OXFORD

Briefings in Bioinformatics, 2023, **24(3)**, 1–11
https://doi.org/10.1093/bib/bbad117
Advance access publication date 24 March 2023
Problem Solving Protocol

Fast and accurate protein function prediction from sequence through pretrained language model and homology-based label diffusion

Qianmu Yuan, Junjie Xie, Jiancong Xie, Huiying Zhao and Yuedong Yang

Corresponding authors: Yuedong Yang, School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510000, China, and Key Laboratory of Machine Intelligence and Advanced Computing of MOE, Sun Yat-sen University, Guangzhou 510000, China. Tel.: +86-020-37106046; Fax: +86-020-37106020; E-mail: yangyd25@mail.sysu.edu.cn; Huiying Zhao, Basic and Translational Medicine Research Center, Sun Yat-sen Memorial Hospital, Sun Yat-sen University, Guangzhou 510000, China. Tel.: +86-020-81332199; Fax: +86-020-81332199; E-mail: zhaohy8@mail.sysu.edu.cn

Abstract

Protein function prediction is an essential task in bioinformatics which benefits disease mechanism elucidation and drug target discovery. Due to the explosive growth of proteins in sequence databases and the diversity of their functions, it remains challenging to fast and accurately predict protein functions from sequences alone. Although many methods have integrated protein structures, biological networks or literature information to improve performance, these extra features are often unavailable for most proteins. Here, we propose SPROF-GO, a Sequence-based alignment-free PROtein Function predictor, which leverages a pretrained language model to efficiently extract informative sequence embeddings and employs self-attention pooling to focus on important residues. The prediction is further advanced by exploiting the homology information and accounting for the overlapping communities of proteins with related functions through the label diffusion algorithm. SPROF-GO was shown to surpass state-of-the-art sequence-based and even network-based approaches by more than 14.5, 27.3 and 10.1% in area under the precision-recall curve on the three sub-ontology test sets, respectively. Our method was also demonstrated to generalize well on non-homologous proteins and unseen species. Finally, visualization based on the attention mechanism indicated that SPROF-GO is able to capture sequence domains useful for function prediction. The datasets, source codes and trained models of SPROF-GO are available at https://github.com/biomed-AI/SPROF-GO. The SPROF-GO web server is freely available at http://bio-web1.nscc-gz.cn/app/sprof-go.

Keywords: sequence-based, protein function prediction, pretrained language model, label diffusion

INTRODUCTION

Proteins play crucial roles within living organisms, including signal transduction, catalysis of metabolic reaction and maintenance of cellular structure. Identification of protein functions benefits disease mechanism elucidation and drug target discovery [1]. Since traditional biochemical experiments to determine protein functions are usually expensive, time-consuming, and of low throughput [2], less than 0.1% of the available protein sequences are currently annotated with reliable information [3], and the gap between unannotated and annotated sequences is expanding at an unparalleled rate [4]. Therefore, it is imperative to develop efficient and effective computational methods for protein function prediction [5].

The functions of proteins are standardized by Gene Ontology (GO) [6], which covers three biological domains: molecular function (MF), biological process (BP), and cellular component (CC), with over 43 000 classes/terms (November 2022). Since a protein is usually associated with multiple GO terms, protein function prediction can be regarded as a large-scale, multi-class, and multi-label problem. Moreover, GO is a directed acyclic graph (DAG), in which if a protein is annotated with a GO term, all its ancestor terms up to the root of the ontology should also be annotated. Therefore, protein function predictors should take the hierarchical structure of GO into account and yield 'consistent' outputs: the predicted probability of a GO term must be equal to or greater than all of its child terms [7]. To facilitate this challenging task, the critical assessment of functional annotation (CAFA) competition has been held four times [5, 8, 9] using a time-delayed evaluation process. Specifically, given the target proteins, participants were required to submit the predictions before T_0. After a few months (T_1), the organizers collected proteins with new experimental annotations as the final test set, consisting of no-knowledge and limited-knowledge proteins. Both types of proteins received their first experimental annotations in the target GO domain

Qianmu Yuan is a PhD student in the School of Computer Science and Engineering at Sun Yat-sen University. His research interests lie in deep learning, graph neural network, protein structure prediction and protein function prediction.
Junjie Xie is a master's student in the School of Computer Science and Engineering at Sun Yat-sen University. His research interests include deep learning, graph neural network and molecule generation.
Jiancong Xie is a master's student in the School of Computer Science and Engineering at Sun Yat-sen University. His research interests include deep learning, graph neural network and knowledge graph.
Huiying Zhao is an associate research fellow in the Sun Yat-sen Memorial Hospital at Sun Yat-sen University. Her research interests include pathogenic gene analysis, protein function and RNA function prediction.
Yuedong Yang is a professor in the School of Computer Science and Engineering at Sun Yat-sen University. Currently he focuses on integrating HPC and AI algorithms for biomedical research.
Received: December 9, 2022. **Revised:** February 8, 2023. **Accepted:** March 7, 2023

between T_0 and T_1. However, no-knowledge proteins did not have any experimental annotations before T_0, while limited-knowledge proteins did in domains other than the target domain. Here we focus on the function prediction for no-knowledge proteins as the vast majority of proteins have no experimental annotations.

Current protein function predictors can be roughly grouped into four categories according to their used information: sequence-based, structure-based, biological network-based, and biomedical literature-based methods. Most sequence-based methods employ sequence similarity, search for sequence domain, or adopt deep learning to capture discriminative features to infer functions. Specifically, a basic way is to transfer annotations directly from homologous sequences with known functions, like Blast2GO [10], since similar sequences tend to have similar functions [5]. Another approach is to search for functional sequence domain or family. For example, GOLabeler [11] utilizes learning to rank algorithm to integrate sequence homology, protein domains and families derived from sequences by BLAST [12] and InterProScan [13]. With the development of deep learning technology, discriminative embeddings can also be automatically extracted from preliminary sequences through designing complex neural networks, including convolutional neural networks in DeepGOPlus [14] and transformer in TALE [15]. However, current sequence-based methods suffer from either low prediction accuracy or the high computational cost (owing to the usage of multi-sequence alignment). On the other hand, recent structure-based methods apply native or predicted protein structures as input, usually followed by graph neural networks (GNN) to learn the local tertiary patterns for function prediction, as in DeepFRI [16] and GAT-GO [17]. Network-based methods exploit the rationale that proteins connected in biological networks (e.g. protein–protein interaction (PPI) or metabolic network) are likely to share the same functions [18]. For example, NetGO [19] integrates multiple protein networks in STRING [20] and transfers annotations from nearest neighbors in the aggregated network. DeepGO [21] adopts knowledge graph embedding algorithm to learn protein features from PPI networks. S2F [22] transfers PPI networks from model organisms to newly sequenced ones, in which label diffusion is employed to propagate initial predictions from several sequence-based component predictors. DeepGraphGO [23] makes the most of both protein sequence domain and high-order protein network information via multispecies GNN strategy. Literature-based methods like DeepText2GO [24] attempt to extract explicit descriptions of protein functions or properties from biomedical texts. NetGO 2.0 [25] incorporates literature and latent sequence information into NetGO to further improve performance. Although CAFA challenge has shown that integrative predictors combining multiple information sources usually outperform sequence-based methods, these extra features are often unavailable, incomplete, or difficult to obtain for most proteins thus limiting their scopes. Therefore, methods that accurately predict protein functions from sequences alone may be more general and applicable to most proteins that have not been extensively studied.

Since protein sequences can be regarded as a language in life, unsupervised pretraining with language models from natural language processing has recently been applied to protein sequence representation learning and has displayed promising results in downstream predictions including secondary structures, tertiary contacts, mutational effects, and protein binding sites [26–29]. Our previous work [29] has shown that sequence representations from pretrained language models can outperform manually engineered evolutionary and structural features for binding site detection. Such results inspire us to develop a fast and accurate sequence-based protein function predictor that does not rely on any features constructed from protein domains, structures, biological networks, or literature. Besides, network propagation approaches have been shown successful to predict protein functions in which existing knowledge is amplified by propagating an initial set of functional labels from experimentally characterized proteins through PPI networks [30]. S2F [22] further presents a label diffusion algorithm accounting for the overlapping communities of proteins with related functions. Therefore, it is promising to advance the performance of sequence-based function predictors by employing label diffusion over homology network built solely on sequence similarities.

In this study, we propose SPROF-GO, a Sequence-based alignment-free PROtein Function predictor, which leverages a pretrained protein language model to efficiently extract informative sequence embeddings and employs self-attention pooling to focus on important residues. Label diffusion algorithm is adopted to exploit the homology information and account for the overlapping communities of proteins with related functions. Besides, a hierarchical learning strategy is applied to produce consistent predictions and improve performance. SPROF-GO was shown to surpass state-of-the-art sequence-based and even network-based approaches by more than 14.5, 27.3 and 10.1% in area under the precision-recall curve (AUPR) on the three sub-ontology test sets, respectively. Our method was further demonstrated to generalize well on non-homologous proteins and unseen species. Finally, visualization based on the attention mechanism indicated that SPROF-GO is able to capture sequence domains useful for function prediction. We suggest that our fast and accurate method could scale with the current fast-growing sequence databases, and provide useful information for biologists studying disease mechanism and chemists interested in targeted drug design.

MATERIALS AND METHODS

Datasets

We adopted the benchmark datasets proposed in [23], in which the training and test sets were collected following the standard protocol of CAFA. Specifically, the protein sequences were downloaded from UniProt [3], and the GO term annotations were extracted and combined from Swiss-Prot [31], GOA [32] and GO [6] in January 2020. Only experimental annotations with the following evidence codes were kept: IDA, IPI, EXP, IGI, IMP, IEP, IC or TA. The annotations were further up-propagated based on the 'is-a' relationship in the hierarchical structure of GO, and the root GO terms were omitted. Then, the training, validation and test sets were split according to the annotation time stamps. The training sets contain proteins annotated before January 2018, while the validation and test sets contain no-knowledge proteins annotated from January to December 2018 and from January 2019 to January 2020, respectively. In this study, we discarded sequences longer than 2000 in the training sets and trimmed sequences to 5000 in the validation and test sets due to the memory limit of GPU. Furthermore, to optimize the training efficiency and predicted accuracy, we only focused on the GO terms with enough training samples ($\geq$ 50 sequences) in the training steps, resulting in 790, 4766 and 667 classes for the MF, BP and CC sub-ontology, respectively. In the evaluation phase, we considered all terms to ensure fair comparison with other methods. Table 1 shows the detailed statistics of the training, validation and test sets for the three

Table 1. Numbers of proteins in the training, validation and test sets used in this study for the three domains in GO

	Train			Valid			Test		
	MF	**BP**	**CC**	**MF**	**BP**	**CC**	**MF**	**BP**	**CC**
HUMAN (9606)	9208	12 095	18 842	86	138	137	41	87	767
MOUSE (10090)	6138	9927	8482	103	299	228	65	156	130
All data	51 549	85 104	76 098	490	1570	923	426	925	1224
Data used by SPROF-GO	50 326	82 793	74 161	490	1570	923	426	925	1224

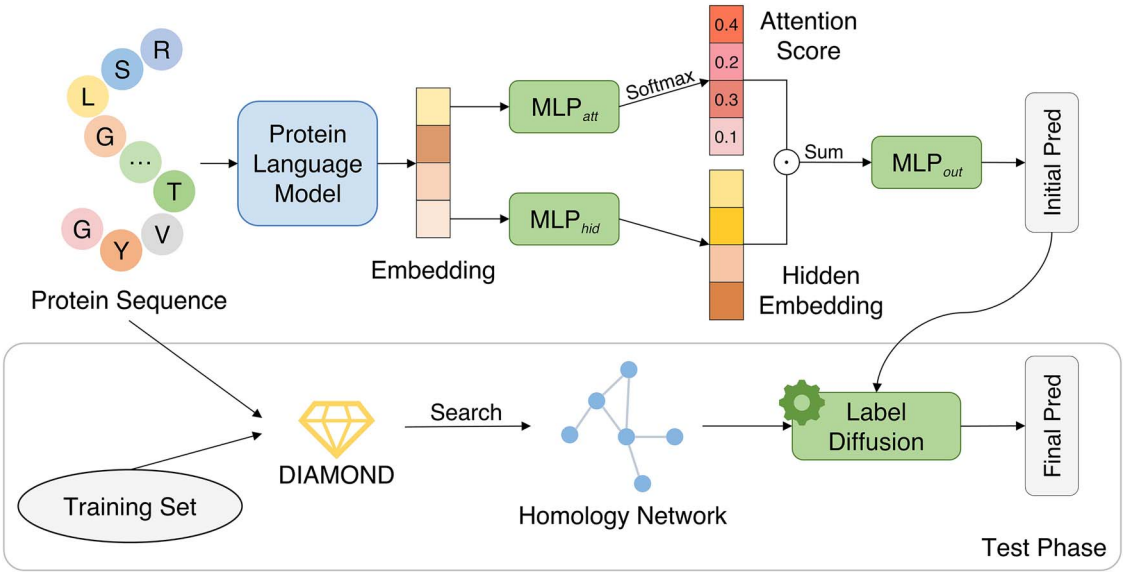

Figure 1. Overview of the SPROF-GO method. First, the protein sequence is input to the pretrained protein language model to extract the initial sequence embedding. Then, the embedding matrix is fed to two MLPs parallelly to learn an attention vector and a hidden embedding matrix. Finally, the hidden embeddings are weighted averaged among different sequence positions based on the attention scores, which is input to the output MLP to predict the GO term probabilities. This initial prediction is used during training to update the model parameters. In the test phase, the input sequence is further searched against the training set using DIAMOND to build a sequence homology network. The initial prediction and the homology network are fed into the label diffusion algorithm, which outputs the final protein function prediction.

domains in GO, as well as the HUMAN and MOUSE subsets used in our downstream analyses.

The architecture of SPROF-GO

The overall architecture of SPROF-GO is shown in Figure 1. First, the protein sequence is input to the pretrained protein language model to extract an initial sequence embedding matrix. Then, the embedding matrix is fed to two multilayer perceptrons (MLPs) parallelly to learn an attention vector and a more informative hidden embedding matrix. Finally, the hidden embeddings are weighted averaged among different sequence positions based on the attention scores to generate an embedding vector, which is input to the output MLP to predict the GO term probabilities. Additionally, a hierarchical learning strategy is applied to ensure the prediction to be consistent. This initial prediction is used during training to update the model parameters. In the test phase, the input sequence is further searched against the training set to build a sequence homology network. The initial prediction and the homology network are fed into the label diffusion algorithm, which outputs the final protein function prediction. Details of these modules are explained in the following sections.

Pretrained protein language model

SPROF-GO leverages the protein language model ProtT5-XL-U50 [27] (denoted as ProtTrans) for efficient feature extraction, thus bypassing the computationally intense sequence alignment to search for sequence domains or produce evolutionary profiles. ProtTrans is a transformer-based auto-encoder named T5 [33] pretrained in a self-supervised manner, essentially learning to predict masked amino acids. Concretely, the ProtTrans model contains 24 layers and 32 heads with 3B parameters, which was first trained on BFD [34] and then fine-tuned on UniRef50 [35]. The BERT's denoising objective [36] was adopted to corrupt and reconstruct single tokens using a masking probability of 15% (details shown in Supplementary Note 1). We extracted the output from the last layer of the encoder part of ProtTrans as the initial sequence representation $H^{(0)} \in \mathbb{R}^{n\times 1024}$, with n denoting the sequence length and 1024 being the feature dimension. Note that the inference cost of ProtTrans is really low, and the feature extraction process for our whole benchmark datasets (~120 000 sequences, ~500 amino acids on average) can be done in about 6 h on an Nvidia GeForce RTX 3090 GPU. The feature values in the sequence representations from ProtTrans were further normalized to scores between 0 and 1 as follows:

$$\upsilon_{\text{norm}} = \frac{\upsilon - \text{Min}}{\text{Max} - \text{Min}} \tag{1}$$

where υ is the original feature value, and Min and Max are the smallest and largest values of this feature type observed in the training set, respectively.

Multilayer perceptron

The sequence embedding matrix output from ProtTrans is fed to two MLPs parallelly to learn an attention vector and a hidden embedding matrix. MLP is a fully connected class of feedforward artificial neural network, which can be generally computed as follows:

$$H^{(l)} = \sigma\left(H^{(l-1)}W^{(l)} + b^{(l)}\right) \tag{2}$$

where $H^{(l-1)} \in \mathbb{R}^{n\times d_{l-1}}$ is the input of the l**th** MLP layer; $W^{(l)} \in \mathbb{R}^{d_{l-1}\times d_l}$ is the weight matrix; $b^{(l)} \in \mathbb{R}^{d_l}$ is the bias term; σ is the non-linear activation function and $H^{(l)} \in \mathbb{R}^{n\times d_l}$ is the output of the l**th** MLP layer. Between two layers of the MLP, we also add layer normalization [37] to stabilize the hidden state dynamics and dropout [38] to avoid overfitting.

Self-attention pooling

Many methods [16, 25] employ global mean pooling or max pooling to convert a residual-level embedding matrix into a protein-level embedding vector for subsequent function prediction, which might either dilute or lose the important features. Here we employ self-attention pooling to automatically focus on important residues as well as provide visualization and interpretability. We set the output dimension of the last layer in MLP_{att} to 1 and the activation function to softmax, so that the output of MLP_{att} is an attention vector $A \in \mathbb{R}^{n\times 1}$. Let $H^{(L)} \in \mathbb{R}^{n\times d}$ denote the hidden embedding output by MLP_{hid}, then the self-attention pooling is calculated as follows:

$$H^{\text{pool}} = A^T H^{(L)} \tag{3}$$

To jointly attend to information from different representation subspaces at different positions, multi-head attention is used in practice to produce h different attention vectors, perform the self-attention pooling in parallel and then concatenate them together. Finally, H^{pool} is input to MLP_{out} with a sigmoid function in the last layer to transform this embedding vector to a K-dimensional function prediction vector H^{out}, where K is the number of GO terms that need to be predicted.

Hierarchical learning

Protein function prediction is a hierarchical multi-label classification problem, in which classes (GO terms) are organized as a DAG, and every prediction must be consistent: the probability of a GO term must be equal to or greater than all of its child terms. Most methods (e.g. [22]) allow inconsistent predictions and require additional post-processing to ensure the consistency at inference time. Here we apply the hierarchical learning strategy proposed by [39] to produce consistent predictions and improve performance, which consists of two elements: (1) a max constraint module (MCM) built upon the network to guarantee consistent predictions inherently; (2) a loss function teaching the network when to exploit the predictions of the lower classes in the hierarchy for making predictions on the upper ones.

Specifically, let H be a $K \times K$ matrix obtained by stacking K rows of the prediction vector H^{out}, and M be a $K \times K$ matrix such that $M_{ij} = 1$ if the j**th** GO term is a subclass of the i**th** GO term, and $M_{ij} = 0$ otherwise. Here, the subclasses of a target term include the child terms in the GO DAG and the target term itself, and only the 'is-a' relationship in GO is considered. Then, the prediction output by MCM is computed as:

$$P = \textbf{MCM}\left(H^{\text{out}}, M\right) = \textbf{max}\left(H \odot M, \textbf{dim} = 1\right) \tag{4}$$

where $\odot$ represents the element-wise product. In the validation and test phases, MCM sets the probability of a GO term to the maximal probability of its subclasses, similar to the post-processing used by other methods. However, if the output of MCM is directly used for training with standard binary cross-entropy loss (BCELoss), as in [40], the network may remain stuck in bad local optima [39]. Thus, max constraint loss (MCLoss) is introduced to control when to exploit the predictive probabilities of the lower classes. Let y be the ground truth function annotation vector, $\overline{y}$ be a $K \times K$ matrix obtained by stacking K rows of y. Then the MCLoss is calculate as:

$$\begin{aligned}&\textbf{MCLoss}\left(H^{\text{out}}, y\right) = \textbf{BCELoss}\\&\left((1-y)\odot P + y \odot \textbf{max}\left(H \odot M \odot \overline{y}, \textbf{dim} = 1\right), y\right)\end{aligned} \tag{5}$$

which means that the probability of a negative class should take the maximal probability of its subclasses, while the probability of a positive class should take the maximal probability of its positive subclasses. Detailed explanations of why MCLoss works are further shown in Supplementary Note 2 using a simple case with only two classes.

Homology-based label diffusion

Proteins rarely perform their functions in isolation. Network propagation methods exploit the fact that groups of proteins connected in functional networks form communities that share similar functions [30]. However, when a protein has more than one function, it may belong to more than one functional group. Such proteins lying at the intersection of communities are generally more functionally similar compared to their neighbors, because they share more functional roles. Therefore, the propagation/diffusion of information between them should be higher. Here, we adopt the label diffusion algorithm proposed by [22] to explicitly model this overlapping community effect, in which we make three modifications: (1) we diffuse annotations over a homology network built solely on sequence similarities, instead of PPI networks in STRING; (2) we incorporate the ground truth annotations from the training set rather than only use test proteins for diffusion; (3) we employ DIAMOND [41] instead of BLAST [12] for similarity search and re-implement the algorithm with sparse matrix operation throughout, to accelerate the computation thus adapting to the large size of the training set.

Specifically, label diffusion is performed only in the test phase to further improve the initial function predictions. We use DIAMOND to search the whole test set against the training set to find training sequences similar to the test sequences, from which a homology network $Q \in \mathbb{R}^{N\times N}$ is built using the sequence identity for each pair of proteins (N is the number of hits in the training set plus the number of test proteins). Then, the weighted Jaccard similarity matrix is defined to measure how much a pair of proteins belong to the same community in network Q:

$$J_{ij} = \frac{\sum_k Q_{ik}Q_{jk}}{\sum_k Q_{ik} + \sum_k Q_{jk} - \sum_k Q_{ik}Q_{jk}} \tag{6}$$

For a target GO term k, we learn the k**th** column of the final annotation matrix F (denoted as F_k) by minimizing the cost function $C(F_k)$:

$$C(F_k) = \sum_{i=1}^{n}(F_{ik} - Y_{ik})^2 + \frac{\lambda}{2}\sum_{i=1}^{n}\frac{1}{d_i}\sum_{j=1}^{n}J_{ij}Q_{ij}\left(F_{ik} - F_{jk}\right)^2 \tag{7}$$

where the first term is to conserve the initial labels/predictions Y_{ik}, the second term accounts for the consistency of the labels/predictions of adjacent nodes in the network, $J_{ij}Q_{ij}$ models the homology information and overlapping community effect, and λ is a regularization parameter. Note that $1/d_i$ is a normalization factor defined as:

$$\frac{1}{d_i} = \frac{1}{\sum_j J_{ij}Q_{ij}} \tag{8}$$

We define Q^* and its Laplacian matrix L as follows:

$$Q^*_{ij} = \frac{1}{2}\left(\frac{1}{d_i} + \frac{1}{d_j}\right)J_{ij}Q_{ij} \tag{9}$$

$$L = D - Q^* \tag{10}$$

where D denotes the diagonal degree matrix of Q^*. Then, the closed-form solution that minimizes $C(F_k)$ can be expressed as:

$$F = (I + \lambda L)^{-1}Y \tag{11}$$

where $I \in \mathbb{R}^{N\times N}$ is an identity matrix, $Y \in \mathbb{R}^{N\times K}$ is the concatenation of the labels of the training set and the initial predictions of the test set, and $F \in \mathbb{R}^{N\times K}$ is the updated annotations for the training samples and the test proteins, from which we retrieve our final predictions for the test sequences. Moreover, as proven in [22], since our initial predictions are consistent with the GO structure, our final predictions output by label diffusion will also be consistent.

Implementation and evaluation

We trained our models to predict GO terms separately for MF, BP and CC ontology. For each training set of the sub-ontology, we trained five models using five different random seeds, and their average performance on the validation set was used to choose the best feature combination and optimize all hyperparameters through grid search (Supplementary Table S1). In the test phase, all five trained models were used to make predictions, which were then averaged as the assembled prediction of SPROF-GO. Specifically, we employed a two-layer fully connected architecture for the three MLPs in SPROF-GO with the following set of hyperparameters: hidden units of 256, attention heads (h) of 8, dropout rate of 0.1 and batch size of 20. The label diffusion regularization parameter λ was simply set to 1. We utilized the Adam optimizer [42] with $\beta_1 = 0.9$, $\beta_2 = 0.99$, weight decay of 10^{-5} and learning rate of 2×10^{-4} for model optimization. We implemented the proposed method with Pytorch 1.13.0 [43]. The training process for one model lasted at most 30 epochs, and we performed early stopping with patience of four epochs based on the validation performance, which took ~40 min for MF and CC ontology, and ~2 h for BP ontology on an Nvidia GeForce RTX 3090 GPU. During the test phase, it took ~2 min to make predictions for all proteins in the three sub-ontology test sets (~2500 sequences).

Similar to the previous studies [14, 23], we used F_{max} and AUPR to evaluate the predictive performance, whose detailed definitions are given in Supplementary Note 3. F_{max} is the maximum protein-centric F-measure computed over all prediction thresholds, which is a major evaluation metrics in CAFA. AUPR is also a suitable measure for highly unbalanced dataset since it emphasizes more on the minority class [44, 45].

RESULTS

The overview of SPROF-GO is shown in Figure 1. For a given protein sequence, the pretrained language model is employed to extract a sequence embedding matrix, which is fed to the self-attention pooling module to generate a protein-level representation for the final output layer. Besides, hierarchical learning is applied to ensure consistent prediction. In the test phase, the prediction is further advanced by exploiting the homology information through label diffusion. The Results section is organized as follows. First, we demonstrated the superiority of the feature from a pretrained language model over other widely used features. Second, we conducted ablation study on several techniques used in SPROF-GO. Third, we compared SPROF-GO with other state-of-the-art methods. Fourth, we evaluated SPROF-GO on non-homologous proteins and unseen species to verify its robustness. Lastly, we interpreted the decision mechanism of SPROF-GO by visualizing the attention scores.

Feature from a pretrained language model is informative for protein function prediction

We evaluated SPROF-GO on the test sets of the three domains in GO (described in Table 1) by F_{max} and AUPR. As shown in Table 2, SPROF-GO achieved F_{max} of 0.647, 0.335 and 0.725, as well as AUPR of 0.622, 0.247 and 0.765 on the MF, BP and CC test sets, respectively. To demonstrate the effectiveness of the language model (ProtTrans) representation employed by SPROF-GO, we conducted feature ablation experiments to compare ProtTrans with other popular features in this field. When adopting the one-hot encoding of amino acid types, the model showed poor performance with AUPR of 0.475, 0.187 and 0.705 on the three sub-ontology test sets, and the performance would further degrade largely when removing the label diffusion module (AUPR of 0.177, 0.130 and 0.582). This suggested that the primary sequences alone are insufficient to characterize protein functions, while sequence homology information is still a valuable source for function inference. We also investigated the widely used [11, 19, 23, 25] InterPro feature generated by InterProScan [13] through sequence alignment, which is a binary protein-level vector indicating the existences of protein domains and families. As shown in Table 2, the model using InterPro obtained AUPR of 0.594, 0.203 and 0.730, surpassing the one using one-hot encoding, which is reasonable since domains often form functional units, such as the calcium-binding EF hand domain of calmodulin. However, the sequence feature by ProtTrans outperformed one-hot, InterPro or the combination of these two features by large margins. Note that the generation of the ProtTrans feature is also much more efficient than that of InterPro since no database searches are needed. Moreover, further integrating one-hot and InterPro features to ProtTrans was redundant and could not attain any further improvements, suggesting that the ProtTrans language model may have potentially captured the protein sequence, domain and family knowledge informative for function prediction.

Model ablation study

To investigate the impacts of the self-attention pooling, hierarchical learning, homology-based label diffusion and model assembly techniques applied by SPROF-GO, we removed one of the four components at a time and then re-trained the model using the same sequence feature. As shown in Table 3, the removal of the assembly strategy caused the largest average AUPR drop (0.017) on the three test sets, while the removals of the attention pooling (using mean pooling instead), hierarchical learning

Table 2. The predictive performance on the test sets of the three domains in GO using different features

Feature	F_{max}			AUPR		
	MF	BP	CC	MF	BP	CC
One-hot	0.555	0.291	0.680	0.475	0.187	0.705
InterPro	0.631	0.300	0.687	0.594	0.203	0.730
One-hot + InterPro	0.634	0.299	0.689	0.589	0.203	0.732
ProtTrans (SPROF-GO)	**0.647**	**0.335**	**0.725**	**0.622**	**0.247**	**0.765**
ProtTrans + one-hot + InterPro	0.645	0.329	0.722	0.620	0.239	0.750

Note: Bold fonts indicate the best results.

Table 3. Ablation study on different techniques used by SPROF-GO on the test sets of the three domains in GO

Method	F_{max}			AUPR		
	MF	BP	CC	MF	BP	CC
SPROF-GO_{base}	0.583	0.307	0.692	0.563	0.209	0.733
SPROF-GO w/o attention	0.628	0.328	0.721	0.613	0.235	0.760
SPROF-GO w/o hierarchical learning	0.638	0.333	0.723	0.612	0.238	0.761
SPROF-GO w/o label diffusion	0.633	0.328	0.721	0.600	0.240	**0.765**
SPROF-GO w/o assembly	0.646	0.327	0.715	0.605	0.235	0.743
SPROF-GO	**0.647**	**0.335**	**0.725**	**0.622**	**0.247**	**0.765**

Note: SPROF-GO_{base} denotes the baseline method that does not use any of the above-mentioned techniques. Bold fonts indicate the best results.

(using post-processing instead) and label diffusion caused average AUPR drops of 0.009, 0.008 and 0.010, respectively. Note that since SPROF-GO is supported by several techniques, removal of a single component seemed to have minor influence on the overall performance. Moreover, some components may have significant benefits on one ontology, but have small impacts on the others. For example, the removal of label diffusion caused the largest AUPR drop of 0.022 on the MF test set, while it had no impact on the AUPR of the CC test set, probably because a pretrained language model is sufficient to capture most discriminative features for the CC ontology (an easier task with more training data and fewer terms to predict compared to the MF ontology). The removal of attention or assembly caused the largest AUPR drops of 0.012 on the BP test set, and the removal of assembly caused the largest AUPR drop of 0.022 on the CC test set. Here, we also report the performance of a baseline method that does not use any of the above-mentioned techniques (SPROF-GO_{base}). SPROF-GO outperformed this baseline significantly with improvements of 0.064, 0.028 and 0.033 on F_{max}, and 0.059, 0.038 and 0.032 on AUPR in the three test sets, further indicating the advantages of the four modules in SPROF-GO. We also repeated this experiment five times using different sets of random seeds for training and got similar results (Supplementary Table S2).

Comparison with state-of-the-art methods

We compared SPROF-GO with four sequence-based (BLAST-KNN, LR-InterPro, DeepGOCNN and DeepGOPlus) and two network-based (Net-KNN and DeepGraphGO) predictors on the test sets of the three domains in GO. The baseline method (SPROF-GO_{base}) that utilizes ProtTrans and MLP with mean pooling is also considered here. The implementation details of these competing methods are introduced in Supplementary Note 4. As reported in Table 4, GO terms in the BP ontology seem harder to predict for all methods, which may be due to the large number of terms and the deep and complex structure of the BP DAG. Howsoever, SPROF-GO outperformed all other sequence-based and even network-based methods on F_{max} and AUPR in all three domains. Undoubtedly, SPROF-GO substantially surpassed the sequence-based method DeepGOPlus by 56.3%, 128.7% and 28.6% on AUPR in the three test sets, respectively. This indicated that representing protein sequence simply by one-hot encoding followed by CNN and mining homology information simply using k-nearest neighbors are not enough to capture the most helpful information for function prediction. Interestingly, though our method is a sequence-based predictor, it outperformed the state-of-the-art network-based method DeepGraphGO by 3.9%, 2.4% and 4.8% on F_{max}, and 14.5%, 27.3% and 10.1% on AUPR. This is expected because the sequence representation from the pretrained language model used by SPROF-GO is more informative and powerful than the handcrafted domain and family features employed by DeepGraphGO (shown in Table 2). Another reason may be that the network information also brought noises since the protein–protein associations in STRING are not always from experiments. In addition, the label diffusion in SPROF-GO could further boost the predictive quality by exploiting the homology information and overlapping community effect. On the other hand, our method is also computationally efficient. Empirically, it takes ~7 min to extract features and make predictions on the three ontologies for 1000 proteins with 500 residues on average using SPROF-GO on an Nvidia GeForce RTX 3090 GPU. However, DeepGraphGO can only predict less than five sequences in the same time since the generation of the InterPro feature requires expensive database searches.

Table 4. Performance comparison of SPROF-GO with state-of-the-art methods on the test sets of the three domains in GO

Method	F_{max}			AUPR		
	MF	**BP**	**CC**	**MF**	**BP**	**CC**
BLAST-KNN	0.590	0.274	0.650	0.455	0.113	0.570
LR-InterPro	0.617	0.278	0.661	0.530	0.144	0.672
Net-KNN	0.426	0.305	0.667	0.276	0.157	0.641
DeepGOCNN	0.434	0.248	0.632	0.306	0.101	0.573
DeepGOPlus	0.593	0.290	0.672	0.398	0.108	0.595
DeepGraphGO	0.623	0.327	0.692	0.543	0.194	0.695
SPROF-GO_{base}	0.583	0.307	0.692	0.563	0.209	0.733
SPROF-GO	**0.647**	**0.335**	**0.725**	**0.622**	**0.247**	**0.765**

Note: SPROF-GO_{base} denotes the baseline method that employs ProtTrans and MLP with mean pooling. Bold and underlined fonts indicate the best and second-best results, respectively.

Table 5. Performance comparison of SPROF-GO with state-of-the-art methods on *difficult* proteins within the test sets of the three domains in GO

Method	F_{max}			AUPR		
	MF	**BP**	**CC**	**MF**	**BP**	**CC**
BLAST-KNN	0.534	0.274	0.521	0.377	0.114	0.354
LR-InterPro	0.589	0.275	0.613	0.493	0.148	0.591
Net-KNN	0.404	0.292	0.595	0.230	0.142	0.560
DeepGOCNN	0.406	0.243	0.578	0.246	0.091	0.478
DeepGOPlus	0.564	0.292	0.602	0.326	0.108	0.454
DeepGraphGO	0.598	0.322	0.625	0.508	0.184	0.607
SPROF-GO	**0.630**	**0.339**	**0.682**	**0.617**	**0.256**	**0.708**

Note: Bold and underlined fonts indicate the best and second-best results, respectively.

To further validate the adaptability of SPROF-GO to other GO datasets and relationships, we adopted another benchmark dataset used by NetGO 2.0 [25] and DeepGOZero [46] to evaluate our method. NetGO 2.0 is a hybrid predictor which incorporates sequence, literature, domain, family and network information, while DeepGOZero is a sequence-based method exploiting formal axioms in GO to make zero-shot predictions. The generation of this NetGO 2.0 dataset is similar to the one used in our study, except that it propagated the annotations using all types of relationships instead of using 'is-a' only (details shown in Supplementary Table S3). We retrained SPROF-GO with simply the same hyperparameters, and the performance comparison of SPROF-GO with other state-of-the-art methods is shown in Supplementary Table S4. Specifically, SPROF-GO obtained F_{max} values of 0.739, 0.453 and 0.729 on the MF, BP and CC test sets, surpassing DeepGOZero (0.662, 0.396, 0.662), DeepGraphGO (0.671, 0.418, 0.679), NetGO 2.0 (0.698, 0.431, 0.662) and other methods consistently, which further demonstrated the adaptability and robustness of our proposed framework.

Generalization on non-homologous proteins and unseen species

To examine the generalization ability of our method for non-homologous proteins, we compared SPROF-GO with other competing methods on *difficult* proteins within the test sets, which are defined by CAFA2 [8] as the test proteins with sequence identity <60% to the training set. The numbers of *difficult* proteins in the MF, BP and CC test sets are 303, 649 and 437, respectively. As shown in Table 5, almost all methods showed performance drops in different degrees compared to the results on the original test sets (Table 4). However, SPROF-GO still outperformed all other methods on F_{max} and AUPR in all three domains. Specifically, SPROF-GO maintained similar performance in the MF/BP ontology, with AUPR of 0.622/0.247 on the full test set and 0.617/0.256 on the subset of *difficult* proteins. By comparison, the AUPR of DeepGraphGO in MF and BP decreased from 0.543 to 0.508 and 0.194 to 0.184, respectively. As for the CC ontology, AUPR decreased by 12.7% for DeepGraphGO but only 7.5% for SPROF-GO on *difficult* proteins. Interestingly, we found that the label diffusion module could still bring improvements in this scenario by mining annotations from dissimilar sequences, and its removal caused AUPR drops of 0.020, 0.011 and 0.003 on the MF, BP and CC test sets, respectively. These results suggested that our method can generalize well on non-homologous proteins, rather than just remember the functions of similar sequences in the training set, thus making it a robust and reliable method for function prediction of novel sequences.

We also investigated the performance of SPROF-GO and other methods over proteins of HUMAN and MOUSE within the test sets (details shown in Table 1), and SPROF-GO again outperformed all competing methods in all twelve settings except one (Supplementary Table S5). More importantly, to explore the generalization ability of our method for unseen species, we further evaluated SPROF-GO and the second-best method DeepGraphGO on the HUMAN and MOUSE proteins when trained with proteins of all species except the target species (denoted by w/o species). As shown in Figure 2, the AUPR of both methods decreased in most cases when excluding the training data of the target species. However, SPROF-$GO_{w/o\ species}$ still surpassed DeepGraphGO$_{w/o\ species}$ in all three domains for the HUMAN and MOUSE proteins. Moreover, even without training data from the target species, SPROF-$GO_{w/o\ species}$ outperformed DeepGraphGO using the whole training set in all six settings except one. For example, SPROF-$GO_{w/o\ HUMAN}$ achieved AUPR of 0.735 for CC ontology on the HUMAN proteins, exceeding the one by DeepGraphGO using the full training set (0.642). Detailed evaluation results are shown in Supplementary Tables S6 and S7, where the performance using only the target

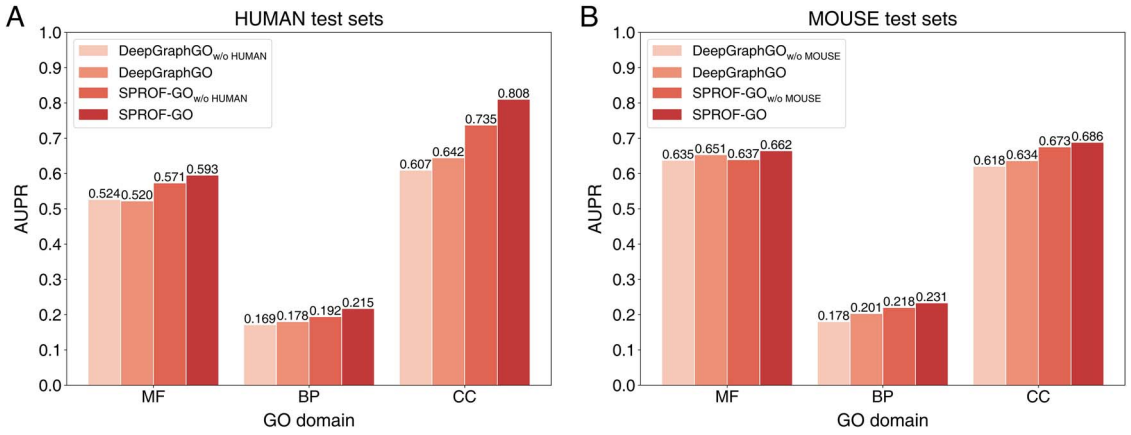

Figure 2. Performance comparison of SPROF-GO and DeepGraphGO on the HUMAN (**A**) and MOUSE (**B**) proteins within the test sets when trained with proteins of all species or all species except the target species (denoted by w/o HUMAN and w/o MOUSE).

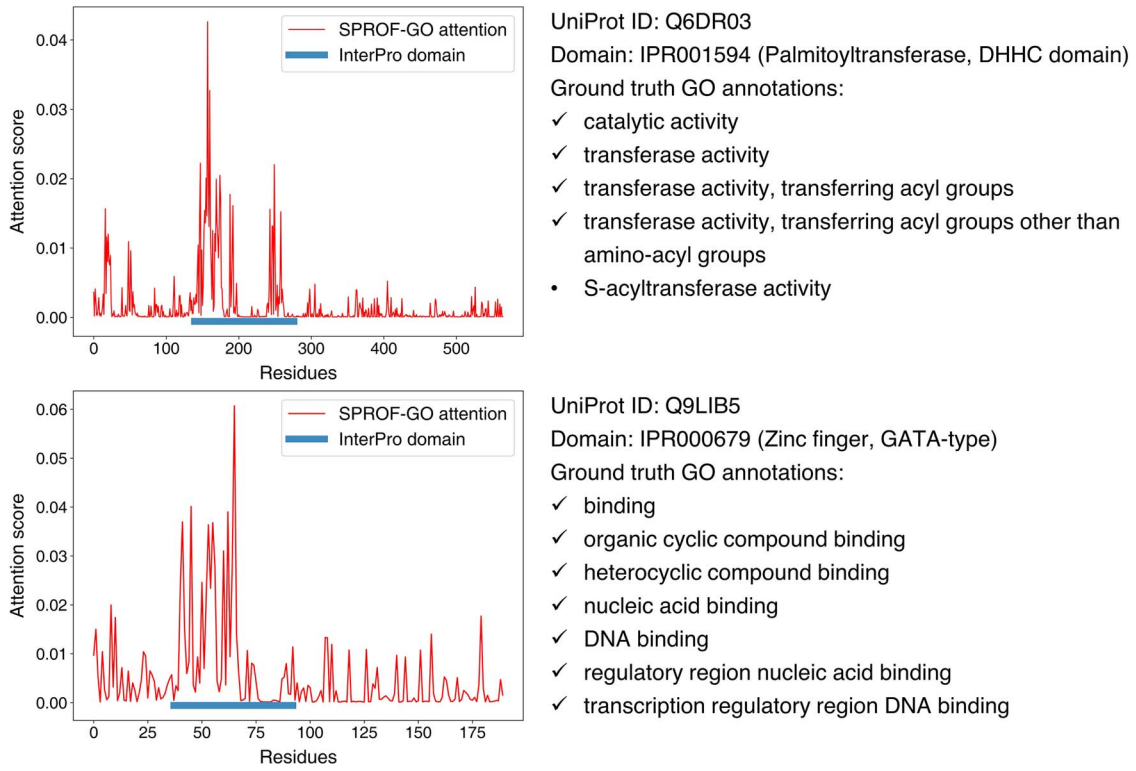

Figure 3. Visualization of two examples (Q6DR03 and Q9LIB5) in the MF ontology test set. The left panels show the attention scores by SPROF-GO at different sequence positions (upper lines), as well as the locations of the domains found by InterProScan (lower bars). The right panels show the information of the test proteins, including UniProt ID, InterPro domain ID, domain name and ground truth function annotation. The GO terms correctly identified by SPROF-GO are marked with ticks, and the root GO term (GO:0003674 molecular function) is omitted.

species for training is also included. These results suggested that our method is robust and can also generalize on proteins of unseen species in the training set, thus having the potential to predict functions for newly sequenced organisms.

Model interpretation by attention visualization

What did SPROF-GO learn? Did the network reason solely by comparing the test proteins with the training samples or did it learn the underlying chemical principles of protein functioning? To better illustrate the decision mechanism of SPROF-GO, we selected two examples (UniProt ID: Q6DR03 and Q9LIB5) in the MF ontology test set and extracted their residue-level attention scores from the self-attention pooling module in SPROF-GO. We averaged the scores from different attention heads and different assembled models as the final attention scores. Besides, we applied InterProScan to search for functional domains in these two sequences. As shown in Figure 3, Q6DR03 contains a DHHC domain of palmitoyltransferases [47] in sequence positions of 140 to 277, where the attention scores are also higher. This protein was annotated hierarchically with five GO terms down to 'S-acyltransferase activity', in which SPROF-GO correctly predicted four terms but missed one specific term, leading to the F-measure of 0.889. Another case Q9LIB5 contains a GATA-type zinc finger domain [48] in sequence positions of 38 to 93, where the attention scores by SPROF-GO are also higher. The presence of the zinc finger domain associates this protein with several functions

involving DNA-binding, in which SPROF-GO correctly predicted all seven terms, leading to an F-measure of 1.000. The attention visualization for these cases suggested that our method could successfully identify and pay more attention to the functional domains in sequences, and thus correctly predicted the correlated functions of the proteins.

DISCUSSION

Protein function prediction benefits disease mechanism elucidation and drug target discovery. Existing sequence-based methods mostly suffer from low predictive accuracy or high computational cost. Although many methods have integrated protein structures, biological networks or literature information to improve performance, these extra features are often unavailable. Here, we propose a Sequence-based PROtein Function predictor SPROF-GO, which has the following five notable features: (1) SPROF-GO leverages a pretrained language model to efficiently extract informative sequence embeddings, thus bypassing expensive database searches; (2) The self-attention pooling is employed to capture sequence domains useful for function prediction and provide interpretability; (3) SPROF-GO applies a hierarchical learning strategy to produce consistent predictions and improve performance; (4) The label diffusion algorithm is adopted to exploit the homology information and overlapping community effect; (5) SPROF-GO is accurate and robust, with better performance than state-of-the-art sequence-based and even network-based approaches, and great generalization ability on non-homologous proteins and unseen species.

However, there is still room for further improvements on SPROF-GO. First, applying GNN [49, 50] on predicted protein structures from sequences by AlphaFold2 [51] or ESMFold [52] may yield better performance [57]. Second, inspired by ATGO [53], contrastive learning [54] could be applied on the PPI networks only in the training phase to maximize the function similarities between network neighbors, thus reflecting the guilt-by-association principle. Third, drug and disease information could be further incorporated using knowledge graph techniques [55], and organism taxa could also be considered as in DeeProtGO [56]. Fourth, SPROF-GO currently only considers around 6200 GO terms that have ≥50 training samples in order to optimize the training efficiency and predicted accuracy (Supplementary Note 5). How to effectively learn the terms with scarce training data remains a challenging and interesting task to solve in our future. The standalone version of SPROF-GO is available at https://github.com/biomed-AI/SPROF-GO, which is easy to set up and run, and the web server is freely available at http://bio-web1.nscc-gz.cn/app/sprof-go. We suggest that our fast and accurate method could scale with the current fast-growing sequence databases, and provide useful information for biologists studying disease mechanism and chemists interested in targeted drug design.

Key Points

- SPROF-GO is a sequence-based protein function predictor which leverages a pretrained language model to efficiently extract informative sequence embeddings, thus bypassing expensive database searches.
- SPROF-GO employs self-attention pooling to capture sequence domains useful for function prediction and provide interpretability.
- SPROF-GO applies hierarchical learning strategy to produce consistent predictions and label diffusion to exploit the homology information.
- SPROF-GO is accurate and robust, with better performance than state-of-the-art sequence-based and even network-based approaches, and great generalization ability on non-homologous proteins and unseen species.

SUPPLEMENTARY DATA

Supplementary data are available online at https://academic.oup.com/bib.

FUNDING

National Key R&D Program of China (2022YFF1203100); National Natural Science Foundation of China (12126610); Guangdong Key Field R&D Plan (2019B020228001 and 2018B010109006); Guangzhou S&T Research Plan (202007030010 and 202002020047).

DATA AVAILABILITY

The datasets, source codes and trained models of SPROF-GO are available at https://github.com/biomed-AI/SPROF-GO. The SPROF-GO web server is freely available at http://bio-web1.nscc-gz.cn/app/sprof-go.

REFERENCES

1. Eisenberg D, Marcotte EM, Xenarios I, *et al.* Protein function in the post-genomic era. *Nature* 2000;**405**:823–6.
2. Costanzo M, VanderSluis B, Koch EN, *et al.* A global genetic interaction network maps a wiring diagram of cellular function. *Science* 2016;**353**:aaf1420.
3. The UniProt Consortium. UniProt: the universal protein knowledgebase in 2021. *Nucleic Acids Res* 2021;**49**:D480–9.
4. Cruz LM, Trefflich S, Weiss VA, *et al.* Protein function prediction, functional. *Genomics* 2017;55–75.
5. Radivojac P, Clark WT, Oron TR, *et al.* A large-scale evaluation of computational protein function prediction. *Nat Methods* 2013;**10**:221–7.
6. Ashburner M, Ball CA, Blake JA, *et al.* Gene ontology: tool for the unification of biology. *Nat Genet* 2000;**25**:25–9.
7. Obozinski G, Lanckriet G, Grant C, *et al.* Consistent probabilistic outputs for protein function prediction. *Genome Biol* 2008;**9**:1–19.
8. Jiang Y, Oron TR, Clark WT, *et al.* An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biol* 2016;**17**:1–19.
9. Zhou N, Jiang Y, Bergquist TR, *et al.* The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome Biol* 2019;**20**:1–23.
10. Conesa A, Götz S, García-Gómez JM, *et al.* Blast2GO: a universal tool for annotation, visualization and analysis in functional genomics research. *Bioinformatics* 2005;**21**:3674–6.
11. You R, Zhang Z, Xiong Y, *et al.* GOLabeler: improving sequence-based large-scale protein function prediction by learning to rank. *Bioinformatics* 2018;**34**:2465–73.
12. Altschul SF, Madden TL, Schäffer AA, *et al.* Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* 1997;**25**:3389–402.

13. Jones P, Binns D, Chang H-Y, *et al.* InterProScan 5: genome-scale protein function classification. *Bioinformatics* 2014;**30**:1236–40.
14. Kulmanov M, Hoehndorf R. DeepGOPlus: improved protein function prediction from sequence. *Bioinformatics* 2020;**36**:422–9.
15. Cao Y, Shen Y. TALE: transformer-based protein function annotation with joint sequence–label embedding. *Bioinformatics* 2021;**37**:2825–33.
16. Gligorijević V, Renfrew PD, Kosciolek T, *et al.* Structure-based protein function prediction using graph convolutional networks. *Nat Commun* 2021;**12**:1–14.
17. Lai B, Xu J. Accurate protein function prediction via graph attention networks with predicted structure information. *Brief Bioinform* 2022;**23**:bbab502.
18. Oliver S. Guilt-by-association goes global. *Nature* 2000;**403**:601–2.
19. You R, Yao S, Xiong Y, *et al.* NetGO: improving large-scale protein function prediction with massive network information. *Nucleic Acids Res* 2019;**47**:W379–87.
20. Szklarczyk D, Gable AL, Nastou KC, *et al.* The STRING database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets. *Nucleic Acids Res* 2021;**49**:D605–12.
21. Kulmanov M, Khan MA, Hoehndorf R. DeepGO: predicting protein functions from sequence and interactions using a deep ontology-aware classifier. *Bioinformatics* 2018;**34**:660–8.
22. Torres M, Yang H, Romero AE, *et al.* Protein function prediction for newly sequenced organisms. *Nat Mach Intell* 2021;**3**:1050–60.
23. You R, Yao S, Mamitsuka H, *et al.* DeepGraphGO: graph neural network for large-scale, multispecies protein function prediction. *Bioinformatics* 2021;**37**:i262–71.
24. You R, Huang X, Zhu S. DeepText2GO: improving large-scale protein function prediction with deep semantic text representation. *Methods* 2018;**145**:82–90.
25. Yao S, You R, Wang S, *et al.* NetGO 2.0: improving large-scale protein function prediction with massive sequence, text, domain, family and network information. *Nucleic Acids Res* 2021;**49**:W469–75.
26. Rives A, Meier J, Sercu T, *et al.* Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc Natl Acad Sci* 2021;**118**:e2016239118.
27. Elnaggar A, Heinzinger M, Dallago C, *et al.* ProtTrans: towards cracking the language of lifes code through self-supervised deep learning and high performance computing. *IEEE Trans Pattern Anal Mach Intell* 2021;**44**:7112–27.
28. Unsal S, Atas H, Albayrak M, *et al.* Learning functional properties of proteins with language models. *Nat Mach Intell* 2022;**4**:227–45.
29. Yuan Q, Chen S, Wang Y, *et al.* Alignment-free metal ion-binding site prediction from protein sequence through pre-trained language model and multi-task learning. *Brief Bioinform* 2022;**23**:bbac444.
30. Cowen L, Ideker T, Raphael BJ, *et al.* Network propagation: a universal amplifier of genetic associations. *Nat Rev Genet* 2017;**18**:551–62.
31. Boutet E, Lieberherr D, Tognolli M, *et al.* UniProtKB/Swiss-Prot, the manually annotated section of the UniProt KnowledgeBase: how to use the entry view. In: *Plant Bioinformatics*. Humana Press, New York, 2016, 23–54.
32. Huntley RP, Sawford T, Mutowo-Meullenet P, *et al.* The GOA database: gene ontology annotation updates for 2015. *Nucleic Acids Res* 2015;**43**:D1057–63.
33. Raffel C, Shazeer N, Roberts A, *et al.* Exploring the limits of transfer learning with a unified text-to-text transformer. *J Mach Learn Res* 2020;**21**:1–67.
34. Steinegger M, Mirdita M, Söding J. Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold. *Nat Methods* 2019;**16**:603–6.
35. Suzek BE, Huang H, McGarvey P, *et al.* UniRef: comprehensive and non-redundant UniProt reference clusters. *Bioinformatics* 2007;**23**:1282–8.
36. Kenton JDM-WC, Toutanova LK. BERT: pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of NAACL-HLT*. Association for Computational Linguistics, USA, 2019, 4171–86.
37. Ba JL, Kiros JR, Hinton GE. Layer normalization. *arXiv preprint* 2016;arXiv:1607.06450.
38. Srivastava N, Hinton G, Krizhevsky A, *et al.* Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res* 2014;**15**:1929–58.
39. Giunchiglia E, Lukasiewicz T. Coherent hierarchical multi-label classification networks. In: *Advances in Neural Information Processing Systems*. Curran Associates, Inc., New York, 2020, 9662–73.
40. Kulmanov M, Hoehndorf R. DeepPheno: predicting single gene loss-of-function phenotypes using an ontology-aware hierarchical classifier. *PLoS Comput Biol* 2020;**16**:e1008453.
41. Buchfink B, Xie C, Huson DH. Fast and sensitive protein alignment using DIAMOND. *Nat Methods* 2015;**12**:59–60.
42. Kingma DP, Ba J. Adam: a method for stochastic optimization. In: *3rd International Conference on Learning Representations (Poster)*. International Conference on Learning Representations, San Diego, 2015.
43. Paszke A, Gross S, Massa F, *et al.* Pytorch: an imperative style, high-performance deep learning library. *Adv Neural Inf Process Syst* 2019;**32**:8026–37.
44. Davis J, Goadrich M. The relationship between precision-recall and ROC curves. In: *Proceedings of the 23rd International Conference on Machine learning*. Association for Computing Machinery, New York, 2006, 233–40.
45. Saito T, Rehmsmeier M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PloS One* 2015;**10**:e0118432.
46. Kulmanov M, Hoehndorf R. DeepGOZero: improving protein function prediction from sequence and zero-shot learning based on ontology axioms. *Bioinformatics* 2022;**38**:i238–45.
47. Fukata M, Fukata Y, Adesnik H, *et al.* Identification of PSD-95 palmitoylating enzymes. *Neuron* 2004;**44**:987–96.
48. Yamamoto M, Ko LJ, Leonard MW, *et al.* Activity and tissue-specific expression of the transcription factor NF-E1 multigene family. *Genes Dev* 1990;**4**:1650–62.
49. Yuan Q, Chen J, Zhao H, *et al.* Structure-aware protein–protein interaction site prediction using deep graph convolutional network. *Bioinformatics* 2021;**38**:125–32.
50. Yuan Q, Chen S, Rao J, *et al.* AlphaFold2-aware protein–DNA binding site prediction using graph transformer. *Brief Bioinform* 2022;**23**:bbab564.
51. Varadi M, Anyango S, Deshpande M, *et al.* AlphaFold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Res* 2022;**50**:D439–44.
52. Lin Z, Akin H, Rao R, *et al.* Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* 2023;**379**:1123–30.
53. Zhu Y-H, Zhang C, Yu D-J, *et al.* Integrating unsupervised language model with triplet neural networks for protein gene ontology prediction. *PLoS Comput Biol* 2022;**18**:e1010793.

54. Chen T, Kornblith S, Norouzi M, *et al*. A simple framework for contrastive learning of visual representations. In: *International Conference on Machine Learning*. Curran Associates, Inc., New York, 2020, 1597–607.
55. Zheng S, Rao J, Song Y, *et al*. PharmKG: a dedicated knowledge graph benchmark for bomedical data mining. *Brief Bioinform* 2021;**22**:bbaa344.
56. Merino GA, Saidi R, Milone DH, *et al*. Hierarchical deep learning for predicting GO annotations by integrating protein knowledge. *Bioinformatics* 2022;**38**:4488–96.
57. Yuan Q, Yang Y. Sequence-based predictions of residues that bind proteins and peptides. In: *Machine Learning in Bioinformatics of Protein Sequences*. World Scientific, Singapore, 2023, 237–63.