnology development. Furthermore, proteins are common targets in drug discovery [3–5] because many proteins are implicated in diseases, and protein function information can facilitates the development of drugs targeting them. As the structure of protein can be determined by experimental techniques such as x-ray crystallography, the function of proteins can also be determined by experimental techniques such as biochemical assays and enzymatic analysis. However, the experimental techniques for protein function determination are expensive, time-consuming, and labor-intensive and can only be applied to a small number of proteins. Therefore, making precise protein function prediction computationally holds the key to address the need of function information for most proteins and has become a critical challenge in bioinformatics.

Currently, hundreds of millions of protein sequences have been generated through numerous genome and transcriptome sequencing projects. However, less than 1% of them have experimentally determined protein function information. This presents a huge gap between known protein sequences and their functions. Therefore, it is critical to devise advanced computational methods to accurately predict protein function to fill the gap as the recent development of deep learning methods has done for protein structure prediction and determination [6–10].

A plethora of various computational methods have been developed to predict protein function, many of which had been reviewed and assessed previously [11–13]. Recently, as AI is transforming many

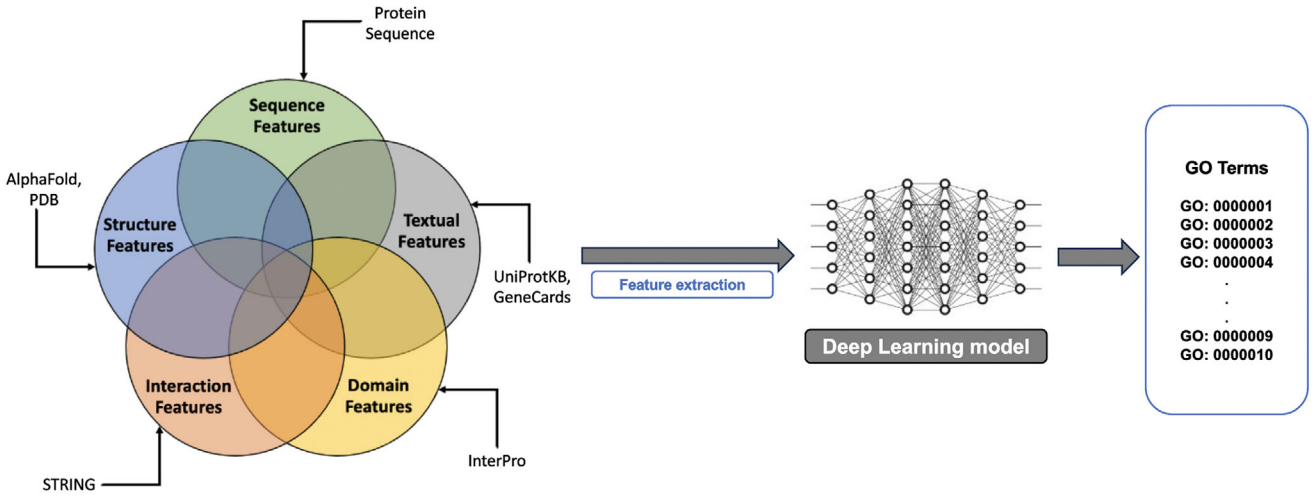

FIGURE 1 The general workflow of deep learning-based protein function prediction. One or multiple sources of data such as protein sequences, protein structures (e.g., structures retrieved from the AlphaFoldDB [17] and the Protein Data Bank [PDB] [18]), protein–protein interaction from the STRING database [19], protein family and domain information from the Interpro database [20], and the textual description of proteins in the literature such as UniProt Knowledgebase (UniProtKB) [21] and GeneCards [22] are presented as input. The features are then extracted from the input data, which are fed into deep learning models to predict protein function as output. Protein function are usually described as GO [14] function terms. Therefore, protein function prediction is essentially a classification problem. Because one protein may have multiple functions described by multiple GO terms, it is a multilabel classification problem.

scientific fields, cutting-edge prediction methods based on deep learning approaches have been thriving in the protein function prediction field, leading to a significant improvement of prediction accuracy over the previous generation of computational protein function prediction methods. Therefore, there is a need of reviewing these latest advances to facilitate the development of more deep learning methods to address the remaining challenges in the field.

Here, we present a comprehensive overview of recent deep learning methods developed to advance protein function prediction. Figure 1 illustrates a general workflow of deep learning-based prediction of protein function defined by the gene ontology (GO) terms [14]. We classify these methods roughly into four main categories based on the input information used by them: (1) sequence-based methods of using only protein sequence as input, (2) structure-based methods of using protein structure as input, (3) interaction-based methods of using protein–protein interaction (PPI) information as input, and (4) integrative methods that use multiple sources of information as input. It is worth noting that structure- or interaction-based methods often also use sequence information implicitly in addition to using structure or interaction information, but they are not classified as integrative methods. Moreover, we also discuss the latest few-shot learning [15, 16] paradigm that improves the prediction of rarely annotated protein function terms associated with few proteins. Table 1 lists the types, input features, neural network architectures, and availability of 30 deep learning protein function prediction methods reviewed in this article. Furthermore, in addition to surveying the deep learning methods, we discuss the data sources, standard benchmarks (i.e., the Critical Assessment of Protein Function Annotation (CAFA) [11]), and evaluation metrics widely used for protein function prediction to assist the AI, machine machine learning, and bioinformatics communities to find necessary resources to develop more protein function prediction methods. Moreover, we identify several major remaining challenges in protein function prediction and envision that developing Large Language Models for Proteins (LLMPs), akin to the Large Language Models (LLMs) used in natural language processing (NLP), such as ChatGPT [15], can be a promising approach to addressing the challenges. These topics are discussed in detail in the sections below.

2 | SEQUENCE-BASED PROTEIN FUNCTION PREDICTION

Sequence-based prediction methods use different kinds of deep learning architectures to take protein sequence information as input to predict protein function. Several deep learning models that have demonstrated effectiveness for dealing with sequential data are (1) convolutional neural networks (CNNs) [23], (2) recurrent neural networks (RNNs) [24, 25], (3) deep neural networks (DNNs) [26, 27], and (4) attention-based transformers [2, 28]. CNNs are effective at identifying motifs (short conserved sequence patterns associated with distinct protein functions), local patterns, and spatial relationships in the protein sequences. RNNs, particularly, long short-term memory networks (LSTMs) [29], can capture sequential dependence between amino acids in protein sequences. DNNs also hold significance in capturing the nonlinear relationships between protein function and sequences through multiple neural network layers. Finally, the attention mechanism and transformer architecture are well-suited for sequence-based function prediction due to their ability to capture long-range dependencies between amino acids in protein sequences.

TABLE 1 The classification of 30 deep learning protein function prediction methods and their input features, architectures, and availability. Sequence, structure, interaction, and domain refers to four types of typical input features: sequence-based features, structure-based features, protein interaction-based features, and other features based on protein family and domain information. RNN stands for both standard recurrent neural networks and advanced ones like gated recurrent unit (GRU) and long short-term memory (LSTM), CNN for convolutional neural networks, and GNN for graph neural networks. Attention denotes the methods utilizing self-attention mechanisms, transformers, and techniques extracting features from pretrained attention- or transformer-based architectures. DNN refers to deep neural networks that use multilayer perceptrons (MLP) as a main part of the model architecture beyond using them in the final classification layer. Few-shot refers to methods specifically designed to utilize deep learning models for predicting GO terms with few annotations. We also include a link to the GitHub repository or webpage of the tool. For tools whose link we cannot find, we use NA.

	Features					Deep learning architecture						
Methods	Sequence	Structure	Interaction	Domain	Text	DNN	CNN	GNN	RNN	Attention	Few-shot	URL
ProLanGO	✓								✓			NA
FUTUSA	✓						✓					GitHub
DeepGOPlus	✓						✓					Web
PFmulDL	✓						✓		✓			GitHub
DEEPred	✓					✓						GitHub
TALE	✓									✓	✓	GitHub
TEMPROT	✓					✓				✓		GitHub
SPROF-GO	✓					✓				✓		GitHubWeb
ATGO	✓					✓				✓		Web
PANDA2	✓							✓		✓		Web
DeepFRI	✓	✓						✓	✓			GitHubWeb
GAT-GO	✓	✓					✓	✓		✓		GitHub
TransFun	✓	✓						✓		✓		GitHub
Struct2GO	✓	✓						✓		✓		GitHub
Mashup			✓									Web
deepNF			✓			✓						GitHub
MELISSA			✓									GitHub
NetQuilt	✓		✓			✓						GitHub
DeepGO	✓		✓			✓	✓					GitHubWeb
STRING2GO			✓			✓						GitHub
DeepGraphGO			✓	✓				✓				GitHub
GRAPH2GO	✓		✓	✓				✓				GitHub
NetGO2	✓			✓	✓				✓			Web
NetGO3	✓			✓	✓					✓		Web
SDN2GO	✓		✓	✓		✓	✓					GitHub
PFP-GO	✓		✓	✓								Web
MultiPredGO	✓	✓	✓			✓	✓					GitHub
DeepGATGO	✓				✓	✓		✓		✓		NA
ProTranslator	✓		✓		✓		✓				✓	GitHub
DeepGOZero				✓		✓					✓	GitHub

Sequence-based **Structure-based** **Interaction-based** **Integrative** **Few-Shot**

Moreover, compared to CNNs, RNNs, and DNNs, transformers can be more interpretable because its attention mechanism can help identify key features (e.g., residues) important for function prediction. Besides directly applying transformer-based architectures to protein function prediction, several methods [30–32] leverage transformer-based pre-trained protein language models to extract representative embeddings from protein sequences for downstream protein function prediction tasks. In the subsequent sections below, we discuss the specific methods that harness these deep learning models to address the intricacies of predicting protein function from sequences.

2.1 | RNN-based protein function prediction

ProLanGO [33] treats the protein function prediction problem as a language translation problem and applies a RNN-based Neural Machine Translation (NMT) model to tackle it. Protein sequences (input) and GO terms (output) are regarded as two separate languages, ProLan and GoLan, respectively. Protein sequences are represented as a series of k-mers (i.e., a substring or word of k amino acids). Protein words are extracted based on the frequency of k-mers. GO function terms are generally represented as a directed acyclic tree structure based on their relationships, with each term uniquely identified by a seven-digit number. ProLanGo allows capturing the hierarchical relationship between GO terms and enables the sequence to function translation through the depth-first search (DFS). Each GO term is assigned to a 26-base Alphabet ID according to its order of being visited during the DFS traversal. Given the Prolan and GOlan languages, an encoder–decoder based on RNNs is trained to predict GOlan from Prolan. The encoder is used to encode a ProLan sentence into fixed-length vectors, and the decoder decodes the representation into a GOLan sentence. The network is trained by maximizing the conditional probability of predicting a GOLan sentence given a ProLan sentence.

2.2 | CNN-based protein function prediction

FUTUSA [34] has following four components: CNN-based embedding layers, CNN-based feature extraction, dense layers, and a classification layer. The embedding layers are used to convert protein sequences to numerical vectors. To alleviate the limitations of one hot encoding such as the inability to capture physiochemical properties of amino acids, a one-dimensional CNN is employed to generate the amino acid embedding vector, followed by another CNN to extract spatial features, whose output is fed into dense layers to generate hidden features. The hidden features are used by the final classification layer to predict GO terms.

DeepGOPlus [35] combines the function prediction from a CNN network and the sequence similarity to improve prediction accuracy. It uses one-dimensional CNN filters to learn similar patterns (motifs) in sequences. An input sequence is transformed into a matrix representation of dimension 21×2000 using a one-hot encoding strategy, where a one-hot vector of 21 binary numbers is used to represent an amino acid and the maximum number of amino acids to be represented is 2000. The input is fed into a set of CNN layers with varying filter sizes to generate features capturing sequence motifs of different size. The features are pooled together and selected by a MaxPooling layer. The output of the MaxPooling layer is forwarded to a fully connected classification layer to predict GO terms. DeepGOPlus is a general sequence-based protein function prediction that can be applied to proteins in any taxa or kingdom of species.

PFmulDL [36] integrates both a multikernel CNN and a gated recurrent unit (GRU) to predict protein function. Like DeepGoPlus, it employs a one-hot strategy to encode an input protein sequence. The encoding serves as input for a multikernel CNN model, which is fine-tuned by a pretraining process. The output layer of the CNN is used as input for the GRU to generate features, which are used as input for a fully connected layer to predict GO terms. In order to prevent issues such as gradient vanishing and overfitting, it uses transfer learning (TL) to improve training, leading to the improved performance of protein function prediction. Particularly, it enhances the prediction accuracy for "rare GO terms (minority class)" without compromising the performance for the "common GO terms (major classes)."

2.3 | DNN-based protein function prediction

DEEPred [37] employs a deep learning model organized as a stack of multitask feed-forward DNNs. Each DNN is independently designed to predict groups of 4 or 5 GO terms. The grouping is based on the levels of GO terms in the GO graph, determined through the topological sorting. Groups are carefully created to ensure that GO terms within the same group have similar numbers of annotations, addressing the variability in protein associations. This approach aims to enhance the model's accuracy and effectiveness in predicting GO terms for diverse biological functions.

2.4 | Attention- and transformer-based protein function prediction

TALE [30] uses a self-attention-based transformer to extract representative features from protein sequence to improve protein function prediction. It also leverage a zero-shot learning paradigm to jointly embed sequence and hierarchical function labels into the latent space, allowing a more cohesive representation of the relationships between features and labels. This joint embedding facilitates TALE to generalize well to novel sequences and unseen function by matching similarities among function labels and sequences. Furthermore, TALE introduces a new loss function to address the issue of hierarchical violation. This loss function includes a hierarchical regularization term, which specifically aims to prevent the predicted scores (probabilities) of child GO terms from surpassing those of its ancestors. Additionally, TALE+, a method that ensembles the top three TALE models and a sequence similarity-based protein function prediction method based on DIAMOND [38], was developed to improve the predictions made by TALE.

TEMPROT [39] is another sequence-based protein function prediction method leveraging ProtBERT-BFD [40], a transformer language model pretrained on the BFD dataset [8, 41, 42]. The pretrained ProtBERT-BFD was first fine-tuned. The fine-tuning process employs a sliding window technique, dividing sequences into 500 chunks to accommodate ProtBERT-BFD's length limitation of 512. After fine-tuning, the backgone of ProtBERT-BFD is used to extract representative features from protein sequences. These features serve as an input for a meta-classifier based on a multilayer perceptron (MLP) to predicting protein function. Furthermore, TEMPROT+ combining TEMPROT

and a sequence-similarity search tool, BLASTp [43], was developed to improve the prediction performance.

SPROF-GO [44] is a sequence-based alignment-free protein function prediction method, which harnesses a pretrained protein language model for efficient extraction of informative sequence embeddings, while applying self-attention pooling to focus on crucial residues. Its prediction has three main stages. First, the pretrained protein language model ProtTrans [40] is used to efficiently extract the initial sequence embedding matrix from sequences. The sequence embedding matrix undergoes parallel processing by two MLPs to acquire an attention vector and a more detailed hidden embedding matrix. The hidden embeddings are then normalized to generate an embedding vector, which is used as an input for an MLP to predict the probabilities of GO terms. SPROF-GO also employs a hierarchical learning strategy to guarantee the consistency among predictions. Furthermore, a label diffusion algorithm is integrated in the test phase to exploit the homology information of proteins with related functions.

ATGO [45] harnesses protein language models trained on extensive sequences in an unsupervised fashion to predict protein function. The strategy aims to address the limitations associated with imbalanced annotated functional data. Specifically, ATGO uses the ESM-1b transformer [46] to extract multilayer feature embeddings from protein sequences. A supervised triplet neural network was trained on these extracted feature embeddings in order to maximize the difference between positive and negative samples. To further enhance ATGO's performance, a composite method, ATGO+ was also introduced. It combines predictions from ATGO and the Sequence Alignment-Based GO Prediction (SAGP).

PANDA2 [47] uses a Graph Neural Network (GNN) to model the GO direct acyclic graph (DAG) representing the hierarchical structure of GO terms. It also incorporates features produced by the transformer-based protein language model ESM [46]. PANDA2 has three blocks serving as fundamental building blocks for refining edge, node, and global features. In the first two blocks, it sequentially updates edge features, node features, and global features by integrating information of all available features in the GNN. Furthermore, it employs a fully connected layer to change the size of ESM features to the number of classes being considered. Then, it merges node features, the output generated by fully connected layer, DIAMOND scores, and priority scores. This comprehensive combination of information is used as input for the third GNN block. The node features of the third GNN block are used by a sigmoid function to predict the probability of each class (GO term). PANDA2 demonstrates the effectiveness of using a GNN architecture for modeling the GO DAG topology and annotating protein functions.

3 | STRUCTURE-BASED PROTEIN FUNCTION PREDICTION

The sequence-based function prediction approach has been more common in protein function prediction than the approaches of using other inputs due to the universal availability of protein sequence, even though other data such as protein structure can provide additional complementary information to improve protein function prediction. Incorporating structure in function prediction provides additional data for models to leverage and enhance their predictive accuracy. For instance, molecular functions are largely determined by protein structures, and proteins with similar structures can have different sequences. BPs and to some extent cellular component (CC) usually rely on multiple proteins and the way they interact. As such, incorporating multiple sources of information in the best possible way will likely improve predictions in these respective domains. In this regard, structure-based prediction methods can utilize structural information to improve predictions, particularly for molecular functions.

With the recent development of high-accuracy protein structure prediction tools such as AlphaFold2 [8, 17], protein structures have become generally available and started to be used more and more in protein function prediction. Most structure-based prediction methods use various GNNs such as Graph Convolutional Network (GCN) and Graph Attention Network (GAT) to represent and process protein structures. GNNs offer powerful capabilities for complex graph-related tasks; however, they come with high computational requirements and scalability issues that must be carefully considered. Libraries such as PyTorch Geometric (PyG) [48] and Deep Graph Library (DGL) [49] provide optimized implementations and tools that significantly enhance the feasibility of using these architectures.

DeepFRI [50] relies on a GCN [51] to integrate protein structures and sequence features extracted from a language model to predict protein function. DeepFRI utilizes known protein structures available in the PDB or homology-based structural models built by SWISS-MODEL [52] as structural input. It uses a language model comprised of a long short-term memory (LSTM) network trained in a self-supervised learning manner to extract residue-level features from protein sequences, followed by the GCN layers merging the reside-level features with the graph built from the contact maps calculated from the input protein structure to generates protein-level feature representations. The protein-level features are used to predict GO terms in each of three function categories: CC, BP, and Molecular Function as well as the Enzyme Commission (EC) numbers, respectively. DeepFRI also employs gradient-weighted Class Activation maps (grad-CAMs) to elevate the representation resolution from protein-level to the region-level, which allows the detection of function-specific structural sites, facilitating the identification of crucial residues correlated with specific functions.

Different from the GCN used by DeepFRI, GAT-GO [53] uses a GAT to integrate both predicted protein structural information and protein sequence embeddings for accurate protein function prediction. The method uses RaptorX [54, 55] to predict protein structural information (i.e., protein contact map) and ESM-1b to generate sequence embeddings. It first uses a one-dimensional CNN to take both sequential features and residue-level sequence embeddings to create per-residue feature embeddings. Then, the CNN-generated embeddings combined with a RaptorX-predicted contact map are fed into GAT, which produces an intermediate embedding that captures both sequential and structural information. The representation constructed by GAT

passes through a dense classifier to predict the probability of protein function terms.

Different from DeepFRI and GAT-GO using earlier protein structure prediction methods to generate structural input, TransFun [31] uses AlphaFold-predicted protein structures as input. It employs a transformer-based protein language model and rotation- and translation-equivariant graph neural networks (EGNNs) [56] to distill information from both protein sequences and structures to predict protein functions. Its prediction process has the following three main stages: (1) building a protein graph from a predicted structure, (2) generating the embeddings from a protein sequence, and (3) using an EGNN model to predict protein functions. In the first stage, protein graphs are generated from protein structures collected from AlphaFoldDB [8, 17] using a K-nearest neighbor (KNN) approach based on the distance between carbon-alpha atoms in a protein structure. In the second stage, per-residue and per-sequence embeddings for proteins are generated from protein sequences by the ESM-1b [46] pretrained language transformer model. In the final stage, both the per-residue and per-sequence features are combined by the EGNNs to predict protein function.

Struct2GO [57] is also a structure-based method that combines sequence features with structural features obtained from Alphafold2-predicted structures. It extracts a two-dimensional (2D) protein contact map for an input protein from the three-dimensional (3D) protein structure according to a distance threshold of 10 Å between carbon-alpha atoms. Additionally, Node2vec [58] algorithm is employed to generate residue-level features for the protein. The contact map serves as the adjacency matrix of the input graph, which are combined with the node features, that is, the residue-level features, to generate a graph representation of the protein. The representation is used by a Graph Convolution Neural (GCN) network to generate hidden structural features. The feature generation is enhanced with a self-attention mechanism and the integration of sum- and max-pooling techniques. Additional sequence features are also extracted using the SeqVec [59]. Finally, the sequence features are fused with the structural features as input for a final classifier to make function prediction.

4 | INTERACTION-BASED PROTEIN FUNCTION PREDICTION

Due to the fact that proteins rarely function in isolation, PPI information can be used to enhance protein function prediction. It is particularly useful for predicting GO terms describing biological processes (BPs) that involve multiple proteins cooperating together. Protein function prediction methods relying on PPIs primarily focus on genome-scale interaction networks, aggregating data from various sources to gain insights into the functional organization of proteins. Some of these methods emphasize the integration of heterogeneous information from diverse interaction networks. A straightforward approach for data integration is to process each network separately and then combine the features generated from each of them. However, this approach often encounters some challenges like increased dimensionality, information loss, and noise accumulation from high-throughput experiments. In this section, we discuss the diverse approaches of integrating multiple heterogeneous networks to predict protein function.

Mashup [60] is an integrative framework designed to extract high-quality and compact topological feature representations from one or more interaction networks constructed from heterogeneous data types. Although Mashup does not inherently use a deep learning technique, it provides a method for extracting features from multiple heterogeneous networks, which are readily used by several interaction-based deep learning methods [61, 62]. The method consists of the following three main stages: a diffusion stage, an embedding stage, and a learning stage. The diffusion stage involves applying a localized network diffusion technique, specifically Random Walks with Restart (RWR), to each individual network to obtain a matrix representation capturing the interactions between nodes denoting proteins. This captures information about topological structure and connectivity of nodes in each network. Next, the embedding phase focuses on obtaining low-dimensional feature vectors that represent the topology of each node, which is achieved by minimizing the difference between observed diffusion states and parameterized multinomial logistic distributions across all networks. Finally, the learned representations are used as input features for various downstream tasks including protein function prediction.

Following a similar approach as Mashup, deepNF [61] integrates diverse heterogeneous protein interaction networks using deep learning techniques. The process begins with the Random Walk with Restart (RWR) algorithm to obtain high-quality vector representations of proteins in each network, capturing their structural information. A Positive Pointwise Mutual Information (PPMI) function is then applied for normalization, and this process is iterated for each network. The subsequent stage focuses on creating a comprehensive representation by integrating the multiple PPMI instances. To achieve this, deepNF employs a Multimodal Data Autoencoder (MDA) network to encode diverse PPMI instances into a representative matrix and reconstruct it through a decoder. The encoder produces low-dimensional nonlinear embeddings for each network, and these representations are concatenated. A common feature representation is computed using multiple nonlinear functions. In the decoding phase, the process is reversed to compute larger common representations from individual ones, followed by the reconstruction of PPMI matrices for each network. The final step predicts protein functions based on the comprehensive representations obtained in the bottleneck layer of the autoencoder network.

Similar to Mashup and deepNF, MELISSA [62] predicts functions from multiple PPI networks. However, the integration of known functional labels during the embedding process sets MELISSA apart from the aforementioned methods. Its prediction unfolds in the following five key steps: Biclustering, Graph Augmentation, Diffusion, Embedding, and Learning. In the initial stage, MELISSA employs a biclustering algorithm to simultaneously cluster proteins and functional labels. This results in biclusters where proteins within clusters share similar functional labels, and functional labels are rarely shared across clusters. In the following step, the PPI graphs undergo augmentation by

introducing auxiliary nodes, each representing a distinct cluster. Nodes in the graph are then connected to their corresponding auxiliary nodes using must-link constraints (positive weighted edges). Additionally, pairwise cannot-link constraints (edges with negative weights) are introduced between the auxiliary nodes. This augmentation transforms the graphs into signed graphs, where auxiliary nodes encode functional information. Nodes within the same cluster are drawn closer, while nodes in different clusters are pushed apart. Following the augmentation stage, diffusion state matrices are generated for each augmented graph using a generalization of the method applied in Mashup, by considering the signed nature of the edges. In the final step, MELISSA follows Mashup's approach to generate embeddings for each node. These embeddings can be effectively utilized by existing function prediction methods to predict function terms.

NetQuilt [63] is a method that integrates protein sequence and PPI information from multiple species. The approach computes similarity scores between proteins across species using a recurrence equation derived from the IsoRank method of multispecies network alignment [64]. A large symmetric similarity matrix is constructed, where IsoRank similarity matrices of all species with themselves are placed along the diagonal, resulting in a block-diagonal matrix. Interspecies protein similarity matrices are placed on the off-diagonal. The matrix then contains the information from all the individual protein interaction networks as well as the links between them.

The matrix constructed, along with sequence-similarity information, is used as input for a maxout neural network to predict protein function.

DeepGO [65] introduces an approach to predict protein function based on protein sequences and known interactions. It integrates features derived from sequences and PPI networks across various species in the STRING database. The combined sequence and PPI network features undergo processing in a fully connected layer, and the resultant output feeds into hierarchically structured neural networks to make function prediction.

STRING2GO [66] employs a deep maxout neural network (DMNN) to acquire functional representations by simultaneously encoding both PPIs and functional annotation information. It uses two methods to generate network embedding representations: (1) a network embedding generation process similar to the one in mashup and (2) node2vec of generating embeddings from the STRING network. After the generation of embeddings, DMNNs are used to simultaneously learn and encode representation information from both the PPI network and protein functional annotations. The functional representations are extracted from the outputs of the third hidden layer of DMNNs, which is used by a support vector machine (SVM) to predict the probability of GO terms.

5 | INTEGRATIVE PROTEIN FUNCTION PREDICTION

In this section, we will delve into the methods of integrating multiple sources of information to predict protein function.

DeepGraphGO [67] aims to tackle the limitation of protein interaction-based methods that did not include sequence information. It introduced a multispecies strategy to incorporate the data of all species to train a single model. This approach significantly augments the number of training samples, surpassing the capabilities of existing network-based methods using less data at the time. Binary input protein features are generated through InterProScan, wherein each element indicates the presence or absence of a protein domain, family, or motif. These binary features are combined with protein network graphs, where proteins serve as the nodes and PPIs form the edges for functional protein annotation. DeepGraphGO prediction comprises three primary steps. First, a fully connected layer is employed to convert the binary features into a nonbinary vector with reduced dimensions, serving as the initial feature representation. Next, updating the representation vector of each node and incorporating new information from network interactions is achieved through a graph CNN. Finally, a fully connected layer is utilized to predict probabilities of GO terms.

Graph2GO [68] is a multimodal graph-based representation learning model that integrates heterogeneous information. This model incorporates multiple types of protein interaction networks derived from sequence similarity and PPI, along with protein features such as amino acid sequence, subcellular location, and protein domains. The Graph2GO pipeline is composed of two Variational Graph Auto-Encoder (VGAE) [69] models for the PPI network and sequence similarity network (SSN). These VGAE models extract representative embeddings, which are subsequently used as input to a final fully connected DNN classifier for the prediction of protein functions.

Three version of NetGO methods, NetGO, NetGO2, and NetGO3 are related to an early integrative method—GOlabeler [70], which encompasses the following five distinct components: Naive prediction (GO term frequency), BLAST-KNN (k-nearest neighbor using BLAST results), LR-3mer (Logistic regression of the frequency of amino acid trigrams), LR-InterPro (Logistic regression of InterPro features utilizing rich domain, family, and motif information), and LR-ProFET (Logistic regression of ProFET features). The outputs of these components are combined through learning to rank (LTR) to predict protein function. NetGO [71] introduces a novel component, Net-KNN, incorporating network information into the system. NetGO2 [72] further enhances the system by incorporating two additional components, LR-Text and Seq-RNN, while excluding the LR-ProFET component. For LR-Text, corresponding text data about proteins are extracted from PubMed, forming a document that is represented using sparse TF-IDF (term frequency-inverse document frequency) and dense semantic representations generated by Doc2Vec [73]. Logistic regression is trained with these text-based features. Meanwhile, Seq-RNN is employed to extract deep representations of protein sequences, using a bi-directional long short-term memory (BiLSTM), followed by a fully connected layer to predict functions. NetGO3 [74] modifies the architecture by replacing the Seq-RNN component with LR-ESM. LR-ESM generates embeddings for each protein using ESM-1b [46].

SDN2GO [75] employs an integrated deep learning model combining protein sequence, protein domains, and PPI networks for protein

function prediction. The model has four parts, a sequence submodel, a domain submodel, a PPI-net submodel, and a weighted classifier. The sequence submodel extracts features from sequence input, which is represented as 2D 3-g-vector-matrix. The model uses one-dimensional CNNs to extract in-depth high-dimensional features. The PPI-net submodel utilizes three-layer trapezoidal neural networks to generate the features of PPI Network input. The domain submodel uses the sorted protein domain information as an input for a sparse layer to generate intermediate features. The output of the Sparse layer represented as 2D matrix enters one-dimensional CNNs to extract features. The output features represented as vectors with same dimensions generated by all the three submodels are combined as input for the weighted classifier to predict functions of protein.

PFP-GO [76] also integrates protein sequence, protein domain, and PPI network information for protein function prediction. It first uses the information separately to rank each individual GO term, and the ranking determines which GO terms are associated with the target proteins. In this method, mapping data from one source to another becomes crucial as three complementary information sources are utilized. It makes predictions in four steps. First, a PPI network for target proteins is obtained. Second, only the level-2 neighborhood graph for each target protein is taken into account, eliminating other nonessential proteins. Thirdly, after acquiring refined PPI for each target protein, GO terms are assigned to the target protein and its neighbors using the sequence-, domain-, and interaction neighbor-based approaches. Lastly, GO terms are ranked based on a function enrichment score, and a consensus score is applied to select GO terms for each target protein.

Like PFP-GO, MultiPredGO [77] predicts protein functions by combining protein sequence, protein structure, and PPI network information. Two individual deep learning models are used for feature extraction from sequence and structure, and a pretrained knowledge graph embedding method is used for PPI network. The sequence is first transformed into a trigram and then processed by an embedding layer. Then, the embedding output passes through one-dimensional convolutional layer for feature extraction. For the structure, a 3D structure is retrieved from Protein Data Bank (PDB) if available, and converted into four distinct 3D voxel representation. Then, an off-shelf residual network, ResNet-50 [78], is employed to extract features from the structure. Lastly, extracted features from sequence and structure are combined with PPI network information to obtain the final features, which are processed by a neuro-symbolic hierarchical classifier to make function prediction.

Finally, DeepGATGO [79] is an integrative function prediction method leveraging a graph attention learning network (GATs) and a contrastive learning [80, 81] approach to combine protein sequence information and structural and semantic information of GO terms to predict protein functions. It utilizes ESM-1b [46] pretrained language model to extract feature embeddings from protein sequences. The structural information of GO terms is extracted using GAT network. The semantic information of GO terms is generated through contrastive learning from embeddings created using their names and textual descriptions by the BioBert [82] pretrained NLP model. The extracted semantic features and structural features of GO terms are concatenated. The resulting concatenation output is then multiplied with the protein sequence features. The concatenated features are used by a classification layer with the triplet loss and binary cross-entropy loss to predict the functions of proteins.

6 | FEW-SHOT LEARNING-BASED PROTEIN FUNCTION PREDICTION

One significant challenge in protein function prediction is to predict GO terms that are associated with few proteins because they are severely underrepresented or not present in the training data. For instance, more than 20,000 GO terms have <100 annotated proteins possessing them as function.

This mirrors the complexities of the few-shot/zero-shot problem, where models must predict classes with minimal or no training examples. To tackle this, effective methods are developed to teach models to recognize both seen and unseen classes without labeled samples of the latter, leveraging knowledge transfer from seen to unseen classes. These methods typically operate in two primary forms: Embedding-based methods, which associate low-level features of seen classes with semantic vectors, facilitating recognition of novel classes through similarity measurements in the embedding space, and Generative-based methods, which generate samples for unseen classes using data from seen classes and semantic representations [16].

In the function prediction domain, most methods tackle this problem by using semantic information of GO terms [30]. That is given the scarcity of labeled examples for rare GO terms, semantic information is harnessed to establish meaningful relationships between rare GO terms and common GO terms. Examples of semantic information include leveraging the hierarchical relationships within the GO graph and utilizing GO textual descriptions. Another way is to apply embedding functions to associate features with labels, projecting both feature and label embeddings into a common space and aligning similar GO terms nearby.

TALE [30] jointly embeds sequence and hierarchical function labels into a latent space, allowing it to generalize to novel/rare terms. Tale focuses on terms that have at least one protein annotation and simultaneously embeds protein sequences and hierarchical function labels using the attention mechanism.

ProTranslator [32] transfers function annotations with similar textual descriptions to annotate a novel function. Leveraging textual descriptions, ProTranslator embeds GO functions using their textual descriptions. The embedding is performed using PubMedBert [83], a language model pretrained on PubMed abstracts and full-text articles. Proteins are embedded to generate the following three widely used features: sequence features, textual description features, and PPI-network features. Similar to deepGOPlus, the sequence features are extracted using CNNs with multiple one-dimensional convolution kernels. Textual descriptions are obtained from GeneCards [22]. The PPI-network features are obtained from pretrained Mashup representations calculated from PPI networks. Ultimately, GO terms and

proteins are projected into the same low-dimensional space using a bilinear layer.

DeepGOZero [84] improves predictions for rare GO classes with limited or zero annotations using a model-theoretic approach (ELEmbeddings [85]) to learn ontology embeddings. The ELEmbeddings represent classes as *n*-balls and relations as vectors to embed ontology semantics into a geometric model. It also uses Interpro domain annotations to generate an embedding of size 1024 for each protein. The protein embeddings and ontology embeddings are combined to predict GO terms.

7 | DATA SOURCES, CRITICAL ASSESSMENT OF PROTEIN FUNCTION ANNOTATION (CAFA), AND EVALUATION METRICS

7.1 | Data sources

Curating high-quality training and test datasets is a key to develop accurate deep learning methods for protein function prediction. Protein sequences and function labels are often sourced from the UniProt Knowledgebase (UniProtKB) [21]. UniProtKB consists of two sections: UniProtKB/Swiss-Prot (reviewed, manually annotated proteins) and UniProtKB/TrEMBL (unreviewed, automatically annotated proteins). The former contains protein sequences and function labels that have been carefully, manually annotated, while the latter includes computationally analyzed records awaiting full manual annotation. To obtain high-quality labels, the proteins in UniProtKB/Swiss-Prot are usually used to create training and test datasets.

The structure for a protein can be directly predicted by protein structure prediction tools such as AlphaFold or collected from PDB [18] and AlphafoldDB [17] if available. PPI networks are usually retrieved from the STRING database integrating huge amounts of experimentally determined and predicted PPIs. InterPro is a valuable source to obtain the family and function motif/site annotations for proteins and domains, which can be used as input features for protein function prediction. InterPro integrates the data from 13 member databases, forming the InterPro consortium, including CATH [86, 87], CDD [88], HAMAP [89], MobiDB Lite [90], Panther [91], Pfam [92], PIRSF [93], PRINTS [94], Prosite [95], SFLD [96], SMART [97], SUPERFAMILY [98, 99], and NCBIfam. All the features for a protein in Interpro can be obtained using the interproscan (a tool to scan sequences against all InterPro's member databases) or downloaded from the InterPro website. Finally, protein textual descriptions can be gathered from UniProtKB and GeneCards.

7.2 | Critical assessment of function annotation (CAFA)

Objectively and rigorously assessing the performance of different protein function prediction methods is important to advance the field. The Critical Assessment of Function Annotation (CAFA) [12, 13], a global, community-wide experiment held every few years to blindly assess protein function prediction methods. It uses proteins whose function annotations are not available as targets for participating methods to predict their function. The prediction results are then evaluated when the true function annotations of the targets become available. Several CAFA experiments have been held, including the inaugural challenge (CAFA1) taking place in 2010–2011 and the most recent challenge, CAFA5, held in 2023. According to the first four rounds of CAFA experiments (CAFA1-4), the performance of protein function prediction has gradually progressed over years. The results of CAFA5 remain to be seen. CAFA employs a comprehensive approach to collect benchmark datasets, focusing on the annotation growth period between two time points, during which proteins acquire experimental annotations.

EVALUATION METRICS

Evaluating protein function prediction using multiple complementary metrics is important to assess the strength and weakness of function prediction methods. A list of commonly used metrics for evaluating GO term predictions including F-measure, weighted F-measure, and semantic distance (S-score) [12, 13], are briefly discussed below.

The F-measure, based on the precision-recall curve whiles the S-score is based on the remaining uncertainty/missing information (RU-MI) curve, where S stands for semantic distance. The remaining uncertainty of the true annotation of protein represents the information that has not been provided or accounted for by the predicted annotation. The misinformation represents a metric that measures the level of misleading information linked to a predicted annotation.

The F-max is used to represent the maximum F-measure across all decision thresholds, and the S-min represents the shortest semantic distance across all thresholds.

AUPR stands for area under the precision-recall curve, which is also a commonly used evaluation metric. Similarly, AUC measuring the area under the receiver operating characteristic (ROC) curve is often used. A ROC curve is a plot of the true positive rate (TPR) against the false positive rate (FPR) across different cutoff values *t*.

- Precision

$$\mathrm{pr}(\tau) = \frac{1}{m(\tau)} \sum_{i=1}^{m(\tau)} \frac{\sum_f \mathbb{I}(f \in P_i(\tau) \wedge f \in T_i)}{\sum_f \mathbb{I}(f \in P_i(\tau))}$$

- Recall

$$\mathrm{rc}(\tau) = \frac{1}{n_e} \sum_{i=1}^{n_e} \frac{\sum_f \mathbb{I}(f \in P_i(\tau) \wedge f \in T_i)}{\sum_f \mathbb{I}(f \in T_i)}$$

- F_1 score

$$F_{1(\tau)} = 2 \times \frac{\mathrm{pr}(\tau) \times \mathrm{rc}(\tau)}{\mathrm{pr}(\tau) + \mathrm{rc}(\tau)}$$

- Maximum F_1 score

$$F_{\max} = \max_{\tau} \left(F_1(\tau)\right)$$

where f is a term, $P_i(\tau)$ is the set of predictions, T_i denotes the corresponding ground-truth, i represents the protein sequence under consideration, and τ is the decision threshold. $m(\tau)$ is the number of proteins sequences with at least one predicted score greater than or equal to the decision threshold τ, $\mathbb{I}(\cdot)$ is an indicator function, and n_e is the number of proteins in the test set for a particular test study.

- Information content (ic) of term f is computed as

$$\mathrm{IC}(f) = \log_2 \frac{1}{\Pr(f|P(f))}$$

- Weighted precision

$$\mathrm{wpr}(\tau) = \frac{1}{m(\tau)} \sum_{i=1}^{m(\tau)} \frac{\sum_f ic(f) \cdot \mathbb{I}(f \in P_i(\tau) \wedge T_i(\tau))}{\sum_f ic(f) \cdot \mathbb{I}(f \in P_i(\tau))}$$

- Weighted recall

$$\mathrm{wrc}(\tau) = \frac{1}{n_e} \sum_{i=1}^{n_e} \frac{\sum_f ic(f) \cdot \mathbb{I}(f \in P_i(\tau) \wedge T_i(\tau))}{\sum_f ic(f) \cdot \mathbb{I}(f \in T_i(\tau))}$$

Here, $\Pr(f|P(f))$ represents the probability that term f in the ontology is associated with a protein given that all of its parents are associated.

- Remaining uncertainty

$$ru(\tau) = \frac{1}{ne} \sum_{i=1}^{ne} \sum_f ic(f) \cdot \mathbb{I}\left(f \notin P_i(\tau) \wedge f \in Ti\right)$$

- Missing information

$$mi(\tau) = \frac{1}{ne} \sum_{i=1}^{ne} \sum_f ic(f) \cdot \mathbb{I}\left(f \in P_i(\tau) \wedge f \notin Ti\right)$$

- S_{min}

$$S_{min} = \min_{\tau} \sqrt{ru(\tau)^2 + mi(\tau)^2}, \tau$$

- Area under precision recall curve (AUPR)

$$\mathrm{AUPR} = \int_0^1 \mathrm{Precision}(R)\, dR$$

where $\mathrm{Precision}(R)$ represents the precision at a given recall level (R).

The CAFA-evaluator [100] is an open-source Python software designed to assess the performance of function prediction methods. The tool evaluates the metrics discussed above. Additionally, it offers a Jupyter Notebook to generate average precision scores, and precision-recall and remaining uncertainty–misinformation curves.

8 | CHALLENGES AND FUTURE DIRECTION

As discussed in the previous sections, substantial advances in developing deep learning methods for protein function prediction have been made by the community in the last several years. However, the accuracy of protein function still has not reached the high-accuracy level of protein structure prediction that has made it an indispensable tool for biomedical research. There are at least three major challenges in protein function prediction that need to be addressed in order to substantially improve its accuracy.

The first major challenge is to develop highly sophisticated deep learning and AI methods to synergistically integrate multiple modalities of input data (e.g., protein sequence, protein structure, protein interaction, protein/domain family information, and biological textual description) to improve protein function. Most existing integrative methods [70, 72, 74–76] simply extract features from each data modality and then concatenate them without letting modalities systematically interact with each other in the feature extraction process. The techniques used by the LLMs such as ChatGPT-4 and Gemini [101] to integrate multiple modality data such as text, image, video, and voice through seamless cross-modality communication may be transferred to the protein function prediction field to integrate multiple modalities of protein data. And it is time to develop multimodal LLMPs as multimodality protein data such as sequences and structures are ubiquitously available nowadays. However, this may introduce its own challenges, such as increased model complexity and scalability issues. It is crucial to consider these factors, especially in the context of large-scale protein function prediction tasks, to ensure the methods remain practical and feasible.

The second major challenge is how to more effectively leverage the evolutionary information hidden in the hundreds of millions of protein sequences better to improve protein function prediction. A promising direction is to develop more sophisticated LLMP sequences that can be directly fine-tuned or promoted to predict protein function [102]. The current application of LLMP such as ESM-1b is still in the early stage and at a shallow level because the pretrained LLMP are mostly used to generate features from sequences as input for protein function prediction. One way to deepen the application of LLMP in protein function prediction is to directly fine tune the weights of the pretrained LLMP component in the protein function prediction system during the training of the system. Another way is to add function prediction into the designing and training of LLMPs in the first place so that they are intrinsically built for protein function prediction. For instance, a LLMP can be designed to predict masked or next amino acids through self-supervised learning as well as function terms through supervised learning. The LLMP can be mainly trained on millions of unlabeled protein sequences to predict masked or next amino acids and auxilinarily trained to predict function terms of thousands

of proteins with function labels at the same time as how a LLM for NLP was trained to predict next (masked) tokens and classify sentences simultaneously [103]. Readers may refer to this recent work [104] for some detailed strategies for fine-tuning and training LLMPs for protein function prediction.

The third major challenge is to improve the prediction accuracy for rare GO terms with low frequency in protein function annotations or novel GO terms that never occur before. Some rare GO terms are highly specific GO terms that occur at the bottom level of the GO graph, which are important for protein function annotation but very hard to predict. As demonstrated by some zero- or few-shot prediction methods such as TALE [30] and ProTranslator [32], zero- or few-shot learning methods [105] used in NLP, computer vision, and image processing may be transferred to the field of protein function prediction. Particularly, we envision that the prompt engineering and in-context learning [106] used with LLMs for NLP can also be used with LLMPs to predict rare or novel GO terms, provided that LLMPs fine-tuned for protein function prediction, akin to LLMs for NLP, are developed in the field. Therefore, a user can use one or a few rare GO terms as examples as prompts to guide the pretrained LLMPs to predict rare or novel GO terms in any context as one uses prompts to instruct ChatGPT to learn new concepts or skills.

In summary, we envision that developing next-generation sophisticated LLMPs that can handle multiple modalities of protein data, be fined tuned directly by function labels, or be customized by prompt-based in-context learning for protein function prediction may be a promising avenue for tackling some major challenges in protein function prediction, such as multimodality data integration, extracting evolutionary information from millions of sequences, and predicting rare/novel GO terms, to push the performance of protein function prediction to the next level.

AUTHOR CONTRIBUTIONS

Jianlin Cheng conceived the review project and the future development directions. Frimpong Boadu and Ahhyun Lee collected the data. Frimpong Boadu, Ahhyun Lee, and Jianlin Cheng wrote the manuscript.

ACKNOWLEDGMENTS

This work is supported in part by grants from the National Science Foundation (NSF Grant #: DBI2308699 and CCF2343612) and the National Institutes of Health (NIH Grant #: R01GM093123 and R01GM146340).

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

This is a review article. It does not contain separate scientific data beyond what has been presented in the article.

ORCID

Jianlin Cheng https://orcid.org/0000-0003-0305-2853

REFERENCES

1. LaPelusa, A., & Kaushik, R. (2022). Physiology, proteins. *StatPearls*. In: *StatPearls [Internet]*. StatPearls Publishing.
2. Giri, N., & Cheng, J. (2024). De novo atomic protein structure modeling for cryoEM density maps using 3D transformer and HMM. *Nature Communications*, *15*(1), 5511.
3. Bull, S. C., & Doig, A. J. (2015). Properties of protein drug target classes. *PLoS One*, *10*(3), e0117955.
4. Santos, R., Ursu, O., Gaulton, A., Bento, A. P., Donadi, R. S., Bologa, C. G., Karlsson, A., Al-Lazikani, B., Hersey, A., Oprea, T. I., & Overington, J. P. (2017). A comprehensive map of molecular drug targets. *Nature Reviews Drug Discovery*, *16*(1), 19–34. Epub 2016 Dec 2.
5. Dhakal, A., McKay, C., Tanner, J. J., & Cheng, J. (2021). Artificial intelligence in the prediction of protein–ligand interactions: Recent advances and future directions. *Briefings in Bioinformatics*, *23*(1), bbab476.
6. Senior, A. W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Žídek, A., Nelson, A. W., Bridgland, A., Penedones, H., Petersen, S., Simonyan, K., Crossan, S., Kohli, P., Jones, D. T., Silver, D., Kavukcuoglu, K., & Hassabis, D. (2019). Protein structure prediction using multiple deep neural networks in the 13th Critical Assessment of Protein Structure Prediction (CASP13). *Proteins: Structure, Function, and Bioinformatics*, *87*(12), 1141–1148.
7. Hou, J., Wu, T., Cao, R., & Cheng, J. (2019). Protein tertiary structure modeling driven by deep learning and contact distance prediction in CASP13. *Proteins: Structure, Function, and Bioinformatics*, *87*(12), 1165–1178.
8. Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., … Hassabis, D. (2021). Highly accurate protein structure prediction with alphafold. *Nature*, *596*(7873), 583–589.
9. Baek, M., DiMaio, F., Anishchenko, I., Dauparas, J., Ovchinnikov, S., Lee, G. R., Wang, J., Cong, Q., Kinch, L. N., Schaeffer, R. D., Millán, C., Park, H., Adams, C., Glassman, C. R., DeGiovanni, A., Pereira, J. H., Rodrigues, A. V., van Dijk, A. A., Ebrecht, A. C., & Baker, D. (2021). Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, *373*(6557), 871–876.
10. Le, N. Q. K., Li, W., & Cao, Y. (2023). Sequence-based prediction model of protein crystallization propensity using machine learning and two-level feature selection. *Briefings in Bioinformatics*, *24*(5), bbad319.
11. Radivojac, P., Clark, W. T., Oron, T. R., Schnoes, A. M., Wittkop, T., Sokolov, A., Graim, K., Funk, C., Verspoor, K., Ben-Hur, A., Orengo, C. A., & Rost, B. (2013). A large-scale evaluation of computational protein function prediction. *Nature Methods*, *10*(3), 221–227.
12. Jiang, Y., Oron, T. R., Clark, W. T., Bankapur, A. R., D'Andrea, D., Lepore, R., Funk, C. S., Kahanda, I., Verspoor, K. M., Ben-Hur, A., Giollo, M., & Piovesan, D. (2016). An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biology*, *17*(1), 1–19.
13. Zhou, N., Jiang, Y., Bergquist, T. R., Lee, A. J., Kacsoh, B. Z., Crocker, A. W., Lewis, K. A., Georghiou, G., Nguyen, H. N., Hamid, M. N., Toppo, S., & Lavezzo, E. (2019). The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome Biology*, *20*(1), 1–23.
14. Consortium, G. O. (2004). The gene ontology (GO) database and informatics resource. *Nucleic Acids Research*, *32*(suppl_1), D258–D261.
15. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., & Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901.

16. Pourpanah, F., Abdar, M., Luo, Y., Zhou, X., Wang, R., Lim, C. P., Wang, X. Z., & Wu, Q. J. (2022). A review of generalized zero-shot learning methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*, 4051–4070.
17. Varadi, M., Anyango, S., Deshpande, M., Nair, S., Natassia, C., Yordanova, G., Yuan, D., Stroe, O., Wood, G., Laydon, A., Hassabis, D., & Velankar, S. (2022). Alphafold protein structure database: Massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Research*, *50*(D1), D439–D444.
18. Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., & Bourne, P. E. (2000). The protein data bank. *Nucleic Acids Research*, *28*(1), 235–242.
19. Szklarczyk, D., Franceschini, A., Wyder, S., Forslund, K., Heller, D., Huerta-Cepas, J., Simonovic, M., Roth, A., Santos, A., Tsafou, K. P., Jensen, L. J., & von Mering, C. (2015). String v10: Protein–protein interaction networks, integrated over the tree of life. *Nucleic Acids Research*, *43*(D1), D447–D452.
20. Hunter, S., Apweiler, R., Attwood, T. K., Bairoch, A., Bateman, A., Binns, D., Bork, P., Das, U., Daugherty, L., Duquenne, L., Wu, C. H., & Yeats, C. (2009). InterPro: The integrative protein signature database. *Nucleic Acids Research*, *37*(suppl_1), D211–D215.
21. Consortium, T. U. (2022). UniProt: The Universal Protein Knowledgebase in 2023. *Nucleic Acids Research*, *51*(D1), D523–D531.
22. Stelzer, G., Rosen, N., Plaschkes, I., Zimmerman, S., Twik, M., Fishilevich, S., Stein, T. I., Nudel, R., Lieder, I., Mazor, Y., Safran, M., & Lancet, D. (2016). The genecards suite: From gene data mining to disease genome sequence analyses. *Current Protocols in Bioinformatics*, *54*(1), 1–30.
23. O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458.
24. Medsker, L. R., & Jain, L. (2001). Recurrent neural networks. *Design and Applications*, *5*(64-67), 2.
25. Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*(11), 2673–2681.
26. Sze, V., Chen, Y. H., Yang, T. J., & Emer, J. S. (2017). Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, *105*(12), 2295–2329.
27. Mahmud, S., Soltanikazemi, E., Boadu, F., Dhakal, A., & Cheng, J. (2022). Deep learning prediction of severe health risks for pediatric Covid-19 patients with a large feature set in 2021 barda data challenge. ArXiv.
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (Vol. 30).
29. Cheng, J., Dong, L., & Lapata, M. (2016). Long short-term memory-networks for machine reading. arXiv preprint arXiv:1601.06733.
30. Cao, Y., & Shen, Y. (2021). TALE: Transformer-based protein function annotation with joint sequence—label embedding. *Bioinformatics*, *37*(18), 2825–2833.
31. Boadu, F., Cao, H., & Cheng, J. (2023). Combining protein sequences and structures with transformers and equivariant graph neural networks to predict protein function. *Bioinformatics*, *39*(suppl_1), i318–i325.
32. Xu, H., & Wang, S. (2022). ProTranslator: Zero-shot protein function prediction using textual description. In International conference on research in computational molecular biology (pp. 279–294). Springer.
33. Cao, R., Freitas, C., Chan, L., Sun, M., Jiang, H., & Chen, Z. (2017). ProLanGO: Protein function prediction using neural machine translation based on a recurrent neural network. *Molecules*, *22*(10), 1732.
34. Ko, C. W., Huh, J., & Park, J. W. (2022). Deep learning program to predict protein functions based on sequence information. *MethodsX*, *9*, 101622.
35. Kulmanov, M., & Hoehndorf, R. (2020). DeepGOPlus: Improved protein function prediction from sequence. *Bioinformatics*, *36*(2), 422–429.
36. Xia, W., Zheng, L., Fang, J., Li, F., Zhou, Y., Zeng, Z., Zhang, B., Li, Z., Li, H., & Zhu, F. (2022). PFmulDL: A novel strategy enabling multi-class and multi-label protein function annotation by integrating diverse deep learning methods. *Computers in Biology and Medicine*, *145*, 105465.
37. Rifaioglu, A. S., Doğan, T., Jesus Martin, M., Cetin-Atalay, R., & Atalay, V. (2019). DEEPred: Automated protein function prediction with multi-task feed-forward deep neural networks. *Scientific Reports*, *9*, 7344.
38. Buchfink, B., Xie, C., & Huson, D. H. (2015). Fast and sensitive protein alignment using diamond. *Nature Methods*, *12*(1), 59–60.
39. Oliveira, G. B., Pedrini, H., & Dias, Z. (2023). TEMPROT: Protein Function Annotation Using Transformers Embeddings and Homology Search. *BMC Bioinformatics*, *24*(1), 242.
40. Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., Bhowmik, D., & Rost, B. (2021). Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(10), 7112–7127.
41. Steinegger, M., Mirdita, M., & Söding, J. (2019). Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold. *Nature Methods*, *16*(7), 603–606.
42. Steinegger, M., & Söding, J. (2018). Clustering huge protein sequence sets in linear time. *Nature Communications*, *9*(1), 2542.
43. Altschul, S., Madden, T., Schäffer, A., Zhang, J., Zhang, Z., Miller, W., & Lipman, D. (1997). Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research*, *25*(17), 3389–3402.
44. Yuan, Q., Xie, J., Xie, J., Zhao, H., & Yang, Y. (2023). Fast and accurate protein function prediction from sequence through pretrained language model and homology-based label diffusion. *Briefings in Bioinformatics*, *24*(3), bbad117.
45. Zhu, Y. H., Zhang, C., Yu, D. J., & Zhang, Y. (2022). Integrating unsupervised language model with triplet neural networks for protein gene ontology prediction. *PLOS Computational Biology*, *18*(12), 1–26.
46. Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., & Fergus, R. (2019). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, *118*(15), e2016239118.
47. Zhao, C., Liu, T., & Wang, Z. (2022). PANDA2: Protein function prediction using graph neural networks. *NAR Genomics and Bioinformatics*, *4*(1), lqac004.
48. Fey, M., & Lenssen, J. E. (2019). Fast graph representation learning with PyTorch geometric. arXiv preprint arXiv:1903.02428.
49. Wang, M., Zheng, D., Ye, Z., Gan, Q., Li, M., Song, X., Zhou, J., Ma, C., Yu, L., Gai, Y., & Karypis, G. (2019). Deep graph library: A graph-centric, highly-performant package for graph neural networks. arXiv preprint arXiv:1909.01315.
50. Gligorijević, V., Renfrew, P., Kosciolek, T., Leman, J. K., Berenberg, D., Vatanen, T., Chandler, C., Taylor, B. C., Fisk, I. M., Vlamakis, H., Cho, K., & Bonneau, R. (2021). Structure-based protein function prediction using graph convolutional networks. *Nature Communications*, *12*(1), 3168.
51. Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In International conference on learning representations (ICLR).
52. Schwede, T., Kopp, J., Guex, N., & Peitsch, M. C. (2003). SWISS-MODEL: An automated protein homology-modeling server. *Nucleic Acids Research*, *31*(13), 3381–3385.
53. Lai, B., & Xu, J. (2022). Accurate protein function prediction via graph attention networks with predicted structure information. *Briefings in Bioinformatics*, *23*(1), bbab502.

54. Peng, J., & Xu, J. (2011). RaptorX: Exploiting structure information for protein alignment by statistical inference. *Proteins: Structure, Function, and Bioinformatics*, *79*(S10), 161–171.
55. Xu, J., Mcpartlon, M., & Li, J. (2021). Improved protein structure prediction by deep learning irrespective of co-evolution information. *Nature Machine Intelligence*, *3*, 601–609.
56. Satorras, V. G., Hoogeboom, E., & Welling, M. (2021). E(n) equivariant graph neural networks. In International conference on machine learning.
57. Jiao, P., Wang, B., Wang, X., Liu, B., Wang, Y., & Li, J. (2023). Struct2GO: Protein function prediction based on graph pooling algorithm and AlphaFold2 structure information. *Bioinformatics*, *39*(10), btad637.
58. Grover, A., & Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 855–864).
59. Heinzinger, M., Elnaggar, A., Wang, Y., Dallago, C., Nechaev, D., Matthes, F., & Rost, B. (2019). Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinformatics*, *20*(1), 1–17.
60. Cho, H., Berger, B., & Peng, J. (2016). Compact integration of multi-network topology for functional analysis of genes. *Cell Systems*, *3*(6), 540–548.
61. Gligorijević, V., Barot, M., & Bonneau, R. (2018). deepNF: Deep network fusion for protein function prediction. *Bioinformatics*, *34*(22), 3873–3881.
62. Wu, K., Zhou, D., Slonim, D., Hu, X., & Cowen, L. (2023). MELISSA: Semi-supervised embedding for protein function prediction across multiple networks. *bioRxiv*.
63. Barot, M., Gligorijević, V., Cho, K., & Bonneau, R. (2021). NetQuilt: Deep multispecies network-based protein function prediction using homology-informed network similarity. *Bioinformatics*, *37*(16), 2414–2422.
64. Singh, R., Xu, J., & Berger, B. (2008). Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proceedings of the National Academy of Sciences*, *105*(35), 12763–12768.
65. Kulmanov, M., Khan, M. A., & Hoehndorf, R. (2018). DeepGo: Predicting protein functions from sequence and interactions using a deep ontology-aware classifier. *Bioinformatics*, *34*(4), 660–668.
66. Wan, C., Cozzetto, D., Fa, R., & Jones, D. T. (2019). Using deep maxout neural networks to improve the accuracy of function prediction from protein interaction networks. *PloS One*, *14*(7), e0209958.
67. You, R., Yao, S., Mamitsuka, H., & Zhu, S. (2021). DeepGraphGO: Graph neural network for large-scale, multispecies protein function prediction. *Bioinformatics*, *37*(suppl_1), i262–i271.
68. Fan, K., Guan, Y., & Zhang, Y. (2020). Graph2GO: A multi-modal attributed network embedding method for inferring protein functions. *GigaScience*, *9*(8), giaa081.
69. Kipf, T. N., & Welling, M. (2016). Variational graph auto-encoders. arXiv preprint arXiv:1611.07308.
70. You, R., Zhang, Z., Xiong, Y., Sun, F., Mamitsuka, H., & Zhu, S. (2018). GOLabeler: Improving sequence-based large-scale protein function prediction by learning to rank. *Bioinformatics*, *34*(14), 2465–2473.
71. You, R., Yao, S., Xiong, Y., Huang, X., Sun, F., Mamitsuka, H., & Zhu, S. (2019). NetGO: Improving large-scale protein function prediction with massive network information. *Nucleic Acids Research*, *47*(W1), W379–W387.
72. Yao, S., You, R., Wang, S., Xiong, Y., Huang, X., & Zhu, S. (2021). NetGO 2.0: Improving large-scale protein function prediction with massive sequence, text, domain, family and network information. *Nucleic Acids Research*, *49*(W1), W469–W475.
73. Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In International conference on machine learning (pp. 1188–1196).
74. Wang, S., You, R., Liu, Y., Xiong, Y., & Zhu, S. (2023). NetGO 3.0: Protein language model improves large-scale functional annotations. *Genomics, Proteomics & Bioinformatics*, *21*, 349–358.
75. Cai, Y., Wang, J., & Deng, L. (2020). SDN2GO: An integrated deep learning model for protein function prediction. *Frontiers in Bioengineering and Biotechnology*, *8*, 391.
76. Sengupta, K., Saha, S., Halder, A. K., Chatterjee, P., Nasipuri, M., Basu, S., & Plewczynski, D. (2022). PFP-GO: Integrating protein sequence, domain and protein-protein interaction information for protein function prediction using ranked go terms. *Frontiers in Genetics*, *13*, 969915.
77. Giri, S. J., Dutta, P., Halani, P., & Saha, S. (2021). MultiPredGO: Deep multi-modal protein function prediction by amalgamating protein structure, sequence, and interaction information. *IEEE Journal of Biomedical and Health Informatics*, *25*(5), 1832–1838.
78. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770–778).
79. Li, Z., Jiang, C., & Li, J. (2023). DeepGATGO: A hierarchical pretraining-based graph-attention model for automatic protein function prediction. arXiv preprint arXiv:2307.13004.
80. Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In International conference on machine learning (pp. 1597–1607).
81. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., & Krishnan, D. (2020). Supervised contrastive learning. *Advances in Neural Information Processing Systems*, *33*, 18661–18673.
82. Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2019). BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, *36*(4), 1234–1240.
83. Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., Naumann, T., Gao, J., & Poon, H. (2021). Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, *3*(1), 1–23.
84. Kulmanov, M., & Hoehndorf, R. (2022). DeepGOZero: Improving protein function prediction from sequence and zero-shot learning based on ontology axioms. *Bioinformatics*, *38*(supplement_1), i238–i245.
85. Kulmanov, M., Liu-Wei, W., Yan, Y., & Hoehndorf, R. (2019). EL embeddings: Geometric construction of models for the description logic EL++. arXiv preprint arXiv:1902.10499.
86. Lewis, T. E., Sillitoe, I., Dawson, N., Lam, S. D., Clarke, T., Lee, D., Orengo, C., & Lees, J. (2017). Gene3D: Extensive prediction of globular domains in proteins. *Nucleic Acids Research*, *46*(D1), D435–D439.
87. Sillitoe, I., Bordin, N., Dawson, N., Waman, V. P., Ashford, P., Scholes, H. M., Pang, C. S. M., Woodridge, L., Rauer, C., Sen, N., Abbasian, M., Le Cornu, S., Lam, S. D., Berka, K., Varekova, I., Svobodova, R., Lees, J., & Orengo, C. A. (2020). CATH: Increased structural coverage of functional space. *Nucleic Acids Research*, *49*(D1), D266–D273.
88. Wang, J., Chitsaz, F., Derbyshire, M. K., Gonzales, N. R., Gwadz, M., Lu, S., Marchler, G., Song, J., Thanki, N., Yamashita, R., Yang, M., Zhang, D., Zheng, C., Lanczycki, C., & Marchler-Bauer, A. (2022). The conserved domain database in 2023. *Nucleic Acids Research*, *51*(D1), D384–D388.
89. Pedruzzi, I., Rivoire, C., Auchincloss, A. H., Coudert, E., Keller, G., de Castro, E., Baratin, D., Cuche, B. A., Bougueleret, L., Poux, S., Redaschi, N., Xenarios, I., & Bridge, A. (2014). HAMAP in 2015: Updates to the protein family classification and annotation system. *Nucleic Acids Research*, *43*(D1), D1064–D1070.
90. Necci, M., Piovesan, D., Clementel, D., Dosztányi, Z., & Tosatto, S. C. E. (2020). MobiDB-lite 3.0: Fast consensus annotation of intrinsic disorder flavors in proteins. *Bioinformatics*, *36*(22-23), 5533–5534.
91. Mi, H., Muruganujan, A., Casagrande, J. T., & Thomas, P. D. (2013). Large-scale gene function analysis with the panther classification system. *Nature Protocols*, *8*(8), 1551–1566.

92. Mistry, J., Chuguransky, S., Williams, L., Qureshi, M., Salazar, G. A., Sonnhammer, E. L., Tosatto, S. C., Paladin, L., Raj, S., Richardson, L. J., Finn, R. D., & Bateman, A. (2021). Pfam: The protein families database in 2021. *Nucleic Acids Research*, *49*(D1), D412–D419.
93. Wu, C. H., Nikolskaya, A., Huang, H., Yeh, L. S. L., Natale, D. A., Vinayaka, C. R., Hu, Z. Z., Mazumder, R., Kumar, S., Kourtesis, P., Ledley, R. S., Suzek, B. E., Arminski, L., Chen, Y., Zhang, J., Cardenas, J. L., Chung, S., Castro-Alvear, J., Dinkov, G., & Barker, W. C. (2004). PIRSF: Family classification system at the protein information resource. *Nucleic Acids Research*, *32*(suppl_1), D112–D114.
94. Attwood, T. K., Coletta, A., Muirhead, G., Pavlopoulou, A., Philippou, P. B., Popov, I., Roma-Mateo, C., Theodosiou, A., & Mitchell, A. L. (2012). The prints database: A fine-grained protein sequence annotation and analysis resource – its status in 2012. *Database*, *2012*, bas019.
95. Sigrist, C. J., De Castro, E., Cerutti, L., Cuche, B. A., Hulo, N., Bridge, A., Bougueleret, L., & Xenarios, I. (2012). New and continuing developments at prosite. *Nucleic Acids Research*, *41*(D1), D344–D347.
96. Akiva, E., Brown, S., Almonacid, D. E., Barber, 2nd., Alan E., Custer, A. F., Hicks, M. A., Huang, C. C., Lauck, F., Mashiyama, S. T., Meng, E. C., Mischel, D., Morris, J. H., Ojha, S., Schnoes, A. M., Stryke, D., Yunes, J. M., Ferrin, T. E., Holliday, G. L., & Babbitt, P. C. (2013). The structure–function linkage database. *Nucleic Acids Research*, *42*(D1), D521–D530.
97. Schultz, J., Copley, R. R., Doerks, T., Ponting, C. P., & Bork, P. (2000). SMART: A web-based tool for the study of genetically mobile domains. *Nucleic Acids Research*, *28*(1), 231–234.
98. Gough, J., Karplus, K., Hughey, R., & Chothia, C. (2001). Assignment of homology to genome sequences using a library of hidden Markov models that represent all proteins of known structure. *Journal of Molecular Biology*, *313*(4), 903–919.
99. Wilson, D., Pethica, R., Zhou, Y., Talbot, C., Vogel, C., Madera, M., Chothia, C., & Gough, J. (2009). Superfamily–comparative genomics, datamining and sophisticated visualisation. *Nucleic Acids Research*, *37*, D380–D386.
100. Piovesan, D., Zago, D., Joshi, P., De Paolis Kaluza, M. C., Mehdiabadi, M., Ramola, R., Monzon, A. M., Reade, W., Friedberg, I., Radivojac, P., & Tosatto, S. C. E. (2024). CAFA-evaluator: A Python tool for benchmarking ontological classification methods. *Bioinformatics Advances*, *4*(1), vbae043.
101. Team, G., Anil, R., Borgeaud, S., Wu, Y., Alayrac, J. B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., Hauth, A., Millican, K., Silver, D., Johnson, M., Antonoglou, I., Schrittwieser, J., Glaese, A., Chen, J., Pitler, E., Lillicrap, T., & Vinyals, O. (2023). Gemini: A family of highly capable multimodal models. arXiv preprint arXiv:2312.11805.
102. Le, N. Q. K. (2023). Leveraging transformers-based language models in proteome bioinformatics. *Proteomics*, *23*(23-24), e2300011.
103. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
104. Zhang, Z., Lu, J., Chenthamarakshan, V., Lozano, A., Das, P., & Tang, J. (2024). Structure-informed protein language model. arXiv preprint arXiv:2402.05856.
105. Wang, Y., Yao, Q., Kwok, J. T., & Ni, L. M. (2020). Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, *53*(3), 1–34.
106. White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., & Schmidt, D. C. (2023). A prompt pattern catalog to enhance prompt engineering with chatGPT. arXiv preprint arXiv:2302.11382.

How to cite this article: Boadu, F., Lee, A., & Cheng, J. (2025). Deep learning methods for protein function prediction. *Proteomics*, *25*, e2300471.
https://doi.org/10.1002/pmic.202300471