Visualizing and Annotating Protein Sequences using A Deep Neural Network

Zhengqiao Zhao
Department of Electrical and Computer Engineering
Drexel University
Philadelphia, USA
zz374@drexel.edu

Gail Rosen
Department of Electrical and Computer Engineering
Drexel University
Philadelphia, USA
glr26@drexel.edu

2020 54th Asilomar Conference on Signals, Systems, and Computers | 978-0-7381-3126-9/20/$31.00 ©2020 IEEE | DOI: 10.1109/IEEECONF51394.2020.9443364

***Abstract*—It is critical for biological studies to annotate amino acid sequences and understand how proteins function. Protein function is important to medical research in the health industry (e.g., drug discovery). With the advancement of deep learning, accurate protein annotation models have been developed for alignment free protein annotation. In this paper, we develop a deep learning model with an attention mechanism that can predict Gene Ontology labels given a protein sequence input. We believe this model can produce accurate predictions as well as maintain good interpretability. We further show how the model can be interpreted by examining and visualizing the intermediate layer output in our deep neural network.**

***Index Terms*—Bioinformatics, Gene Ontology, Deep Learning, Data Visualization**

I. Introduction

Understanding protein function at the molecular level has great implications in the biomedical and pharmaceutical industry [1]. For example, protein annotation facilitates the development of novel tools for disease prevention, diagnosis, and treatment [2]. Researchers can design experiments to characterize the function of a protein (for example, researchers can design an assay to measure the execution of a given molecular function and show if the protein serves as an agent in such executions) [3]. Knowing the diversity and full space of the protein universe will be helpful, and the number of genomic sequences collected is exponentially growing due to recent advances in sequencing technology [4]. However, experimental methods can not efficiently annotate protein sequences at large scale.

Researchers have shown that the knowledge of the biological role of common proteins in one organism can often be transferred to other organisms [5]. Therefore, the Gene Ontology Consortium proposed to use a dynamic collection of controlled vocabulary to describe the functions of proteins. Such a Gene Ontology (GO) database lays a foundation for computational analysis of large-scale molecular biology and genetics experiments in biomedical research [6].

There are three major branches in GO, namely, Biological Process (BP), Molecular Function (MF), and Cellular Component (CC) to describe the function of proteins from different aspects. GO is hierarchical, "children" terms are more specific functions compared with a "parent" term. In addition, individual terms can have not only multiple descendants, but also multiple parents [7]. This controlled vocabulary of protein functions has enabled computational methods for functional annotation. And the experimental function information of annotated protein sequences can be used to infer the functions of protein sequences that have not yet been characterized. One of the challenges in GO term prediction is that it's essentially a multi-task learning problem as the model predicts the presence of multiple GO terms simultaneously.

Recent advances in deep learning have made huge successes in Computer Vision (CV) and Natural Language Processing (NLP) fields. In this paper, we propose a deep learning model to predict the GO terms of protein sequences. We show that such model can outperform baseline methods and can be interpreted by extracting the output of intermediate layers. We show that our model can learn sequential information from input proteins and demonstrate multiple methods for model interpretation and data visualization. The rest of this paper is organized as follows. In Section II, we give an overview of related research. Section III discusses detailed information on our proposed model. Experiments are presented in Section IV, followed by our results and discussion in Section V. In section VI, we draw our conclusions.

II. Related Work

One of the most effective ways to computationally determine the functions of an unknown protein is to find the most similar sequence in the reference sequences with experimental functional annotations and use its functions to annotate the query sequence. Similarity comparison programs such as Basic Local Alignment Search Tool (BLAST) [8] can identify biologically relevant sequence similarities. Many computational methods have been proposed based on sequence similarity [1], [9]. However, the similarity search process involves alignment which is relatively computationally expensive. In addition, the similarity search method doesn't generalize to distantly related sequences. Therefore, the prediction of novel protein sequences can be challenging for similarity based methods [10].

Machine learning based methods, on the other hand, have the advantage of better generalizability to predict the function of remotely relevant proteins and the homologous proteins of distinct functions [11]. The function prediction performance

are further improved by deep learning based models in recent years [2], [12], [13]. However, the proposed models mainly use convolutional neural networks (CNN) instead of recurrent neural networks (RNN) which are more suitable to capture sequential order information [14]. Furthermore, recurrent neural networks, especially long short-term memory (LSTM) network, with an attention mechanism [15] have demonstrated better performance and model interpretability in not only the NLP field [16]–[18] but also the bioinformatics field [19], [20]. Therefore, we believe it is beneficial to incorporate a recurrent neural network with an attention mechanism for protein function prediction.

The contributions of our work are 3-fold: 1) We develop a deep learning model with attention mechanism that can predict Gene Ontology labels given a protein sequence input; 2) We show this model can produce more accurate predictions by comparing it against other baseline methods; 3) We show how the model can be interpreted by examining and visualizing the intermediate layer output in our deep neural network.

III. The Proposed Model

Our model consists of several convolutional residual network, a Bi-directional LSTM network with an attention mechanism [15] and a hierarchical dense layer described in [12]. The convolutional residual network (ResNet) is proposed to capture local patterns using convolutional filters and avoid vanishing gradients by shortcut connections. Therefore, we place it at the beginning of our model to learn the lower level patterns directly from the input data. Bi-directional LSTM network is then used to learn the high level features and sequential information from the previous step. One of the key component in our proposed model is the feed-forward attention mechanism, which is inspired by previous work in neural language processing field [15], [18]. The previous works have shown that the attention mechanism can enable the model for better interpretability in addition to accurate predictions. Therefore, we add this mechanism to the Bi-directional LSTM layer to get a dense representation of the input sequence. Finally, a hierarchical dense layer described in [12] is used to perform the final prediction. The advantage of such hierarchical dense layer is that it captures the hierarchical structure of GO and ensures that a GO term will be predicted if one of its "child" terms is predicted. We have applied a deep neural network with similar architecture in 16S rRNA datasets for phenotype prediction and demonstrated that such deep learning architecture can learn informative regions from 16S rRNA reads that are predictive for "Phenotype" [20].

Our proposed model is described in Fig. 1. The input is a one-hot coded protein sequence with length T. Since there are 20 different amino acids in our input sequences [21], the dimension of the a one-hot coded sequence is a $T \times 20$ dimensional matrix. The input is fed to one 1-dimensional convolutional blocks with window size of W and the number of output channels of N_c followed by 4 ResNet layers with the same configuration. The resultant output is a $T \times N_c$ dimensional matrix which is processed by a Bidirectional LSTM

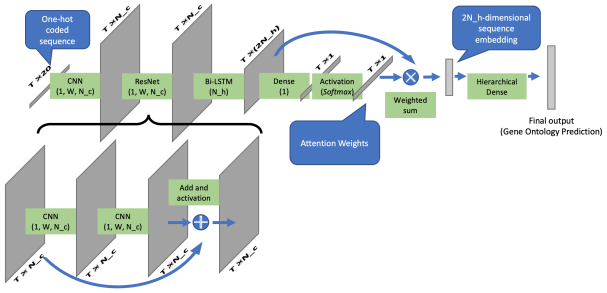

Fig. 1. The proposed model diagram

layer with the number of hidden nodes of N_h and the output from both directions are concatenated together. Therefore, the hidden states output, H, is a $T \times 2N_h$ dimensional matrix. The attention layer is a time distributed dense layer with 1 hidden unit, and it assigns a attention score for each position in the input along the sequence axis. The attention scores are activated by a `softmax` function to generate attention weights so that the weights sum to 1. The attention weights, α, is $1 \times T$ dimensional. Then, the attention weights and the hidden states output, H, are used compute a sequence embedding vector, E, as shown in (1) where E is $1 \times 2N_h$ dimensional.

$$E = \alpha H \tag{1}$$

Finally, a hierarchical dense layer is used to produce the final GO term predictions. Since one sequence can have multiple GO term annotation, we need a prediction layer that can predict different GO terms simultaneously. Here we used the hierarchical dense layer proposed in DeepGO [12]. In the hierarchical dense layer, each dense node, which outputs a scalar, corresponds to a GO term. We use `sigmoid` activation function after each dense node and use the binary cross-entropy as the cost function.

IV. Experiments

A. Baseline Methods

Kulmanov *et al.* proposed a novel deep learning model, DeepGO, that predicts protein function from sequence that outperforms the similarity based baseline method, BLAST [12]. The proposed model can take the protein sequences only as input and predict GO terms (referred as DeepGOSeq in the original paper). In addition, their model can be trained to take both a protein sequence and a protein-protein interaction (PPI) network embedding vector as inputs for GO term prediction. However, in order to retrieve the PPI information of the input sequence, similarity searches are required. In our paper, we focus on developing a model without similarity search steps. Therefore, we choose to compare our method with DeepGOSeq model which only takes a protein sequence as input. BLAST is another baseline method we consider in this paper which relies on sequence similarity comparison.

B. Dataset and Evaluation Metric

SwissProt provides manually curated protein sequences with Gene Ontology annotation [22] which was used by Kulmanov

et al. as the experimental dataset which contains 60,710 proteins annotated with 27,760 classes (19,181 in BP, 6221 in MF and 2358 in CC) [12]. They further filtered out very specific GO terms with only small number of annotations and selected the top 932 terms for BP, 589 terms for MF and 436 terms for the CC ontology to train their models [12] which results in three sets of training and testing datasets labeled by the three GO term branches. The authors have released the processed dataset[1]. We use their filtered dataset to develop our model and further visualization.

The performance of our model is evaluated by a protein centric maximum F-measure, F_{max}. It is widely used for Gene Ontology prediction evaluation [1], [2], [12]. We adapted the F-measure computation defined in DeepGO [12]. The output vectors of both our model and DeepGO model are vectors of decimal values between 0 and 1. To determine the predicted GO terms, a threshold is needed and protein centric maximum F-measure can be used to find the best threshold that maximizes the F-measure of the resultant GO term predictions. To be specific, Kulmanov *et al.* compute the F_{max} measure using the following formulas:

1) The precision and recall of an individual sequence, i, can be evaluated by (2) and (3) where f is a GO term, $P_i(t)$ is a set of predicted GO terms for the protein i determined by a threshold t, and Ti is a set of annotated GO terms for the protein i (ground truth). Note that one sequence can be annotated by multiple GO terms, therefore, precision and recall can be calculated per sequence.

$$pr_i(t) = \frac{\sum_f I(f \in P_i(t) \wedge f \in T_i)}{\sum_f I(f \in P_i(t))} \tag{2}$$

$$rc_i(t) = \frac{\sum_f I(f \in P_i(t) \wedge f \in T_i)}{\sum_f I(f \in T_i)} \tag{3}$$

2) The averaged precision and recall can be calculated among testing proteins by (4) and (5) where $m(t)$ is the total number of proteins that the model predicts at least one term using the threshold t and n is a number of all testing proteins.

$$AvgPr(t) = \frac{1}{m(t)} \sum_{i=1}^{m(t)} pr_i(t) \tag{4}$$

$$AvgRc(t) = \frac{1}{n} \sum_{i=1}^{n} rc_i(t) \tag{5}$$

3) Finally, the protein centric maximum F-measure, F_{max}, can be calculated by choosing the threshold t that maximizes the averaged F-measure as shown in (6).

$$F_{max} = \max_i \frac{2 \cdot AvgPr(t) \cdot AvgRc(t)}{AvgPr(t) + AvgRc(t)} \tag{6}$$

[1]The DeepGO experimental dataset is available at https://github.com/bio-ontology-research-group/deepgo [12]

C. Experimental Setup

The experimental dataset has been split into a train and test set by Kulmanov *et al.* We further split the training set into 80% training set and 20% validate set. We train three different models to predict GO terms associated with three main branches (BP, MF, CC) respectively similar to DeepGO models [12]. The best set of parameters are selected for each model to maximize F_{max} in validation set through a grid search of possible combinations of parameters listed in Table I. The best parameters for three models are pretty consistent: $N_c = 128$, $W = 3$, $D = 0.1$ and $HD = Yes$. However, the optimal N_h is 64 for both BP and MF model and 128 for CC model.

TABLE I
HYPERPARAMETER SEARCH SPACE

Hyperparameter	Value
Number of conv filters, N_c	64, 128, 256
Window size of conv filters, W	3, 9, 27
Number of units in LSTM, N_h	32, 64, 128
Dropout probability for Dropout Layer, D	0, 0.1, 0.2
Hierarchical dense layer, HD	No, Yes

V. RESULTS AND DISCUSSION

The models are trained with the best parameters selected in Section Experimental Setup and compare with the F_{max} values reported in DeepGO paper [12]. We also implement the DeepGOSeq model ourselves and evaluate our own implementation with the experimental dataset. The performance is shown in Table II. From Table II, we can see that BLAST has the best performance for BP related GO terms prediction. However, it is not performing well in MF and CC prediction tasks. The performance of the DeepGOSeq model reported in the original paper notably outperforms BLAST in the CC prediction task. However, its performance is slightly worse than BLAST in BP and MF tasks. Our implementation of DeepGOSeq model outperforms BLAST in both MF and CC tasks with a some performance drop in BP. Lastly, our proposed model produces better or comparable Gene Ontology terms prediction performance. It is comparable to BLAST in the BP task and it also stands out in MF and CC prediction tasks compared with DeepGOSeq model.

TABLE II
PROTEIN CENTRIC MAXIMUM F-MEASURE FOR DIFFERENT MODELS – THE PROPOSED MODELS OUTPERFORM THE BASELINE MODELS

Method	BP	MF	CC
BLAST (reported in [12])	0.314	0.372	0.362
DeepGOSeq (reported in [12])	0.293	0.364	0.568
DeepGOSeq (our implementation)	0.293	0.356	0.539
Proposed model	0.304	0.419	0.598

In addition, to produce accurate GO predictions, the intermediate layer output of our model, namely the attention weights, α, and the sequence embedding vector, E, can facilitate data visualization and model interpretation. First, our model can convert a raw amino acid sequence into a meaningful numerical embedded vector, E, that encodes protein function signals. To understand how our model transforms

the protein sequences in numerical space, we extract the embedding vectors, H, for all testing sequences and reduce the dimensionality to 2-dimension using t-SNE [23] for visualization. Fig. 2 and 3 show the testing sequence embedding vectors generated by MF and CC models respectively. We can observe that there are several clusters which might correspond to biologically meaningful functions. To interpret the clusters, we perform k-means [24], with $k = 5$ clusters, to assign cluster labels to sequences. Then we find the GO term that dominates each clusters and color code all testing sequences with that dominant GO term with a distinctive color. Sequences that are not associated with the selected dominant GO terms are color coded by gray color. In Figs. 2 and 3, each point is a sequence and colored by a color that corresponds to a dominant GO term. From these figures, we can see that the model learns the amino acid sequences information and can embed sequences associated with the same functions closely. To be specific, Fig.

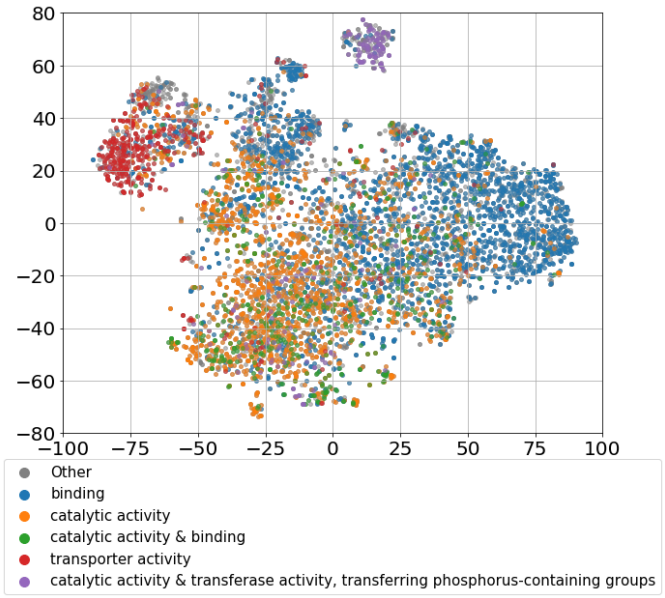

Fig. 2. 2D projection of the testing sequences embedding generated by Molecular Function (MF) model. Sequences annotated with transporter activity are embedded in a small cluster at the left side of the figure and sequences annotated with transferase activity and transferring phosphorus-containing groups are embedded in a dense cluster at the top of the figure.

2 shows the embedding of testing sequences of the MF model. From the figure, we can see that binding related sequences (in blue) are clustered in the right side of the figure and sequences with catalytic activity annotation (in orange) are embedded in the lower side of the figure. And these two clusters overlap with each other. We further show that some sequences have both catalytic activity and binding labels (in green) which contributes to the overlap between the two clusters. Sequences with transporter activity form their own cluster (in red) and sequences with one specific catalytic activity related GO term (transferase activity, transferring phosphorus-containing groups) form the purple cluster. In Fig. 3, clusters are formed based on Organelle. Proteins in the Mitochondria form a small cluster (in green). There is a symbiogenesis theory

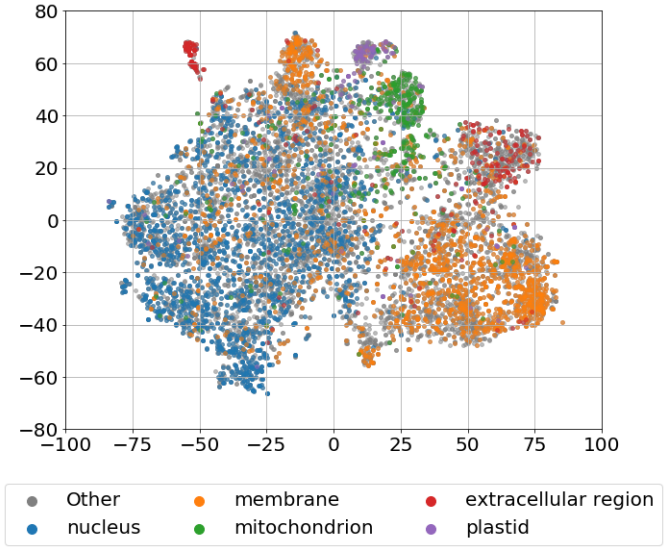

Fig. 3. 2D projection of the testing sequences embedding generated by Cellular Component (CC) model. Sequences annotated with Mitochondrion, Plastid and extracellular region forms their own clusters.

hypothesizing that all mitochondria derive from a common ancestral organelle that originated from the integration of an endosymbiotic *alphaproteobacterium* into a host cell related to *Asgard Archaea* [25]. Our proposed model picks up this information by embedding protein sequences in Mitochondria together in the embedding space. Membrane associated sequences, on the other hand, are more widely spread which implies that the membrane related protein sequences also participate in other functions.

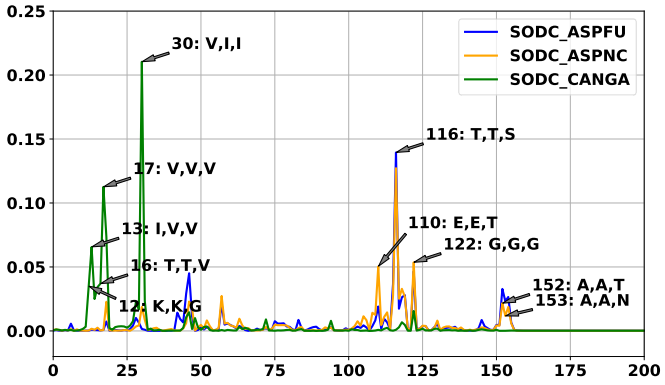

Fig. 4. Attention weights for three `sodC` gene sequences. The model pays more attention weights towards the middle of the two genes from Aspergillus species while the attention is placed at the beginning of the `sodC` gene for the Candida species

We further explore the attention weights extracted from our model. We focus on three *sodC* gene sequences from three different *fungi*, namely, *Aspergillus fumigatus Af293* (SODC_ASPFU), *Aspergillus niger CBS 513.88* (SODC_ASPNC) and *Candida glabrata CBS 138* (SODC_CANGA). According to Uniprot [22], *sodC* gene can destroy toxic radicals which are normally produced within the cells. Fig. 4 shows the attention weights for these three sequences. The protein sequence SODC_ASPFU (in blue) and

SODC_ASPNC (in orange) are pretty similar (15 amino acid differences out of 154 amino acids) whereas SODC_CANGA (in green) sequence has more amino acid variations compared with the other two sequences. In the figure, we also label the amino acid for those three sequences at high attention positions (from left to right: SODC_ASPFU, SODC_ASPNC and SODC_CANGA). From the figure, we can see that similar sequences have similar attention weights whereas different amino acid configurations can result in different attention weights. For example, in position 116, both SODC_ASPFU (in blue) and SODC_ASPNC (in orange) have high attention weights while the model doesn't pay high attention to that position in SODC_CANGA (in green) sequence. And only the SODC_CANGA (in green) has an S at that position which is different from the amino acid T shared by SODC_ASPFU (in blue) and SODC_ASPNC (in orange).

VI. Conclusion

This paper presents a deep learning based Gene Ontology prediction model that combines convolutional neural network and recurrent neural network with feed-forward attention mechanism. We evaluate our new approach using an experimental dataset and demonstrate that our method performs comparably or outperforms the baseline methods in different GO branch prediction tasks. We then show that the output of intermediate layer can be used to interpret the model. We demonstrate that the sequence with the same Gene Ontology annotation are clustered together in the embedding space by the model. Furthermore, with the help of attention weights produced by the model, users can explore the sequential information space to identify predictive regions in the sequences. As a future direction, we suggest to predict other protein labels such as protein families [26] with this framework.

Acknowledgments

This work used the Extreme Science and Engineering Discovery Environment (XSEDE) [27], which is supported by National Science Foundation grant number ACI-1548562. Specifically, it used the Bridges system, which is supported by NSF award number ACI-1445606, at the Pittsburgh Supercomputing Center (PSC). This work was supported by NSF awards #1919691 and #1936791.

References

[1] P. Radivojac, W. Clark, T. Oron, A. Schnoes, T. Wittkop, A. Sokolov, K. Graim, C. Funk, K. Verspoor, A. Ben-Hur, G. Pandey, J. Yunes, A. Talwalkar, S. Repo, M. Souza, D. Piovesan, R. Casadio, J. Cheng, and I. Friedberg, "A large-scale evaluation of computational protein function prediction," *Nature Methods*, vol. 10, pp. 221–227, 01 2013.

[2] A. Rifaioglu, T. Dogan, M. Martin, R. Cetin-Atalay, and M. V. Atalay, "Deepred: Automated protein function prediction with multi-task feed-forward deep neural networks," *Scientific Reports*, vol. 9, 12 2019.

[3] D. Hill, B. Smith, M. McAndrews-Hill, and J. Blake, "Gene ontology annotations: what they mean and where they come from," *BMC Bioinformatics*, vol. 9, pp. S2 – S2, 2008.

[4] Z. Zhao, A. Cristian, and G. Rosen, "Keeping up with the genomes: efficient learning of our increasing knowledge of the tree of life," *BMC Bioinformatics*, vol. 21, 2020.

[5] M. Ashburner, C. Ball, J. Blake, D. Botstein, H. Butler, J. Cherry, A. Davis, K. Dolinski, S. Dwight, J. Eppig, M. Harris, D. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. Richardson, M. Ringwald, G. Rubin, and G. Sherlock, "Gene ontology: tool for the unification of biology," *Nature Genetics*, vol. 25, pp. 25–29, 2000.

[6] T. G. O. Consortium, "The gene ontology resource: 20 years and still going strong," *Nucleic Acids Research*, vol. 47, pp. D330 – D338, 2019.

[7] F. Supek and N. Skunca, "Visualizing go annotations." *Methods in molecular biology*, vol. 1446, pp. 207–220, 2017.

[8] S. Altschul, T. L. Madden, A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. Lipman, "Gapped blast and psi-blast: a new generation of protein database search programs." *Nucleic acids research*, vol. 25 17, pp. 3389–402, 1997.

[9] S. M. Sahraeian, K. R. Luo, and S. Brenner, "Sifter search: a web server for accurate phylogeny-based protein function prediction," *Nucleic Acids Research*, vol. 43, pp. W141 – W147, 2015.

[10] J. Hong, Y. Luo, Y. Zhang, J. Ying, W. Xue, T. Xie, L. Tao, and F. Zhu, "Protein functional annotation of simultaneously improved stability, accuracy and false discovery rate achieved by a sequence-based deep learning," *Briefings in Bioinformatics*, vol. 21, pp. 1437 – 1447, 2020.

[11] C. Y. Yu, X. Li, H. Yang, Y. H. Li, W. W. Xue, Y. Chen, L. Tao, and F. Zhu, "Assessing the performances of protein function prediction algorithms from the perspectives of identification accuracy and false discovery rate," *International Journal of Molecular Sciences*, vol. 19, 2018.

[12] M. Kulmanov, M. A. Khan, and R. Hoehndorf, "Deepgo: predicting protein functions from sequence and interactions using a deep ontology-aware classifier," *Bioinformatics*, vol. 34, pp. 660 – 668, 2018.

[13] M. L. Bileschi, D. B. Belanger, D. H. Bryant, T. Sanderson, B. Carter, D. Sculley, M. A. DePristo, and L. J. Colwell, "Using deep learning to annotate the protein universe," *bioRxiv*, 2019.

[14] F. Emmert-Streib, Z. Yang, H. Feng, S. Tripathi, and M. Dehmer, "An introductory review of deep learning for prediction models with big data," in *Frontiers in Artificial Intelligence*, 2020.

[15] C. Raffel and D. P. W. Ellis, "Feed-forward networks with attention can solve some long-term memory problems," *ArXiv*, vol. abs/1512.08756, 2015.

[16] Q. Liu, H. Zhang, Y. Zeng, Z. Huang, and Z. Wu, "Content attention model for aspect based sentiment analysis," in *Proceedings of the 2018 World Wide Web Conference*, ser. WWW '18. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2018, pp. 1023–1032. [Online]. Available: https://doi.org/10.1145/3178876.3186001

[17] Z. Yang, D. Yang, C. Dyer, X. He, A. J. Smola *et al.*, "Hierarchical attention networks for document classification," in *HLT-NAACL*, 2016.

[18] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "Attention-based bidirectional long short-term memory networks for relation classification," in *ACL*, 2016.

[19] L. Deming, S. Targ, N. Sauder, D. Almeida, and C. J. Ye, "Genetic Architect: Discovering Genomic Structure with Learned Neural Architectures," *ArXiv*, may 2016. [Online]. Available: http://arxiv.org/abs/1605.07156

[20] Z. Zhao, S. Woloszynek, F. Agbavor, J. Mell, B. A. Sokhansanj, and G. Rosen, "Learning, visualizing and exploring 16s rrna structure using an attention-based deep neural network," *bioRxiv*, 2020.

[21] NIH. Amino Acids. (2020, October 22). [Online]. Available: https://www.genome.gov/genetics-glossary/Amino-Acids

[22] E. Boutet, D. Lieberherr, M. Tognolli, M. Schneider, P. Bansal, A. Bridge, S. Poux, L. Bougueleret, and I. Xenarios, *UniProtKB/Swiss-Prot, the Manually Annotated Section of the UniProt KnowledgeBase: How to Use the Entry View*, 01 2016, vol. 1374, pp. 23–54.

[23] L. van der Maaten and G. Hinton, "Viualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 11 2008.

[24] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," vol. 8, 01 2007, pp. 1027–1035.

[25] A. Roger, S. A. Muñoz-Gómez, and R. Kamikawa, "The origin and diversification of mitochondria," *Current Biology*, vol. 27, pp. r1177–r1192, 2017.

[26] S. El-Gebali, J. Mistry, A. Bateman, S. Eddy, A. Luciani, S. Potter, M. Qureshi, L. Richardson, G. A. Salazar, A. Smart, E. Sonnhammer, L. Hirsh, L. Paladin, D. Piovesan, S. Tosatto, and R. Finn, "The pfam protein families database in 2019," *Nucleic Acids Research*, vol. 47, pp. D427 – D432, 2019.

[27] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, R. Roskies, J. R. Scott, and N. Wilkins-Diehr, "Xsede: Accelerating scientific discovery," *Computing in Science Engineering*, vol. 16, no. 5, pp. 62–74, Sep. 2014.