Systems biology

TransformerGO: Predicting protein-protein interactions by modelling the attention between sets of gene ontology terms

Ioan Ieremie\textsuperscript{1,\ast}, Rob M. Ewing\textsuperscript{2} and Mahesan Niranjan\textsuperscript{1}

\textsuperscript{1}Vision, Learning & Control Group, University of Southampton, Southampton, SO17 1BJ, UK
\textsuperscript{2}Biological Sciences, University of Southampton, Southampton, SO17 1BJ, UK.

\textsuperscript{\ast}To whom correspondence should be addressed.

Abstract

Motivation: Protein-protein interactions (PPIs) play a key role in diverse biological processes but only a small subset of the interactions have been experimentally identified. Additionally, high-throughput experimental techniques that detect PPIs are known to suffer various limitations such as exaggerated false positives and negatives rates. The semantic similarity derived from the Gene Ontology (GO) annotation is regarded as one of the most powerful indicators for protein interactions. However, while computational approaches for prediction of PPIs have gained popularity in recent years, most methods fail to capture the specificity of GO terms.

Results: We propose TransformerGO, a model that is capable of capturing the semantic similarity between gene ontology sets dynamically using an attention mechanism. We generate dense graph embeddings for GO terms using an algorithmic framework for learning continuous representations of nodes in networks called node2vec. TransformerGO learns deep semantic relations between annotated terms and can distinguish between negative and positive interactions with high accuracy. TransformerGO outperforms classic semantic similarity measures on gold standard PPI datasets and state-of-the-art machine learning-based approaches on large datasets from S. cerevisiae and H. sapiens. We show how the neural attention mechanism embedded in the transformer architecture detects relevant functional terms when predicting interactions.

Availability and implementation: https://github.com/Ieremie/TransformerGO
Contact: ii1g17@soton.ac.uk

1 Introduction

Identifying protein-protein interactions is a major challenge in molecular biology, because it is fundamental for our understanding of biological processes and cellular activities such as metabolism, signal transduction pathways and immune response. Advances in high-throughput methods allowed the discovery of PPIs at the genome scale. However, experimental methods are time-consuming, labour-intensive and the results suffer from high false positive and negative rates. Yeast-two-hybrid (Y2H) experiments (Hu et al., 2001) report direct physical interactions and generate binary interactome network maps. The noise found in the final datasets comes primarily from the inability of the method to capture interactions between proteins that rely on intermediary proteins (protein complexes, post-translational modifications) and on expression levels. On the other hand, experiments using affinity-purification and mass spectrometry (AP-MS) (Ewing et al., 2007, Gavin et al., 2006) generate datasets complementary to the previously mentioned method by detecting interactions appearing in protein complexes. However, AP-MS is limited in its ability to detect transient interactions (Cafarella et al., 2017). Therefore, computational approaches have been developed to infer protein-protein interactions in silico.

Multiple studies approached the prediction of protein-protein interactions using various sources of information such as the
primary structure of the protein (Chen et al., 2019, Li et al., 2018, Hashemifar et al., 2018), the three-dimensional protein structure (Bepler and Berger, 2021), gene expression profiles (Chin et al., 2010) and Gene Ontology (GO) annotation (Bandyopadhyay and Mallick, 2016, Jain and Bader, 2010, Kulmanov et al., 2019, Smaili et al., 2018, Smaili et al., 2019, Zhang et al., 2020, Zhang et al., 2018).

The Gene Ontology project is a collaborative effort to annotate genes and the products of genes with useful descriptions of biological processes across multiple databases and species (Consortium, 2015). GO is composed of the ontology graph and annotation databases. The graph is structured as a directed acyclic graph (DAG) and is divided into three orthogonal sub-ontologies, cellular component (CC), biological process (BP) and molecular function (MF). Nodes inside the graph denote GO terms which are descriptions of biological concepts and the edges (‘is_a’, ‘part_of’, ‘regulates’, ‘has_part’) represent relations between GO terms (Consortium, 2015). Annotation databases contain GO terms and the gene products they annotate to. The semantic similarity in GO annotation is regarded as one of the most powerful descriptors of protein-protein interactions (Consortium, 2015, Miller et al., 2005, Patil and Nakamura, 2005). The idea behind this is that interacting protein pairs such as protein complexes interact in the same cellular location and functional modules participate in the same cellular processes or molecular functions at different times. These two types of interactions are closely related in terms of GO annotation (Zhao et al., 2020).

Multiple semantic similarity measures on GO have been proposed over the years (Resnik, 1995, Jain and Bader, 2010, Zhang et al., 2018) that predict protein-protein interactions using semantic similarity in GO annotation. However, classic semantic similarity measures are in general handcrafted and fail to fully capture the specificity of GO terms. It has also been shown that semantic similarity measures are difficult to compare and are only performing well on some datasets (Kulmanov and Hoehndorf, 2017). Depending on the downstream application, different features should be more or less relevant in defining the notion of similarity (Smaili et al., 2018). On the other hand, while machine learning approaches (Jansen et al., 2003, Rhodes et al., 2005, Stelzl et al., 2005) can be trained in a supervised fashion, the similarity is encoded as a simple feature vector indicating the common GO terms. Disregarding the structure of the ontology would not allow for a correct evaluation of proteins that have common terms but are too general (Qiao et al., 2006). Several studies apply techniques from the field of Natural Language Processing to extract dense feature vectors for GO terms (Smaili et al., 2018, Smaili et al., 2019, Zhong et al., 2019, Zhang et al., 2020, Zhao et al., 2020). We find that previous work comparing feature vectors using cosine similarity or using a fully connected neural network fails to capture deep semantic similarity between the gene ontology terms.

Inspired by previous work based on gene ontology terms and current advancements made in NLP, we propose a trainable approach called TransformerGO that predicts PPIs using information extracted from the GO graph. We apply node2vec (Grover and Leskovec, 2016) to generate dense feature vectors for gene ontology terms and then use the Transformer model (Vaswani et al., 2017) to dynamically learn a deep semantic similarity between sets of GO terms. We demonstrate that TransformerGO outperforms classic similarity measures and recent models that use a similar way of encoding the gene ontology graph. Furthermore, experiments that analyse the attention weights show how semantic similarity is learned in the decoder and provide useful visualisations that could aid future research on comparing proteins at a functional level.

Table 1. Node2vec hyper-parameters

| Parameter       | Value |
|-----------------|-------|
| Window/Neighbourhood size | 10    |
| Walk length     | 80    |
| Number of walks | 10    |
| Search bias P   | 1     |
| Search bias Q   | 1     |
| Iterations      | 10    |
| Dimensions      | 64    |

2 System and Methods

2.1 Protein embeddings

With recent research on unrestrained representation learning, new methods for creating latent representations of nodes and edges in networks have emerged. DeepWalk (Perozzi et al., 2014) uses information from truncated random walks to learn link features by extending the Skip-gram model (Mikolov et al., 2013). The network is represented as a document, and the nodes in the random walks are the equivalent of words forming sentences. Node2vec (Grover and Leskovec, 2016) is a framework that learns continuous feature representations for nodes by maximizing the log-likelihood of preserving the network neighbourhoods. Compared to the previous method, Node2vec adds flexibility when defining the neighbourhoods by using biased random walks. The model has proven successful in multi-label classification and link prediction, such as PPI (Grover and Leskovec, 2016).

We use $G = (V, E)$ to denote the Gene Ontology graph, where $V$ represents the set of GO terms and $E$ all the undirected edges named ‘is_a’ and ‘part_of’ appearing between the terms. Node2vec seeks to optimise the objective function $f$, which maximises the log-probability of observing the neighbourhood $N_v(u)$ of a node $u$, given its feature representation.

$$
\max_{f} \sum_{v \in V} \log P(N_v(u)|f(u))
$$

To create continuous feature representations for proteins, we use $S = \{GO_1, GO_2, \ldots, GO_n\}$ to denote the set of all GO terms that are annotated to a protein. Using the learned matrix defined by the function $f$ from node2vec, we replace the GO IDs with the corresponding rows, such that the resulting set would be of size $|S| \times d$.

We use the code of node2vec provided by the authors to train the model on the Gene Ontology graph. The hyperparameters used are summarised in Table 1.

2.2 The architecture of TransformerGO

We introduce a framework developed for GO based PPI prediction, capable of analysing complex relations between sets of gene ontology terms. The model proposed is based on the Transformer (Vaswani et al., 2017) which uses an attention mechanism to solve seq2seq tasks such as translation. In the recent years, multiple models have been published.
that use the Transformer architecture to achieve state-of-the-art results on natural language processing tasks: the Vanilla Encoder is used to train deep bidirectional representations from unlabelled text by conditioning on both left and right context (Devlin et al., 2018, Liu et al., 2019, Dai et al., 2019). While the Transformer architecture is designed for seq2seq tasks, with different modifications it can be used for biological prediction tasks. The model proposed by Lian, TransformerCPI (Chen et al., 2020), uses the Transformer Decoder to model the protein-compound interaction, while the Encoder is replaced by a set of convolution blocks. To the best of our knowledge, the Transformer model has not been used to predict protein-protein interactions at a functional level. Inspired by the ability of the model to capture deep connections between sequences, we developed TransformerGO to predict PPIs from sets of gene ontology terms. An overview of the model proposed is in Figure 1, where we made modifications to the attention heads and changed the final layers.

Both the Encoder and Decoder receive as input an embedded set of GO terms which are used to define the protein at a functional level. Through self-attention mechanisms, GO terms are weighted accordingly to the contribution they have as an interaction descriptor. Replacing the Masked Multi-Head Attention with a Multi-Head attention allows the model to attend to subsequent positions, a change that transfers the architecture from an autoregressive task to a classification task (Chen et al., 2020).

Given that the order of the GO terms should not be relevant in predicting interactions at the functional level, the positional encodings are not injected in our input, which enables annotated GO terms to be treated as a set.

The key component of the Transformer network is the Scaled-Dot-Product-Attention, which allows the model to focus on certain parts of the input. The attention function can be described as mapping a query and a set of key-value pairs to an output. In practice, the attention is computed on a set of queries using a matrix as follows:

$$\text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) V$$  \hspace{1cm} (2)

where $Q, K, V$ are matrices for the set of queries, keys, and values and $\sqrt{d_k}$ is a scalar factor.

Multi-Head-Attention blocks allow the model to attend at different positions and subspaces by linearly projecting the input using learned weights before performing the attention function. This mechanism enables the Decoder to focus dynamically on certain parts of the Encoder’s output to learn semantic similarities between gene ontology terms that could be important when predicting PPIs.

After extracting deep semantic similarities between GO terms, the output of the Transformer consists of an interaction sequence which has the same shape as the input of the Decoder. We apply the same methodology used in TransformerCPI (Chen et al., 2020) to transform this sequence into a final probability vector. Given the interaction sequence $\{X_1, X_2, ..., X_n\}$, we modify the volume by using a weighted sum of the attention vectors. Therefore, the input to the final Linear layer is given by Equation 3, where $\text{aff}(i) = \sum_{j=1}^{n} \text{attn}(i)_{X_j}$:

$$\text{output} = \sum_{i=1}^{n} \text{aff}(i), X_i$$  \hspace{1cm} (3)

The entire architecture is trained to optimise the Binary Cross-Entropy Loss, given the binary nature of the protein-protein interaction prediction:

$$\text{Loss} = -[y_a \log x_a + (1 - y_a) \log (1 - x_a)]$$  \hspace{1cm} (4)

where $y_a$ is the label of the class and $x_a$ a function returning the predicted probability of class “1”.

TransformerGO was implemented in PyTorch (Paszke et al., 2017) and trained using the Adam optimiser (Kingma and Ba, 2014). We reduced the size of the Transformer from 6 layers to 3 and the embedding size from 512 to 64, but we kept the number of heads to 8 as this did not increase training time. The training hyperparameters and the model settings are summarised in Table 2.

**Table 2.** TransformerGO hyper-parameters and settings

| Parameter          | Value |
|--------------------|-------|
| Embedding size     | 64    |
| Number of layers   | 3     |
| Feed Forward dimension | 64*4  |
| Learning Rate      | $1e-04$ |
| Batch size         | 32    |
| Dropout            | 0.2   |
2.3 Datasets

2.3.1 Gene ontology graph & GO annotation data

- Gene ontology graph (Ashburner et al., 2000) - a filtered version of the ontology graph is downloaded, which guarantees that the generated graph is acyclic and there are no relationships that cross the 3 GO hierarchies. The file’s release date is September 19, 2018 to ensure a fair comparison with previous work.

- GO annotation data - we adopted the files provided by Jain’s work (Jain and Bader, 2010). Electronically annotated interactions that lack manual review are included and named IEA+.

2.3.2 PPI datasets

Jain’s datasets (Jain and Bader, 2010) - contain positive and negative interactions for S. cerevisiae and H. sapiens organisms. A number of 4598 positive interactions and 2077 respectively, were retrieved using the Database of Interacting Proteins (DIP, Xenarios et al., 2000). These were further split into 3 smaller datasets (BP, MF, CC) with both proteins annotated to terms (other than root) in their respective ontologies (Table 3, 4). For S. cerevisiae, an equal number of negative interactions are generated at random by selecting protein pairs from the GO annotation file that are not part of all known yeast PPIs appearing in the iRefWeb database (45,448 yeast PPIs) (Razick et al., 2008). For H. sapiens, an equal number of negative interactions are generated at random from a pool of all possible interactions and then removing those which appear in the iRefWeb database (43,935 human PPIs).

STRING-DB datasets: To analyse the performance of TransformerGO on larger datasets, we obtained two protein interaction networks for S. cerevisiae and H. sapiens from the STRING database (Szklarczyk et al., 2016, downloaded on November 17, 2021) which account for 1,988,592 and 11,938,498 interactions. After filtering the interactions that have a score above 700 (high confidence), the S. cerevisiae dataset is composed of 120,386 interactions from 5,966 proteins and H. sapiens is composed of 252,984 interactions from 16,814 proteins. We used gene annotation files for both organisms from the STRING website along with the gene ontology graph. We filtered down annotations inferred electronically (IEA) and annotations where there is no biological data available (ND). We generated an equal number of negative interactions by randomly choosing pairs of proteins from the positive dataset that do not appear in the STRING database (Table 5).

To allow for a fair comparison with previous methods, we also train and test the TransformerGO model on a benchmark containing 420,534 human interactions and 11,937 yeast interactions (Kulmanov et al., 2021) retrieved from the STRING database. The difference between our proposed dataset and the benchmark is the use of up-to-date files and the method of generating negative interactions. We consider negative interactions, pairs of proteins for which there is no association in the STRING database, while the previous proposed benchmark considers interactions under the confidence score (700) to be negative. We also found the use of ‘mirror’ interactions where we consider both A → B and B → A interactions to be redundant for training, and we do not include them in our proposed datasets.

3 Results & Discussion

3.1 The order of the input sequence

Zhang’s work (Zhang et al., 2020) proposes a new method called protein2vec that differs from classic semantic similarity measures between gene ontology terms. Here, a protein is characterised by a vector according to the annotated GO terms. A network embedding algorithm is applied to generate dense feature vectors for each GO term, then the resulting sequence of embeddings becomes input to a long short-term neural network. To model the interaction between two proteins, the outputs from the LSTM network are fed into a feed-forward neural network which outputs an interaction probability.

We argue that the annotated GO terms should not be modelled as a sequence due to the nature of the information they hold. Gene ontology terms are simple labels that should not act as words in a sentence. We re-implemented the model proposed by Zhang et al. (2020) and performed experiments by changing the order of the terms in the sequence. As shown in Table 6, there is a decrease in accuracy for all subsets, showing that the LSTM is modelling the interaction based on the order of the GO terms in the sequence. Therefore, some improvement in accuracy comes from the model’s bias over the ordering of the input.

3.2 TransformerGO performance on Jain’s datasets

We choose two classic semantic similarity measures, TCSS and Resnik (Jain and Bader, 2010, Resnik, 1995), to compare performance on Jain’s datasets along with two more recent approaches, HVSM and protein2vec (Zhang et al., 2018, Zhang et al., 2020). We use the receiving operating characteristics ROC curve, which is a method used widely to measure the
performance of binary classifiers. ROC is defined by plotting the true-positive rate (sensitivity) $TPR = \frac{TP}{TP+FN}$ against the false-negative rate (1-specificity) $FPR = \frac{FP}{TP+FP}$, where $TP$ stands for the number of true-positives, $FP$ for false positives and $TN$ for true negatives. We report the AUC (area under the curve) as a measurement of performance (Fawcett, 2006). Note that semantic similarity measures do not require a training phase of the algorithm, therefore the validation results are reported on the entire dataset. To fairly compare our work with these measures, we perform a 5-fold-cross-validation on $S$. cerevisiae and $H$. sapiens and report the average AUC values.

In Table 7 and 8, the results of the proposed methods are shown, with the TransformerGO performance marked in bold. We can observe that classic semantic similarity measures along with the Hierarchical Vector Space Model perform relatively poor compared to machine learning models. TransformerGO improves the performance on average with 5% across all subsets, indicating that semantic similarities methods fail to capture the true meaning of the Gene Ontology graph and use it to predict PPIs. While protein2vec uses a similar way of encoding GO terms as TransformerGO, the LSTM is designed to capture features from sequences where the order of the ‘words’ have a deep meaning. Furthermore, protein2vec only uses a feed-forward network to capture the semantic similarity features between two gene products. The decoder in our approach focuses its attention dynamically on the output of the encoder to capture interaction features of the two sets of GO terms.

### 3.3 TransformerGO performance on STRING-DB datasets

Taking into consideration that the Transformer network requires a large training corpus, and it is easy to overfit on small datasets (Qiu et al., 2020), we trained our model on two considerable larger datasets retrieved from STRING database (Szklarczyk et al., 2016). We randomly split the datasets into 80% training, 20% testing, and use 20% of the training dataset as a validation set to choose the best performing model. This would allow for a better analysis of the TransformerGO performance on an external test set and the effects of a larger training corpus.

We benchmark our model against recent work on generating feature vectors for gene ontology terms and proteins. Oto2vec (Smaili et al., 2018) is a method that learns representations for classes in an ontology and biological entities annotated with these classes. To generate feature vectors for GO terms and proteins, it trains a skip-gram model on the set of all axioms appearing in the gene ontology. Opa2vec (Smaili et al., 2019) extends this method by including meta-data from the ontology (natural language statements) into the training corpus and using transfer learning from biomedical literature. El Embeddings (Kulmanov et al., 2019) embeds classes by minimising a set of loss functions that preserve the axioms inside an ontology. In Table 9, we observe that the classic semantic similarity such as Resnik outperforms some unsupervised methods. El Embeddings has good performance on the task of PPI prediction by exploiting more axioms from the ontology. Adding all the GO terms generated by node2vec together, combined with cosine similarity outperforms opa2vec. This suggests that most of the information contained in the Gene Ontology can be captured by exploiting the graph neighbourhoods generated by the ‘is_a’ and ‘part_of’ relations. As it has been shown before (Smaili et al., 2018, Smaili et al., 2019), the main advantage of generating embeddings for gene ontology terms is that it allows a downstream model to learn semantic similarity in a supervised way. TransformerGO improves AUC values on the $S$. cerevisiae and $H$. sapiens STRING benchmark with up to 7% compared to the second-best method. Using a simple feed-forward neural network with 3 layers (200, 400 and 200 neurons) on top of the embeddings performs better than cosine similarity, but it fails to match the performance of our model. This again validates the importance of the Decoder in our network, which is able to capture deep semantic similarities between GO terms.

Semantic similarity has been used to capture similarities between gene products from different perspectives. In the case of GO-based semantic similarity, the cellular component terms build up a context which could be used to validate physical interactions and localisation-dependant functions or processes. Furthermore, the biological process aspect of the GO hierarchy is used to validate physical interactions and localisation-dependent functions or processes. accordingly, the biological process aspect of the GO hierarchy is used to validate physical interactions and localisation-dependent functions or processes. Similarly to previous work (Xu et al., 2008, Zhang and Tang, 2016), we find that the performance for BP and CC is better than MF. However, proteins in the dataset have on average less MF annotations than BP and CC. This raises the question if annotation size has an effect on performance.

### Protein interaction networks

The hotel interaction networks have been used to infer properties and functions of proteins through a “guilt by association” principle, which

### Table 7. AUC values on H. sapiens from a 5-fold cross-validation experiment

| IEA+ | IEA- |
|------|------|
| CC   | BP   | MF   | CC   | BP   | MF   |
| CTCSS-max | 0.82 | 0.92 | 0.85 | 0.80 | 0.89 | 0.80 |
| Resnik-max | 0.81 | 0.92 | 0.84 | 0.80 | 0.89 | 0.80 |
| HVSM  | 0.84 | 0.93 | 0.88 | -    | -    | -    |
| Protein2vec | 0.85 | 0.87 | 0.82 | 0.85 | 0.89 | 0.82 |
| TransformerGO | 0.936 | 0.933 | 0.939 | 0.912 | 0.927 | 0.912 |

HVSM does not report the results on the datasets without electronically inferred annotations. The standard deviation of TransformerGO is less than 0.01 across all datasets.

### Table 8. AUC values on S. cerevisiae from a 5-fold cross-validation experiment

| IEA+ | IEA- |
|------|------|
| CC   | BP   | MF   | CC   | BP   | MF   |
| CTCSS-max | 0.83 | 0.89 | 0.75 | 0.83 | 0.89 | 0.73 |
| Resnik-max | 0.83 | 0.89 | 0.75 | 0.83 | 0.89 | 0.73 |
| HVSM  | 0.83 | 0.90 | 0.74 | -    | -    | -    |
| Protein2vec | 0.91 | 0.89 | 0.88 | 0.90 | 0.90 | 0.87 |
| TransformerGO | 0.927 | 0.929 | 0.924 | 0.921 | 0.926 | 0.926 |

The best AUC values are marked in bold. Node2vec, COS adds all the embeddings together and predicts PPI using cosine similarity. Node2vec, NN trains a simple feed-forward neural network on top of the embeddings.

### Table 9. AUC values on S. cerevisiae and H. sapiens - STRING benchmark

| S. cerevisiae | H. sapiens |
|---------------|------------|
| Resnik        | 0.87       | 0.89       |
| Oto2vec       | 0.80       | 0.77       |
| Opa2vec       | 0.81       | 0.87       |
| El Embeddings | 0.93       | 0.90       |
| Node2vec, COS | 0.847      | 0.845      |
| Node2vec, NN  | 0.952      | 0.958      |
| TransformerGO | 0.963      | 0.974      |

The best AUC values are marked in bold. Node2vec, COS adds all the embeddings together and predicts PPI using cosine similarity. Node2vec, NN trains a simple feed-forward neural network on top of the embeddings.
states that proteins that are associated (interact) are more likely to have similar functions (Oliver, 2000). Recent studies (Gillis and Pavlidis, 2011, Gillis and Pavlidis, 2012) show that function can be extracted from interactions networks without using “guilt” and only using the node degree as input. This proved to be successful because genes that have more interacting partners are more likely to have multiple functions (GO terms). To explore if TransformerGO is biased towards proteins with more annotations (multifunctional), we train and test the model using datasets filtered at different annotation sizes. We consider the annotation size to be the size of the GO-set containing all the GO terms annotated to both interacting proteins. In Table 11, we observe that in the most conservative case where interactions are defined by only up to 6 terms, there is a considerable drop in performance. This suggests that interactions between proteins with few known functions are difficult to predict, due to a decrease in chance of finding similar or semantic similar GO terms. For interactions with a higher number of GO terms associated, there is a clear improvement in AUC values. However, it is interesting to see that the performance drops when the model is trained and evaluated using interactions which contain proteins with a high number of annotations (above 20 for S. cerevisiae and 30 for H. sapiens). This suggests that the model is not biased towards proteins with a high number of annotations, but capturing semantic similarity between proteins with a few annotations proves to be a difficult task.

### 3.4 Model interpretation

One desirable outcome along a semantic similarity value between two proteins would be to determine what GO terms are more important and in what manner they relate to each other when predicting an interaction. For example, when modelling the interaction between two binding proteins, the cellular component should be more important than their ability to regulate other proteins (Smalls et al., 2018). This is one of the reasons why negative interactions are usually generated to include proteins from different cellular components or biological processes (Bandyopadhyay and Mullick, 2016). Our understanding of the success behind the Transformer networks is limited, but recent work in the field of NLP brings light in the interpretation of attention (Kovaleva et al., 2019, Rogers et al., 2020). In the field of Bioinformatics, attention has been used to show how it captures the folding structure of proteins and target binding sites (Vig et al., 2020). We follow this work to analyse the attention weights of TransformerGO and observe the patterns learned when modelling semantic similarity between sets of gene ontology terms. We define an indicator function $f$ that takes as input interacting proteins $A, B$ and returns 1 if the pair of GO terms determined by the index pair $(i, j)$ is present in the interaction, and zero otherwise. We compute the attention matrix $M$ that aggregates the attention weights over all the GO term pairs in the ontology and all the interactions in dataset $X$ as follows:

$$M_{i,j} = \sum_{(A,B) \in X} f(A,B) \alpha_{i,j}(A,B) / \sum_{(A,B) \in X} f(A,B)$$

where $\alpha_{i,j}$ denotes the attention from GO term $i$ to GO term $j$ in the input interaction.

We use a model trained on the *H. Sapiens* STRING-DB benchmark dataset and analyse the attention using Equation 5. Compared to the training corpus used in NLP models, the dataset is relatively small, therefore we only analyse the average attention over all layers and attention-heads. In Figure 2 we can observe that the attention of Cellular Component and Molecular Function terms are highly dependent on the background frequency in the dataset. This could suggest that the model is considering frequent terms a good indicator of interaction. However, there is a considerable discrepancy in GO term frequency and attention values for Biological Process terms: the Decoder allows the model to capture complex semantic relations between gene ontology terms, disregarding information that comes in the form of noise which does not contribute towards interaction prediction. On the right side of Figure 2, the top 25 terms that have attention values greater than the background frequency are shown. It is interesting that the model is paying more attention to Cellular Component terms which define complex like structures. In other words, the interaction is easier to predict if we know that both proteins are part of the same complex. Similar for Biological Process, a few terms which focus on the binding activity are being picked up. One important question that arrives is why does the model disregard most of the Molecular Function information, considering that training on this part of the ontology alone showed top performance. Our intuition suggests that TransformerGO finds it ‘easier’ to model terms which define the interaction in broader terms than focusing on terms which have a more granular view of a possible association between proteins.

| Table 10: AUC values on S. cerevisiae and H. sapiens - STRING dataset |
|-----------------------------|-----------------------------|
| **S. cerevisiae** | **H. sapiens** |
| CC | BP | MF | ALL | CC | BP | MF | ALL |
| 0.941 | 0.966 | 0.935 | 0.973 | 0.924 | 0.948 | 0.900 | 0.958 |

Each column represents the AUC value of TransformerGO trained and evaluated on a filtered dataset with ALL or only GO terms from a specific sub-ontology.

| Table 11: AUC values on S. cerevisiae and H. sapiens - STRING dataset |
|-----------------------------|-----------------------------|
| **S. cerevisiae** | **H. sapiens** |
| GO-set size | AUC | GO-set size | AUC |
| (0, 6) | 0.822 | (0, 6) | 0.711 |
| (0, 10) | 0.948 | (0, 10) | 0.840 |
| (10, 20) | 0.973 | (10, 30) | 0.953 |
| (20, ∞) | 0.963 | (30, ∞) | 0.951 |

AUC values of TransformerGO trained and evaluated on a filtered dataset containing only interactions where the aggregated number of GO terms is within a specific range.
an increase in similarity score to other terms. To evaluate the effects of depth when it comes to modelling gene ontology terms, we consider the depth as the longest path to the root of the ontology and analyse the attention accumulated per level. In Figure 3 we can see that the attention values are similar with the background frequency, suggesting that the model is not focusing on certain parts of the gene ontology graph to extract information about protein interactions. Another interesting aspect is that TransformerGO ‘pays’ the same attention to all Biological Process terms, regardless of the depth at which they appear in the gene ontology graph. This shows again that BP terms are more granular and when used in isolation, they provide more information about protein interactions, therefore, achieving top performance. We also observe that in the case of CC and MF terms, there is high attention at the root of ontology, highlighting the fact that terms which define complexes act as ‘clique-like’ structures in contrast to other terms (Gillis and Pavlidis, 2012). In other words, GO terms which have high learnability in the network data can act in ‘reverse’ and be used to predict interactions. CC terms which define complexes and MF terms which define binding processes.

To further analyse what the model is paying attention to when predicting the interactions, we map the positive interactions from the H. sapiens dataset (String benchmark) to the BioGRID database (Oughtred et al., 2019). This allows us to compare which GO terms are more significant in predicting the interaction depending on the type of experiment that determined it. A number of 21 923 interactions are labelled as ‘High Throughput’ and 11 729 as ‘Low Throughput’, while the rest of the interactions were not found in the BioGRID database. In Figure 4 we computed the attention heatmap using the self-attention weights when the model gets as input interactions labelled as ‘High Throughput’. We further filtered down the heatmap to contain only the top 30 GO terms that have high attention values. We can observe that a diagonal pattern appears similar to previous work of analysing BERT’s self-attention weights (Kovaleva et al., 2019). This could suggest the model’s inability to capture semantic similarity at an early stage when the GO terms are analysed in isolation (the other protein in the interacting pair is not seen). However, there are similarities captured within the Cellular Component ontology between terms which are one level apart: ‘Extracellular region’ → ‘Extracellular space’ and ‘Endoplasmic reticulum’ → ‘Endoplasmic reticulum membrane’. One common motif that appears across the heatmaps is the importance of gene ontology terms that define binding functions, e.g., ‘DNA, RNA, protein, enzyme binding’. This could suggest that proteins which have a binding behaviour are more likely to interact.

The heatmap in Figure 5 drawn from the source-attention weights highlights specific semantic relations between GO terms which are part of different sub-ontologies. Such important terms appear as vertical line patterns, e.g., ‘(BP) Neutrophil degranulation’, ‘(CC) Cytoplasm’. Similar patterns appear in the attention heatmaps for yeast, e.g., ‘(CC) Nucleus’, ‘(MF) structural constituent of ribosome’ (Supplementary Figures 1–4).

The difference in important GO terms when predicting interactions determined by ‘Low Throughput’ methods compared to ‘High Throughput’ is minimal (see Figure 6). In both cases, Cellular Component terms have high attention values even if the terms are close to the root of the ontology. While the background frequency of CC terms is higher than BP and MF terms, the model has the ability of disregarding unimportant terms. Therefore, the presence of large number of CC terms with high attention is not necessarily due to a background frequency. One explanation could be that the model views Cellular Component terms as valuable because proteins that are localised in the same cellular component are more likely to interact. This is similar to previous work (Shin et al., 2009) that demonstrated that observed frequencies of co-location do not arise by chance.

4 Conclusions

Multiple methods that compute semantic similarity between gene ontology terms have been proposed in the recent years, but the choice of appropriate use still depends on the application, as the performance can vary for different applications. Thus, they fail to answer which method is the most appropriate measure given the biological question (Mazandu et al., 2017). Furthermore, classic similarity measures have been shown to be biased due to the number of annotations, difference in annotation size and depth of specificity of annotation classes when predicting protein-protein interactions (Kulmanov and Hoehndorf, 2017). We proposed TransformerGO, a method that uses recent advancement from Deep Learning to predict PPIs using network information extracted
from the gene ontology graph. One clear advantage of our model compared to semantic similarity measures is the ability to use generalised feature vectors of GO terms and then weight them according to the relation between terms in the training phase using an attention mechanism. This overcomes a limitation of manually created semantic similarity measures to judge how each relation between terms should contribute towards the end goal (Smaili et al., 2018). TransformerGO improves performance compared to recent machine learning approaches due to a careful design that captures the semantic similarity between gene ontology sets. Onto2vec and OpA2vec (Smaili et al., 2018, Smaili et al., 2019) encode the GO terms similarly as TransformerGO, but the prediction of the interaction is modelled by a simple cosine similarity or a shallow fully connected neural network. While protein2vec (Shin et al., 2009) uses an LSTM to model the protein representation, the input is considered a sequence of terms and the interaction is still predicted by a fully connected layer.

A new trend is to allow these methods to take into consideration modern high-throughput technologies from various datasets (Mazandu et al., 2017). TransformerGO takes as input feature vectors of gene ontology terms and can be trained to solve other biological questions such as predicting the type of interactions between protein pairs.

Transformers are neural networks that use attention to accelerate training (Vaswani et al., 2017) and are the main component of state-of-the-art NLP architectures such as BERT (Devlin et al., 2018). Interpreting attention is an active and well-known area of research (Wiegreffe and Pinter, 2019, Jain and Wallace, 2019), but the application to biological sequences is still lagging behind. We propose a visualisation of the attention heads that extends previous work to the field of semantic similarity based prediction of protein interactions. Unlike classic semantic similarity measures, the source attention offers valuable insights that explain the similarity between gene ontology terms. We find that Cellular Component terms are an important indicator of interacting proteins and that there are cases where semantic similarity is being picked up across different ontologies.

While we demonstrated TransformerGO’s performance on the task of protein–protein interaction prediction, semantic similarity is still far from reaching the status of other similarity measures between gene products, such as the sequence based ones (Pesquita et al., 2009).

We expect future research on attention based models to offer more comprehensive analysis of protein to protein interactions, thorough model interpretation of the semantic similarity at a more granular level.

Funding
This work was supported by a studentship from the Engineering and Physical Sciences Research Council (EPSRC) via the University of Southampton. MN’s contribution is partially funded by EPSRC grant “Artificial and Augmented Intelligence for Automated Scientific Discovery” (EP/S000356/1).

References
Ashburner, M. et al. (2000). Gene ontology: tool for the unification of biology. Nature genetics, 25(1), 25–29.
Bandyopadhyay, S. and Mallick, K. (2016). A new feature vector based on gene ontology terms for protein-protein interaction prediction. IEEE/ACM transactions on computational biology and bioinformatics, 14(4), 762–770.
Bepler, T. and Berger, B. (2021). Learning the protein language: Evolution, structure, and function. Cell Systems, 12(6), 654–669.
Cafarella, T. et al. (2017). Mapping, modeling, and characterization of protein–protein interactions on a proteomic scale. Current Opinion in Structural Biology, 44, 201–210.
Chen, L. et al. (2020). Transformerspc: improving compound–protein interaction prediction by sequences-based deep learning with self-attention mechanism and label reversal experiments. Bioinformatics, 36(16), 4406–4414.
Chen, M. et al. (2019). Multifaceted protein–protein interaction prediction based on siamese residual cnn. Bioinformatics, 35(14), 2305–2314.
Chin, C.-H. et al. (2010). A hub-attachment based method to detect functional modules from confidence-scored protein interactions and expression profiles. BMC bioinformatics, 11(1), 1–9.
Consortium, G. O. (2015). Gene ontology consortium: going forward. *Nucleic acids research*, 43(D1), D1049–D1056.

Dai, Z. et al. (2019). Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02560*.

Devlin, J. et al. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ewing, R. M. et al. (2007). Large-scale mapping of human protein–protein interactions by mass spectrometry. *Molecular systems biology*, 3(1), 89.

Fawcett, T. (2006). An introduction to roc analysis. *Pattern recognition letters*, 27(8), 861–874.

Gavin, A.-C. et al. (2006). Proteome survey reveals modularity of the yeast cell machinery. *Nature*, 440(7084), 631–636.

Gillis, J. and Pavlidis, P. (2011). The impact of multifunctional genes on guilt by association analysis. *PloS one*, 6(2), e17258.

Gillis, J. and Pavlidis, P. (2012). "guilt by association" is the exception rather than the rule in gene networks. *PLoS computational biology*, 8(5), e1002444.

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.

Guo, X. et al. (2006). Assessing semantic similarity measures for the characterization of human regulatory pathways. *Bioinformatics*, 22(8), 967–973.

Hashemi, S. et al. (2018). Predicting protein–protein interactions through sequence-based deep learning. *Bioinformatics*, 34(17), 1802–1810.

Ito, T. et al. (2001). A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proceedings of the National Academy of Sciences*, 98(9), 4569–4574.

Jain, S. and Bader, G. D. (2010). An improved method for scoring protein–protein interactions using semantic similarity within the gene ontology. *BMC bioinformatics*, 11(1), 1–14.

Jain, S. and Wallace, B. C. (2019). Attention is not explanation. *arXiv preprint arXiv:1902.10186*.

Jansen, R. et al. (2003). A bayesian networks approach for predicting protein–protein interactions from genomic data. *Science*, 302(5644), 449–453.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kovalova, O. et al. (2019). Revealing the dark secrets of bert. *arXiv preprint arXiv:1908.08593*.

Kulmanov, M. and Hochndorf, R. (2017). Evaluating the effect of annotation size on measures of semantic similarity. *Journal of biomedical semantics*, 8(1), 1–10.

Kulmanov, M. et al. (2019). El embeddings: geometric construction of models for the description logic e+ω. *arXiv preprint arXiv:1902.10499*.

Kulmanov, M. et al. (2021). Semantic similarity and machine learning with ontologies. *Briefings in bioinformatics*, 22(4), biaa199.

Li, H. et al. (2018). Deep neural network based predictions of protein interactions using primary sequences. *Molecules*, 23(8), 1923.

Liu, Y. et al. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Mazandu, G. K. et al. (2017). Gene ontology semantic similarity tools: survey on features and challenges for biological knowledge discovery. *Briefings in bioinformatics*, 18(5), 886–901.

Mikolov, T. et al. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Miller, J. P. et al. (2005). Large-scale identification of yeast integral membrane protein interactions. *Proceedings of the National Academy of Sciences*, 102(34), 12123–12128.

Oliver, S. (2000). Guilt-by-association goes global. *Nature*, 403, 6770, 601–602.

Oughtred, R. et al. (2019). The biogrid interaction database: 2019 update. *Nucleic acids research*, 47(D1), D529–D541.

Paszke, A. et al. (2017). Automatic differentiation in pytorch.

Patil, A. and Nakamura, H. (2005). Filtering high-throughput protein–protein interaction data using a combination of genomic features. *BMC bioinformatics*, 6(1), 1–13.

Perozzi, B. et al. (2014). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710.

Pesquita, C. et al. (2009). Semantic similarity in biomedical ontologies. *PLoS computational biology*, 5(7), e1000443.

Qiu, X. et al. (2020). Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, pages 1–26.

Razick, S. et al. (2008). irfindex: a consolidated protein interaction database with provenance. *BMC bioinformatics*, 9(1), 1–19.

Riesen, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*.

Rhodes, D. R. et al. (2005). Probabilistic model of the human protein–protein interaction network. *Nature biotechnology*, 23(8), 951–959.

Rogers, A. et al. (2020). A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8, 342–366.

Shin, C. J. et al. (2009). Protein–protein interaction as a predictor of subcellular location. *BMC systems biology*, 3(1), 1–20.

Smaill, F. Z. et al. (2018). Onto2vec: joint vector-based representation of biological entities and their ontology-based annotations. *Bioinformatics*, 34(13), 152–160.

Smaill, F. Z. et al. (2019). Op2vec: combining formal and informal content of biomedical ontologies to improve similarity-based prediction. *Bioinformatics*, 35(12), 2133–2140.

Stefl, U. et al. (2005). A human protein–protein interaction network: a resource for annotating the proteome. *Cell*, 122(6), 957–968.

Sklarczyk, D. et al. (2016). The string database in 2017: quality-controlled protein–protein association networks, made broadly accessible. *Nucleic acids research*, page gkw937.

Vaswani, A. et al. (2017). Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Vig, J. et al. (2020). Bertology meets biology: Interpreting attention in protein language models. *arXiv preprint arXiv:2006.15222*.

Wiegreffe, S. and Pinter, Y. (2019). Attention is not explanation. *arXiv preprint arXiv:1908.04626*.

Xenarios, I. et al. (2000). Dip: the database of interacting proteins. *Nucleic acids research*, 28(1), 289–291.

Xu, T. et al. (2008). Evaluation of go-based functional similarity measures using s. cerevisiae protein interaction and expression profile data. *BMC bioinformatics*, 9(1), 1–10.

Zhang, Y. et al. (2018). An improved approach to infer protein–protein interaction based on a hierarchical vector space model. *BMC bioinformatics*, 19(1), 1–14.

Zhang, J. et al. (2020). protein2vec: predicting protein–protein interactions based on lstm. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*.

Zhang, S.-B. and Tang, Q.-R. (2016). Protein–protein interaction inference based on semantic similarity of gene ontology terms. *Journal of theoretical biology*, 401, 30–22.

Zhao, L. et al. (2020). Conjoint feature representation of go and protein sequence for ppi prediction based on an inception rnn attention network. *Molecular Therapy-Nucleic Acids*, 22, 198–208.

Zhang, X. et al. (2019). Go2vec: transforming go terms and proteins to vector representations via graph embeddings. *BMC genomics*, 20(9).