

# GORetrieve: reranking protein-description-based GO candidates by literature-driven deep information retrieval for protein function annotation

Huiying Yan<sup>1</sup>, Shaojun Wang<sup>1</sup>, Hancheng Liu<sup>1</sup>, Hiroshi Mamitsuka<sup>2,3</sup>, Shanfeng Zhu<sup>1,4,5,6</sup>

<sup>1</sup>Institute of Science and Technology for Brain-Inspired Intelligence and MOE Frontiers Center for Brain Science, Fudan University, Shanghai 200433, China

<sup>2</sup>Bioinformatics Center, Institute for Chemical Research, Kyoto University, Uji, Kyoto Prefecture 611-0011, Japan

<sup>3</sup>Department of Computer Science, Aalto University, Espoo 00076, Finland

<sup>4</sup>Key Laboratory of Computational Neuroscience and Brain-Inspired Intelligence (Fudan University), Ministry of Education, Shanghai, 200433, China

<sup>5</sup>Shanghai Key Lab of Intelligent Information Processing and Shanghai Institute of Artificial Intelligence Algorithm, Fudan University, Shanghai, 200433, China

<sup>6</sup>Zhangjiang Fudan International Innovation Center, Shanghai, 200433, China

Corresponding author. Institute of Science and Technology for Brain-Inspired Intelligence and MOE Frontiers Center for Brain Science, Fudan University, Shanghai 200433, China. E-mail: zhuf@fudan.edu.cn (S.Z.)

## Abstract

**Summary:** The vast majority of proteins still lack experimentally validated functional annotations, which highlights the importance of developing high-performance automated protein function prediction/annotation (AFP) methods. While existing approaches focus on protein sequences, networks, and structural data, textual information related to proteins has been overlooked. However, roughly 82% of SwissProt proteins already possess literature information that experts have annotated. To efficiently and effectively use literature information, we present GORetrieve, a two-stage deep information retrieval-based method for AFP. Given a target protein, in the first stage, candidate Gene Ontology (GO) terms are retrieved by using annotated proteins with similar descriptions. In the second stage, the GO terms are reranked based on semantic matching between the GO definitions and textual information (literature and protein description) of the target protein. Extensive experiments over benchmark datasets demonstrate the remarkable effectiveness of GORetrieve in enhancing the AFP performance. Note that GORetrieve is the key component of GOCurator, which has achieved first place in the latest critical assessment of protein function annotation (CAFA5: over 1600 teams participated), held in 2023–2024.

**Availability and implementation:** GORetrieve is publicly available at <https://github.com/ZhuLab-Fudan/GORetrieve>.

## 1 Introduction

Understanding the functions of proteins plays a crucial role in biomedical research and is essential for comprehending life processes, disease mechanisms, and drug development. To systematically describe the functions of proteins, Gene Ontology (GO) has been developed. GO has three branches, i.e. Molecular Function Ontology (MFO), Biological Process Ontology (BPO), and Cellular Component Ontology (CCO), with over 50 000 terms (Ashburner *et al.* 2000). However, due to the high cost of biochemical experiments, only <0.1% of more than 250 million proteins collected in UniProt (The UniProt Consortium 2023), the current largest protein database, have experimental functional annotations. Therefore, it is imperative to develop high-performance computational methods for automated function prediction/annotation (AFP).

AFP is typically achieved by associating the proper GO terms with a target protein. This process can be regarded as a multi-label classification task. Many methods have been proposed to tackle AFP by using different types of data sources (Makrodimitris *et al.* 2020). Most methods concentrate on sequence information, including sequence alignment, domains, family, motifs (Marchler-Bauer *et al.* 2015, Sillitoe *et al.* 2015), and sequence-based features obtained by deep

learning (Kulmanov and Hoehndorf 2020, 2022, Cao and Shen 2021). Protein language models extract deep semantic information from protein sequences by pre-training (Rao *et al.* 2019, Rives *et al.* 2021), which have achieved competitive performance on AFP (Zhu *et al.* 2022, Wang *et al.* 2023). In addition, protein–protein interactions (PPIs) (You *et al.* 2021) and protein 3D structures (Gligorijevic *et al.* 2021, Lai and Xu 2021, Boadu *et al.* 2023) are also broadly used for AFP. Moreover, as demonstrated in a recent critical assessment of protein function annotation (CAFA) (Radivojac *et al.* 2013, Jiang *et al.* 2016, Zhou *et al.* 2019), the state-of-the-art AFP methods, such as GOLabeler and NetGO (You *et al.* 2018b, 2019, Yao *et al.* 2021, Wang *et al.* 2023), were achieved by ensemble methods that integrate different types of data sources.

To further advance AFP, we focus on textual information related to proteins, such as description in UniProt and expert-curated biomedical literature in SwissProt, since this information has not been considered extensively despite its usefulness for AFP. Protein functions are typically annotated by human curators. Despite the existence of relevant literature, a significant portion of proteins still awaits expert annotation using the literature review. Specifically, over 80% of proteins

(around 470 000) in SwissProt have expert annotated literature, whereas merely approximately 16% (around 73 900) have experimental functional annotations (The UniProt Consortium 2023). Therefore, developing an efficient AFP method is important to mitigate this gap and save the time and cost of human curators.

Currently, very few methods use textual information for AFP. A notable example is DeepText2GO, which uses expert annotated biomedical literature to improve the prediction performance (You et al. 2018a). DeepText2GO generates text representations by concatenating Document to Vector (D2V) (Le and Mikolov 2014) and Term Frequency-Inverse Document Frequency (TFIDF), and then trains a classifier for each GO term (You et al. 2018a). A significant drawback of DeepText2GO is that the text representations based on D2V and TFIDF cannot well capture contextual information (Minaee et al. 2021).

In addition, ProTranslator (Xu and Wang 2022) and ProtST (Xu et al. 2023) showed the effectiveness of protein textual descriptions, where many of them were generated by ProtNLM (Gane et al. 2022) with high efficiency. Specifically, ProtST proposes a multi-modal framework to enhance the protein language model by protein descriptions for multiple downstream tasks. On the other hand, ProTranslator embeds proteins based on the protein sequence, description, and network, which puts the focus on annotating proteins with novel GO terms (zero-shot protein function prediction).

Both ProtST and ProTranslator use PubMedBERT (Gu et al. 2021) to encode protein descriptions but ignore the abundant information in the annotated literature, which limits their performance for AFP. In light of the above, there are three challenging issues in using literature data for better AFP: (i) how to construct an effective protein representation using annotated literature, especially when the number of annotated literature for a target protein is large; (ii) how to conduct semantic matching between proteins (descriptions, literature) and GO terms (definitions); and (iii) how to make the whole procedure feasible for the huge number of all possible protein-GO pairs.

To address the above challenges, we propose GORetriever (Fig. 1), a two-stage framework using a deep information retrieval (IR) model with three components for high-performance AFP. To the best of our knowledge, this is the first deep-learning-based IR framework for AFP. In

GORetriever, the target protein and GO terms are regarded as a query and documents, respectively: AFP is a problem of finding relevant GO terms (documents), given a target protein (query). In the first stage, an initial small set of candidate GO terms is retrieved by the **Retrieval** component from annotated proteins in the training data, which have similar descriptions to those of a target protein. At the same time, to construct effective protein representation and reduce noise, the **Sentence Extraction** component extracts the most informative sentences from the annotated literature for each protein. In the second stage, the **Rerank** component re-scores the result of the first stage, which prioritizes the most relevant GO terms by deep semantic matching of proteins and GO terms using their textual information. We note that the candidate GO terms in the first stage can be generated from another AFP method using different data sources, such as BLAST-KNN (Altschul et al. 1997) (sequence alignment), and LR-ESM (protein language model), which are two major components of NetGO3.0 (Wang et al. 2023). We validated the effectiveness of GORetriever on a large-scale dataset and observed a significant improvement compared to the state-of-the-art methods using a single source. Finally, note that incorporating GORetriever into the “learning to rank” framework of NetGO led to a significant achievement already: first place out of over 1600 teams in CAFA5 (Friedberg et al. 2023).

## 2 Materials and methods

### 2.1 Overview

We formulate AFP as a problem of ranking documents relevant (semantically matching) to a query, where a query denotes a target protein and documents correspond to GO terms. Specifically, we represent the textual feature of a target protein as pair  $x \in \mathbb{R}^d \times \mathbb{R}^d$ , where  $p$  encapsulates the descriptions in UniProt, including recommended names  $p_{\text{name}}$ , species  $p_{\text{species}}$ , entry name  $p_{\text{entry}}$ , and so on,  $L$  denotes annotated literature in SwissProt. To characterize GO terms, we leverage GO definitions as candidate documents. The problem can be formally defined as follows: Given protein feature  $x \in \mathbb{R}^d \times \mathbb{R}^d$  and the whole set of GO terms  $\mathbf{G} = \{g_1, g_2, \dots, g_n\}$  with their respective definitions  $\mathbf{D} = \{d_1, d_2, \dots, d_n\}$ , retrieve the most relevant subset  $\mathbf{G}^0$  to  $x$ , out of  $\mathbf{G}$ .

Figure 2 shows the three major components of GORetriever. First, due to the fact that many sentences in the

**Figure 1.** The workflow of GORetriever, with three components: Retrieval, Sentence Extraction, and Rerank. Retrieval and Sentence Extraction can be processed in parallel to generate candidate GO terms and informative sentences, respectively, for Rerank.

**Figure 2.** Three components of GORetriever. (a) Sentence Extraction from annotated literature. Sentences are colored to represent different literature resources, where the number of sentences for each color can be changed. (b) Retrieve candidate GO terms from proteins with similar descriptions. Similar proteins  $X_r$  are first obtained and then annotated GO terms of these proteins are added to candidate set  $G_r$ . (c) Rerank candidate GO terms, according to Cross-Encoder (CE). Each input of CE is a concatenation of “protein name,” “informative sentences” (from a) and “GO definitions” (from b). Candidate GO terms are reordered by the output of CE.

annotated literature are not related to protein functions, we introduce the **Sentence Extraction** component to extract informative sentences from the literature, minimizing extraneous noise and reducing cost. Secondly, we obtain a small set of candidate GO terms based on the annotated proteins with similar descriptions through the **Retrieval** component. Finally, the **Rerank** component prioritizes the most relevant GO terms using the textual interaction of extracted protein annotations and GO terms with a deep learning model.

## 2.2 Sentence extraction: extract the most informative sentences from annotated literature

Before implementing our retrieval model, for each target protein, we collect annotated literature data  $L$  from SwissProt, which gives the PubMed identifiers under protein entries (Boutet *et al.* 2016), and focus on the “Title” and “Abstract” fields only. As many sentences within the annotated literature are unnecessary for protein function annotation, the **Sentence Extraction** component generates effective protein representations as well as reduces noise and computational cost.

As shown in Fig. 2a, extracting all sentences from the annotated literature and mixing them, the goal of the Sentence Extraction component is to select the most informative sentences  $L_i$  for each GO branch  $b_i$ . To achieve this, we score the sentences with a pre-trained Seq2seq ranking model without any additional training (Nogueira *et al.* 2020). The input is defined as:

Query : q Sentence : s Relevance :  $\tilde{r}$

where q is “What is the [branch description] of protein [protein]?” and “[branch description]” can be replaced by “Molecular Function” (MF), “Biological Process” (BP) or “Cellular Component” (CC), and for “[protein],” we can

choose only the recommended protein names in SwissProt to avoid noise. Then we compute the relevance score for each sentence  $s$  in the annotated literature  $L$  by computing the log-likelihood probability of generating “true” in the next [token]. Finally, we reorder all sentences with respect to their corresponding probabilities and obtain a part of top-scoring sentences as the extracted context  $L_i$  (see Section 3.3 for more detail).

## 2.3 Retrieval: retrieve candidate GO terms from proteins in the training data with similar descriptions

Based on the assumption that similar proteins generally have similar GO functions, we develop our Retrieval component. As shown in Fig. 2b, given the features of target protein  $x$ , we retrieve a subset of candidate GO terms  $G_r$  from a subset  $X_r$  (with top  $k$  most similar descriptions) of annotated proteins in the training data, where both  $p_{\text{name}}$  and  $p_{\text{species}}$  are used to measure the textual similarity. That is, for each protein  $x_i$  in  $X_r$ , we add the corresponding GO term set  $G_i$  to the current set  $G_r$ . Iterating this step, we obtain the set of candidates  $G_r = \bigcup_{i=1}^k G_i$  and its definition  $D_r = \bigcup_{i=1}^k D_i$ . In order to reduce the time and space cost, we use a classical IR approach of BM25 (Lin *et al.* 2021) for measuring textual similarity.

## 2.4 Rerank: rerank retrieved GO terms using similarity between textual features and GO definitions

As shown in Fig. 2c, to calculate the relevance score between a target protein and retrieved GO terms, we consider a Cross-Encoder architecture, which is a BERT-based framework and computes interactions among tokens with self-attention mechanism (Reimers and Gurevych 2019). For each branch  $b_i$ , given the features of a target protein  $x_i$  and one of

the GO definitions  $d_i$  from its corresponding retrieved set  $D_r$ , the input of Cross-Encoder is defined as:

$$\text{Input}_{i,j} = \text{concat}(b_i, \text{CLS} > ; \text{concat}(x, b_j) < \text{SEP} > ; d_i) \quad (1)$$

and then we compute the semantic embedding of the input pair and apply a linear classifier to output the relevance score:

$$E_{i,j} = \text{BERT}_{\text{CLS}}(\text{Input}_{i,j}) \quad (2)$$

$$S_{i,j} = \text{Sigmoid}(W^T E_{i,j}) \quad (3)$$

where  $\text{BERT}_{\text{CLS}}$  denotes the embedding of the [CLS] token to express the interaction between  $\text{concat}(x, b_j)$  and  $d_i$ . Also, to connect the protein description  $p$  and literature description  $L$ , we define the query  $\text{concat}(x, b_j)$  as:

$$\text{concat}(x, b_j) = \text{concat}(p, L) \quad (4)$$

Here we choose the recommended names  $p_{\text{name}}$  of a protein rather than the whole description  $p$ , since the species information is too shallow to capture functional similarity.  $L$  is the extracted literature context of target protein  $x$  from the Sentence Extraction component.

## 2.5 Model training

Let  $D^+$  denotes the set of definitions of the relevant GO terms for target protein  $x$  and  $D_r$  denotes the set of definitions of all candidate GO terms obtained in Retrieval, the negative sample set can be defined as  $D^- = D_r \setminus D^+$ . Given a sample pair  $(x, D^+)$ , for each branch  $b_i$ , the loss function is:

$$\mathcal{L}_{i,j} = \frac{1}{|D^+|} \sum_{d_i \in D^+} \log S_{i,j} - \frac{1}{|D^-|} \sum_{d_i \in D^-} \log S_{i,j} \quad (5)$$

where  $S_{i,j}$  is the score function given in (3) and  $\text{concat}(x, b_j)$  is the extracted query given in (4). To achieve better performance, we train a separate Cross-Encoder for each GO branch. Then the total loss for all proteins in training set  $X_{\text{train}}$  to train the Cross-Encoder for a branch  $b_j$  is defined as:

$$\mathcal{L}_{\text{train}} = \frac{1}{|X_{\text{train}}|} \sum_{(x, D^+) \in X_{\text{train}}} \mathcal{L}_{i,j} \quad (6)$$

In training, we keep  $|D^+| : |D^-| = 1 : 1$  (where  $|D^+|$  denotes the number of examples in  $D^+$ ) for optimal training results. So we randomly select  $|D^+|$  negative samples from  $D^-$ .

## 2.6 Alternative approaches for generating candidate GO terms

In Retrieval, candidate GO terms are generated based on the textual similarity using protein descriptions. It is worth noting that we can explore alternative approaches using other data sources rather than text to generate the candidate GO terms and investigate the possible enhancement of results by Rerank. In fact, in our experiments, we examine two alternative approaches for generating candidate GO terms: BLAST-

KNN and LR-ESM, which are both two major component methods in NetGO3.0 (Wang et al. 2023).

## 3 Experimental setup

### 3.1 Datasets and evaluation metrics

We collected experimental annotations of proteins from SwissProt (Boutet et al. 2016), GOA (Huntley et al. 2015), and GO (Ashburner et al. 2000) in July 2023. Following the settings of CAFA5, we only consider functional annotations validated by experimental (or high-throughput) evidence (or traceable author statements) or inferred by curators (Friedberg et al. 2023).

The sequences of proteins with annotated GO terms are extracted from UniProt. We extract PubMed identifiers from each protein entry in SwissProt, following DeepText2GO (You et al. 2018a). Similarly, we obtain protein descriptions from SwissProt which include three parts: "Protein names," an exhaustive list of all names of the corresponding protein; "Gene names," the name list of genes that encode a protein; "Organism names," showing the species information of each protein (The UniProt Consortium 2023). Besides, we downloaded the GO of 1 January 2023 and obtained definitions for every GO term.

To validate the effectiveness of GORetriever, following DeepText2GO, we randomly select 1000 SwissProt proteins for testing based on the species distribution of the CAFA5 test superset (around 140 000 proteins), and choose proteins that have literature information in SwissProt, resulting in three test datasets of 882, 861, and 811 proteins on MFO, BPO, and CCO, respectively. They are regarded as *text proteins*. The size of the test dataset is approximately 2 and 1.5 times that of NetGO2.0 and NetGO3.0, respectively, which were both demonstrated as a reliable performance indicator in previous CAFA challenges. Table 1 shows the statistics of datasets. Following the setting of CAFA5, we use the weighted maximum *F*-measure to evaluate the performance. It is calculated based on weighted precision and recall, in which the weights are the information content of the terms (Friedberg et al. 2023). The information content is defined as follows (Clark and Radivojac 2013):

**Table 1.** The number of proteins of different species on three branches for training and testing.

	Train			Test		
	MFO	BPO	CCO	MFO	BPO	CCO
Human ( <i>Homo sapiens</i> )	16 237	11 759	22 783	139	142	118
Mouse ( <i>Mus musculus</i> )	9560	11 045	10 361	111	113	109
Drome ( <i>Drosophila melanogaster</i> )	7134	10 985	9154	18	16	14
Arath ( <i>Arabidopsis thaliana</i> )	8444	8756	8756	101	103	100
Rat ( <i>Rattus norvegicus</i> )	5932	7000	5737	47	43	50
All species (not only the above)	79 866	90 014	95 988	882	861	811



$$IC_{i|v}^{w|v} \log_2 \frac{1}{Pr_{i|v}^{w|v} Pr_{i|v}^{w|v} Pr_{i|v}^{w|v}} \quad (7)$$

where  $Pr_{i|v}^{w|v}$  is the ancestor terms of term  $v$  and  $Pr_{i|v}^{w|v} Pr_{i|v}^{w|v} Pr_{i|v}^{w|v}$  denotes the conditional probability of term  $v$  given its ancestor terms in GO. Using the information content, we obtain the weighted  $F$ -measure as follows (Jiang *et al.* 2016):

$$wF_{\max}^{i|v} = \max \frac{2 \cdot wp_{i|v}^{w|v} \cdot wr_{i|v}^{w|v}}{wp_{i|v}^{w|v} + wr_{i|v}^{w|v}} \quad (8)$$

where  $wp_{i|v}^{w|v}$  and  $wr_{i|v}^{w|v}$  are weighted-precision and weighted-recall under a threshold  $\tau$ , given as follows:

$$wp_{i|v}^{w|v} = \frac{1}{m_{i|v}^{w|v}} \sum_{p_i \in P} \frac{IC_{i|v}^{w|v} \cdot S_{i|v}^{w|v}(p_i) \cdot I_{i|v}^{w|v}(p_i)}{\sum_{p_i \in P} IC_{i|v}^{w|v} \cdot S_{i|v}^{w|v}(p_i) \cdot I_{i|v}^{w|v}(p_i)}$$

$$wr_{i|v}^{w|v} = \frac{1}{n_e} \sum_{p_i \in P} \frac{IC_{i|v}^{w|v} \cdot S_{i|v}^{w|v}(p_i) \cdot I_{i|v}^{w|v}(p_i)}{\sum_{p_i \in P} IC_{i|v}^{w|v} \cdot S_{i|v}^{w|v}(p_i) \cdot I_{i|v}^{w|v}(p_i)}$$

where  $m_{i|v}^{w|v}$  is the number of proteins with scores not smaller than  $\tau$  for at least one GO term,  $n_e$  is the number of test proteins,  $S_{i|v}^{w|v}(p_i)$  is the prediction score of protein  $p_i$  on term  $v$  and  $I_{i|v}^{w|v}(p_i)$  denotes that protein  $p_i$  has the function of term  $v$  (You *et al.* 2018b).

### 3.2 Competing methods

We compare our method with important component methods in NetGO3.0: BLAST-KNN, LR-Text, LR-InterPro, LR-ESM (Wang *et al.* 2023). We also compare with two state-of-the-art sequence-based deep learning methods, DeepGOCNN (Kulmanov and Hoehndorf 2020), and ATGO (Zhu *et al.* 2022), which are re-trained on the same dataset of our method using their open sources. Note that all these competing methods are based on a single source of information. Furthermore, to explore the effect of incorporating protein descriptions, we add one more competing method, LR-ProtST, which uses ProtST to generate protein embedding from both protein sequences and descriptions.

**BLAST-KNN** aims to find similar proteins of a target protein and assigns annotated terms of the similar proteins to the target. We used BLAST to search similar proteins set  $H_i$  for protein  $p_i$  and the prediction score is computed as follows:

$$S_{i|v}^{w|v}(p_i) = \frac{\sum_{p_j \in H_i} B_{i|v}^{w|v}(p_i, p_j)}{\sum_{p_j \in H_i} B_{i|v}^{w|v}(p_i, p_j)} \quad (9)$$

where  $B_{i|v}^{w|v}(p_i, p_j)$  is the bit score (similarity) between two proteins,  $p_i$  and  $p_j$ , from BLAST.

**LR-InterPro** uses protein families, domains, and motifs to generate a binary feature vector for each protein, which is then used to train logistic regression (LR) classifiers for each GO term (You *et al.* 2018b).

**Net-KNN** finds similar proteins from a PPI network and the prediction score is obtained as follows:

$$S_{i|v}^{w|v}(p_i) = \frac{\sum_{p_k \in N_i} I_{i|v}^{w|v}(p_k) \cdot w_{i|v}^{w|v}(p_k, p_i)}{\sum_{p_k \in N_i} I_{i|v}^{w|v}(p_k) \cdot w_{i|v}^{w|v}(p_k, p_i)} \quad (10)$$

where  $N_i$  denotes the neighborhood of node  $p_i$  in a PPI network (STRING) and  $w_{i|v}^{w|v}(p_k, p_i)$  is the weight of the association between  $p_k$  and target protein  $p_i$  (You *et al.* 2019).

**LR-Text** is the text component of DeepText2GO.

**LR-ESM** uses ESM-1b to generate protein embedding (Rives *et al.* 2021) and trains LR classifiers for each GO term (Wang *et al.* 2023). Then we predict protein functions based on the trained classifiers.

**LR-ProtST** uses ProtST to generate protein embedding and train LR classifier for each GO term.

**DeepGOCNN** uses 1D convolutional neural networks (CNNs) with different filter sizes to scan protein sequences and learns a feature vector with the size of 8192 for function prediction (Kulmanov and Hoehndorf 2020).

**ATGO** extracts functional features by a pretrained protein language model, ESM-1b, and implements high-precision function prediction with a triplet neural network (Zhu *et al.* 2022).

### 3.3 Model parameters setup

In Sentence Extraction, we choose the MonoT5 (Nogueira *et al.* 2020) model, which is trained based on MS MARCO corpus (Bajaj *et al.* 2018), as the Seq2seq backbone to compute semantic similarity between sentences from literature and protein descriptions. We sort all sentences and pick the top 50% as the informative sentences. In Retrieval, we choose the name and species information of proteins in UniProt to build the BM25 (Lin *et al.* 2021) index. In Rerank, for Cross-Encoder, we set the batch size as 8 and the warm-up ratio as 0.1. To embed the textual description, we use PubMedBERT (Gu *et al.* 2021), which is pre-trained by millions of biomedical papers to obtain state-of-the-art performance on specialized tasks. In training, we randomly select 10% of the data as the validation set. For Retrieval, we extract around 1000 proteins for each branch as the validation set, because obviously the more training data in Retrieval, the better results can be achieved. Another important parameter for Retrieval is the number of retrieved proteins  $k$ , that have similar descriptions as those of the target protein. Based on the  $F_1$  scores over GO terms retrieved by Retrieval, we set  $k=3$  for MFO and BPO and  $k=2$  for CCO that has less annotated GO terms (see Section 4.2 for more details).

## 4 Results and analysis

### 4.1 Comparison with competing methods over text proteins in the test set

The key idea of GORetriever is to annotate the functions of a protein with textual information. In the left part of Table 2, we report the results of GORetriever and the competing methods over *text proteins* in the test set. We have three main findings: (i) GORetriever achieves the best performance in all three branches, especially for BPO. For example, GORetriever achieves 7.1% and 12.1% improvements over the second and third-best methods, respectively. Moreover, GORetriever achieves the highest  $wF_{\max}$  on average. This result demonstrates that GORetriever makes good use of textual information on proteins and GO terms in a reasonable and efficient way. (ii) Sequence-based methods are effective for MFO. BLAST-KNN (0.644) using sequence alignment performs better than LR-ESM (0.632) only utilizing ESM-1b. In contrast, LR-ProtST (0.649) using both ESM-1b and protein descriptions for embeddings achieves the second best. This suggests that protein embedding allows to capture functional information from amino acid sequences, and protein

**Table 2.** Performance comparison of competing methods.<sup>a</sup>

Method	Text proteins				Difficult proteins			
	MFO (882)	BPO (861)	CCO (811)	Ave. wF <sub>max</sub>	MFO (404)	BPO (419)	CCO (362)	Ave. wF <sub>max</sub>
<i>Sequence-based method</i>								
BLAST-KNN	0.644	0.471	0.595	0.570	0.603	0.454	0.513	0.523
LR-InterPro	0.627	0.465	0.591	0.561	<u>0.615</u>	0.450	0.525	0.530
LR-ESM	0.632	0.449	0.607	0.563	0.592	0.448	0.591	0.544
LR-ProtST <sup>b</sup>	<u>0.649</u>	0.462	<u>0.625</u>	0.579	0.610	0.461	<u>0.613</u>	0.561
DeepGOCNN	0.573	0.409	0.547	0.510	0.532	0.402	0.515	0.483
ATGO	0.642	<u>0.509</u>	0.589	<u>0.580</u>	0.611	0.487	0.599	<u>0.566</u>
<i>Network-based method</i>								
Net-KNN	0.400	0.458	0.596	0.485	0.422	<u>0.490</u>	0.611	0.508
<i>Text-based method</i>								
LR-Text	0.520	0.486	0.619	0.541	0.502	0.479	0.606	0.529
GORetriever	<b>0.659</b>	<b>0.545</b>	<b>0.653</b>	<b>0.619</b>	<b>0.619</b>	<b>0.573</b>	<b>0.651</b>	<b>0.614</b>

<sup>a</sup> The bold and underlined numbers denote the best and second-best performances, respectively. The figures in brackets denote the number of test proteins.

<sup>b</sup> LR-ProtST is also a text-related method, where ProtST uses both protein sequence and descriptions to generate embedding.

description is very helpful in generating better representation for AFP. (iii) Text information plays a significant role in predicting function on CCO. GORetriever, LR-ProtST and LR-Text, three text-related methods, outperform all sequence-based methods on CCO. This result indicates that the locations related to the cellular structures of proteins may have been annotated or implied in the scientific literature or protein descriptions.

Furthermore, we focus on *difficult proteins* [called by CAFA (Zhou *et al.* 2019)] in the test set with the BLAST identity of <0.6 to any proteins in training data. In the right part of Table 2, we present the prediction performance of competing methods over *difficult proteins*. We have the following three observations: (i) GORetriever achieves the best performance in all three branches, indicating the robustness of GORetriever over *difficult proteins*. (ii) LR-ProtST outperforms LR-ESM again in all three branches. This is a strong indication of the effectiveness of incorporating protein descriptions into protein embedding for better AFP. (iii) Most sequence-based methods show heavy performance decreases, especially BLAST-KNN. BLAST-KNN achieves an average wF<sub>max</sub> of only 0.523, suggesting that their performances heavily rely on homologous proteins.

All these results demonstrate that GORetriever is the most effective and robust among all competitive methods over *text proteins* and especially *difficult proteins*. We also present the prediction performances by other metrics ( $S_{min}$ ) in the supplement, where GORetriever also achieves the highest performances.

## 4.2 Robustness analysis

To determine the value of top  $k$ , we evaluate the  $F_1$  score (a standard performance measure in information retrieval) of Retrieval on the validation set. Note that the  $F_1$  score allows us to balance accuracy and cost. Figure 3 shows the  $F_1$  scores on the validation set, first increasing and then decreasing with increasing  $k$ , as a consistent trend over all three sets. Thus, we adopt  $k=2$  for CCO and  $k=3$  for MFO and BPO as the parameter values. We compute wF<sub>max</sub> on the test set with different  $k$  values to explore the robustness of GORetriever. Figure 3 shows the wF<sub>max</sub> curves, implying that (i) the trends of wF<sub>max</sub> and  $F_1$  scores are consistent,

suggesting that choosing the optimum parameter values of  $k$  would be possible, implying that similar proteins have similar functions, and (ii) the change of wF<sub>max</sub> (when changing  $k$ ) is smoother than  $F_1$  scores. We believe that Rerank in the second stage could further improve the annotation ability of GORetriever and more importantly, make the model, GORetriever, more robust against the change of parameter  $k$ .

## 4.3 Ablation experiment




*GORetriever is improved progressively during two stages.* We conducted ablation experiments to understand the contribution of each component in GORetriever. Given a target protein  $x$ , for GO term  $g_i$  associated with relevant proteins in Retrieval, we compute the score  $S$  between  $g_i$  and  $x$ , following the idea of BLAST-KNN (You *et al.* 2018b):

$$S_{i,j} = \frac{\sum_{x^j \in X_r} I_{i,j} \cdot g_i \cdot x^j \cdot B_{i,j} \cdot x \cdot x^j}{\sum_{x^j \in X_r} B_{i,j} \cdot x \cdot x^j}, \quad (11)$$

where  $B_{i,j} \cdot x \cdot x^j$  stands for the relevance score obtained by the BM25 algorithm, and  $I_{i,j} \cdot g_i \cdot x^j$  is an indicator whether protein  $x^j$  has the function of  $g_i$ . We use the same setting of  $k$  as discussed in Section 4.2. Table 3 shows the results, implying that (i) both components (Retrieval and Rerank) are beneficial to the result. Using Retrieval only has achieved better performances than LR-ESM (4.8%) and BLAST-KNN (3.5%), and (ii) Adding Rerank improves the annotation ability for *difficult proteins*. The improvement from Retrieval by Rerank is 4.9% on *text proteins*, but this improvement becomes 6.8% on *difficult proteins*.

*GORetriever enhances the performance of sequence-based methods via Rerank.* We choose the top 100 GO terms from the predictions of each of GORetriever, LR-ESM and BLAST-KNN as the input of Rerank. Table 3 shows the results (wF<sub>max</sub>), implying that (i) GORetriever yields different levels of enhancement for sequence-based methods by Rerank. The improvement is especially significant on BPO, which has more GO terms with deeper parts in the ontology structure, making prediction more challenging for traditional classification methods, and (ii) GORetriever shows robustness: Compared to *text proteins*, the results on *difficult*

**Figure 3.**  $F_1$  score and  $wF_{\max}$  scored by Retrieval only and GORetriever, respectively, changing  $k$ .**Table 3.** Ablations of using Retrieval or sequence-based AFP methods and then Rerank.<sup>a</sup>

Method	Text proteins				Difficult proteins			
	MFO (882)	BPO (861)	CCO (811)	Ave. $wF_{\max}$	MFO (404)	BPO (419)	CCO (362)	Ave. $wF_{\max}$
Retrieval	0.621	0.519	0.629	0.590	0.570	0.535	0.619	0.575
 Rerank	<b>0.659</b>	<b>0.545</b>	<b>0.653</b>	<b>0.619</b>	<b>0.619</b>	<b>0.573</b>	<b>0.651</b>	<b>0.614</b>
LR-ESM	0.632	0.449	0.607	0.563	0.592	0.448	0.591	0.544
 Rerank	0.628	<b>0.496</b>	<b>0.615</b>	<b>0.579</b>	<b>0.608</b>	<b>0.510</b>	<b>0.614</b>	<b>0.577</b>
BLAST-KNN	0.644	0.471	0.595	0.570	0.603	0.454	0.513	0.523
 Rerank	0.633	<b>0.495</b>	<b>0.598</b>	<b>0.575</b>	<b>0.605</b>	<b>0.480</b>	<b>0.566</b>	<b>0.550</b>

<sup>a</sup> A denser shadow means more performance improvement.

*proteins* are reduced for all methods, but the performance reduction is significantly alleviated by using Rerank.

#### 4.4 Performance over CAFA5 blind test set

In CAFA5, the organizers provided a large number of protein sequences (test superset), which consists of around 140 000 proteins, where around 92% of these proteins have annotated literature. Participants of CAFA5 made their predictions over all proteins in the test superset, and then  $wF_{\max}$  was used to rank the submissions of all participants over a blind test set (an unknown number of proteins out of test superset). Following this protocol, we examine the performance of GORetriever and the competing methods over the CAFA5 blind test set. [Figure 4](#) shows the performance results of all methods as well as, additionally, the top three submissions in CAFA5 (GOCurator, U900 and tito) for reference. The best method is GOCurator, an ensemble method (based on NetGO3.0), allowing to integrate protein structure-, sequence-, and text-based methods, including BLAST-KNN, LR-ESM, and so on. Note that we have proposed GOCurator for CAFA5, and GORetriever was the new and key component of GOCurator. From this result, we have the following three observations: (i) GORetriever outperforms LR-Text significantly, highlighting the methodological advantage of GORetriever over LR-Text. (ii) Incorporating GORetriever into NetGO3.0 successfully improves the  $wF_{\max}$  performance (from 0.587 to 0.604). This demonstrates that the strategy of GORetriever for predicting GO functions based on literature annotations is robust and can complement traditional AFP methods (which were used as other components in NetGO3.0). (iii) Without any knowledge of the proportion of text proteins in the blind test set, GORetriever

**Figure 4.** Results over the CAFA5 blind test set. The top three methods (using diagonal lines) in CAFA5: GOCurator, U900, and tito (both U900 and tito are sequence-based), are added. Our methods are indicated by the dark color.

achieves a competitive performance against BLAST-KNN, LR-ESM, and LR-InterPro, which all do not use any text information. This is consistent with the results of all test proteins (including both text and nontext proteins) in this study (see [Supplementary Material](#)).

#### 4.5 Case study of protein O82234

[Table 4](#) shows the prediction results of GORetriever and six typical competing methods for protein O82234 (“Translation initiation factor IF3-2”) in BPO (results of all 11 competing methods are shown in [Supplementary Table S3](#)). Methods based on protein embedding information perform unfavorably, such as LR-ESM, which could predict

only one correct term ( $F_1$  score of 0.076). ATGO using ESM-1b (with a triple neural network) outperforms LR-ESM with an  $F_1$  score of 0.357. In contrast, GORetrieve based on textual information achieves the best  $F_1$  score of 0.667 out of all 12 methods, and LR-text obtains a favorable  $F_1$  score of 0.47. Enhancing the base model by Rerank using textual information is also shown: BLAST-Rerank (0.296) and ESM-Rerank (0.545) yield significant improvements over BLAST-KNN (0.250) and LR-ESM (0.076), respectively. However, BLAST-Rerank and ESM-Rerank could not perform at the same level as GORetrieve (0.667), due to the unsatisfactory performances of base models, i.e. BLAST and ESM. This result highlights the importance of generating good candidate GO terms in the first stage.

To further investigate the impact of using literature annotation, we focus on two GO terms “chloroplast organization” (GO:0009658) and “system development” (GO:0048731), which are both correctly predicted only by GORetrieve or its variants (e.g. ESM-Rerank). Table 5 shows sentences (relevant to these two GO terms), which are extracted based on computations by Rerank. We can see that GORetrieve effectively captures the semantic correlation between extracted sentences and GO definitions, such as “disassembly of the chloroplast” and “chloroplast development,” resulting in facilitating the annotation of GO functions. Supplementary Fig. S2 illustrates the ontology graph based on the relations among 16 GO terms that are annotated to O82234 in BPO. Compared to the competing methods, both the text-based LR-Text and GORetrieve correctly predict the “developmental process” branches. This result clearly demonstrates the advantage of incorporating textual information for AFP.

To further elucidate the influence of textual information on predictive outcomes, we focus on the “false positive” GO categories identified by GORetrieve. Notably, several of

these categories are broad and general in nature, such as GO:0065007 (biological regulation), GO:0043170 (macromolecule metabolic process), and GO:0044238 (primary metabolic process). Although these categories may not precisely correspond to the target protein, the associated textual descriptions are sufficiently broad and ambiguous, thereby achieving a semantic match that may still capture some aspects of the protein’s biological functions. These results also suggest that integrating GORetrieve with other models could yield unexpectedly nuanced outcomes.

5 Conclusion

We have presented GORetrieve for accurate automated function annotation of proteins. For a given target protein, GORetrieve first generates GO term candidates using protein descriptions and also relevant informative sentences (from literature), and then reranks the GO terms by deep information retrieval on these informative sentences. Through extensive experiments on *text proteins*, we demonstrate that GORetrieve can annotate protein functions accurately by using textual information, especially powerful for predicting *difficult proteins*. Furthermore, we replace the Retrieval of GORetrieve with existing sequence-based AFP methods and find that the performance of these methods can be always enhanced by Rerank of GORetrieve, indicating that Rerank is universally effective for AFP. Possible future work would be to create an approach to incorporate the interactions between/among GO terms to improve the current AFP performance further.

Supplementary data

Supplementary data are available at *Bioinformatics* online.

Conflict of interest

None declared.

Funding

This work was supported by the National Natural Science Foundation of China [62272105], Shanghai Municipal Science and Technology Major Project [2018SHZDZX01], and ZJ Lab and Shanghai Center for Brain Science and Brain-Inspired Intelligence Technology, the 111 Project [B18015]; in part by MEXT KAKENHI [19H04169, 20F20809, 21H05027 and 22H03645] and the AIPSE program of the

Table 4. Prediction results for O82234 in BPO.<sup>a</sup>

Method	TP	FP	$F_1$ -score
BLAST-KNN	4	14	0.250
LR-Text	8	12	0.471
LR-ESM	1	11	0.076
ATGO	5	9	0.357
BLAST-Rerank	4	9	0.296
ESM-Rerank	9	10	0.545
GORetrieve	11	8	<b>0.667</b>

<sup>a</sup> TP and FP are the numbers of true and false positives out of all 16 GO terms. The bold number denote the best performance. The root term “biological process” (GO:0008150) is excluded. Detailed results of all 12 methods are shown in the Supplementary material.

Table 5. GO term, GO definition and predicted relevant sentences of O82234: translation initiation factor IF3-2 in BPO.

GO term	GO definition	Relevant Sentences
chloroplast organization (GO:0009658)	A process that is carried out at the cellular level which results in the assembly, arrangement of constituent parts, or <b>disassembly of the chloroplast</b> .	Genetic and molecular evidence indicate that SVR9 and its close homolog SVR9-LIKE1 (SVR9L1) are functionally interchangeable and their combined activities are essential for <b>chloroplast development</b> and plant survival.
system development (GO:0048731)	The process whose specific outcome is <b>the progression of an organismal system over time</b> , from its formation to the mature structure. A system is a regularly interacting or interdependent group of organs or tissues that work together to carry out a given biological process.	Interestingly, we found that SVR9 and SVR9L1 are also involved in <b>normal leaf development</b> . Genetic analysis established that SVR9/SVR9L1-mediated <b>leaf margin development</b> is dependent on CUP-SHAPED COTYLEDON2 activities and is independent of their roles in <b>chloroplast development</b> .

Bolded text represents evidence that can support the final function annotation.



Academy of Finland to H.M. This paper was published as part of a supplement financially supported by ECCB2024.

## Data availability

The data is available at <https://github.com/ZhuLab-Fudan/GORetriever/tree/main/data>.

## References

- Altschul SF, Madden TL, Schaffer AA *et al.* Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* 1997;**25**:3389–402.
- Ashburner M, Ball CA, Blake JA *et al.* Gene ontology: tool for the unification of biology. *Nat Genet* 2000;**25**:25–9.
- Bajaj P, Campos D, Craswell N *et al.* MS MARCO: a human generated machine reading comprehension dataset. arXiv, arXiv:1611.09268, 2018, preprint: not peer reviewed.
- Bateman A, Martin M-J, Orchard S *et al.*; The UniProt Consortium. UniProt: the universal protein knowledgebase in 2023. *Nucleic Acids Res* 2023;**51**:D523–31.
- Boadu F, Cao H, Cheng J *et al.* Combining protein sequences and structures with transformers and equivariant graph neural networks to predict protein function. *Bioinformatics* 2023;**39**:i318–25. ISSN 1367–4811.
- 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. *Plant Bioinf Methods Protoc* 2016;**1374**:23–54.
- Cao Y, Shen Y. TALE: transformer-based protein function annotation with joint sequence–label embedding. *Bioinformatics* 2021;**37**:2825–33.
- Clark WT, Radivojac P. Information-theoretic evaluation of predicted ontological annotations. *Bioinformatics* 2013;**29**:i53–61.
- Friedberg I, Radivojac P, De Paolis C *et al.* CAFA 5 Protein Function Prediction, 2023. <https://kaggle.com/competitions/cafa-5-protein-function-prediction>
- Gane A, Bileschi ML, Dohan D *et al.* ProtNLM: model-based natural language protein annotation, 2022. <https://www.uniprot.org/help/ProtNLM>
- Glorigijevic V, Renfrew PD, Kosciolk T *et al.* Structure-based protein function prediction using graph convolutional networks. *Nat Commun* 2021;**12**:3168.
- Gu Y, Tinn R, Cheng H *et al.* Domain-specific language model pretraining for biomedical natural language processing. *ACM Trans Comput Healthcare* 2021;**3**:1–23.
- 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.
- 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.
- Kulmanov M, Hoehndorf R. DeepGOPlus: improved protein function prediction from sequence. *Bioinformatics* 2020;**36**:422–9.
- Kulmanov M, Hoehndorf R. DeepGOZero: improving protein function prediction from sequence and zero-shot learning based on ontology axioms. *Bioinformatics* 2022;**38**:i238–45. <https://doi.org/10.1093/bioinformatics/btac256>
- Lai B, Xu J. Accurate protein function prediction via graph attention networks with predicted structure information. *Brief Bioinf* 2021;**23**:bbab502. <https://doi.org/10.1093/bib/bbab502>
- Le Q, Mikolov T. Distributed representations of sentences and documents. In: *ICML*, Beijing 2014, 1188–96.
- Lin J, Ma X, Lin S *et al.* Pyserini: a python toolkit for reproducible information retrieval research with sparse and dense representations. In: *SIGIR*, Canada, 2021, 2356–62.
- Makrodimitris S, van Ham RCHJ, Reinders MJT *et al.* Automatic gene function prediction in the 2020's. *Genes (Basel)* 2020;**11**:1264.
- Marchler-Bauer A, Derbyshire MK, Gonzales NR *et al.* CDD: NCBI's conserved domain database. *Nucleic Acids Res* 2015;**43**:D222–6.
- Minaee S, Kalchbrenner N, Cambria E *et al.* Deep learning–based text classification: a comprehensive review. *ACM Comput Surv* 2021;**54**:1–40.
- Nogueira R, Jiang Z, Pradeep R *et al.* Document ranking with a pre-trained sequence-to-Sequence model. In: *EMNLP 2020*, Online 2020, 708–18.
- Radivojac P, Clark WT, Oron TR *et al.* A large-scale evaluation of computational protein function prediction. *Nat Methods* 2013;**10**:221–7.
- Rao R, Bhattacharya N, Thomas N *et al.* (2019). Evaluating Protein Transfer Learning with TAPE. *Advances in neural information processing systems*, 32, 9689–9701.
- Reimers N, Gurevych I. Sentence-BERT: sentence embeddings using Siamese BERT-Networks. In: *EMNLP-IJCNLP 2019*, Hong Kong, China, 2019, 3982–92.
- 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 USA* 2021;**118**: e2016239118.
- Sillitoe I, Lewis TE, Cuff A *et al.* CATH: comprehensive structural and functional annotations for genome sequences. *Nucleic Acids Res* 2015;**43**:D376–81.
- Wang S, You R, Liu Y *et al.* NetGO 3.0: a protein language model improves large-scale functional annotations. *Genomics. Proteomics Bioinf* 2023;**21**:349–58.
- Xu H, Wang S. ProTranslator: zero-shot protein function prediction using textual description. In: *RECOMB*. La Jolla, USA, 2022, 279–94.
- Xu M, Yuan X, Miret S *et al.* ProtST: multi-modality learning of protein sequences and biomedical texts. In: *ICML*. Hawaii, USA, 2023, 38749–67.
- 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. ISSN 0305–1048.
- You R, Huang X, Zhu S *et al.* DeepText2GO: improving large-scale protein function prediction with deep semantic text representation. *Methods* 2018a;**145**:82–90.
- You R, Zhang Z, Xiong Y *et al.* GOLabeler: improving sequence-based large-scale protein function prediction by learning to rank. *Bioinformatics* 2018b;**34**:2465–73.
- 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.
- You R, Yao S, Mamitsuka H *et al.* DeepGraphGO: graph neural network for large-scale, multispecies protein function prediction. *Bioinformatics* 2021;**37**:i262–71.
- 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**:244–23.
- 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.