Bioinformatics, 2023, 1–8
doi: DOI HERE
Advance Access Publication Date: Day Month Year
Original Paper

DualNetGO: A Dual Network Model for Protein Function Prediction via Effective Feature Selection

Zhuoyang Chen[1] and Qiong Luo[1,*]

[1]Data Science and Analytics Thrust, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, Guangdong, 511400, China

*Corresponding author. Her other affiliation is the HongKong University of Science and Technology. luo@ust.hk

FOR PUBLISHER ONLY Received on Date Month Year; revised on Date Month Year; accepted on Date Month Year

Abstract

Motivation: Protein-protein Interaction (PPI) networks are crucial for automatically annotating protein functions. As there are different types of evidence to define PPI networks, multiple PPI networks exist for the same set of proteins to capture their properties from different aspects, creating challenges in effectively utilizing these heterogeneous graphs for protein function prediction. Recently, several deep learning models have combined PPI networks from all evidence, or concatenated all graph embeddings. However, the lack of a delicate selection procedure prevents the effective harness of information from different PPI networks as they vary in densities, structures and noise levels. Consequently, combining protein features indiscriminately could increase the noise level, leading to decreased model performance.
Results: We develop DualNetGO, a dual network model comprised of a classifier and a selector, to predict protein functions by effectively selecting features from different sources including graph embeddings of PPI networks, protein domain and subcellular location information. Evaluation of DualNetGO on human and mouse datasets in comparison with other network-based models show at least 4.5%, 6.2% and 14.2% improvement on Fmax in BP, MF and CC Gene Ontology categories respectively for human, and 3.3%, 10.6% and 7.7% improvement on Fmax for mouse. We further show that our model is insensitive to the choice of graph embedding method and is time- and memory-saving. These results demonstrate that combining a subset of features including PPI networks and protein attributes selected by our model is more effective in utilizing PPI network information than only using one kind of or concatenating graph embeddings from all kinds of PPI networks.
Availability and implementation: The source code of DualNetGO and some of the experiment data are available at: https://github.com/georgedashen/DualNetGO.

1 Introduction

Proteins are the main player in biological processes, and their functions can be categorized into three aspects by Gene Ontology (GO): biological process (BP), molecular function (MF) and cellular component (CC) (Aleksander *et al.*, 2023). Knowing a protein's function helps explain its role and evaluate its importance in a biological process, and is also useful for enzyme and drug design (Radivojac *et al.*, 2013). However, by 2023 less than 1% of over 200 million known proteins have been revealed their functions (Uniprot, 2023; Aleksander *et al.*, 2023) because experimental protein annotation is laborious, time consuming and costly (Luck *et al.*, 2020). Thus, automatically annotating protein functions becomes a meaningful and yet challenging task. In this paper, we explore how to predict protein functions with deep learning networks.

With the development of the Critical Assessment of Functional Annotation (CAFA) community, dozens of advanced algorithms have been proposed for automatic protein function annotation (Zhou *et al.*, 2019). Generally, these algorithms adopt (1) sequence-based, (2) structure-based, (3) network-based, or (4) mixed source methods. Sequence-based methods are based on the hypothesis that if two protein sequences are similar, they will share similar functions. Such prior models assign protein functions based on similarity in raw sequences (Hamp *et al.*, 2013) or annotated motifs (Gong *et al.*, 2016), similarity in learned features from neural networks (Kulmanov *et al.*, 2018; Kulmanov and Hoehndorf, 2021; Cao and Shen, 2021) or even protein language models (Wang *et al.*, 2023; Oliveira *et al.*, 2023). However, protein functions can not be determined when the query protein shares low sequence similarity with others, and proteins with similar functions are not necessarily similar in sequences.

While structure-based methods could predict protein functions using structure similarity even when sequence similarity is low (Gligorijević *et al.*, 2021; Lai and Xu, 2022), the amount of experimentally determined protein structure data is small and the predicted structures using tools such as AlphaFold could be inaccurate (Jumper *et al.*, 2021; Varadi *et al.*, 2022). In comparison, network-based methods (Milenković and Pržulj, 2008; Cho *et al.*, 2016; Gligorijević *et al.*, 2018) utilize protein-protein interaction (PPI) networks to predict protein functions based on the assumption that if two proteins have interactions or share similar roles in the network, they will share similar functions. A limitation of this approach is that PPI networks are incomplete and noisy and therefore difficult to extract useful information (Pandey *et al.*, 2012). Finally, mixed-source methods are models that combine information from different sources such as sequences, structures, PPI networks and literature. Several mixed source models achieved top performance in previous CAFA challenges (You *et al.*, 2018; Yao *et al.*, 2021).

PPI networks provide additional information into how proteins work cooperatively to exert a certain function, which is difficult to determine directly from protein sequences or structures. Therefore, several deep learning methods have been proposed to combine PPI networks with sequence features (Fan *et al.*, 2020; Wu *et al.*, 2023) and achieve comparable performance with other state-of-the-art (SOTA) mixed-source ensemble models. According to the STRING PPI database (Szklarczyk *et al.*, 2023) there are seven types of evidence to define an interaction between two proteins: neighborhood, fusion, cooccurence, coexpression, experimental, database and textmining. However, most of existing network-based methods simply concatenate or average over latent factors encoded from PPI networks of all types of evidence (Cho *et al.*, 2016; Gligorijević *et al.*, 2018), or use a combined PPI network that integrates edges from all evidence (Fan *et al.*, 2020; Wu *et al.*, 2023). As PPI networks from different evidence can be represented as graphs with an equal number of nodes but different numbers of edges, these networks have different topological properties including density and connectivity. Consequently, indiscriminate use of different PPI network features could further increase the noise level of the data and result in decreased model performance. How to properly and effectively utilize different PPI networks is still to be explored for protein function prediction.

In this study, we develop a dual network model named DualNetGO to predict protein function by effectively determining the combination of features from PPI networks and protein attributes without enumerating each possibility. We design a feature matrix space that includes eight matrices: seven for graph embeddings of PPI networks from different evidence and one for protein domain and subcellular location. After encoding each PPI network into low-dimensional latent factors, the two multilayer perceptron (MLP) components of DualNetGO, a classifier and a selector, are trained alternately to evaluate the importance of each matrix and choose a proper combination to predict protein functions. Experiment results on human and mouse datasets show that DualNetGO outperforms other network-based methods and is insensitive to the choice of graph embedding methods. Further evaluation shows that with proper settings DualNetGO takes less time and requires less memory in data preprocessing and training. These results demonstrate that DualNetGO is an efficient and effective model for protein function prediction, outperforming those only using one kind of or simply concatenating latent factors from all kinds of PPI networks, and providing insight into better ways of utilizing heterogeneous PPI network data.

2 Materials and methods

2.1 Dataset

PPI data are retrieved from STRING database version 11 on STRINGv11.5 (Szklarczyk *et al.*, 2021) with the taxonomy code 9606 for human and 10090 for mouse. GO functional annotations are downloaded from the gene ontology website (version 2022-01-13 release). Protein attributes that include subcellular location and **Pfam** protein domain annotation are retrieved from the Uniprot database (v3.5.175). Following the CAFA challenge setting (Jiang *et al.*, 2016), we only retain protein functions with experimental evidence 'IDA', 'IPI', 'EXP', 'IGI', 'IMP', 'IEP', 'IC' or 'TA' as one-hot encoded labels, and define proteins with annotations before 2018-01-01 as the training set, those between 2018-01-02 and 2020-12-31 as the validation set and those after 2021-01-01 as the test set. This temporal holdout method to split data was proposed in the CAFA challenge to mimic a real-life application scenario instead of random splitting (Jiang *et al.*, 2016). To make sure there are a sufficient number of proteins for each label, we retain labels with at least 10, 5 and 1 proteins in the training, validation and test set, respectively, following a previous study (Wu *et al.*, 2023). To further reduce the correlation or dependency between over 5000 GO terms in total, any labels containing more than 5% of the number of proteins in human and mouse PPI network are removed as well. The statistics of the final training, validation and test set are shown in Supplementary Table S1.

2.2 Method

2.2.1 PPI networks of different types of evidence

PPI networks can be defined by seven available types of evidence: neighborhood, fusion, cooccurence, coexpression, experimental, database and textmining according to the STRING database (Szklarczyk *et al.*, 2023). A score is assigned to each of these evidence for a protein pair and there is also a *combined* score that integrates scores from all evidence. If there is a positive score, the interaction between two proteins is considered to exist for a specific evidence. Totally 19385 and 21373 proteins were included in the PPI network of human and mouse, respectively, in this study, and proteins with annotated GO functions but absent in the PPI network were removed. Statistics of different PPI networks from different evidence are shown in Supplementary Table S2. Different evidence of PPI are important for protein function prediction as they differ in topological information such as density, average degree and clustering coefficient, thus they can provide valuable information about the nature of a particular association.

2.2.2 Transformer-based autoencoder for PPI and protein attributes

DualNetGO contains two components: a graph encoder and a predictor (Figure 1a,b). The graph encoder is a previously published transformer-based autoencoder (denoted as **TransformerAE**) (Wu *et al.*, 2023) that takes protein attributes and PPI networks as input and outputs low-dimensional embeddings. PPI networks are transformed into weighted adjacency matrices using minmax-normalized scores of the corresponding evidence. For protein attributes, we include

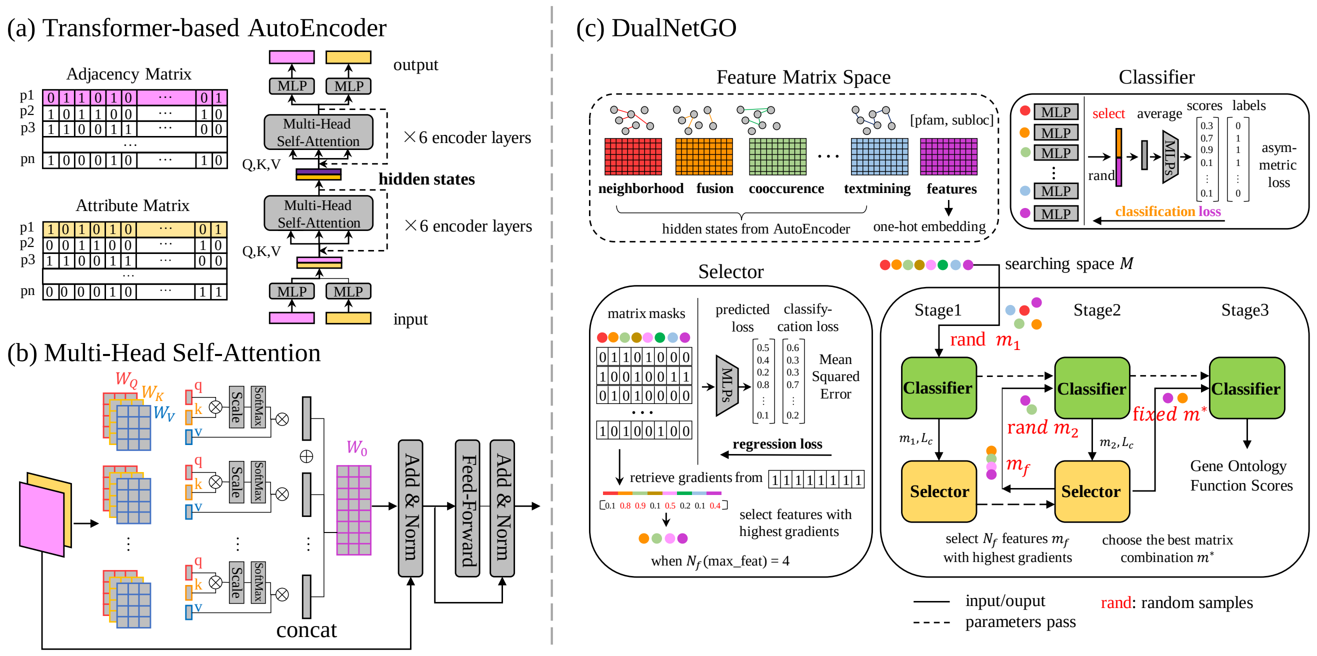

Fig. 1: The workflow of DualNetGO. (a) Architecture of TransformerAE for generating PPI network graph embeddings. (b) Multi-head attention layer mechanism. (c) Architecture of DualNetGO to select features for protein function prediction.

protein Pfam domains and subcellular locations for their easy retrieval from Uniprot data and wide use in previous studies. Domain terms that appear in fewer than 6 across proteins are removed. In the TransformerAE, the adjacency matrix and protein attribute matrix together go through 6 multi-head attention layers for the encoder and another 6 layers for the decoder to fuse information from the two sources. The core of the attention mechanism is the Scaled Dot-Product Attention, where Q is query, K is key and V is value matrix (Vaswani *et al.*, 2017), and d_k is the dimension of query and key vectors in the matrix:

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{1}$$

During the self-supervised learning process of TransformerAE, differences between the original input before being passed into the encoder and the reconstructed output after the decoder are minimized by binary cross-entropy. Two 512-dimension matrices (one for adjacency matrix and another for protein attributes) of latent factors are retrieved after being encoded by the multi-head attention encoder, but only the graph embeddings for PPI networks are included in the feature matrix space. Seven embeddings of PPI networks from seven types of evidence and the original unencoded protein attributes together form the final feature matrix space for feature selection in the predictor.

Since there are different graph embedding methods such as node2vec (Grover and Leskovec, 2016) and GAE (Kipf and Welling, 2016), the effects of using different graph embedding methods will be also in the experiment section.

2.2.3 Dual network architecture of DualNetGO

The predictor is a dual network comprised of a *classifier* and a *selector* (Figure 1c). The classifier maintains a one-layer MLP with ReLU as the nonlinear activation function for each matrix in the feature matrix space to further reduce the dimension, and a two-layer MLP (denoted as the **prediction head**) with softmax activation function in the second layer to output a score for each GO term. Selected feature matrices will first go through their own MLP modules and be averaged, then pass through the prediction head. For protein function prediction, asymmetric loss (ASL) (Ridnik *et al.*, 2021) is used as the loss function to achieve better performance for a multi-label task, since in a typical multi-label dataset sample numbers could be severely imbalanced across labels, eg. a term may only have a few proteins out of several thousand.

ASL is defined as:

$$ASL = \frac{1}{N_{train} \times K} \sum_{i=1}^{N_{train}} \sum_{k=1}^{K} -y_{ik}L_{+} - (1-y_{ik})L_{-} \tag{2}$$

$$\begin{cases} L_{+} = (1-p_{ik})^{\gamma^{+}} log(p_{ik}) \\ L_{-} = (p_{ik})^{\gamma^{-}} log(1-p_{ik}) \end{cases} \tag{3}$$

N_{train} is the number of proteins in the training set, K is the number of functions in a specific category, γ^{+} and γ^{-} are the focusing parameters for positive and negative samples, respectively. When both parameters are set to 0, ASL equals to a binary cross entropy. In this study, we set γ^{+} to 0 and γ^{-} to 2 to reduce the contribution of easy negative samples, encouraging the model to make more positive predictions.

The selector is a two-layer MLP that takes the choice of feature matrices which are presented as one-hot encoding (denoted as **mask**), as the input, and outputs a scalar value as the predicted classification loss from the classifier using the selected matrices in the mask. Mean squared error (MSE) is used as the loss function between the predicted classification loss and the true loss from the classifier. The selector learns to evaluate the classifier's performance with the selected subset of feature matrices, and is expected to output a lower value when the selected input subset is more suitable for predicting protein function.

2.2.4 Training process of DualNetGO

The training process is divided into three stages, and two networks are trained alternately in the first two stages. The number of training epochs for each stage is defined as $E1$, $E2$ and $E3$, respectively, and the maximum number of feature matrices to be included for prediction is defined as N_f. These numbers are set as hyperparameters before training.

In the first stage, a random combination of feature matrices is sampled from the matrix space M with no more than N_f matrices at the beginning of each epoch. These matrices go through the classifier and an ASL prediction loss on validation set L_c is calculated. L_c is further backpropagated to update weights of corresponding MLP modules in the classifier for each selected matrix. The combination of selected matrices is encoded as a one-hot encoding mask $m1$, and together with L_c is used as input for the selector. An MSE loss L_s is calculated between the output of the selector and L_c, and backpropagated to update weights in the selector. In total $E1$ combinations will be evaluated, and weights in the MLP module for each matrix in the classifier will be updated. The number of rounds for the weights in an MLP module to be updated depends on how many times the corresponding matrix is sampled.

In the second stage, in each epoch we first create a mask with equal weights of 0.5 representing an equal chance for each matrix to be selected, and then we use this mask as input for the selector and calculate the gradient of each element in the mask with respect to the model. Given a trained selector model, gradients of the input are expected to reflect the importance of each element. We select the corresponding matrices with top N_f absolute gradient values, indicated by the indices in the mask, to form a new matrix space M_f. Because the optimal combination is a subset of M_f, 10 combinations are sampled from M_f and evaluated by the classifier on the validation set. The lowest loss and the corresponding mask $m2$ are recorded, and $m2$ is used for a similar process in stage 1 to train the classifier and then the selector. Different epochs in stage 2 may generate different M_fs as weights are still updated for the two networks. In total, a number of $E2$ losses and masks are recorded.

In the third stage, the mask with the minimal validation loss across the record is identified as $m*$, and the training process for the classifier is continued only using the corresponding matrices with $m*$. Weights in the MLP modules with respect to $m*$ and the prediction head in the classifier will be updated for $E3$ epochs. Performance on test set data is reported as the final results when the classifier achieves the best Fmax score on the validation set.

2.3 Evaluation metrics

Metrics from two aspects are used to evaluate model performance for protein function prediction (Jiang *et al.*, 2016). From the protein-centric aspect, predicting functions for a protein is a multi-label task where the number of predicted labels is not fixed. Macro-F1 (M-F1), F1, Fmax score and accuracy that we use in this study belong to this category to evaluate how well a model can reveal a protein's functions. Fmax is the most widely used metric for protein function prediction and defined as:

$$Fmax = \max_{\tau} \left\{ \frac{2 \times precision(\tau) \times recall(\tau)}{precision(\tau) + recall(\tau)} \right\} \tag{4}$$

where τ is a flexible threshold for both recall and precision to obtain the maximum Fmax score.

Precision and recall for a multi-label task are defined as:

$$\begin{cases} precision(\tau) = \frac{1}{s(\tau)} \sum_{i=1}^{s(\tau)} \frac{\sum_k I(p_{ik} > \tau \wedge y_{ik} \equiv 1)}{\sum_c I(p_{ik} > \tau)} \\ recall(\tau) = \frac{1}{n} \sum_{i=1}^{n} \frac{\sum_c I(p_{ik} > \tau \wedge y_{ik} \equiv 1)}{\sum_c I(y_{ik} \equiv 1)} \end{cases} \tag{5}$$

$s(\tau)$ denotes the number of proteins that are predicted with at least one function. k is the total number of labels for a specific functional category. p_{ik} is the predicted score for the function and y_{ik} is the ground truth with 1 indicating the existence of the function. n is the total number of proteins to be evaluated.

Accuracy for this multi-label task is defined as the proportion of proteins with all functions correctly predicted (no missing label or extra prediction). While M-F1, F1, Fmax accept scores as input, accuracy needs a threshold to generate binary labels. The threshold is determined by achieving the highest Fmax score on the validation set.

Term-centric evaluation can be viewed as a series of binary classifications to predict what proteins belong to a specific GO term. Two types of area under the precision-recall (AUPR) curve are among this aspect: micro-AUPR (m-AUPR) is the AUPR calculated across all true labels and predictions, and macro-AUPR (M-AUPR) is the average of AUPR values over AUPRs of all GO terms.

2.4 Experiment setup

DualNetGO is only trained on the training set data defined by a temporal hold-out method as described in the **Dataset** section. The validation set is used to evaluate the model performance during training and when the model attains the best Fmax score, its performance on the test set is reported as the final results. Data in the validation and test set is not used for training the model. We set $E2 + E3 = 100$ for easier implementation and limit N_f ranging from 2 to 5.

We compare DualNetGO with two baseline methods and four network-based models. The naive method simply assigns the relative frequency of a term over all proteins in the training set as the score for this term for all proteins in the test set. The BLAST method transfers GO terms of a target protein in the training set to the query protein in the test set via the *blastp* software, and the identity score of alignment is used as a coefficient for all assigned terms. *Mashup* (Cho *et al.*, 2016) generates balanced hidden diffusion states across different PPI networks; *deepNF* (Gligorijević *et al.*, 2018) is a deep learning model that uses MLP-based autoencoders (MLPAE) for different PPI networks and concatenates latent factors; *Graph2GO* (Fan *et al.*, 2020) utilizes variational graph autoencoders (GAE) on a combined of PPI network and sequence similarity network with protein attributes as input features; *CFAGO* (Wu *et al.*, 2023) designs a transformer-based autoencoder (TransformerAE) to cross-fuse the combined PPI graph and protein attributes. Details can be found in Supplementary Section 4.

DualNetGO is implemented using Pytorch in Python. All hyperparameters of the TransformerAE graph encoder and data preprocessing follow the *CFAGO* paper (Wu *et al.*, 2023). For DualNetGO, hidden dimensions of the classifier are set to 512 except for the prediction head whose dimension is set to the number of GO terms, and 256 for the selector. Dropout rates are set to 0.5 for all components except for the prediction head, which is set to 0.1. Learning rates are set to 0.01 for both the classifier and the selector. AdamW is chosen as the optimizer for the classifier and the selector with weight decays set to 0 and 1e-5, respectively. Hyperparameters are determined by manual grid search on the validation data (see Table **??**). All experiments are conducted with a single RTX 3090 GPU with 24G memory. Single run for training DualNetGO takes 2-5 minutes and 2-6 GB depending on the number of iterations and and size of training data.

3 Experiments

3.1 DualNetGO outperforms competing network-based models

Figure 2 shows that DualNetGo outperforms other models on most of the metrics across GO aspects and organisms except MF in mouse. Specifically, DualNetGO gains improvement of at least 0.045, 0.062 and 0.142 (up to 0.459, 0.226 and 0.464) in terms of Fmax for BP, MF and CC respectively on human, and 0.027, 0.077 (up to 0.296 and 0.502) for BP and CC on mouse. For m-AUPR, DualNetGO achieves at least 0.058, 0.026 and 0.141 higher (up to 0.364, 0.133 and 0.422) for BP, MF and CC respectively on human, and 0.001, 0.147 (up to 0.186 and 0.459) for BP and CC on mouse. Improvements in M-AUPR, M-F1 and F1 can also be observed in half of the scenarios (more details in Supplementary Table S3,S4).

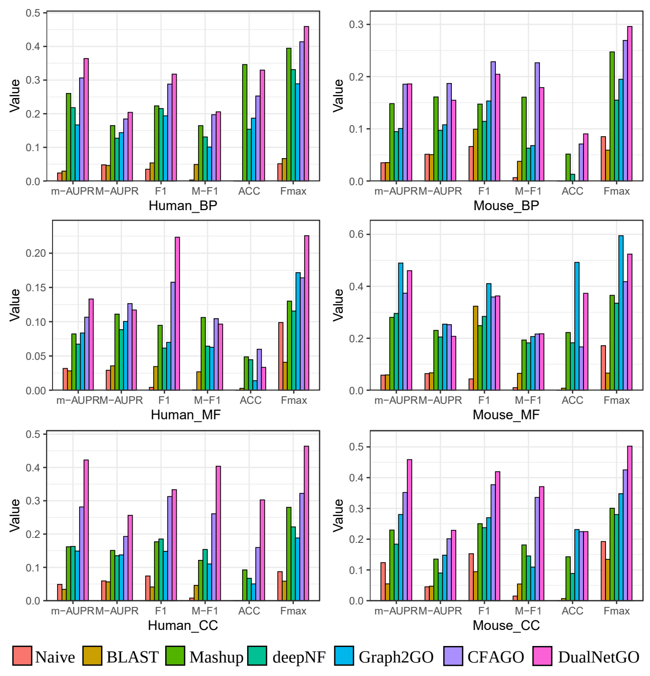

Fig. 2: Performance of DualNetGO for protein function prediction compared with other network-based models.

In the MF category of mouse, DualNetGO produces slightly worse results than Graph2GO, which is not surprising because Graph2GO not only uses GAE on the PPI network, but also on the sequence similarity network which provides additional and valuable information that is not included in other models in the comparison. Several studies (Fan *et al.*, 2020; Oliveira *et al.*, 2023) suggest that MF is more related to sequence patterns that may not be reflected by Pfam protein domains and PPI networks. However, retrieval of sequence similarity through BLAST requires quadratic time with respect to the number of sequences, which is costly when this number scales up.

The accuracy of DualNetGO is not distinct compared to other metrics, due to how accuracy is measured for the multi-label task. It requires all labels (GO terms) to be correctly predicted for one protein, and needs a threshold on the function score to define the binary class. The threshold for a model is determined by the evaluation on the validation set with the highest Fmax score, but the data distribution between the validation set and the test set may be different. It is not surprising that BLAST and Naive attain 0 accuracy, as they tend to make extra or missing predictions.

3.2 DualNetGO benefits from other graph embedding methods

We implement other graph embedding methods including node2vec, GAE and MLPAE to replace the TransformerAE in the PPI network preprocessing step, in order to investigate whether DualNetGO is affected by the choice of graph embedding methods. To make a more comprehensive comparison, we also consider situations when only using protein attributes without PPI networks (denoted as **Feature** in figures), using randomly generated latent factors (**random**) or not using any graph embedding techniques but only the raw adjacency matrix (**NoEmbed**) as input for each PPI network. Figure 3 shows that the effects of graph embedding methods differ by PPI usage scenarios, but DualNetGO outperforms other scenarios on the human dataset. The superior performance of DualNetGO is not affected by the choice of graph embedding methods, as we can see that DualNetGO performs better than the best scenario of other methods, regardless of the choice of graph embedding methods in all three categories, except for the use of node2vec in BP. Even simply using raw adjacency matrices of PPI networks as input for DualNetGO still yields comparable results, with the highest Fmax score on BP. While TransformerAE does not always produce the highest Fmax score in all scenarios, it has decent performance on both BP and CC. The use of protein attributes yields better results than using some of the PPI networks especially on CC, and information of PPI networks extracted via graph embedding is useful rather than random noise. Similar observation is found on the mouse dataset (Supplementary Figure S2). These results show the effectiveness of DualNetGO on protein function prediction with the matrix selection strategy is robust and could benefit from advanced graph embedding algorithms in the future.

3.3 Analysis of model training time and validation loss

The update of parameters in the classifier can be viewed as a stochastic process, which strikes a balance between determining the optimal subset and reducing the training time. While fluctuating loss is observed during stage 1 and 2 (Figure 5), the overall training loss decreases over time and Fmax on the validation set is reaching the plateau.

Time for preprocessing data and training is also compared for various models. Results (Figure 4) show that the data preprocessing time needed for CFAGO and DualNetGO_TransformerAE is more than one magnitude longer than the other models.

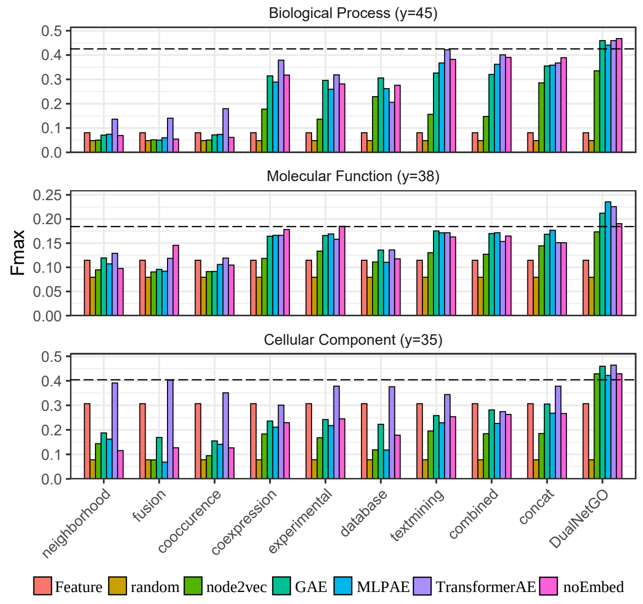

Fig. 3: Fmax scores across different graph embedding methods and different PPI usage settings on human dataset.

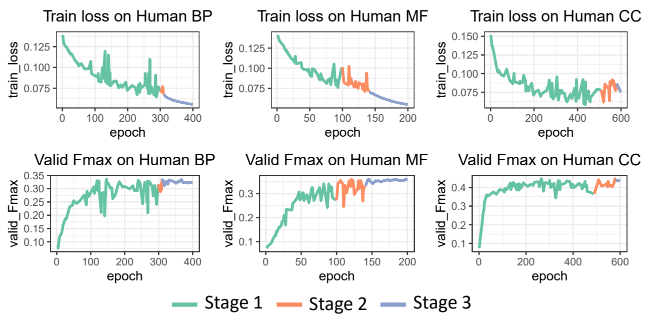

Fig. 4: Training loss and validation Fmax of DualNetGO across different training stages on the human dataset.

CFAGO, the previous SOTA model, adopts a 12-layer transformer-based autoencoder to cross-fuse a combined PPI network and protein attributes, and has much more parameters (up to 82 million) than other models (Supplementary Table S5). DualNetGO utilizes the same graph encoder for processing PPI networks and thus shares the same preprocessing time with CFAGO. The long running time for TransformerAE is due to the choice of a large number of epochs in the original paper, but in practice a much smaller number of epochs such as 500 is applicable. Furthermore, TransformerAE is not the only option for DualNetGO, with GAE and node2vec among the most time-efficient graph embedding algorithms. CFAGO's much deeper model and the fine-tuning strategy also contribute to the long training time. Graph2GO, which performs well on mouse MF, is the second time consuming model in preprocessing PPI data, due to the exhaustive sequence alignments between any two sequences out of about 20,000 sequences and the choice of using 100 epochs to train the GAE. Overall, recent models are more memory-friendly, except for Mashup, which is implemented with Matlab and runs completely on the CPU, and deepNF, which simultaneously processes seven PPI networks and uses a larger batch size.

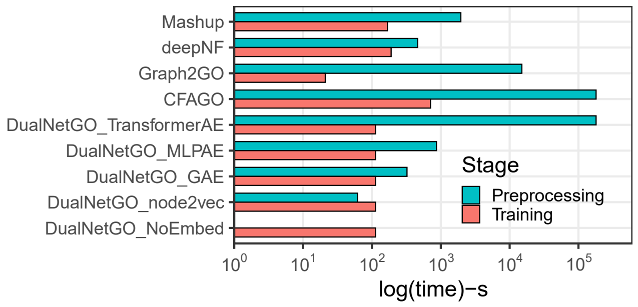

Fig. 5: Comparison of preprocessing and training time across models.

Theoretically there are $\sum_{i=1}^{N_f} \binom{|M|}{i}$ combinations. Assuming each combination needs 100 epochs to train if enumerating all combinations to determine the best one, the total training time will be intractable (given $|M|$=8 and N_f=4, the total number of epochs is 16200). However, in DualNetGO only one combination is sampled in each epoch of stage 1 and 10 combinations in each epoch for stage 2. The total number of training epochs is $E1 + 10 * E2 + E3$. According to our possible hyperparameter settings (Supplementary Table S8) the maximum number of weight updates for the classifier is 1410 epochs. We acknowledge that the final subset calculated by DualNetGO is may not the optimal one due to the sampling nature in stage 1 and 2. As training DualNetGO is a stochastic process, a careful choice of the hyperparameters of the epochs in stages 1 and 2 may be necessary to approach the optimal solution, as suggested by another dual-network model that deals with heterophilic graph data (Maurya *et al.*, 2022). Fortunately, the search of hyperparameters costs little time, and DualNetGO is still more efficient in determining a suitable combination of features than enumerating all possibilities.

3.4 Embedding-centric setting versus evidence-centric setting for DualNetGO

We denote the previous search space for searching across different PPI networks given a specific graph embedding method (TransformerAE) as the **embedding-centric** setting. We call another way of utilizing DualNetGO model the **evidence-centric** setting, which searches across different graph embedding algorithms given a PPI network from a specific type of evidence (Figure **??**). As an extension to the DualNetGO, we also implement the model in the evidence-centric setting using only the combined PPI network and evaluate the performance on both human and mouse dataset. Results (Supplementary Table S3) show that the evidence-centric setting further improves Fmax on BP and MF on human, and BP on mouse. Performance gains are also seen on m-AUPR of human BP and mouse MF, M-AUPR on mouse CC. Collectively, DualNetGO with embedding-centric and evidence-centric settings produces the top-2 results in almost all metrics in all datasets except in mouse MF where Graph2GO generally performs better. We also evaluate the use of other graph embedding methods for the embedding-centric model and other PPI networks for the evidence-centric model. The best performance and the corresponding parameters can be found in Supplementary Table S6,S7.

3.5 Analysis of features selected for predicting protein functions

As DualNetGO can determine a suitable subset of PPI for protein function prediction, the chosen combination reflects the importance of different PPI networks. As for the embedding-centric model, one embedding method is used and generates the subset for each of the three GO functional categories. In total, we implement five different graph embedding algorithms, and then we count the occurrences of each PPI network in these experiments. As shown in Figure 6a, textmining provides the most valuable information in Biological Process (BP) prediction for both human and mouse, and Cellular Component (CC) is more related to protein attributes (denoted as **feature**). CC's close relation to protein attributes is not surprising because the protein attributes used in this study include Pfam domain and subcellular location, and subcellular location could cause information leakage due to its close relation to CC. In the evidence-centric model, the number of graph embeddings chosen as features is counted across different PPI networks. In Figure 6b we observe that TransformerAE generally performs better than the other methods, demonstrating its superior performance. However, the simpler deep learning autoencoder, MLPAE, is also effective on extracting PPI information for MF prediction. Search space for features can be found in Supplementary Table S9.

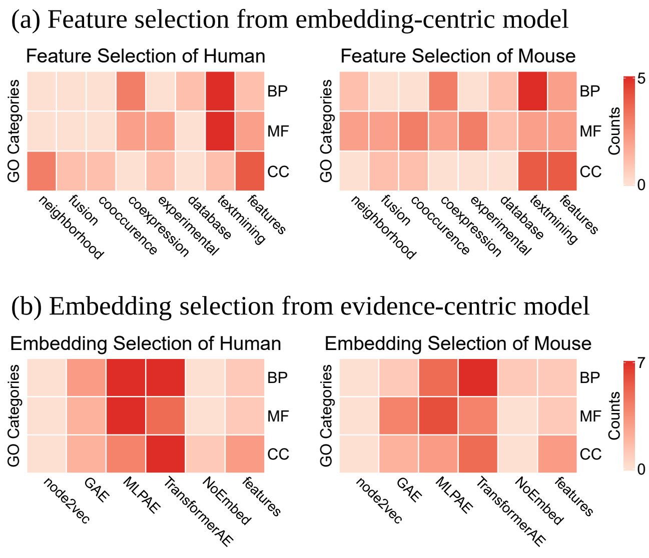

Fig. 6: Feature importance across different settings of DualNetGO.

4 Discussion

DualNetGO's ability to select and integrate multi-source features from different types of PPI networks is a major step forward in addressing the challenge of effectively utilizing heterogeneous PPI network data. The results of this study demonstrate that DualNetGO achieves state-of-the-art performance on protein function prediction in comparison to other single-species, PPI network-based methods. The model's intelligent PPI network selection strategy is necessary for achieving this level of performance and is insensitive to the choice of graph embedding methods, which makes it a versatile framework for dealing with multi-modal data when additional information such as embeddings from protein language models, knowledge graphs and 3D structures of proteins are available. Our comprehensive study evaluating the effects of different graph embedding methods on different PPI networks for protein function prediction provides valuable insight for future research. However, one limitation of applying PPI network-based models is it is confined to proteins in the network. Any newly identified protein that lacks information in the PPI network will be hard to predict its function based on the current network. In addition, different data filtering strategies may affect the performance of models dramatically. Depending on the degree of the dependent or hierarchical relationships among GO terms and the number of proteins that a term must include, or even a subset of functions that are of special interest, different numbers of proteins and GO terms are considered in different studies. Such differences are tolerated as different models may have their own use scenarios (Jiang *et al.*, 2016). In this paper, we consider a relatively small range of GO terms to demonstrate the importance for the handling of different PPI networks, and have not tested DualNetGO in the CAFA competition setting, which includes proteins from over 20 species and over 5000 GO terms. According to previously published studies such as (Gligorijević *et al.*, 2018), training a deep learning model on the CAFA dataset took about 30-40 minutes.

5 Conclusion

In this study, we demonstrate that carefully and efficiently utilizing the heterogeneous PPI network data is necessary to achieve better performance on protein function prediction. The superior performance of DualNetGO in comparison with other methods attributes to the efficiency of using a dual network structure to intelligently select features from different sources. The versatility of DualNetGO is reflected by including different feature matrices such as those from different graph embedding methods, or from potential sequence embeddings and 3D structural information. Integrating different graph embedding methods is also important as they capture different topological information from different PPI networks that each has unique density, connectivity and clustering pattern. How the performance of graph embedding methods on protein function prediction is related to the properties of different PPI networks, which PPI evidence to pay more attention to are both opening questions for future exploration.

References

Aleksander, S., Balhoff, J., Carbon, S., Cherry, J., Drabkin, H., Ebert, D., Feuermann, M., Gaudet, P., and Harris, N. (2023). The gene ontology knowledgebase in 2023. *Genetics*, **224**(1), iyad031.

Cao, Y. and Shen, Y. (2021). TALE: Transformer-based protein function annotation with joint sequence-label embedding. *Bioinformatics*, **37**(18), 2825–2833. Type: Journal Article.

Cho, H., Berger, B., and Peng, J. (2016). Compact integration of multi-network topology for functional analysis of genes. *Cell systems*, **3**(6), 540–548.

Fan, K., Guan, Y., and Zhang, Y. (2020). Graph2go: a multi-modal attributed network embedding method for inferring protein functions. *GigaScience*, **9**(8), giaa081.

Gligorijević, V., Barot, M., and Bonneau, R. (2018). deepnf: deep network fusion for protein function prediction. *Bioinformatics*, **34**(22), 3873–3881.

Gligorijević, V., Renfrew, P. D., Kosciolek, T., Leman, J. K., Berenberg, D., Vatanen, T., Chandler, C., Taylor, B. C., Fisk, I. M., Vlamakis, H., *et al.* (2021). Structure-based protein function prediction using graph convolutional networks. *Nature communications*, **12**(1), 3168.

Gong, Q., Ning, W., and Tian, W. (2016). Gofdr: a sequence alignment based method for predicting protein functions. *Methods*, **93**, 3–14.

Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864.

Hamp, T., Kassner, R., Seemayer, S., Vicedo, E., Schaefer, C., Achten, D., Auer, F., Boehm, A., Braun, T., Hecht, M., *et al.* (2013). Homology-based inference sets the bar high for protein function prediction. In *BMC bioinformatics*, volume 14, pages 1–10. Springer.

Jiang, Y., Oron, T. R., Clark, W. T., Bankapur, A. R., D'Andrea, D., Lepore, R., Funk, C. S., Kahanda, I., Verspoor, K. M., Ben-Hur, A., *et al.* (2016). An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome biology*, **17**(1), 1–19.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., *et al.* (2021). Highly accurate protein structure prediction with alphafold. *Nature*, **596**(7873), 583–589.

Kipf, T. N. and Welling, M. (2016). Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*.

Kulmanov, M. and Hoehndorf, R. (2021). DeepGOPlus: improved protein function prediction from sequence. *Bioinformatics*, **37**(8), 1187. Type: Journal Article.

Kulmanov, M., Khan, M. A., Hoehndorf, R., and Wren, J. (2018). DeepGO: predicting protein functions from sequence and interactions using a deep ontology-aware classifier. *Bioinformatics*, **34**(4), 660–668.

Lai, B. and Xu, J. (2022). Accurate protein function prediction via graph attention networks with predicted structure information. *Briefings in Bioinformatics*, **23**(1), bbab502.

Luck, K., Kim, D.-K., Lambourne, L., Spirohn, K., Begg, B. E., Bian, W., Brignall, R., Cafarelli, T., Campos-Laborie, F. J., Charloteaux, B., *et al.* (2020). A reference map of the human binary protein interactome. *Nature*, **580**(7803), 402–408.

Maurya, S. K., Liu, X., and Murata, T. (2022). Not all neighbors are friendly: Learning to choose hop features to improve node classification. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 4334–4338, New York, NY, USA. Association for Computing Machinery.

Milenković, T. and Pržulj, N. (2008). Uncovering biological network function via graphlet degree signatures. *Cancer informatics*, **6**, CIN–S680.

Oliveira, G. B., Pedrini, H., and Dias, Z. (2023). Temprot: protein function annotation using transformers embeddings and homology search. *BMC bioinformatics*, **24**(1), 1–16.

Pandey, G., Manocha, S., Atluri, G., and Kumar, V. (2012). Enhancing the functional content of protein interaction networks. *arXiv preprint arXiv:1210.6912*.

Radivojac, P., Clark, W. T., Oron, T. R., Schnoes, A. M., Wittkop, T., Sokolov, A., Graim, K., Funk, C., Verspoor, K., Ben-Hur, A., *et al.* (2013). A large-scale evaluation of computational protein function prediction. *Nature methods*, **10**(3), 221–227.

Ridnik, T., Ben-Baruch, E., Zamir, N., Noy, A., Friedman, I., Protter, M., and Zelnik-Manor, L. (2021). Asymmetric loss for multi-label classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 82–91.

Szklarczyk, D., Gable, A. L., Nastou, K. C., Lyon, D., Kirsch, R., Pyysalo, S., Doncheva, N. T., Legeay, M., Fang, T., Bork, P., *et al.* (2021). The string database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets. *Nucleic acids research*, **49**(D1), D605–D612.

Szklarczyk, D., Kirsch, R., Koutrouli, M., Nastou, K., Mehryary, F., Hachilif, R., Gable, A. L., Fang, T., Doncheva, N. T., Pyysalo, S., *et al.* (2023). The string database in 2023: protein–protein association networks and functional enrichment analyses for any sequenced genome of interest. *Nucleic acids research*, **51**(D1), D638–D646.

Uniprot (2023). Uniprot: the universal protein knowledgebase in 2023. *Nucleic Acids Research*, **51**(D1), D523–D531.

Varadi, M., Anyango, S., Deshpande, M., Nair, S., Natassia, C., Yordanova, G., Yuan, D., Stroe, O., Wood, G., Laydon, A., *et al.* (2022). Alphafold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic acids research*, **50**(D1), D439–D444.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, **30**.

Wang, S., You, R., Liu, Y., Xiong, Y., and Zhu, S. (2023). Netgo 3.0: Protein language model improves large-scale functional annotations. *Genomics, Proteomics & Bioinformatics*.

Wu, Z., Guo, M., Jin, X., Chen, J., and Liu, B. (2023). Cfago: cross-fusion of network and attributes based on attention mechanism for protein function prediction. *Bioinformatics*, **39**(3), btad123.

Yao, S., You, R., Wang, S., Xiong, Y., Huang, X., and Zhu, S. (2021). Netgo 2.0: improving large-scale protein function prediction with massive sequence, text, domain, family and network information. *Nucleic acids research*, **49**(W1), W469–W475.

You, R., Zhang, Z., Xiong, Y., Sun, F., Mamitsuka, H., and Zhu, S. (2018). GOLabeler: improving sequence-based large-scale protein function prediction by learning to rank. *Bioinformatics*, **34**(14), 2465–2473. Type: Journal Article.

Zhou, N., Jiang, Y., Bergquist, T. R., Lee, A. J., Kacsoh, B. Z., Crocker, A. W., Lewis, K. A., Georghiou, G., Nguyen, H. N., Hamid, M. N., *et al.* (2019). The cafa challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome biology*, **20**(1), 1–23.