Paper Review For deepSimDEF
by Anurag Banerjee

deepSimDEF: deep neural embeddings of gene products and gene ontology terms for functional analysis of genes[1]

Ahmad Pesaranghader et. al.

1 Jargon Glossary

Biological Process (BP) - A biological macro-function such as *DNA repair*, which is achieved via multiple molecular activities. [2]

Cellular Component (CC) - The location occupied by a macro-molecular machine, relative to the cellular compartments and structures. [2]

Functional Similarity (FS) (*of genes*) - A quantitative measure to enable comparison of genes for their roles in biological processes and molecular functions. Most measures make use of semantic similarity in *Gene Ontology*.

Gene Ontology (GO) - Whereas an *ontology* is a formal representation of a body of knowledge within a given domain; a *gene ontology* describes our knowledge of the biological domain w.r.t. *molecular function*, *cellular component* and *biological process*. [2]

Gene Ontology *annotations* - A statement about the function of a particular gene. At the very minimum, a GO annotation consists of: *gene product*, *GO term*, *reference* and *evidence*. [2]

Gene Ontology *term* - A GO *term* or class consists primarily of a definition, a label and a unique identifier. Whereas the ontology itself is a loosely hierarchical graph, the terms are nodes in this graph. [2]

Molecular Function (MF) - Describes activities that occur at the molecular level (*e.g., catalysis*). These activities are performed by an individual gene product *or complexes*. Part of the process involves learning low dimensional vector embeddings for gene products and GO terms (*using which the FS is computed*). [2]

Sequence Homology - Similarity due to shared ancestory, between DNA, RNA or protein sequences.

Semantic Similarity (SS) (*of GO terms*) - It is a means of comparing the similarity of GO terms based on ontology (graph) structure and annotation corpora.

2 Problem Description

Given a *set* of genes and their **Gene Ontology annotation**s, this paper delves into discovering the process of *learning* **functional similarity** estimation for any gene-pair.

Learned embeddings for GO terms and gene-products are the *by-products* of the training process.

3 Problem Relevance

In general, GO (*Gene Ontology*) based FS (*Functional Similarity*) measures depend on slow FS computation and empirical SS (*Semantic Similarity*) metric engineering.

The following provides important context!

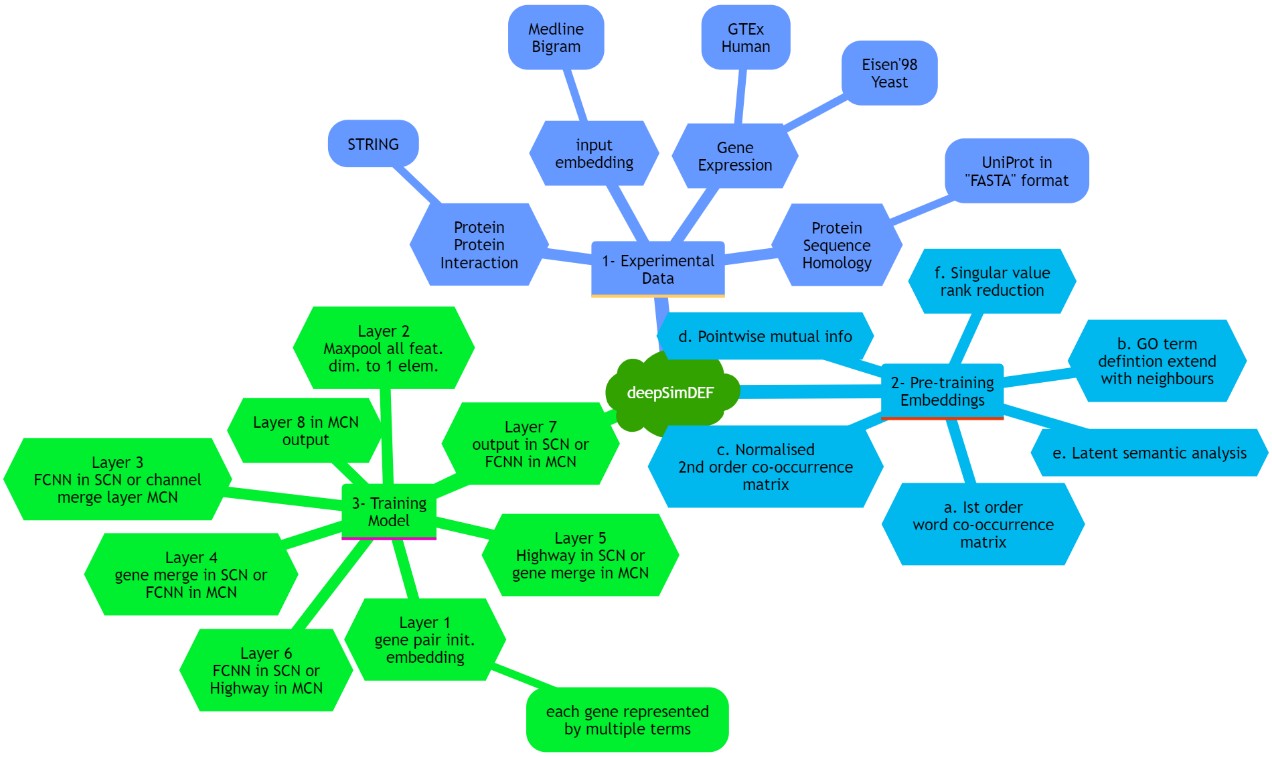
Medline Bigram
GTEx Human
Eisen'98 Yeast
STRING
input embedding
Gene Expression
UniProt in "FASTA" format
Protein Protein Interaction
1- Experimental Data
Protein Sequence Homology
f. Singular value rank reduction
Layer 2 Maxpool all feat. dim. to 1 elem.
d. Pointwise mutual info
b. GO term defintion extend with neighbours
deepSimDEF
2- Pre-training Embeddings
Layer 8 in MCN output
Layer 7 output in SCN or FCNN in MCN
c. Normalised 2nd order co-occurrence matrix
e. Latent semantic analysis
Layer 3 FCNN in SCN or channel merge layer MCN
3- Training Model
a. Ist order word co-occurrence matrix
Layer 4 gene merge in SCN or FCNN in MCN
Layer 5 Highway in SCN or gene merge in MCN
Layer 1 gene pair init. embedding
Layer 6 FCNN in SCN or Highway in MCN
each gene represented by multiple terms

Figure 1: Mindmap

In literature, two computational classes of GO based FS measures are available, viz., **Ontology-based methods** and **Distributional-based methods**. The former utilizes either pair-wise *Information Content* measures, which is computationally costly, or group (set) wise measures which are less compute intensive, but also, less accurate. In the latter, the similarity measures are based on the comparison of *text definitions of a term* with that of its neighbours. Both of these classes rely on **manual metric** (*e.g., MAX, BMA*) **& feature engineering** for *aggregating GO terms' SS scores, prior to computing gene FS scores.*

4 Proposed Solution

The proposed solution[3] sets up a supervised neural network which takes in GO-term embeddings (*from some other source*), utilizes all sub-ontologies (BP, MF, CC) either individually (**single channel**) or simultaneously (**multi channel**) and performs *classification* or *regression* for some downstream biological task, generating learned embeddings in the due course. The main components of the logic presented in the paper may be visualized through the *mindmap* in Figure 1; wherein, under **Training Model**, `SCN` means *Single Channel Network* and `MCN` means *Multi-Channel Network*.

Three downstream tasks have been used, where, **Protein-Protein Interactions** (PPI) has been setup as a *classification* task and **Gene Expression** (GE) and **Sequence Homology** (SH) have been setup as *regression* tasks.

4.1 Experimental Data

The data for the experimental setup has been acquired from the sources as described below:

1. The **Gene Ontology** information has been downloaded from `www.geneontology.org/page/download-ontology`
2. The term definition texts were enriched using the **MEDLINE** abstracts downloaded from `www.nlm.nih.gov/databases/download/data_distrib_main.html`
3. For PPI classification downstream task, **STRING** dataset was obtained from `string-db.org/cgi/download.pl`

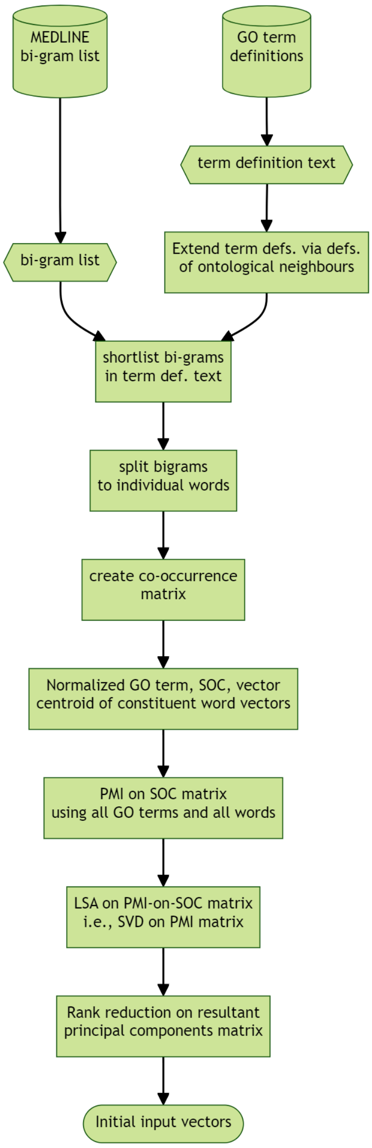
MEDLINE bi-gram list
GO term definitions
term definition text
bi-gram list
Extend term defs. via defs. of ontological neighbours
shortlist bi-grams in term def. text
split bigrams to individual words
create co-occurrence matrix
Normalized GO term, SOC, vector centroid of constituent word vectors
PMI on SOC matrix using all GO terms and all words
LSA on PMI-on-SOC matrix i.e., SVD on PMI matrix
Rank reduction on resultant principal components matrix
Initial input vectors

Figure 2: Pre-training Embedding

4. For SH regression downstream task, **UniProt in FASTA format** was obtained from `www.uniprot.org/proteomes/*`, where, * is replaced by a *code* for either ***yeast*** or for ***human***

5. For GE regression downstream task, for *yeast*, data was obtained from *supplementary material* of [4] and for *human*, **GTEx** data was obtained from `www.gtexportal.org`

4.2 Pre-training Embeddings

In Figure 2, the overall process for creating the input vectors for the neural network has been depicted via a *flowchart*. In the figure, the term `SOC` means *Second Order Co-occurrence* (matrix), `PMI` means *Pointwise Mutual Information*, `LSA` means *Latent Semantic Analysis* and `SVD` means *Singular Value Decomposition*.

The purpose of the initialization of the embeddings is to ensure that gene products have good initial representation in the neural network so that meaningful gradients are generated.

4.3 Training Model

Whereas, the layers in the *Single Channel* and *Multi Channel* versions have been summarised in the *mindmap* in Figure 1, in this section we will capture the logic of the neural network via the *flowchart* in Figure 3.
In the Figure 3 the data sources indicate the different downstream tasks that have already been discussed above. The process may be further enumerated with certain more details as follows (*the multi-channel process is described*):

1. A data source for a particular task is selected, and pairs of relevant gene-products are selected

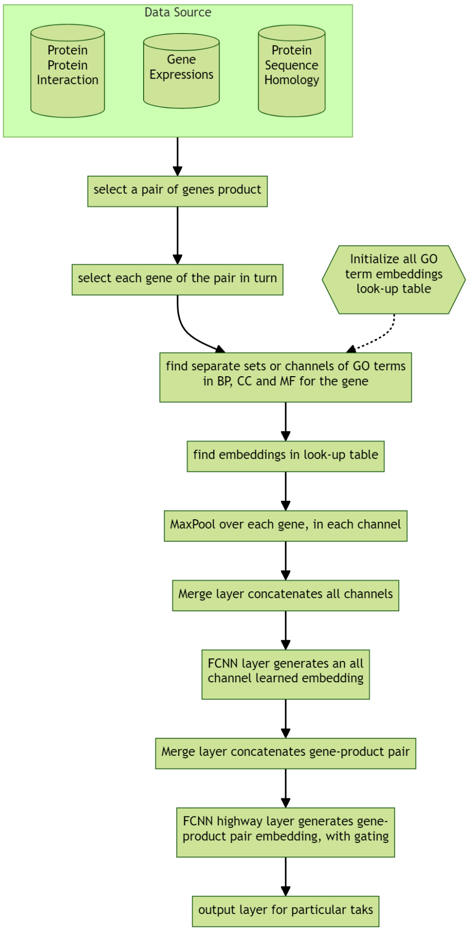
Data Source
Protein Protein Interaction
Gene Expressions
Protein Sequence Homology
select a pair of genes product
select each gene of the pair in turn
Initialize all GO term embeddings look-up table
find separate sets or channels of GO terms in BP, CC and MF for the gene
find embeddings in look-up table
MaxPool over each gene, in each channel
Merge layer concatenates all channels
FCNN layer generates an all channel learned embedding
Merge layer concatenates gene-product pair
FCNN highway layer generates gene-product pair embedding, with gating
output layer for particular taks

Figure 3: Training Model

2. Each gene product may contain multiple GO terms. Further, the gene terms are different under each sub-ontology (BP, CC and MF). So, 3 sets of GO terms are formed for each gene-product

3. The term embeddings are looked-up in the tables formed in the pre-training embedding step, and matrices are formed. When the number of terms are not same, padding of large negative numbers are used

4. The first maxpool layer selects the max value from the fixed 100 dimension features for each term and prepares a 1-D vector from the matrix

The remaining steps are as per the *flowchart*. In case of the **Single Channel** version, the channel merge layer is not required. The **Gene Embedding** can be extracted right before the *gene-pair merge layer*.

5 Positive Points

- The setup is a robust method to generate embeddings for gene-prducts.

6 Negative Points

- The actual task, viz., classification and regression, are not well defined; as in, the loss functions could have been better explained along with the ways to prepare the **groundtruth**

7 Questions

1. In the experiments described, how were the groundtruths prepared?

2. The exact description of the loss functions?

References

[1] A. Pesaranghader, S. Matwin, M. Sokolova, J.-C. Grenier, R. G. Beiko, and J. Hussin, “deepSimDEF: deep neural embeddings of gene products and gene ontology terms for functional analysis of genes,” *Bioinformatics*, vol. 38, pp. 3051–3061, May 2022.

[2] P. Gaudet, “Gene ontology overview.” Available: `https://geneontology.github.io/docs/ontology-documentation/`, Oct 2023. Accessed: 11 June 2024.

[3] A. Pesaranghader, “deepsimdef github repository.” Available: `https://github.com/ahmadpgh/deepSimDEF`, May 2022. Accessed: 14 June 2024.

[4] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, “Cluster analysis and display of genome-wide expression patterns,” *Proceedings of the National Academy of Sciences*, vol. 95, no. 25, pp. 14863–14868, 1998.