Protein Function Identification and Analysis

Yuqing Qin
qing0002@umn.edu

Anlan He
he000193@umn.edu

Jesse Elder
elder099@umn.edu

Arnav Solanki
solan053@umn.edu

April 13, 2021

Abstract

Prediction on protein function is a popular task these days due to its increasing volume of sequenced data. Given the protein sequence, obtaining the Gene Ontology(GO) terms that describe the protein's function is the main prediction task. To deal with this problem, we implemented k nearest neighbor method and compared its performance with the methods in Kulmanov et al's paper[1]. Also, we did a GO term frequency analysis and principle component analysis(PCA) to analyze the most common and the most significant GO term in protein function prediction task.
Keywords: *Protein-Sequence, Protein-Function, Gene Ontology, Classification.*

1 Introduction

Proteins are chains of amino acids that are the product of RNA translation by ribosomes. Most cellular operations and mechanisms, from metabolism to gene repair, are carried out by proteins built from the same 22 amino acids. Given how all proteins, regardless of their length and structure, arise from their coding in DNA (as understood in the central dogma of Biology), it is possible and feasible to compare and analyze different proteins and their properties. A key property of interest is the function of a protein - what does it do? Methods such as DNA sequence comparison, Protein-Protein interactions, Phylogenetic analysis, Protein structural analysis and Gene Expression profiles can be used to predict the function of a protein. Just using one of these techniques is usually not sufficient for this task, but sequence-based prediction allows a fast, interpretable means of achieving it. Our project aims to deal with this protein function prediction problem by using sequence data and dive deeper into analyzing prediction results.

We implemented a Naive kNN classifier with BLAST similarity metric on proteins to predict GO terms for their functions. We based our classifier of and compared it to Kulmanov et al[1] 's DeepGoPlus method by using Precision-Recall curves. We tested the relevance of GO terms and variation in prediction scores for proteins. We also did a GO term frequency analysis by generating a frequency histogram and a word cloud of GO term names to show the most common GO terms in the predictions. Principal Components Analysis(PCA) was also implemented to determine which predicted GO terms were most significant in explaining the variation seen in protein function prediction.

2 Methodology

2.1 Data Files

We operated upon data from Kulmanov et al's original dataset - the file we chose was the CAFA dataset available here `http://deepgoplus.bio2vec.net/data/` and the scripts available here `https://github.com/bio-ontology-research-group/deepgoplus/blob/master/README.md`. The CAFA dataset contains a training set and testing set of $66,000$ and $3,300$ proteins in FASTA format adapted from Swissprot16, and a Gene ontology file for protein function annotations. Their dataset, neural network and BLAST scores were already saved as python variables, so were easier to import than parsing the input files.

2.2 Naive K Nearest Neighbors Classifer

Our naive k nearest neighbor method combined the BLAST similarity score for sequence data to tackle the multi-label classification problem. BLAST alignment measures how similar a given sequence is to another - we can compare all testing proteisn with the entire training set to generate a matrix of BLAST scores. To find the nearest neighbors during our prediction process, we used One Over All cross validation and used the similar proteins' GO-term labels to generate the prediction output. We combine all the GO-terms from its nearest neighbors as a prediction result giving the probability of each GO-term annotating a testing protein.

2.3 Analysis Methods

We performed various methods of analysis to confirm the accuracy of Kulmanov et al's DeepGoPlus classifier -

- Area Under Precision Recall Curve - similar to a ROC curve, the precision and recall of prediction results can be calculated for different prediction thresholds and the area under its Precision-Recall curve estimates how well the classifier performs (larger area is better).
- Relevance Analysis - relevance tracks how relevant a GO-term or protein is in the prediction result. We define it as the sum of the prediction probabilities for one instance (either a GO-term or Protein) over all orthogonal values. The higher and more prevalent the probability scores of an instance, the higher its relevance.
- Frequency Analysis - frequency is a well understood metric, keeping track of how often a certain GO-term is predicted. We filtering out probabilities that cross a prediction threshold and analyze the frequency of such GO-terms. We finally generate a wordcloud for the annotations of these GO-terms. Annotations were carried out with the use of the python dependency at `https://github.com/tanghaibao/goatools`.
- Principle Component Analysis - The dataset of 3329 proteins and 19323 GO term prediction scores from DeepGoPlus poses a challenge in terms of visualizing and qualifying groups of proteins by function. In an effort to tease out any clusters of proteins, we performed Principal Components Analysis. PCA allows for visualization of high dimension data in a low dimension space. Additionally it extracts information on the GO terms that account for the most variance in the dataset in the form of variable "loadings."

3 Results

3.1 Comparison with Blast-kNN and DeepGoPlus

We tried different k values on the Naive kNN method. The results are shown in Figure 1. The figure suggests that there is not a significant difference in the predictions from different k values, but the green line for k=40 value works a bit better than the others. The details for the precision and recall value with the best operation point are shown in the following table (Table 1).

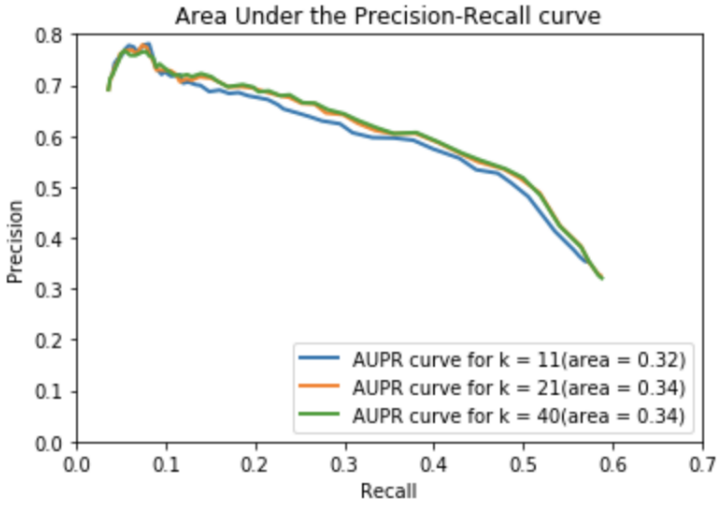

Figure 1: Precision-recall curve with different k values for Naive kNN method

Table 1: Precision and Recall Results for Different k Values in Naive kNN

k value	Best Threshold	Precision	Recall
11	0.14	0.527	0.471
21	0.12	0.517	0.496
40	0.12	0.518	0.500

We also compared the Naive kNN method with the methods mentioned in the paper, which is the Blast-kNN method, and DeepGoPlus method. Blast-kNN is the method mentioned in the paper[1], and it utilizes the Blast scores to get the nearest neighbors. However, this combines all the similar neighbors' labels as the output and is not restricted by k value. DeepGoPlus is the method combining Blast-kNN and a Convolutional Neural Network (CNN) which performs one-hot feature generated from the amino acids sequence. The CNN forms a 128*2000*21 feature table for each layer - batch size times protein sequence length limit times number of amino acids. They augment the BLAST scores with these features to get the final predictions.

We reran these two models and generated the Blast-kNN and DeepGoPlus results from the same data set. The precision-recall curves are generated by using Blast-kNN and DeepGoPlusWe correspondingly. The curves are shown in Figure 2 and Figure 3. We then picked the best prediction model for Naive kNN ($k = 40$) and compared its performance with those models. The comparison results are shown in the Table 2. From the table, it is obviously that the DeepGoPlus model works the best, and our naive kNN with large k value works similar to the Blast-knn model.

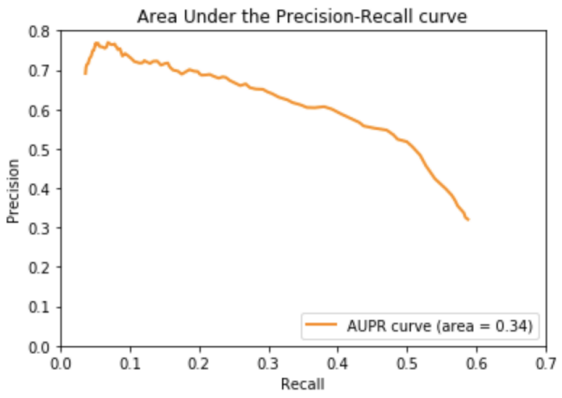

Figure 2: Precision-recall curve for Blast-kNN

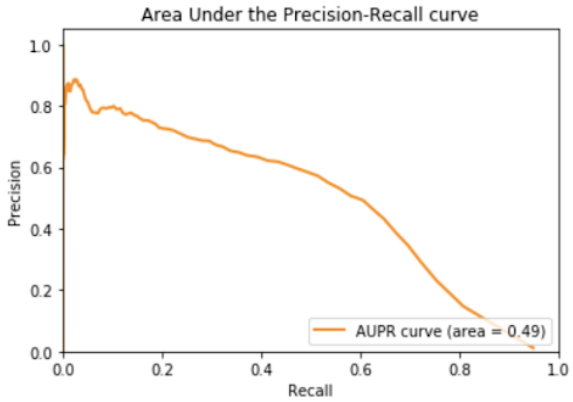

Figure 3: Precision-recall curve for DeepGoPlus

Table 2: Precision and Recall Results for Different Model comaprison

Models	Best Threshold	Precision	Recall
Naive KNN	0.12	0.518	0.500
Blast-KNN	0.12	0.517	0.499
DeepGoPlus	0.10	0.531	0.559

3.2 Relevance of GO Terms and Variation in Prediction Scores

The predictions scores of the protein function classifier provides an estimate of how likely a GO-term is annotated to a testing protein. While this is a useful measure, it is possible this does not scale well with a large ontology simply because these probabilities are not normalized, and hence many different GO-term predictions could be made concurrently. To demonstrate this, we generated a metric for every GO-term called relevance which measures the sum of all prediction probabilities for it across all proteins. Figure 4 shows the relevance for all GO-Terms while Figure 5 shows the histogram of these relevance scores. Clearly the first 5000 GO-terms show high relevance compared to the others as they are repeatedly predicted for multiple proteins. The histogram demonstrates that very few GO-terms have high relevance and almost seem to be outliers.

The 20 most relevant GO-terms in order of increasing relevance were - plasma membrane, obsolete cytoplasmic part, intracellular membrane-bounded organelle, metabolic process, cell periphery, membrane-bounded organelle, molecular function, cytoplasm, membrane, intracellular organelle, cellular process, organelle, biological process, obsolete intracellular part, cellular process, intracellular, biological process, cell part, cell, and cellular component.

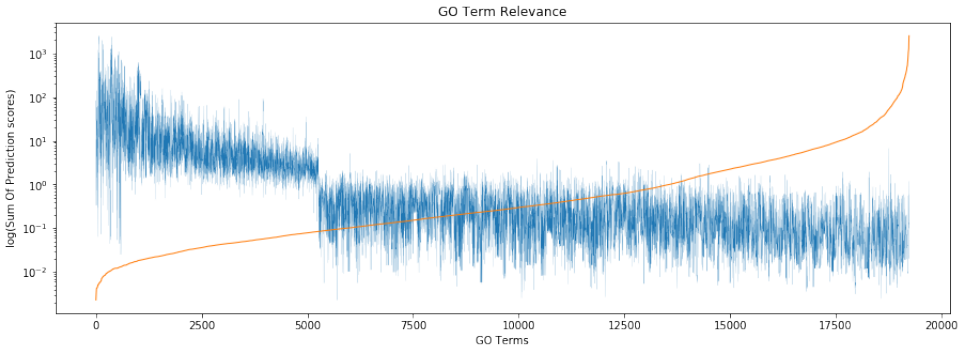

Figure 4: GO-term Relevance vs GO-term index in Blue. The Orange line shows the sorted relevance to demonstrate the distribution.

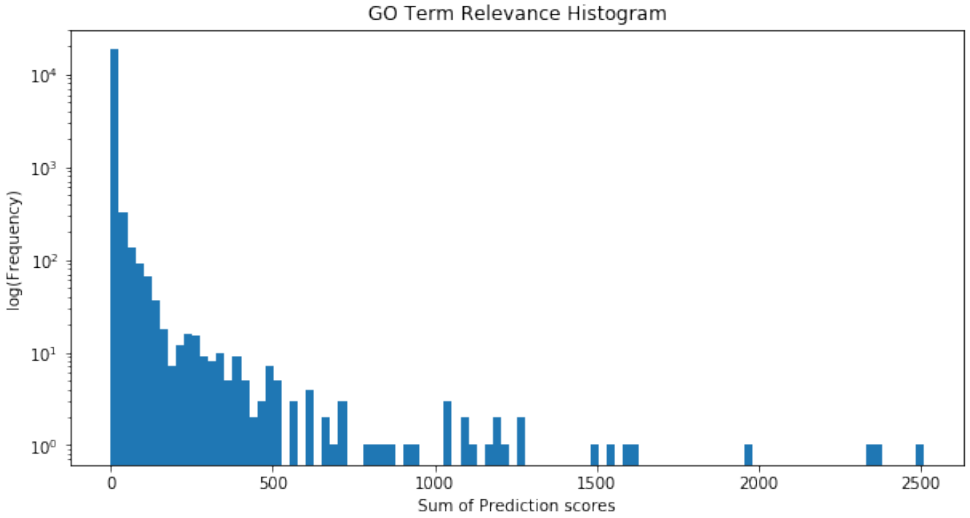

Figure 5: Histogram of all the GO-term relevance sums.

A similar analysis of relevance was done for every protein as well. The relevance for a protein was measured as the sum of all the GO-term prediction probabilities for that protein. Figure 6 shows the relevance for all proteins in the testing set and Figure 7 shows the corresponding histogram. Most proteins had a similar relevance in the range of 25-60, but the histogram shows that a few proteins had hardly any relevant predictions.

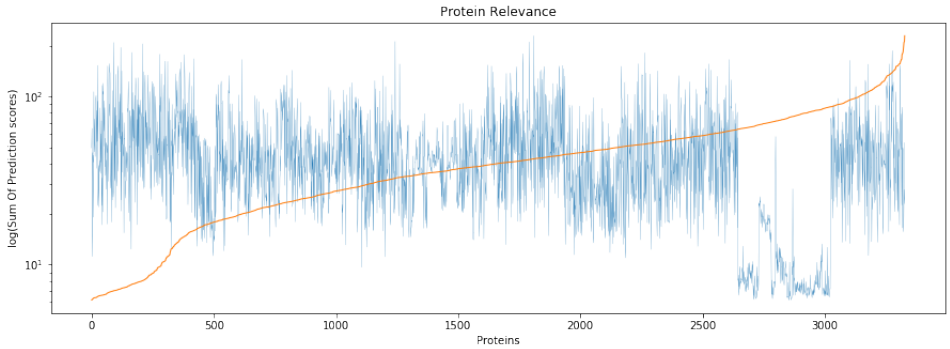

Figure 6: Protein Relevance vs Protein index in Blue. The Orange line shows the sorted relevance to demonstrate the distribution.

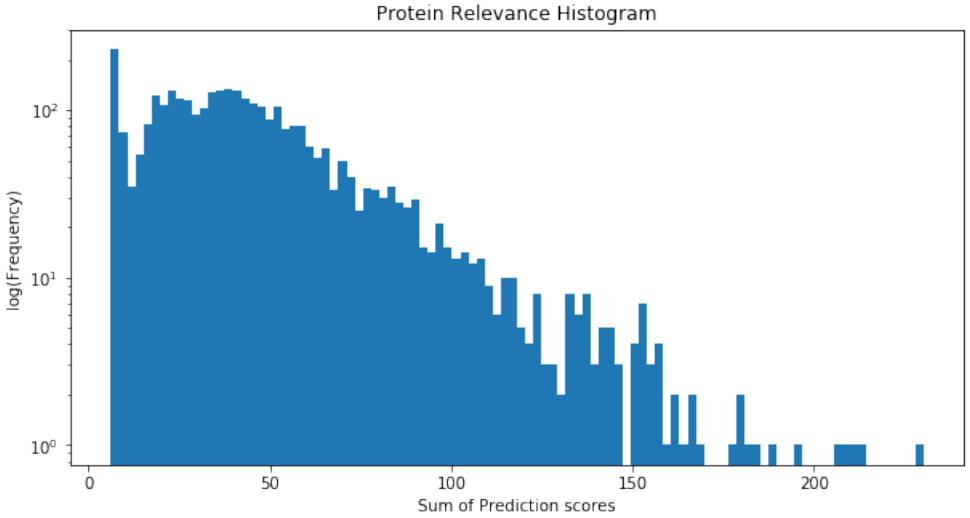

Figure 7: Histogram of all the testing Protein relevance sums.

3.3 GO Term Frequency Histogram and Word Cloud

We conducted a GO term frequency analysis based on the prediction results of the DeepGoPlus model. We firstly plotted a GO term frequency histogram based on the prediction results of the DeepGoPlus model, as shown in Figure 8. According to Figure 3, to achieve a relative high value in both the precision and recall measure, we found the best choice of threshold is 0.1. When the prediction score of a GO term is higher than 0.1, we assume that the protein's functional characteristics can be annotated by the GO term. Otherwise, we will not include the GO term as a member of the annotation set of the given protein. After we generated the classifications, we found that that most of the GO terms have very low frequency, and thus the majority of the data gathered in the narrow interval, making it difficult for us to have a better visualization of the whole distribution. To fix the issue and optimize the data visualization results, we found that adding a lower bound filter value of 100 would help. If the frequency of the GO term is lower than 100, we will not include this GO term as a part of the histogram. From the graph, we can tell that most of the GO term having frequencies lie in the interval from 0 - 500. The highest GO term frequency goes beyond 3000.

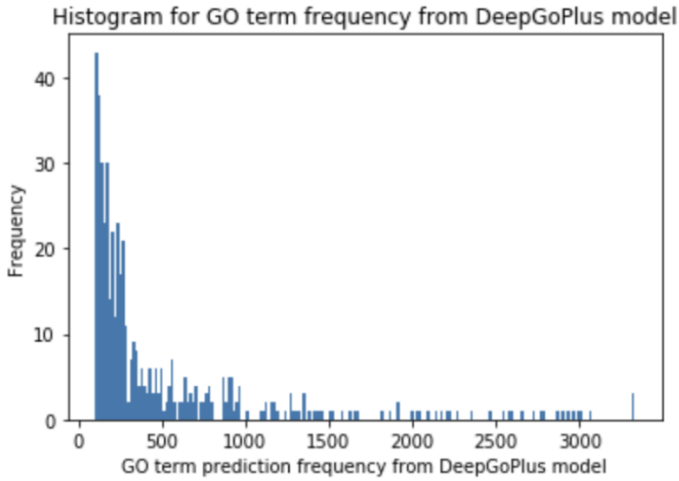

Figure 8: GO Term Frequency Histogram of DeepGoPlus Prediction Results

Furthermore, we selected the top 20 GO terms with the highest frequencies and found the GO term names respectively. The size of the word in the wordcloud shown in 9 represents the frequency of the

GO term. The most frequent GO terms are related to the most significant biological functions including metabolic process, biosynthetic process, positive/negative regulations, which are reasonable and correspondent to what we expected.

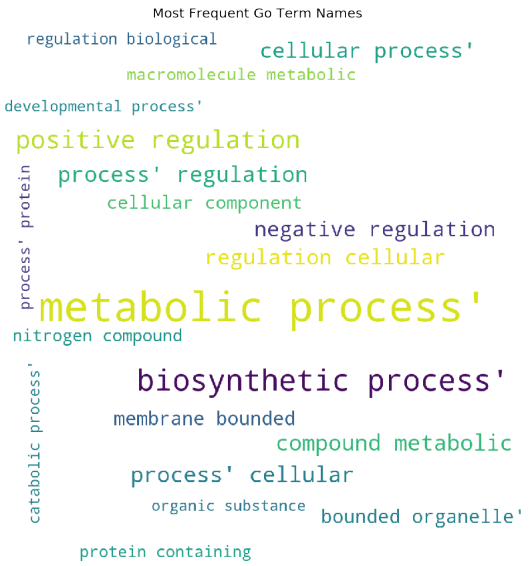

Figure 9: GO Term Frequency Histogram of DeepGoPlus Prediction Results

3.4 GO Term Prediction Score PCA

PCA offered limited results in terms of clustering. Analysis presented no clear, distinct protein clusters. The bulk of proteins loosely centered around (0,0), however, there is a small, dense grouping of proteins around (-0.2,0.2) that coincide with the direction of GO:0005886. It is not sufficiently distinct from the central group to consider it its own cluster. Although PCA did not demonstrate obvious clustering tendencies, it did extract the variable loadings of each GO term. The 20 GO terms with the highest variable loadings were selected for further analysis. 10 representative GO terms were selected and visualized with a red arrow.

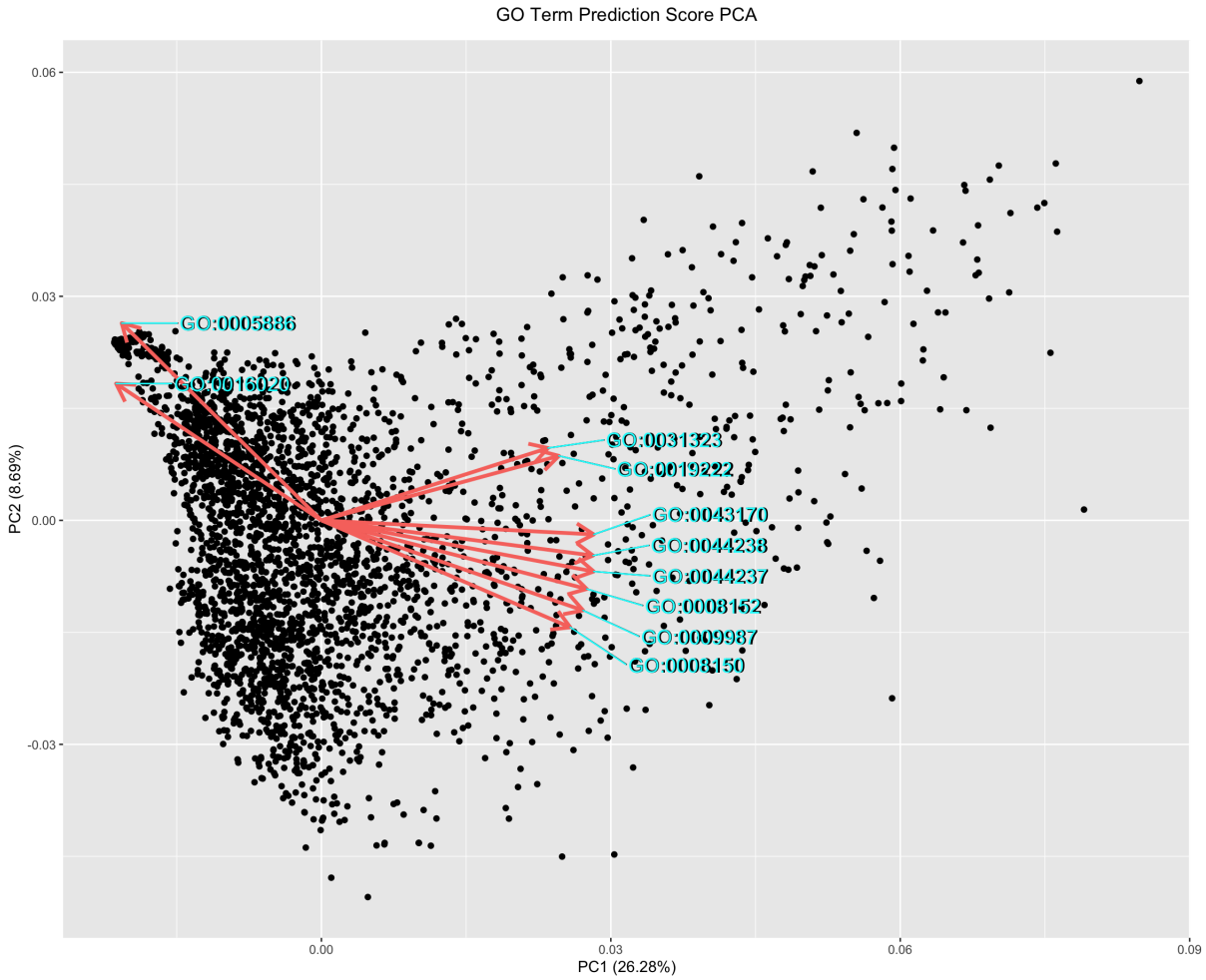

Figure 10: PCA on DeepGoPlus Prediction Results

3.5 Word Cloud on the Most Significant GO Terms

Given the PCA results, we sorted the GO terms with respect to the absolute values of their components scores and select the top 20 GO terms which could most explain the variance seen in the output proteins and generated a word cloud showing the relative significance of these GO terms (Figure 6). The GO names with bigger size have higher variable loadings (i.e. they account for more variance). The word cloud reveals that the most significant GO terms include metabolic process, biosynthetic process, compound metabolic.

If we compare the most frequent GO term word cloud and the most significant word cloud, we would find a large overlap between the GO names contained in the two graphs, which means many of the GO terms with high frequencies are also the most significant in explaining the variations.

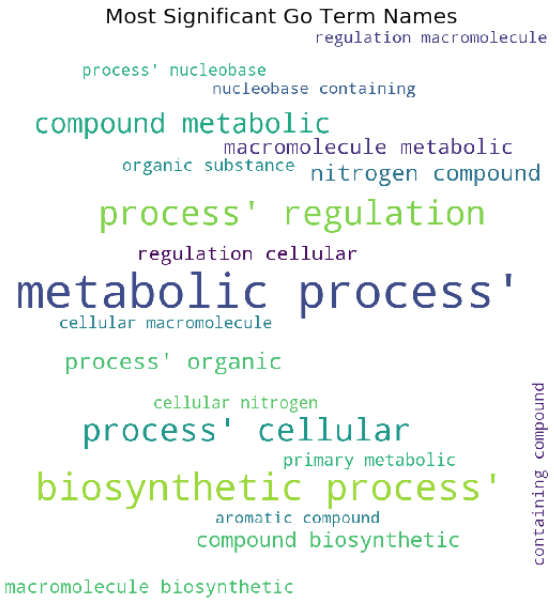

Figure 11: GO Term Frequency Histogram of DeepGoPlus Prediction Results

4 Conclusion

Our k-NN classifier performs similar to the BLAST k-NN method with Area under the Curve's of 0.32 and 0.34 respectively - these are still outperformed by DeepGoPlus CNN based classifier. Our choice of k was not significant in the performance of our classifier.

Principal Components Analysis was unable to separate the protein GO term prediction score data into distinct clusters. The anticipated result was for PCA to separate proteins into functional groups based on the similarity of predicted GO terms. Due to the inability of the method to separate proteins as expected, it can be concluded that this data is not linearly separable. Despite this, the most significant GO terms offer insight into the behavior of the DeepGoPlus model and the characteristics of Gene Ontology terms.

Our various methods of analysis on protein function reported the same GO-terms as significant - Metabolic Process, Biosynthetic Process, Regulation Cellular and Cellular Process were all GO-terms that were predicted frequently.

5 Future Works

Our work could be augmented by numerous additions -

- Use t-SNE alongside PCA. Since the data is not linearly separable, t-SNE could potentially generate more observable clusters. In addition, it can be used for further variable loadings analysis and can help infer the significance of GO-terms.
- Use a custom feature extraction method different from the CNN proposed by Kulmanov et al to allow for training more complex classifiers such as SVMs. This would allow a better understanding of what similarity means for different proteins.
- In the process for computing prediction probabilities from BLAST scores, the DeepGoPlus method uses different computation constants for the three major GO classes - Molecular Function, Biological Process and Cell Component. These classes had varying distributions in the Ontology file and the use of different constants to scale them was left untouched in our classifier when adapting from their

research. The distance metric for classification could be improved by accounting for the dependency on these 3 classes in a different format.

References

[1] Maxat Kulmanov and Robert Hoehndorf. Deepgoplus: improved protein function prediction from sequence. *Bioinformatics*, 36(2):422–429, 2020.