OXFORD

Briefings in Bioinformatics, 2022, **23**, 1–9

https://doi.org/10.1093/bib/bbab502

Problem Solving Protocol

Accurate protein function prediction via graph attention networks with predicted structure information

Boqiao Lai and Jinbo Xu

Corresponding author: Jinbo Xu, Toyota Technological Institute at Chicago, Chicago, IL 60637, USA. E-mail: jinboxu@gmail.com

Abstract

Experimental protein function annotation does not scale with the fast-growing sequence databases. Only a tiny fraction (<0.1%) of protein sequences has experimentally determined functional annotations. Computational methods may predict protein function very quickly, but their accuracy is not very satisfactory. Based upon recent breakthroughs in protein structure prediction and protein language models, we develop GAT-GO, a graph attention network (GAT) method that may substantially improve protein function prediction by leveraging predicted structure information and protein sequence embedding. Our experimental results show that GAT-GO greatly outperforms the latest sequence- and structure-based deep learning methods. On the PDB-mmseqs testset where the train and test proteins share <15% sequence identity, our GAT-GO yields Fmax (maximum *F*-score) 0.508, 0.416, 0.501, and area under the precision-recall curve (AUPRC) 0.427, 0.253, 0.411 for the MFO, BPO, CCO ontology domains, respectively, much better than the homology-based method BLAST (Fmax 0.117, 0.121, 0.207 and AUPRC 0.120, 0.120, 0.163) that does not use any structure information. On the PDB-cdhit testset where the training and test proteins are more similar, although using predicted structure information, our GAT-GO obtains Fmax 0.637, 0.501, 0.542 for the MFO, BPO, CCO ontology domains, respectively, and AUPRC 0.662, 0.384, 0.481, significantly exceeding the just-published method DeepFRI that uses experimental structures, which has Fmax 0.542, 0.425, 0.424 and AUPRC only 0.313, 0.159, 0.193.

Keywords: Machine Learning, Gene Ontology, Protein Function Prediction, Deep Learning, Graph Attention Networks

Introduction

High throughput sequencing technology has yielded an explosive number of sequences, but only a tiny fraction of them have experimentally determined functional annotations [1]. There is a dire need for fast and accurate protein function annotation tools for the community to study the growing sequence databases [2–4]. Many computational methods have been developed to annotate protein functions based on primary sequences [5–9], protein family and domain annotations [10–12], protein–protein interaction (PPI) networks [7, 9, 10], and other hand-crafted features [8, 9, 13]. Critical Assessment of Functional Annotation (CAFA), a community-driven benchmark effort for automated protein function annotation, has shown that integrative prediction methods that combine multiple information sources usually outperform sequence-based methods [2–4]. Sequence-based methods use sequence similarity to transfer functional information and thus, do not work well on novel sequences that are not similar to any annotated sequences [2, 4, 14]. Domain and family annotations as well as PPI information are useful for function prediction, but they are often missing or incomplete for the vast majority of unannotated sequences [12, 15]. In addition, structure-based methods such as local surface match [16–19] have been successfully applied to protein function and PPI inference with high-resolution structure data based on binding site characterization.

Proteins acquire their function by folding into certain 3-dimensional structures *in vivo [20, 21]*. Two structurally similar proteins may share similar functions even with dissimilar sequences [22–25]. That is, purely sequence-based approaches may not work well in transferring functions between structural homologs. To bridge the gap between sequence and function, it is crucial to develop methods that can directly utilize structural information for function prediction. Methods that leverage protein structure databases such as Funfam and DeepFRI [15, 26] have shown promising results in structure-based protein function annotation. Although only a very small percentage of proteins have experimental structures, recent breakthroughs in protein contact and structure prediction [27–29] allow us to generate accurate structure information for a large portion of proteins, which can be used for large-scale automated protein function annotations.

Deep learning such as convolutional neural networks (CNN) [30] and residual neural networks (ResNet) [31] has been widely adopted by the computational biology community and showed immense success in some

Boqiao Lai is a Ph.D. student at Toyota Technological Institute at Chicago. His research focuses on developing machine learning tools to model and analyze biological sequences.

Jinbo Xu is a Professor at Toyota Technological Institute at Chicago. His research lies in the interface of machine learning, optimization and computational biology.

Received: August 14, 2021. **Revised:** October 13, 2021. **Accepted:** November 2, 2021

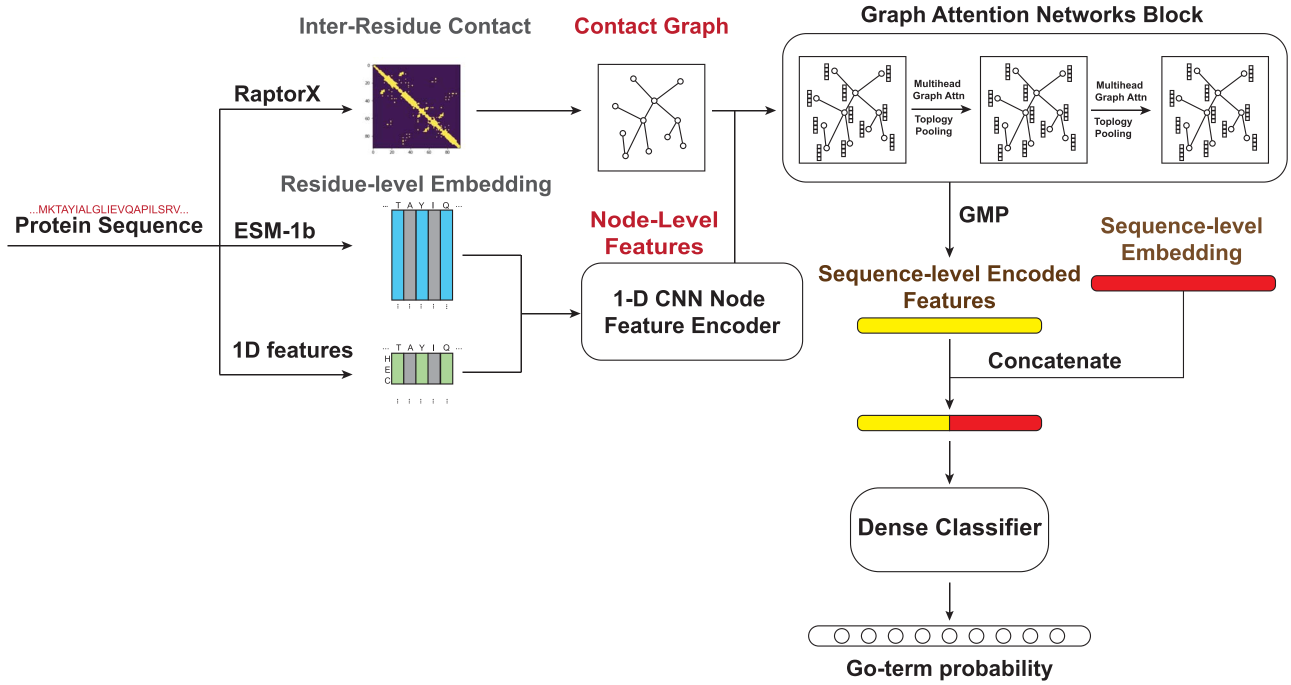

Figure 1. Overall architecture of GAT-GO: Input sequences are first processed into 1-D features (SA/SS/PSSM) and fed into a 1-D CNN feature encoder to produce node-level feature embeddings. GAT combines the inter-residue contact graphs with node-level feature embeddings to generate a protein-level feature vector, which is then used by a dense classifier to predict GO term probability.

areas such as profiling epigenomic landscapes from DNA sequences [32–35] and protein folding [27, 28, 36]. Graph convolutional networks (GCNs) [37] are able to learn representation from arbitrarily structured graph input [38, 39]. Graph attention network (GAT) [40] is a type of graph neural network (GNN) that performs graph convolution with self-attention [41]. GAT and GNN are used to model gene expression and study protein structure refinement [42, 43]. Unsupervised protein sequence models are used to capture inter-residue relationships for protein contact prediction and have become an integral part of many protein structure prediction methods [27, 28, 36]. Recently, deep protein language models have been developed [44–46] to encode context and global information of a protein for downstream tasks such as stability prediction and contact prediction.

To leverage predicted structure information and protein embeddings for function prediction, we have developed a GAT-based method called GAT-GO that uses RaptorX [47] to predict structure information of a protein and Facebook's ESM-1b [44] to generate its embedding. GAT-GO outperforms traditional homology-based algorithms such as BLAST [48] and previous deep learning methods [6], even when test proteins have low sequence identity with training proteins. Two recent studies [15, 49] have explored GCN and protein embeddings for protein function prediction, but they show limited improvement over sequence-only methods. Our method differs from the GCN method DeepFRI [15] as follows. We use GAT [40] instead of conventional GCN. GAT enhances model capacity by allowing flexible node feature aggregation through self-attention. In addition, we use topological pooling [50] to enable more efficient downsampling that improves model generalizability. Moreover, GAT-GO uses predicted inter-residue contacts for both training and test, while DeepFRI uses some native contact graphs in training.

Results

GAT-GO: predicting protein function via GATs

As shown in Figure 1, GAT-GO integrates protein sequence representations and predicted inter-residue contact graphs using a CNN-based feature encoder and a GAT-based graph encoder. GAT allows capturing interactions among spatially close residues, which may be missed in sequence-only methods. GAT-GO consists of three major modules: (i) a CNN that takes sequential features and residue-level sequence embedding as input to produce per residue feature representation. (ii) A GAT that takes a predicted contact graph and the CNN-generated representation vector as input. Each GAT layer is followed by an attention-based topological pooling layer [50] to perform topology-aware downsampling. A global pooling layer is used at the end of GAT to extract protein-level representation. (iii) A dense classifier that predicts the probability of functional annotations from the GAT-generated representation and protein-level sequence embedding.

We evaluate the performance across all three gene ontology domains (MFO, BPO, CCO) using both protein-centric metric F_{max} (Maximum F-score) and GO term-centric metric area under precision-recall curve ($AUPRC$). F_{max} measures how well a method retrieves relevant function annotations across all tested proteins and

Table 1. F_{max} and *AUPRC* of the tested methods on the PDB-cdhit dataset. DeepFRI(Native) indicates the native contact map of a test protein is used

	F_{max}			AUPRC		
Model	**MFO**	**BPO**	**CCO**	**MFO**	**BPO**	**CCO**
Naive	0.156	0.244	0.318	0.075	0.131	0.158
BLAST	0.498	0.400	0.398	0.120	0.120	0.163
1D CNN(DeepGO)	0.359	0.295	0.420	0.368	0.210	0.302
DeepFRI(Native)	0.542	0.425	0.424	0.313	0.159	0.193
GAT-GO(Ours)	0.633	0.492	0.547	0.660	0.381	0.479

AUPRC assesses the precision-recall tradeoff across all GO terms predicted.

GAT-GO improves protein function prediction

We test GAT-GO on the PDB-cdhit dataset and compare it to sequence-only methods including BLAST and the standard Naive baseline used in the CAFA benchmark [2, 4] and a just-published structure-based GCN method DeepFRI, which uses contact graphs extracted from native structures. We have also implemented a 1D CNN method to represent the state-of-the-art sequence-only deep learning method DeepGO [6]. As shown in Table 1, GAT-GO vastly outperforms BLAST, 1D CNN and DeepFRI across all three gene ontology domains. GAT-GO has F_{max} 0.637, 0.510, 0.542 on the MFO, BPO, CCO ontology domains, respectively, whereas BLAST has F_{max} 0.497, 0.399, 0.390. Despite using predicted contacts and being trained on a much smaller dataset, GAT-GO has AUPRC 0.662, 0.384, 0.481 on the MFO, BPO, CCO ontology domains, respectively, substantially better than DeepFRI that has AUPRC 0.313, 0.159, 0.193. The higher the *AUPRC* value, the more confidence we have in the predicted protein-function pairs. GAT-GO has a much better *AUPRC* than the competing methods, indicating the function annotations predicted by GAT-GO are much more reliable than the other methods across all predicted GO terms.

GAT-GO's performance with respect to sequence identity

Past studies often use time-gated temporal datasets to evaluate model generalizability to previously unannotated sequences [6–8]. However, this temporal evaluation approach may have similar sequences in the training and test data and thus, inflate the test result. Some recent studies use sequence identity to split the training and test data [15, 49] and thus, may provide a more accurate view of model generalizability to novel sequences. This evaluation practice is widely adopted by other fields such as protein structure prediction [51]. We generate training and test datasets (See Methods; Data split and processing) using five different sequence identity thresholds (15%, 25%, 35%, 45%, 55%) and compare GAT-GO with BLAST and two sequence-based methods implemented by ourselves: (i) a 1D CNN method that predicts protein function from primary sequence only; (ii) a 1D ResNet method that predicts protein function from both sequential features and protein sequence embeddings. Neither 1D CNN nor 1D ResNet uses predicted inter-residue contact graphs. Here we cannot compare GAT-GO with DeepFRI since the latter does not have pretrained models for a specific sequence identity threshold. As shown in Figure 2, the performance of almost all the test methods increases with respect to sequence identity, but BLAST performs badly at low sequence identity zones. GAT-GO consistently outperforms the other methods on all three ontology domains regardless of sequence identity. When the training and test sequences share ≤15% sequence identity, GAT-GO has F_{max} 0.501, 0.406, 0.508 for the MFO, BPO, CCO ontology domains, respectively, and AUPRC 0.427, 0.253, 0.411, much better than 1D ResNet (F_{max} 0.408, 0.331,0.450 and *AUPRC* 0.294, 0.184, 0.318) and 1D CNN (F_{max} 0.154, 0.161, 0.270 and AUPRC 0.048, 0.067, 0.112). The sequence-only methods 1D CNN and BLAST do not fare well at low sequence identity due to a lack of explicitly shared sequence patterns between the training and test sequences. On the other hand, structural features like predicted inter-residue contact graphs and protein sequence embedding can drastically improve protein function prediction for novel sequences.

Predicted contacts and sequence embedding improve protein function prediction

To thoroughly investigate the contributions of individual factors, we evaluate the performance of four deep models (1D ResNet, GCN and GAT) with different feature combinations on the PDB-cdhit testset. As shown in Table 2, both 1D ResNet and GAT-GO can leverage protein sequence embeddings to improve function prediction. For example, compared to 1D ResNet using only primary sequence, 1D ResNet using both primary sequence and protein-level embeddings may improve F_{max} by 0.204, 0.136, 0.109 for the MFO, BPO, CCO ontology domains, respectively. This observation is consistent with [49] that found protein sequence embeddings encode useful information for protein function prediction and can significantly enhance model performance.

To investigate the contribution of protein-level embeddings to the performance of GAT-GO, we compare two GAT models. One uses sequential features and residue-level embeddings as input and the other uses both protein-level and residue-level embeddings on top of sequential features. Using protein-level embeddings improves F_{max} by 0.070, 0.020, 0.052, and *AUPRC* by 0.097, 0.083, 0.101 for the MFO, BPO, CCO ontology domains, respectively. To demonstrate the improvement of the GAT-based model architecture over the GCN-based architecture developed by [15], we compare a GAT model that uses one-hot encoded primary sequences, residue-level embeddings, and predicted inter-residue contacts with a GCN model that uses the same set of features. The

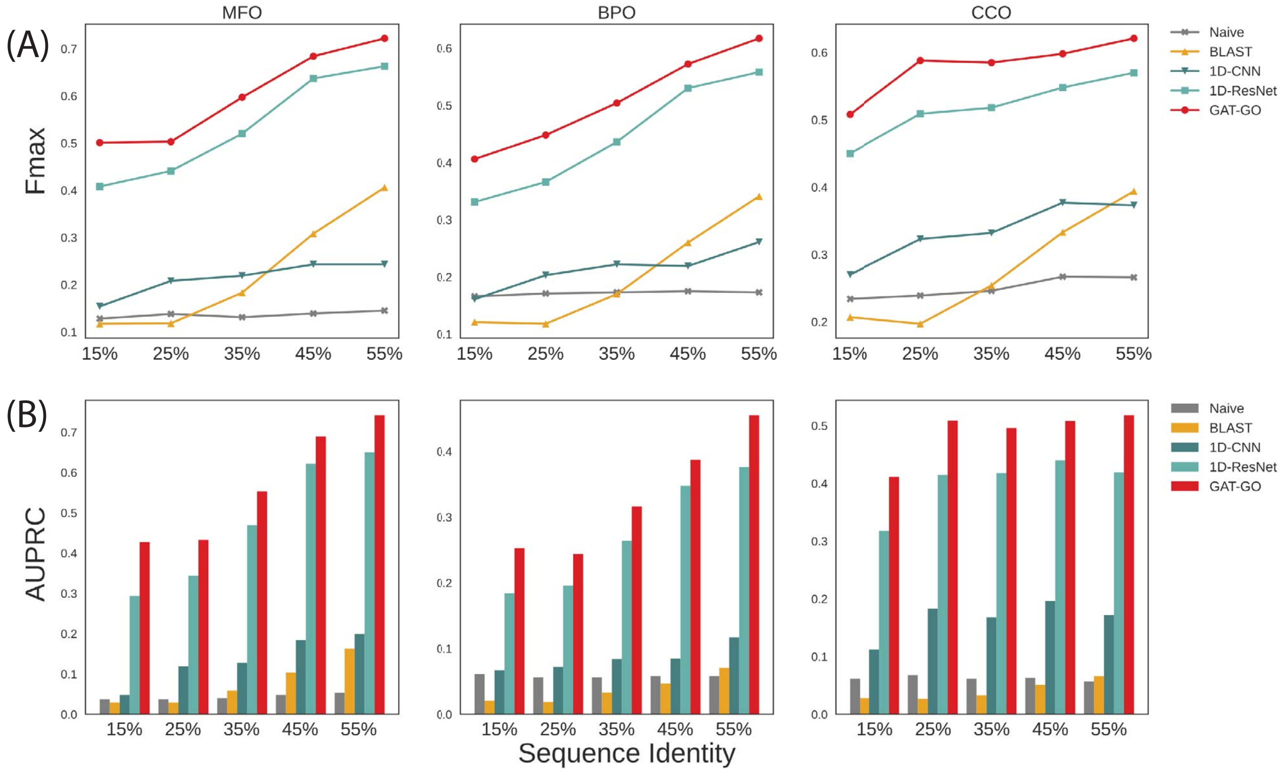

Figure 2. (A) Fmax and (B) AUPRC performance comparison on the PDB-mmseq dataset across different sequence identity thresholds.

GAT model yields F_{max} of 0.551, 0.472, 0.490, and *AUPRC* of 0.558, 0.289, 0.364 whereas the GCN model yields F_{max} of 0.459, 0.443, 0.461, and *AUPRC* of 0.408, 0.245, 0.315 for the MFO, BPO, CCO ontology domains, respectively.

To study how much predicted inter-residue contacts may help, we compare the ResNet model and the GAT model, both of which use sequential features and protein-level and residue-level embeddings. But the GAT model uses predicted contact graphs whereas the ResNet model does not. The GAT model has F_{max} 0.637, 0.501, 0.542, and AUPRC 0.662, 0.384, 0.481 for the MFO, BPO, CCO ontology domains, respectively, whereas the ResNet model has F_{max} 0.548, 0.416, 0.500 and AUPRC 0.559, 0.293, 0.393. See Supplementary Tables S2 and S3 for more detailed comparisons. This result suggests that predicted structural information can improve protein function prediction in addition to protein sequence embeddings.

Explicit structural information amends function interpretation in longer sequences and high-specificity GO terms

To better understand how predicted inter-residue contacts enhance protein function prediction, we compare GAT-GO with ResNet, both using the same set of input features except that ResNet does not use predicted contact graphs. To measure how GAT-GO improves function prediction accuracy, we calculate the precision difference between GAT-GO and ResNet on each tested sequence and study its relationship with sequence length. The precision improvement by GAT-GO is positively correlated with sequence length. Their spearman's correlation coefficient is 0.312, 0.259, 0.221 for the MFO, BPO, CCO ontology domains, respectively, as shown in Figure 3. This result implies that the longer the sequence, the greater impact predicted structural information has on prediction accuracy. This is because GAT (with predicted contacts) may capture interactions among residues that are well separated along the primary sequence. In contrast, sequence-based methods such as 1D ResNet often focus on local sequence patterns and are not good at capturing interactions of residues that are far away from each other along the primary sequence. Studies have shown that explicitly modeling long-range residue interactions can greatly improve performance in various tasks [42] in addition to functional prediction.

Using predicted structural information may also improve the prediction accuracy of GO terms with high specificity. We measure the specificity of one GO term using information content (IC) (See Methods; Evaluation metrics). A Go term with high IC is more specialized and thus, rarer in occurrence. On the other hand, a GO term of low IC represents a broad function that is more common in annotations. As shown in Figure 4, GO terms with high IC (IC > 12) can benefit from the inclusion of inter-residue contact graphs.

Discussion

In this study, we have presented GAT-GO, a structure-based deep learning method that integrates predicted

Table 2. F_{max} and *AUPRC* of different methods with different feature combinations on the PDB-cdhit dataset

		F_{max}			AUPRC		
Model	Input features	MFO	BPO	CCO	MFO	BPO	CCO
1D ResNet	Primary sequence	0.337	0.284	0.369	0.334	0.188	0.267
	Primary sequence and protein-level embeddings	0.541	0.416	0.478	0.541	0.289	0.386
	Sequential features andpretrained embeddings[1]	0.548	0.416	0.500	0.559	0.293	0.393
GCN	Predicted contact andPrimary sequence andresidue-level embeddings	0.459	0.443	0.461	0.408	0.245	0.315
GAT-GO	Predicted contact andprimary sequence andresidue-level embeddings	0.551	0.472	0.490	0.558	0.289	0.364
	Predicted contact and sequential features andresidue-level embeddings	0.567	0.481	0.495	0.565	0.301	0.380
	Predicted contact and sequential features and pretrained embeddings	0.637	0.501	0.542	0.662	0.384	0.481

[1] Pretrained embeddings refer to both residue-level and protein-level pretrained embeddings (See Methods; Input features).

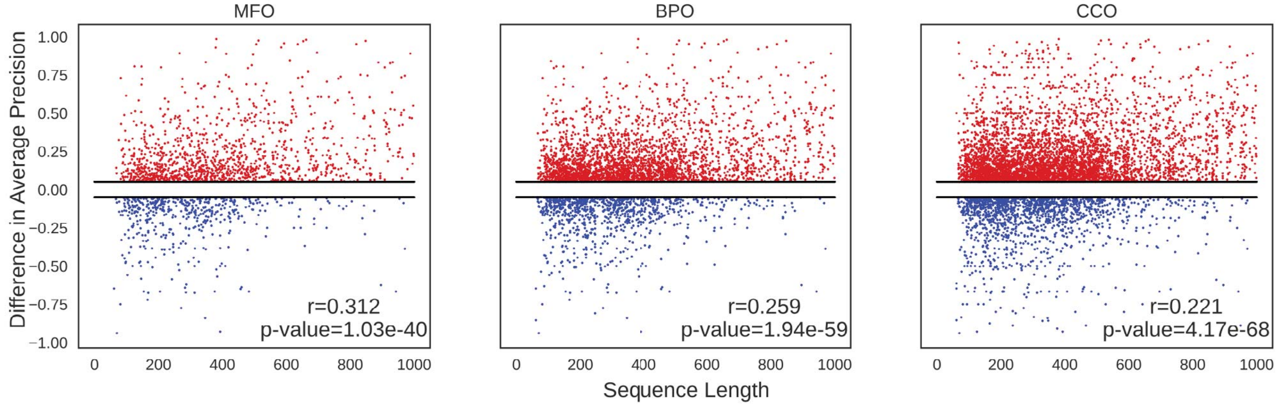

Figure 3. Performance improvement of GAT-GO over the ResNet-based sequence method versus sequence length. Sequences with increased average precision (red) and decreased average precision (blue) are plotted against the length of the sequences. Spearman's correlation coefficients for the MFO, BPO, CCO ontology domains are 0.312, 0.259, 0.221 with *P*-value of 1.03e−40, 1.94e−50, 4.17e−68, respectively. Where the two-sided *P*-values are calculated under the student's t distribution with *n*−2 degree of freedom.

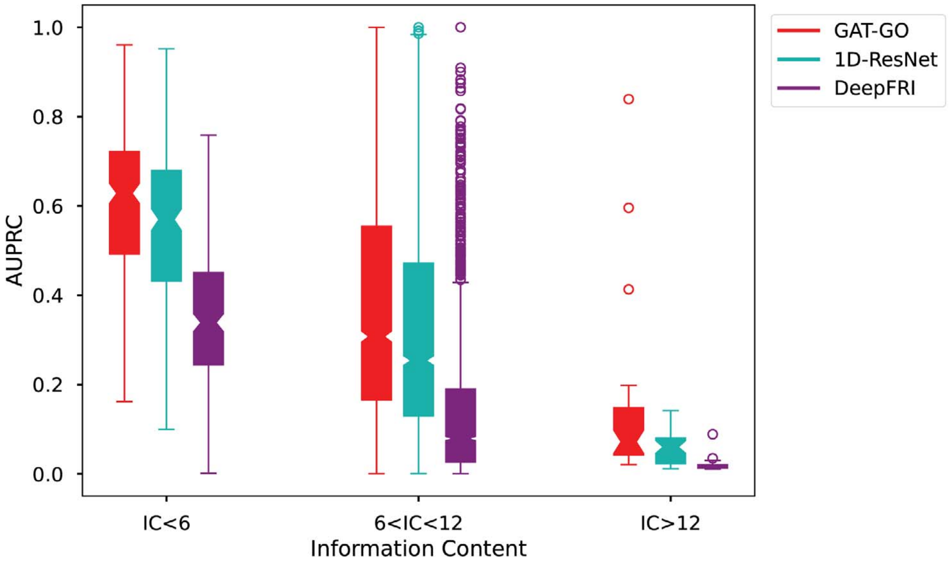

Figure 4. Distribution of AUPRC score on GO Terms covered by GAT-GO over different specificity levels (Information Content) for different methods: GAT-GO (red), 1D-ResNet (green), and DeepFRI (purple). The box representation has the center as the median, upper and lower edges are the interquartile range and the whiskers are the data range.

inter-residue contact graphs, protein embedding and sequential features for protein function prediction. By integrating predicted contacts and protein embedding through GATs, our method may accurately and efficiently map protein sequences to function annotation at a large scale, especially when the test sequences are not similar to the training sequences. By combining sequential features, protein embeddings and inter-residue contact graphs, GAT-GO may predict protein function from both local and global information. In contrast, sequence-based methods cannot make use of predicted structure information and thus, are not good at handling a test sequence that is not similar to any training sequences.

In this study, we did not use the very large metagenomic databases in generating multiple sequence alignments for inter-residue contact prediction since it needs much more computing power to search through these databases. That is to say, our prediction accuracy may further improve if we detect more sequence homologs for our test proteins from the metagenomic databases. Although GAT-GO can predict protein function from predicted structures, we benchmark our methods on the proteins with experimental structures to fairly compare with other structure-aware methods. To make full use of protein structure prediction, we plan to train and test our method using some larger protein function prediction datasets such as the one used in CAFA [2].

GAT-GO outperforms existing function prediction methods by utilizing high-resolution structure information and high-capacity pretrained protein embeddings. Our experiment shows that protein embeddings, predicted contact graphs, and the GAT network architecture all are important for improving function prediction. To further improve structure-based protein function annotation, instead of using predicted contact graphs, we may use predicted inter-residue distance graphs or 3D structure coordinates as the structure representation. Other network-based protein features such as PPI networks may also be used.

Methods

Data split and processing

We download the data used by DeepFRI [15] at https://github.com/flatironinstitute/DeepFRI. We denote it as PDB-cdhit since the train/test split is generated by cd-hit [52] with 40% sequence identity. Each sequence is annotated with memberships of 2752 GO terms across three ontology domains. CD-hit [52] is often used to remove redundancy between training and test proteins, but its greedy clustering method and local-alignment-based sequence similarity calculation can still lead to redundancy between the training and test set and thus, inflate the test results [52]. To fix this, we employ another sequence clustering tool MMseqs to generate a new dataset denoted as PDB-mmseqs. MMSeqs uses a nongreedy clustering scheme and profile-based alignment method to ensure there is no higher than desired sequence identity across clusters [53]. To split data by sequence identity, we use MMseqs to cluster all protein sequences with a given sequence identity threshold, and then select a representative sequence from each cluster to build a seed sequence pool. Then we split the seed sequence pool uniformly at random with an 8/2 ratio to form the train and test seed sets. The final train/test sequences are determined by including sequences from the respective clusters of the train/test seeds. Validation sequences are generated by sampling 10% of the sequences from the training set uniformly at random. We generate five different data splits with five different sequence identity thresholds (15%, 25%, 35%, 45%, 55%). That is, in each dataset, no training and testing protein share higher than the respective sequence identity. See Supplementary Table S1 for details of the datasets.

To measure the IC for an individual Go term, we compute the Shannon Information from its frequency in the training set. See Supplementary Section 5 for more details.

Input features

Sequential features

One-hot encoding of the primary protein sequence is the most commonly used input feature for sequence-based methods. We encode a sequence with 25 different symbols including the 20 common amino acid symbols and five compound ambiguous symbols. We also use position specific scoring matrix (PSSM) as sequence profiles derived from the profile HMM generated by HHblits [54] with *E*-value = 0.001 and uniclust30 dated in August 2018. We use secondary structure and solvent accessibility predicted by RaptorX-Property [55] from PSSM. In the article, sequential features are referred to as the combination of one-hot encoded primary sequence, sequence profile, secondary structure annotations and solvent accessibility annotations.

Predicted protein structure information

To obtain inter-residue contact graphs, we predict protein Cb-Cb distance using RaptorX [27] and define inter-residue contact probability as the probability of the Cb-Cb distance <8 Å. To build the contact graph, we add an edge between two residues if and only if they have >50% predicted contact probability. We have evaluated performance with respect to different contact probability cutoffs in Supplementary Table S4.

Protein embeddings

To obtain residue-level sequence embeddings, we use ESM-1b [44], a deep protein language model trained on over 250 million protein sequences with the UniRef UR50/50 database [56]. ESM-1b embeddings have been successfully applied in protein engineering tasks such as guided directed evolution [57]. Since the number of functionally annotated protein sequences is very limited, we hope the ESM-1b embeddings can leverage the huge protein sequence database by exposing the functionally annotated proteins to a much larger protein landscape. We generate the protein-level embedding from the residue-level embedding by averaging across all residue positions. The residue-level embeddings are incorporated in the sequential features and the protein-level embeddings are integrated along with the learned representation into the dense classifier.

Graph attention networks

GNN is a powerful tool for extracting information from arbitrarily structured graph data [58]. GCN uses spectral convolution on the graph Fourier domain to aggregate neighboring representation for feature learning [37]. In this study, we use the first-order approximation of the spectral convolution: $H^{l+1} = \hat{D}^{-1/2}\hat{A}\hat{D}^{-1/2}H^{l}\Theta^{l+1.}$Where $\hat{D}$ is the degree matrix such that $\hat{D}_{ii} = \sum_{l=1}^{L}\hat{A}_{il}$, and $\hat{A}$ is the adjacency matrix of the input contact graph with self-loops, i.e. $\hat{A} = A + I_L$ where I_L is an $L \times L$ identity matrix. H^l is the graph representation at layer l, and Θ is the trainable weight of the neural network.

GAT aim to improve the flexibility and capacity of the graph spectral convolution by employing a self-attention mechanism to parameterize the feature aggregation process [40]. Compared to GCN which aggregates neighboring features with fixed weights determined by the degrees of the respective nodes, GAT uses learnable self-attention-based weights. First, GAT calculates the pairwise importance scores for all node pairs as $e_{ij} = a(\Theta h_i, \Theta h_j)$ where ${h^l}_i$ is the hidden representation for node i at layer l and Θ^l is the trainable weight for layer l. We use a single-layer feedforward neural network as our attention function, i.e. $a(z_i, z_j) = LeakyReLU(w^T[z_i \| z_j])$ where w is the trainable attention weight vector and $\|$ denotes concatenation. Graph structure information is then injected by using the mask attention score $\alpha_{ij} \forall j \in N_i$ on top of the pairwise importance score with softmax as $\alpha_{ij} = \frac{\exp(a(e_{ij}))}{\Sigma_{k \in N_i} \exp(a(e_{ik}))}$, where N_i is the neighborhood of node i. The hidden representation of each node is then updated with ${h_i}^{l+1} = \sigma(\Sigma_{j \in N_i} \alpha_{ij} \Theta {h_j}^{l})$where $\sigma(\cdot) = ReLU(\cdot)$ is the activation function.

GAT-GO network architecture

GAT-GO consists of three major components: (i) a CNN-based sequence feature encoder. It takes three sequence features as input and encodes them into residue-level feature vectors. The three input features are the one-hot encoding of primary sequences, sequence profile and residue-level sequence embedding derived from the protein language model. A 512-channel CNN is used to encode each input feature and then summed to generate the final encoding. The CNN-based encoder can capture locally sensitive patterns from the sequential features. Local patterns such as active sites and sequence motifs are important in inferring protein functions. (ii) A GAT-based graph encoding module comprising four GAT layers with 512, 512, 1024, 1024 hidden channels and 12 attention heads with a 0.5 dropout rate. Following each GAT layer, there is a topological pooling layer [50] that computes node-level self-attention scores with graph convolution and the top 50% of the nodes are retained as input to the next GAT layer. A global mean pooling layer is then used to pool residue-level feature encoding into sequence-level feature encoding. This GAT-based graph encoder helps our model to integrate long-range inter-residue interactions and residue-level features to generate a more informative sequence-level representation for function prediction. The number of layers and hidden channels are decided by a hyperparameter sweep over multiple combinations. (iii) A dense classifier that predicts protein function jointly from the learned sequence-level feature by GAT-GO and the sequence-level protein embeddings. By default, GAT-GO uses RaptorX-predicted contact graphs and protein embeddings generated by ESM-1b.

Evaluation metrics

We evaluate models by two main metrics, protein-centric F_{max} and GO-term-centric AUPRC. F_{max} is the maximum F_1 score across all prediction thresholds in the range of [0,1] with a step size of 0.01. The AUPRC is a summarization of the precision-recall curve by calculating the weighted mean of the precision achieved at each threshold. We use AUPRC to measure the precision-recall tradeoff in a label imbalanced environment. See Supplementary Section 2 for more details.

Competing methods

DeepFRI

It is a recently published GCN-based method that uses autoregressive protein embeddings and contact graphs derived from 3-D structures [15]. We obtain DeepFRI's predictions following the instruction at https://github.com/flatironinstitute/DeepFRI using experimentally solved structures downloaded from https://www.rcsb.org/. In contrast, our GAT-GO is a GAT-based method with multilevel topological pooling that uses transformer-based residue- and protein-level embeddings. GAT-GO is trained on inter-residue contact graphs predicted by RaptorX [47] and thus, does not rely on experimentally solved structures. For the purpose of comparing the DeepFRI architecture with different feature combinations, we also implemented a 3-layer GCN model as described in [15] that we could train and test on our custom input features.

Blast

Following the protocol used in the CAFA benchmark [4], we obtain the BLAST score for each sequence by first running blastp against the corresponding training database. For each hit, we transfer the corresponding GO terms with the sequence identity as the predicted probability. When multiple hits contain the same GO term, the maximum sequence identity is retained.

CNN and ResNet

We have implemented a 1D CNN model with 16 parallel one-layer convolution operations where each convolution has kernel sizes of [8,16, …,128] and 512 filters.

This CNN is very similar to the state-of-the-art sequence-only deep learning method DeepGOplus [6]. We have also implemented a ResNet model with 21 convolution layers that predict protein function from both primary amino acid sequences and sequential features. See Supplementary for more details.

Model training

We train our models with binary cross-entropy as the loss metric and the AdamW optimizer [59] with a learning rate of 1e−4 for 30 epochs. We implemented all models in the Pytorch and Pytorch geometric library [60, 61]. A validation set is used to employ an early stopping scheme with patience of 10 epochs. All models are trained with an NVIDIA 2080Ti GPU on a Linux machine.

Key Points

- Our method GAT-GO outperformed state-of-the-art structure-based and sequence-based methods in predicting protein functions by a large margin even when test proteins are not similar to the training proteins.
- GAT-GO leveraged both predicted protein structure information and protein language models for function prediction.
- Graph Attention Networks is a powerful deep model that may jointly encode protein structure and sequence information.

Supplementary data

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

Funding

This work was supported by funding from National Institute of Health under award R01GM089753 to Jinbo Xu.

References

1. Consortium, U., Others. UniProt: the universal protein knowledgebase. *Nucleic Acids Res* 2018;**46**:2699.
2. Zhou N, Jiang Y, Bergquist TR, *et al.* Others: the CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome Biol* 2019;**20**:1–23.
3. Jiang Y, Oron TR, Clark WT, *et al.* Others: an expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biol* 2016;**17**:1–19.
4. Radivojac P, Clark WT, Oron TR, *et al.* Others: a large-scale evaluation of computational protein function prediction. *Nat Methods* 2013;**10**:221–7.
5. Fa R, Cozzetto D, Wan C, *et al.* Predicting human protein function with multi-task deep neural networks. *PLoS One* 2018;**13**:e0198216.
6. Kulmanov M, Hoehndorf R. DeepGOPlus: improved protein function prediction from sequence. *Bioinformatics* 2020;**36**:422–9.
7. Kulmanov M, Khan MA, Hoehndorf R. DeepGO: predicting protein functions from sequence and interactions using a deep ontology-aware classifier. *Bioinformatics* 2018;**34**:660–8.
8. You R, Huang X, Zhu S. DeepText2GO: improving large-scale protein function prediction with deep semantic text representation. *Methods* 2018;**145**:82–90.
9. You R, Zhang Z, Xiong Y, *et al.* GOLabeler: improving sequence-based large-scale protein function prediction by learning to rank. *Bioinformatics* 2018;**34**:2465–73.
10. Duong DB, Gai L, Uppunda A, *et al.* Annotating gene ontology terms for protein sequences with the transformer model. *bioRxiv* 2020. https://doi.org/10.1101/2020.01.31.929604.
11. Cai Y, Wang J, Deng L. SDN2GO: an integrated deep learning model for protein function prediction. *Front Bioeng Biotechnol* 2020;**8**:391.
12. Cozzetto D, Minneci F, Currant H, *et al.* FFPred 3: feature-based function prediction for all gene ontology domains. *Sci Rep* 2016;**6**: 1–11.
13. 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.
14. Huberts DHEW, van der Klei IJ. Moonlighting proteins: an intriguing mode of multitasking. *Biochim Biophys Acta* 2010;**1803**: 520–5. https://doi.org/10.1016/j.bbamcr.2010.01.022.
15. Gligorijevic V, Renfrew PD, Kosciolek T, *et al.* Structure-based function prediction using graph convolutional networks. *Nature communications* 2021;**1**:1–14.
16. Tseng YY, Dundas J, Liang J. Predicting protein function and binding profile via matching of local evolutionary and geometric surface patterns. *J Mol Biol* 2009;**387**:451–64. https://doi.org/10.1016/j.jmb.2008.12.072.
17. Tseng YY, Liang J. Estimation of amino acid residue substitution rates at local spatial regions and application in protein function inference: a Bayesian Monte Carlo approach. *Mol Biol Evol* 2006;**23**:421–36. https://doi.org/10.1093/molbev/msj048.
18. Zhao J, Dundas J, Kachalo S, *et al.* Accuracy of functional surfaces on comparatively modeled protein structures. *J Struct Funct Genomics* 2011;**12**:97–107. https://doi.org/10.1007/s10969-011-9109-z.
19. Binkowski TA, Freeman P, Liang J. pvSOAR: detecting similar surface patterns of pocket and void surfaces of amino acid residues on proteins. *Nucleic Acids Res* 2004;**32**:W555–8. https://doi.org/10.1093/nar/gkh390.
20. Mitchell AL, Attwood TK, Babbitt PC, *et al.* Others: InterPro in 2019: improving coverage, classification and access to protein sequence annotations. *Nucleic Acids Res* 2019;**47**: D351–60.
21. Dawson NL, Lewis TE, Das S, *et al.* CATH: an expanded resource to predict protein function through structure and sequence. *Nucleic Acids Res* 2017;**45**:D289–95.
22. Krissinel E. On the relationship between sequence and structure similarities in proteomics. *Bioinformatics* 2007;**23**:717–23.
23. Rost B. Twilight zone of protein sequence alignments. *Protein Eng* 1999;**12**:85–94.
24. Brenner SE, Chothia C, Hubbard TJP, *et al.* Understanding protein structure: using scop for fold interpretation. *Methods Enzymol* 1996;**266**:635–43.
25. Holm L, Sander C. Mapping the protein universe. *Science* 1996;**273**:595–602.
26. Das S, Lee D, Sillitoe I, *et al.* Functional classification of CATH superfamilies: a domain-based approach for

protein function annotation. *Bioinformatics* 2016;**32**:2889. https://doi.org/10.1093/bioinformatics/btw473.

27. Wang S, Sun S, Li Z, *et al.* Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLoS Comput Biol* 2017;**13**:e1005324.
28. Senior AW, Evans R, Jumper J, *et al.* Others: improved protein structure prediction using potentials from deep learning. *Nature* 2020;**577**:706–10.
29. Xu J. Distance-based protein folding powered by deep learning. *Proc Natl Acad Sci U S A* 2019;**116**:16856–65. https://doi.org/10.1073/pnas.1821309116.
30. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. *Adv Neural Inf Process Syst* 2012;**25**:1097–105.
31. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. *In: Proceedings of the IEEE conference on computer vision and pattern recognition*. Manhattan, New York, NY, US. 2016 pp. 770–8.
32. Lai B, Qian S, Zhang H, *et al.* X.: predicting epigenomic functions of genetic variants in the context of neurodevelopment via deep transfer learning. *bioRxiv* 2021. https://doi.org/10.1101/2021.02.02.429064.
33. Zhou J, Troyanskaya OG. Predicting effects of noncoding variants with deep learning–based sequence model. *Nat Methods* 2015;**12**: 931–4.
34. Kelley DR, Snoek J, Rinn JL. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Res* 2016;**26**:990–9.
35. Grønning AGB, Doktor TK, Larsen SJ, *et al.* DeepCLIP: predicting the effect of mutations on protein–RNA binding with deep learning. *Nucleic Acids Res* 2020;**48**:7099–118.
36. Yang J, Anishchenko I, Park H, *et al.* Improved protein structure prediction using predicted interresidue orientations. *Proc Natl Acad Sci* 2020;**117**:1496–503.
37. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. *arXiv preprint* 2016 *arXiv*:1609.02907.
38. Bruna J, Zaremba W, Szlam A, *et al.* Spectral networks and locally connected networks on graphs. *arXiv*1312.6203 [cs.LG] 2013.
39. Henaff M, Bruna J, LeCun Y. Deep convolutional networks on graph-structured data. *arXiv*:1506.05163 2015.
40. Veličković P, Cucurull G, Casanova A, *et al.* Graph attention networks. *arXiv preprint*arXiv:1710.10903 2017.
41. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint*arXiv:1409.0473 2014.
42. Karbalayghareh A, Sahin M, Leslie CS. Chromatin interaction aware gene regulatory modeling with graph attention networks. *bioRxiv* 2021. https://doi.org/10.1101/2021.03.31.437978.
43. Jing, X., Xu, J.: *Fast and effective protein model refinement by deep graph neural networks*, https://www.biorxiv.org/content/10.1101/2020.12.10.419994v1.abstract, (2020). https://doi.org/10.1101/2020.12.10.419994.
44. Rives A, Meier J, Sercu T, *et al.* Others: biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc Natl Acad Sci* 2021;**118**:e2016239118; doi: 10.1073/pnas.2016239118.
45. Alley EC, Khimulya G, Biswas S, *et al.* Unified rational protein engineering with sequence-only deep representation learning. *Nature methods* 2019;**16.12**:1315–322.
46. Madani A, McCann B, Naik N, *et al.* Progen: language modeling for protein generation. *arXiv*: 2020.03497 [q-bio.BM].
47. Xu J, McPartlon M, Li J. Improved protein structure prediction by deep learning irrespective of co-evolution information. *Nat Mach Intell* 2021;**1–9**. https://doi.org/10.1038/s42256-021-00348-5.
48. Altschul SF, Gish W, Miller W, *et al.* Basic local alignment search tool. *J Mol Biol* 1990;**215**:403–10.
49. Villegas-Morcillo A, Makrodimitris S, van Ham R, *et al.* Unsupervised protein embeddings outperform hand-crafted sequence and structure features at predicting molecular function. *Bioinformatics* 2021;**37.2**: 162–70.
50. Lee J, Lee I, Kang J. Self-Attention Graph Pooling. In: Chaudhuri K, Salakhutdinov R (eds). *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 2019, 3734–43. *arXiv*:1904.08082.
51. AlQuraishi M. ProteinNet: a standardized data set for machine learning of protein structure. *BMC Bioinform* 2019;**20**: 1–10.
52. Fu L, Niu B, Zhu Z, *et al.* CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics* 2012;**28**: 3150–2.
53. Steinegger M, Söding J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat Biotechnol* 2017;**35**:1026–8.
54. Steinegger M, Meier M, Mirdita M, *et al.* HH-suite3 for fast remote homology detection and deep protein annotation. *BMC Bioinformatics* 2019;**20**:473. https://doi.org/10.1186/s12859-019-3019-7.
55. Wang S, Peng J, Ma J, *et al.* Protein secondary structure prediction using deep convolutional neural fields. *Sci Rep* 2016;**6**:18962. https://doi.org/10.1038/srep18962.
56. Suzek BE, Wang Y, Huang H, *et al.* UniProt consortium: UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics* 2015;**31**:926–32. https://doi.org/10.1093/bioinformatics/btu739.
57. Yang KK, Wu Z, Arnold FH. Machine-learning-guided directed evolution for protein engineering. *Nat Methods* 2019;**16**:687–94. https://doi.org/10.1038/s41592-019-0496-6.
58. Bronstein MM, Bruna J, LeCun Y, *et al.* Geometric deep learning: going beyond Euclidean data. *IEEE Signal Process Mag* 2017;**34**: 18–42. https://doi.org/10.1109/MSP.2017.2693418.
59. Loshchilov, I., Hutter, F.: *Decoupled Weight Decay Regularization*, http://arxiv.org/abs/1711.05101, (2017), *arXiv*:1711.05101.
60. Fey, M., Lenssen, J.E.: *Fast Graph Representation Learning with PyTorch Geometric*, http://arxiv.org/abs/1903.02428, (2019), *arXiv*:1903.02428.
61. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: *PyTorch: An Imperative Style, High-Performance Deep Learning Library*, *arXiv*:1912.01703. http://arxiv.org/abs/1912.01703, (2019).