



University
of Glasgow | School of
Computing Science

CAFA 5 - Protein Function Prediction

Aayush Nagpal

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

A dissertation presented in part fulfilment of the requirements of the
Degree of Master of Science at The University of Glasgow

1st September, 2023

Abstract

Protein function annotation is one of the most complex tasks in computational biology. And with the availability of high-throughput sequencing technologies, there are a large number of protein sequences that are yet to be annotated. It is important to understand the functionality of proteins in order to understand new diseases and their mechanisms, the creation of new drugs, and finding drug targets. Experimental functional characterization of proteins is an expensive and laborious process. To resolve this problem, researchers have developed computational approaches to predict protein functions. This is called automated function prediction (AFP).

This research proposes a feature-based approach to protein function prediction where protein sequences are encoded using the physicochemical and biochemical properties of the amino acids present in the sequence. The aim is to exploit the protein sequences by creating embeddings. These encodings are then fed to an LSTM network to extract features. The extracted features are used as input to a dense fully connected network. This model is evaluated using the Computational Assessment of Function Annotation (CAFA) dataset. The model reports an F_{max} score of 0.271 which is slightly inferior to BLAST ($F_{max} = 0.34$) but at the same time, it is not competitive with some of the leading approaches in the field of Protein Function Prediction.

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic format. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Name: Aayush Nagpal Signature: *Aayush*

Acknowledgements

I would like to express my gratitude to Dr. Kevin Bryson for his support and guidance throughout the project. Also, I would like to express my gratitude to my family and friends for their support during this time.

Contents

1	Introduction	5
1.1	Background	5
1.2	Critical Assessment of Function Annotation (CAFA)	6
1.3	Research Proposal	7
2	Literature Survey	8
2.1	Introduction	8
2.2	Literature Review	8
2.2.1	Homology Based Methods	8
2.2.2	Sequence Based Methods	9
2.2.3	The DeepGO Models	10
2.2.4	Feature Based Methods	11
3	Methodology	13
3.1	System Specification	13
3.2	Assumptions	13
3.3	Evaluation Criteria	13
3.4	Dataset Description	14
3.5	Data Pre-processing	15
3.6	Protein Sequence Embeddings	16
3.7	Model Architecture	18
4	Results	22
4.1	Experimental Setup	22
4.2	Results	22
4.3	Discussion	23

5	Conclusion	24
5.1	Conclusion	24
5.2	Future Scope	24
	Bibliography	25

Chapter 1: Introduction

Proteins are extremely complex, naturally occurring substances that are known as the building blocks of the body because they are needed for the growth and development of most organisms. It can be found in different parts of the body and is an important component in the making of biological components like antibodies, enzymes, hormones, etc. Proteins are made up of amino acids, which are small molecules with a common backbone composed of a few key elements: Carbon, Hydrogen, Nitrogen, and Oxygen. These amino acids join to form long chain-like structures where each amino acid is linked to the next using peptide bonds (Radivojac et al., 2013). 20 standard amino acids are commonly found in proteins. Out of these, 11 amino acids are marked as non-essential as they can be synthesized by the body, and the remaining 9 are categorized as essential because either the body cannot synthesize them or produce them in sufficient volumes. The standard amino acids have distinct chemical structures and properties contributing to their various biological roles.

Proteins often have multi-functional roles within cells, and participate in various molecular functions, cellular components, and biological processes. The specific function performed by the protein is dependent on various factors like amino-acid sequence, 3-D structure of the protein, protein-protein interactions, active sites, binding sites, etc. It's essential to understand the functionality of proteins not only from the biological aspect but also from the perspective of understanding diseases, the discovery of new drugs, functional genomics, and biotechnology.

1.1 Background

There are a wide variety of biological processes, cellular components, and molecular functions performed by proteins. The Gene Ontology (GO) (Ashburner et al., 2000) helps to structure our knowledge of the field of biology by categorising it into three major ontologies: Biological Process (BP), Molecular Function (MF), and Cellular Component (CC). These ontologies are further subdivided into multiple sub-graphs forming a Directed Acyclic Graph (DAG), where every node represents a GO term and each connecting edge represents the parent-child relationship between the corresponding GO terms. The relationships are categorized into “is_a”, “part_of”, “has_part”, or “regulates” and can be used to infer additional information. Currently, the GO database is one of the most comprehensive databases that describe the properties of proteins using a controlled vocabulary. Each GO term consists of a “Term Name” that describes the molecular activity, a unique identifier called “Term ID” that is assigned to organize the data, and a concise ‘description’ of the molecular activity.

Figure 1.1 provides a better understanding of hierarchical relationships between GO terms. It illustrates a subset of relationships for the term GO:0060491 that represents “regulation of cell project assembly” using a DAG. At its core, it is observed that there is a root node representing biological process (GO:0008150) and the graph has multiple “layers” or “levels”. There are no cycles in the graph which means that a parent-child relationship can be established between nodes present at distinct layers. For example - GO:0060491 is a (edge represents “is_a” relationship) biological regulation (GO:0065007). As mentioned earlier, the root node represents GO:0008150, encompassing a broad range of biological processes. As one delves into deeper levels, the definitions of GO terms become more specific and nuanced.

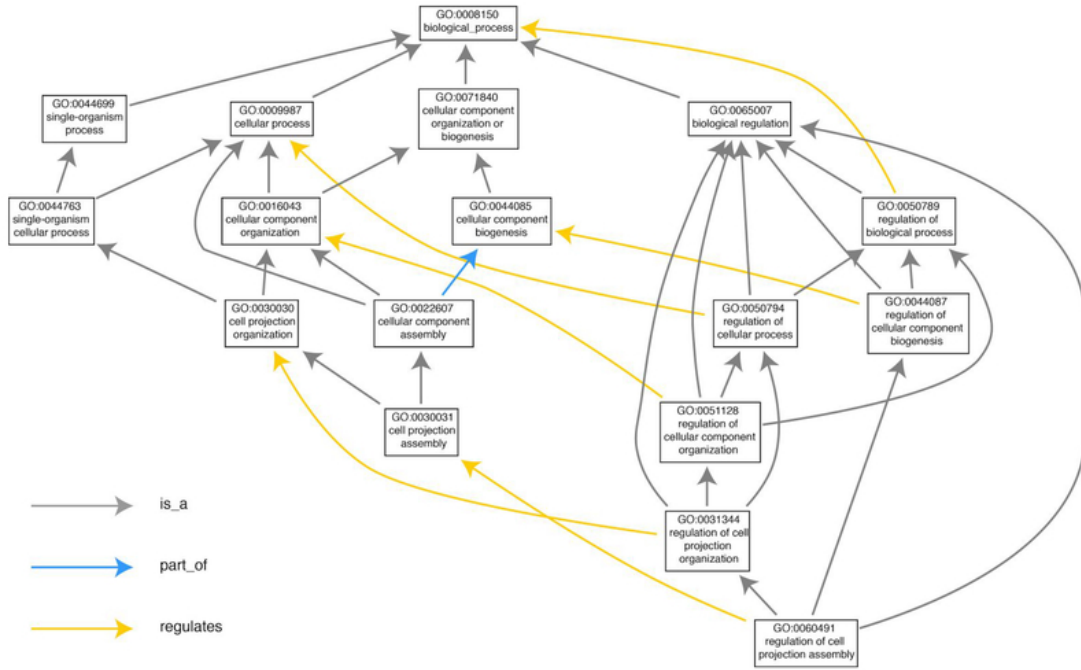


Figure 1.1: The Gene Ontology (GO) structure is explained using a subset of paths of the term “regulation of cell projection assembly,” GO:0060491, to its root term. (Gaudet et al., 2017)

The annotation of protein sequences using the GO terms is called Protein Function Identification. It is an intricate task that involves a combination of computational and experimental methods. The latter approach yields high-quality results as the experiments are performed by subject matter experts - biologists, but is a highly laborious and costly process. Another issue is that with advancements in sequencing technologies like Gene Sequencing, there is a large volume of data on protein sequences available in databases like UniProt (<https://www.uniprot.org/>), which contains more than 100 million sequences that can be synthesized. Therefore, there is a need for more rapid and cost-effective methods to predict protein function. This is where computational methods using machine learning and deep learning algorithms help by annotating large amounts of newly generated protein sequences. Computationally, protein function annotation can be treated as a multi-label classification problem where each protein sequence can be mapped to multiple GO terms. These computational models allow biologists to quickly develop hypotheses about the various functions that the newly generated protein sequences can play and/or fill the gaps between the unknown properties of proteins. As highlighted earlier, the GO terms form a DAG where the terms at deeper levels are more complex. This means that nodes at upper levels are more frequently used for annotation compared to the terms at deeper levels. This makes the terms at a deeper level challenging to predict.

1.2 Critical Assessment of Function Annotation (CAFA)

As discussed, utilizing Gene Ontology terms for Automated Protein Function Annotation is a notably intricate and demanding issue within the realm of bioinformatics. In response, research teams and organizations have endeavoured to leverage crowdsourcing platforms to devise an effective resolution. One such example is the Critical Assessment of Function Annotation (CAFA) challenge (Radivojac et al., 2013). The goal of this challenge is to predict the GO terms for a set of proteins by developing models trained on protein sequences. This challenge aims to understand the current state of the field of computational protein function prediction and drive this field forward. The evaluation metric used in CAFA (outlined in

section 3.4) is designed in a manner that gives preference to models capable of providing more precise predictions for GO terms situated at deeper levels, resulting in higher scores being assigned to such models.

1.3 Research Proposal

This research paper leverages the transformative capabilities of the Long Short-Term Memory (LSTM) networks, a class of RNN, to process sequential data of the protein sequences. The key is the transformation of amino acids present in the protein sequences into sequential data using the physicochemical or biological properties of amino acids. This data can be used as an input to the LSTM network that can then capture patterns over a long temporal range. The functionality of a protein can be determined by the sequence in which amino acids are arranged. This makes the LSTM network an ideal candidate for processing sequential information in protein sequences to predict protein functionality.

Further, the research paper is divided into 4 sections. In the subsequent section, the Literature Review discusses the previous work done in the field of Automated Protein Function Annotation and CAFA. The Methodology section provides comprehensive details on the dataset, our approach – LSTM networks, and the process of converting protein sequences into encodings. The remaining segments focus on providing a detailed analysis of the model's results and concluding the research.

Chapter 2: Literature Survey

2.1 Introduction

The complexity of the protein function prediction problem stems from multiple factors. Firstly, it arises due to the extensive array of Gene Ontology (GO) terms available for association with each protein, necessitating a multi-label classification approach. Over the past few decades, with the increase in computation power and the rise of advanced machine learning algorithms, researchers have tried to develop computational methods to solve this problem. These methods can be classified into 3 different categories: Homology-based approach, subsequence-based approach and feature-based approach.

The homology-based approach depends on the similarity score between two protein sequences. These methods assume the similarity score is correlated to function similarity which is a very weak assumption because regions of low complexity can give pseudo-high similarity scores and there are large variations observed within the proteins of the same family. Many authors ((Vu and Jung, 2021),(Kulmanov and Hoehndorf, 2020)) have mentioned the use of the Basic Local Alignment Search Tool (BLAST) (Altschul et al., 1990). It utilizes a predefined scoring matrix (example - BLOSUM62) to search a query sequence against the existing proteins in databases like UniProt and then predict the function annotations for the query sequence based on the functions of these homologous proteins. An improvement on this method is PSI-BLAST (Altschul, 1997) which uses the Position Specific Scoring Matrix (PSSM).

The subsequence-based approach looks for hidden recurrent patterns in protein sequences. This involves breaking down the protein sequences into smaller units called k-mers (the k-mer rule is described in Figure 2.1), where each k-mer represents a group of amino acids. Given that the standard set consists of 20 types of amino acids, the vocabulary size is fixed to 20. But techniques based on this approach generate k-mers (where k is greater than or equal to 2) which leads to an exponential increase in the vocabulary size. However, proteins are composed of amino acid chains that exhibit varying lengths, spanning from as small as 20 amino acids in the case of trp-cage protein to as large as 38,000 amino acids in titin. This variability in protein length poses a challenge when attempting to identify meaningful and consistent sequences of amino acids that hold structural or functional significance.

The feature-based approaches are dependent on feature extraction. The significance of the features extracted determines the quality of the results. The transformation of raw protein sequences using selective features can aid in the characterization of proteins. These techniques can also use k-mers and PSSMs as features.

2.2 Literature Review

This section explores the previous research conducted in the realm of automated protein function annotation employing diverse methodologies.

2.2.1 Homology Based Methods

Gong, Ning and Tian, (2016), proposed a homology-based approach called GoFDR which was amongst one of the top performers in CAFA-2. This algorithm uses the query sequence

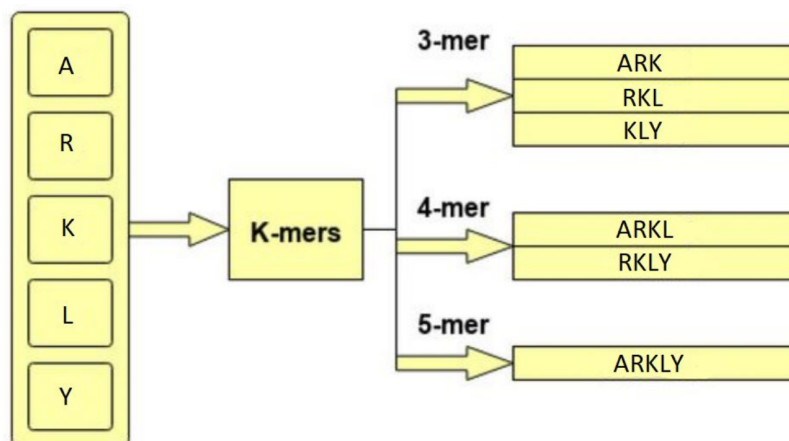


Figure 2.1: k-mer rule

and BLAST algorithm to create multiple sequence alignment (MSA). The next step is to collect all the relevant GO terms for sequences in MSA. For all the GO terms, find functionally discriminating residues (FDR) using EFICAz’s approach (Tian, Arakaki and Skolnick, 2004) with modifications and score the query sequence based on the Position-Specific Scoring Matrix (PSSM) constructed for FDR. The raw scores of the PSSM are converted to probability scores using the score-to-probability table generated using the training dataset. GoFDR underwent evaluation using a large benchmark dataset, demonstrating exceptional performance in comparison to the three established sequence-based methods: ConFunc (Wass and Sternberg, 2008), Gotcha (Martin, Berriman and Barton, 2004) and PFP (Hawkins et al., 2009).

2.2.2 Sequence Based Methods

You et al., (2018) developed a new framework called GOLabeler which is a sequence-based approach for automated function prediction (AFP) of protein sequences. GOLabeler uses 5 different sequence-based information for different components – GO term frequency for the Naïve Bayes model, BLAST-KNN (k- nearest neighbour algorithm applied on BLAST results), Logistic Regression on amino acid trigrams (3-mer), Logistic Regression on ProFET features, and Logistic Regression on InterPro features. All these components are combined using “weighted voting” to get the GO term predictions. The model was evaluated against BLAST (Altschul et al., 1990) and GoFDR (Gong, Ning and Tian, 2016). GOLabeler outperformed BLAST and GoFDR across almost all experimental settings. Another advantage of GOLabeler was its ability to annotate protein sequences using GO terms that are present at deeper levels in the gene ontology hierarchy.

With the increasing popularity of NLP (Natural Language Processing) and language models, Cao et al., (2017) published a novel method to transform a function prediction problem into a language translation problem. This method is called “ProLanGO”, and it exploits the advances made by Neural Machine Translation (NMT by Google) (Bahdanau, Cho and Bengio, 2014). ProLanGO first encodes the protein function and protein sequences into “GOLan” and “ProLan” language respectively and then uses these encodings to train the NMT model which is built on recurrent neural networks. This translates the protein sequences (ProLan) into protein functions or GO terms (GOLan). The authors introduced two novel encoding techniques: “GOLan” and “ProLan”. “ProLan” utilizes the k-mer rule and incorporates more than 400,000 amino acid k-mers. On the other hand, “GOLan” is rooted in a Depth-First Search (DFS) approach applied to the directed acyclic graph of the Gene Ontology to encode GO terms alphabetically. The model was trained using 419,192 protein

sequences that were available in the UniProt database at that point and used 130,787 protein sequences that were released as part of CAFA-3 to test the model. The model performed better than some of the prevalent function prediction methods at the time but there was still a large gap between ProLanGO and other homologous methods.

2.2.3 The DeepGO Models

Kulmanov, Khan and Hoehndorf (2018) proposed the use of protein sequences and protein-protein interaction (PPI) networks for function prediction using the DeepGO model. It combines two different forms of feature representation, one from protein sequences and the other from PPI networks. Using multiple layers of convolutional neural network (CNN) on protein sequence embeddings results in a feature map that contains redundant information that was removed through temporal max-pooling. The features obtained from max-pooling are combined with knowledge graph embeddings (PPI networks) to form a combined vector of length 832. This is then fed to a hierarchical classification layout via a fully connected layer. The hierarchically structured classification layout forms a directed acyclic graph where each node is a fully connected layer that helps in classifying a specific GO term. The DeepGO model with a hierarchical classification system outperformed the BLAST (Altschul et al., 1990) model but this slight performance improvement was not going to revolutionize the field of function prediction.

However, DeepGO has its limitations. It cannot handle protein sequences with sequence lengths greater than 1002 and it uses PPI interaction information which might not be available for all protein sequences.

To handle these shortcomings, Kulmanov and Hoehndorf, (2020) developed DeepGOPlus – an alternate approach to function prediction. DeepGOPlus uses multiple CNN layers with fixed filter lengths. Instead of using the dropout or activation function, the MaxPooling layer is used which results in every filter in the CNN layers returning at least a single value. This makes CNN learn similar patterns and if the filter finds a pattern in the sequence, then it results in a high MaxPooling value. To get the final prediction, a dense layer with a sigmoid activation function is added at the end. DeepGOPlus uses only one hot encoding of protein sequences which helps in reducing the number of parameters in comparison to DeepGO. To evaluate the model, the authors used a benchmark CAFA-3 dataset where DeepGOPlus was not only successful in beating Naïve Bayes, DiamondBLAST (variation of BLAST method), and GOLabeler but also had an edge against its predecessor DeepGO.

Apart from an improved performance, DeepGOPlus tries to minimize the limitations of its predecessor as it can handle protein sequences with 2000 amino acids. As mentioned earlier PPI interactions might not be available for novel protein sequences and this is where the sequence-motif-based function prediction ability of DeepGOPlus is more suitable.

As part of the evolution of DeepGO architecture, Zhang et al., (2021) proposed DeepGOA, an improved variation of DeepGO model that makes predictions about protein functions by utilizing protein sequences and protein-protein interaction (PPI) networks. The protein sequences are vectorized using ‘word2vec’ and a Bi-LSTM network is used to extract global features. Then, a multi-scale Convolutional Neural Network (CNN) is used to obtain local-level features. To extract topological features from the PPI network, the Deepwalk algorithm (Perozzi, Al-Rfou and Skiena, 2014) was used. Finally, all the distinct features are combined and fed to the classification section. The classification section is a dense layer with a sigmoid activation function.

The DeepGOA model is an enhancement as the DeepGO model completely ignores global

information present in the sequence. Also, the DeepGOA model uses a filtered PPI network which is about 4% the size of the original PPI network used in DeepGO. On testing the model with the benchmark evaluation dataset of CAFA-3, the model outperformed the DeepGO, FFPred3 (Cozzetto et al., 2016), and BLAST (Altschul et al., 1990) across all 5-evaluation methods (F_{max} , Average Precision, Average Recall, MCC, and Area Under Curve (AUC)) but DeepGOPlus (Kulmanov and Hoehndorf, 2020) still had a slight edge over it.

2.2.4 Feature Based Methods

You, Huang and Zhu, (2018) also proposed a close competitor of DeepGOPlus in the form of DeepText2GO. It uses citations from the MEDLINE database and its protein sequences to train the model. Based on the UniProt IDs of the target protein the authors extract all PubMed identifiers (PMIDs) to retrieve corresponding citations to form a single document. To extract features from this document the authors use text-based techniques – TFIDF (Term Frequency- Inverse Document Frequency), D2V (Document to Vector), D2V-TFIDF. These features are used to obtain a GO term score for the corresponding target protein using a Logistic Regression (LR) model. Now, to obtain features from protein sequences the authors used InterPro as it combines 14 different databases including Pfam, CCD, and CATH-Gene3D. The features from InterPro are used to train another Logistic Regression (LR) model. Additionally, BLAST-KNN – a homology-based method is used to calculate protein similarity score. DeepText2GO uses a weighted consensus approach to integrate the results of different models. DeepText2GO can predict 28,000 GO terms which is a significant improvement over models like DeepGO (Kulmanov, 2018). When evaluated using a set of 1600 protein sequences, the model proved to be superior to DeepGO, BLAST, BLAST-KNN, and Naïve Bayes across all 3-evaluation metrics - F_{max} , S_{min} , and AUPR (Area Under Precision-Recall) for all three categories (BP, MF, and CC).

Ranjan et al., (2019) developed a novel approach to generate protein vectors called ‘ProtVecGen’. This approach was founded on the assumption that a protein’s functionality is contingent on a limited set of subsequences referred to as “conserved sequences.” These specific subsequences hold a direct influence over the protein’s functionality, while the remaining subsequences, termed “non-conserved” are seen as inconsequential to the protein’s function and are treated as extraneous noise. As the length of the protein sequence increases, the “noise” also escalates.

ProtVecGen leverages the identified conserved sequences and employs a bi-directional Long Short-Term Memory (Bi-LSTM), a type of Recurrent Neural Network (RNN), to construct protein vectors. These vectors are then channelled into a fully connected dense network characterized by a single hidden layer, governed by a sigmoid activation function. The resultant output is subsequently fed to a layer of Support Vector Machines (SVMs), which are separately trained. The model was evaluated using precision, recall and F1-score, and demonstrated superior performance compared to numerous existing models. Nonetheless, a notable limitation in this research pertains to the authors’ utilization of a small subset of 295 Gene Ontology (GO) terms, which represents a minuscule fraction of the 40,000 possible GO terms.

Kipf and Welling, (2016) introduced the concept of Graph Convolutional Networks (GCN) as part of their research on “Semi-Supervised Classification with Graph Convolutional Networks”. Their work laid the foundation for researchers to utilize neural networks for learning representations from graph-structured data. You et al., (2021) proposed DeepGraphGO, an end-to-end, multi-species graph neural network-based method for protein function prediction. The model takes in N binary feature vectors, generated by InterProScan and a protein

network graph which has N nodes. The model has a fully connected layer which converts the binary features into non-binary vectors. There are 2 GCN layers that use the non-binary vector representation of each node to capture high-order information and the final layer is a fully connected layer used to predict the confidence scores for the GO terms. Across all the evaluation metrics DeepGraphGO was far ahead of its competitors like LR-InterPro, DeepGOCNN, DeepGoPlus, BLAST-KNN, and Net-KNN with the highest F_{max} and AUPR score across all the categories – BP, MF and CC.

In this section, the focus was on different methods developed over the last few years. Since the inception of the CAFA (Critical Assessment of Function Annotation) challenge researchers have been trying to develop a deep-learning model that can reliably predict protein functionality. Over the years, continuous improvement has been observed in the abilities of models to correctly annotate the GO terms. From GoFDR to DeepGraphGO, there has been a substantial enhancement in the performance of these models. However, there is a huge scope for improvement in the performance of these models.

Chapter 3: Methodology

The primary objective of this research paper is to develop a robust technique that can aid in the prediction/annotation of protein functions. To achieve this task, a simplified yet novel approach is designed. As discussed earlier, proteins contain sequential information in the form of chains of amino acids. The idea is to encode these sequences using aaIndex1 properties (Kawashima et al., 1999) (explained in section 3.6) and then use a special type of recurrent neural network (RNN) known as Long Short-Term Memory (LSTM) for classification.

But before delving into the details of the model, it is imperative to discuss some of the other key aspects. These include system specification used during model development, underlying assumptions guiding the research, evaluation criteria, a brief description of the dataset used and data pre-processing strategy.

3.1 System Specification

For model development, Kaggle notebooks that are equipped with 13 GB of RAM and a powerful Nvidia Tesla P100 GPU with a capacity of 15.9 GB were used. Our model was meticulously crafted within this environment, and once trained, it was saved for future use. Subsequently, a separate script was employed to load the trained model and perform inference on our test data. This strategy was chosen primarily because Kaggle allows 30 hours per week for GPU utilization. However, it is important to note that the inference phase, in contrast to the resource-intensive training phase, does not necessarily require a GPU. Hence, a computing system equipped with 30 GB of RAM proved to be more than sufficient for conducting inference tasks and hence, saved the GPU hours for training purposes only.

3.2 Assumptions

Due to the nature of the data, associated complexities, and system limitations, the following assumptions have been considered:

1. The protein sequences in the CAFA-5 dataset contain ambiguous amino acid codes represented using B, J, O, U, X, and Z. These letters represent more than one amino acid. To deal with this ambiguity, these amino acids are excluded from the protein sequences.
2. The dataset contains amino acid sequences that are longer than 35000 amino acids. Since such large sequences comprise only a small fraction (about 1%) of the dataset we truncate the sequences to a maximum length of 1000 amino acids (AA) and assume that this should not impact the model's ability to predict the GO terms correctly.

3.3 Evaluation Criteria

The evaluation approach is the same as the one adopted in the CAFA challenge. Each output of the predictor model is a value between 0 to 1 which represents the confidence score of the model for the respective GO term. The greater the confidence score, the higher the level of certainty in prediction. Because the model assigns confidence scores to all potential GO

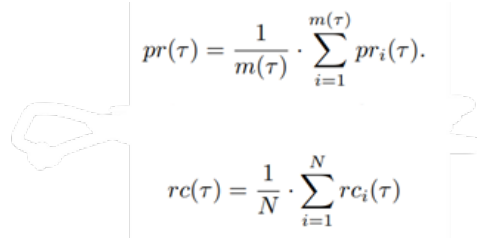
terms (in this case 1500 GO terms), it is important to define a decision threshold, represented using τ that can help to determine the set of predicted terms. For a given protein i with threshold τ , precision and recall can be defined as follows:

$$pr_i(\tau) = \frac{\sum_{v \in \mathcal{O}} I(v \in P_i(\tau) \wedge v \in T_i)}{\sum_{v \in \mathcal{O}} I(v \in P_i(\tau))}$$

$$rc_i(\tau) = \frac{\sum_{v \in \mathcal{O}} I(v \in P_i(\tau) \wedge v \in T_i)}{\sum_{v \in \mathcal{O}} I(v \in T_i)},$$

Figure 3.1: Precision and Recall for protein i (Jiang et al., 2016)

where (Figure 3.1) I represent an indicator function, $P_i(\tau)$ denotes the set of GO terms for which the confidence score is greater than τ and T_i denotes the ground truth values (Jiang et al., 2016). To find the average precision and recall the following formula is used:



$$pr(\tau) = \frac{1}{m(\tau)} \cdot \sum_{i=1}^{m(\tau)} pr_i(\tau).$$

$$rc(\tau) = \frac{1}{N} \cdot \sum_{i=1}^N rc_i(\tau)$$

Figure 3.2: Average Precision and Recall (Jiang et al., 2016)

where $m(\tau)$ are proteins over which at least one prediction is made, and N represents all the proteins in the test dataset. To provide a single score for the evaluation of the model F-measure is used. It is defined as a harmonic mean between precision and recall at a specific threshold and has a value between 0 to 1. F-measure is applied over all thresholds.

$$F_{\max} = \max_{\tau} \left\{ \frac{2 \cdot pr(\tau) \cdot rc(\tau)}{pr(\tau) + rc(\tau)} \right\}$$

Figure 3.3: F_{\max} score formula (Jiang et al., 2016)

The F_{\max} values are calculated as an average across the three categories: BP, MF and CC. The average F_{\max} value indicates the quality of the model.

3.4 Dataset Description

This study uses two datasets: one for training and the other for testing. These datasets were officially released as part of the CAFA-5 challenge. The organisers of CAFA-5 provide participants with GO term annotations which were determined experimentally for about 150,000 protein sequences. These annotations are used as class assignments for each protein. This means that if a protein is labelled with a GO term, then the protein has this function validated using either an experiment designed by biologists or high-throughput evidence or traceable author statement (TAS) or inferred by curator (IC). The absence of a term annotation does not necessarily always mean that the protein does not perform this function, it could imply

that the annotation has not been established yet or that the protein genuinely does not perform that specific function. As mentioned earlier, each protein can be mapped to multiple GO terms and each GO term can be categorized as either BP, MF or CC. Figure 3.4 shows the distribution of protein sequences across these categories.

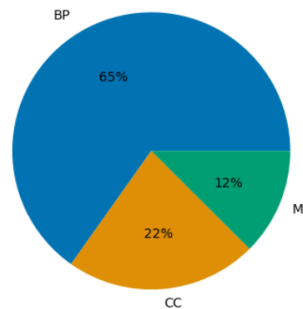


Figure 3.4: Distribution of protein sequences across BP, MF or CC

Source: CAFA-5 starter notebook shared by organizers

Similarly, the test dataset contains about 142,000 protein sequences. For about half the sequences the ground truth values are available but it would be interesting to see if the models can help scientists test the protein for a new functionality. The annotations for the other half of protein sequences are completely unknown and scientists would develop a hypothesis based on the submission made by participants during the CAFA-5 challenge.

3.5 Data Pre-processing

Deep learning algorithms require a large amount of training data for each distinct category to learn complex patterns, recognize hierarchical representation and avoid overfitting. The given training dataset contains 31,466 unique GO terms. Unfortunately, a considerable proportion of these GO terms are mapped to only a limited number of protein sequences, rendering them inadequate to train a model.

To curate a more meaningful and representative dataset, a strategic approach was adopted. This approach concentrates on the 1,500 most frequently annotated GO terms in the dataset. Because of this, the trained model will only be able to forecast inside this subset of 1500 GO terms. This deliberate selection means that the model's predictions should exhibit higher frequency and relevance.

Figure 3.5 shows the top 100 frequent GO terms and the number of protein sequences associated with them. It clearly shows that the dataset is imbalanced because a significant portion of the data is represented by a set of specific labels, while the remaining GO terms remain under-represented. However, this skewness can be attributed to the hierarchical nature of the labels.

This imbalance in the dataset can be attributed to the fact the GO terms are represented by using DAG (Direct Acyclic Graph). In other words, when a protein sequence is annotated using a GO term then the likelihood of it being mapped to its parent GO term also increases.

The selection of the most frequent GO terms means that the distribution of BP, MF and CC in the training dataset also changes which is highlighted in the chart [Figure 3.6]

Another important step to solving such a complex computational problem is to understand the underlying data and clear any obvious noise in the data. Based on the assumptions defined

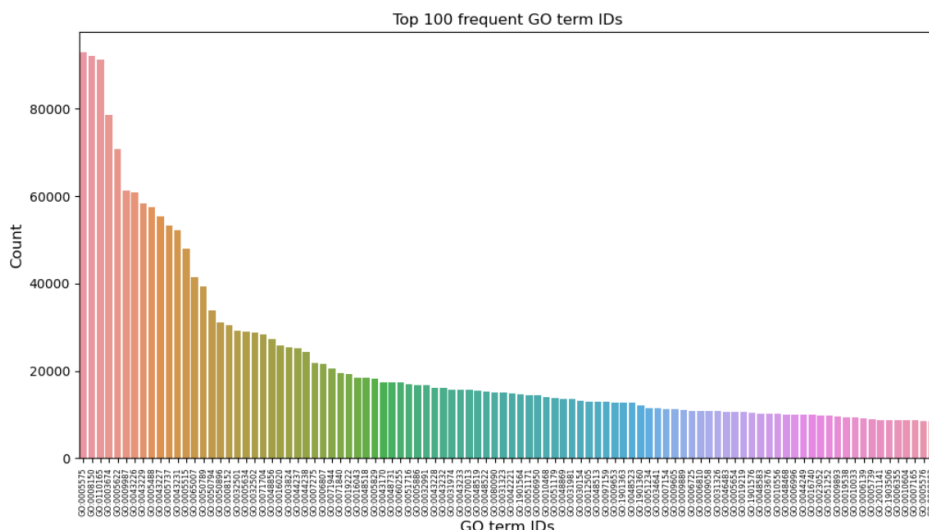


Figure 3.5: Distribution across GO term IDs
Source: CAFA-5 starter notebook shared by organizers

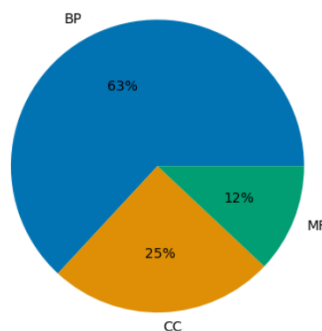


Figure 3.6: Distribution of protein sequences across BP, MF or CC after data pre-processing
Source: CAFA-5 starter notebook shared by organizers

in the section above, the first step is to remove any uncertainty in the data by removing the ambiguous amino acids that are part of the protein sequences and then truncating any protein sequence with more than 1000 amino acids chained together. This process is applied to both training and testing data.

3.6 Protein Sequence Embeddings

As described earlier, proteins are complex molecules that are formed when a set of amino acids form chain-like structures. The sheer complexity and large volume of data in protein sequences pose a challenge in extracting meaningful features from these protein sequences. This is where the concept of embeddings proves to be valuable. Protein sequence embeddings provide a way to transform raw amino acid sequences into a numerical representation that has the potential to capture essential structural and/or functional properties.

This research proposes the use of the AAindex database (Kawashima et al., 1999) to build protein sequence embeddings. The AAindex database is a flat-file database. It consists of numerical indices representing various physicochemical and biochemical properties of amino acids or pairs of amino acids. This database contains 3 sections: AAindex1 contains more than 550 physicochemical and biochemical properties of amino acids represented using

an index of 20 numerical values, AAindex2 represents an amino acid mutation matrix of 210 numerical values and AAindex3 is for the statistical protein contact potentials.

Access to the AAindex database is facilitated through a lightweight 'aaindex' Python package. It simplifies access to physicochemical and biochemical properties which can be accessed using a record code. Each record code allows access to the property values using a dictionary structure. The single-letter representations of the amino acids are the 20 keys in this dictionary, while the associated values provide access to the specific property information.

To create embeddings, we encode each amino acid in a protein sequence using a small portion of selected AAindex1 properties as described below:

1. **Hydrophobicity index** – Hydrophobicity is the ability of molecules or part of molecules to repel water. In the context of proteins, hydrophobic amino acids often cluster together in the protein's interior, contributing to protein stability and protein folding. Amino acids with higher values are more hydrophobic.
2. **Hydrophilicity value** - Hydrophilicity is the ability of molecules or parts of molecules to interact with water. Hydrophilic amino acids are often found on the protein's surface where they can interact with water.
3. **Flexibility parameter for no rigid neighbours** - The Karplus-Schulz model's proposed flexibility parameter for no stiff neighbours assigns a numerical value to each amino acid that indicates its level of relative flexibility. When compared to a less flexible amino acid with a lower value, an amino acid with a higher value is more likely to undergo conformational changes or movements.
4. **Polarizability parameter** - It refers to the ability of molecules to generate induced electric dipole moments when subjected to the electric field. It defines the behavior of amino acids and influences the interaction with surrounding molecules which influences phenomena such as protein-ligand interactions, protein-protein binding, etc.
5. **Normalized van der Waals volume** - It can be defined as atomic or molecular volume defined by van der Waals radii. The van der Waals radius is the distance from the nucleus where the electron cloud ends.
6. **Polarity** - Zimmerman defined polarity as the distribution of electric charge in a molecule's overall dipole moment. It influences the way amino acids interact at the intermolecular level and impacts the solubility of proteins.
7. **Isoelectric point** - It is defined as the pH level at which the amino acid has no charge. Accordingly, the proteins are either positively charged, negatively charged or neutral.
8. **Bulkiness** – Some of the amino acids have long bulky side chains, making them bulky whereas other amino acids might occupy less space because of smaller chains. This leads to an increase in the radius of the molecule.
9. **Molecular weight** - It is the sum of the weight of all atoms in a molecule. It is expressed using the atomic mass unit or Dalton (Da).
10. **Average accessible surface area** - A protein folds in such a way that the hydrophobic parts are inside the protein's core whereas the hydrophilic part is exposed to the surroundings. The average surface area provides information on how exposed or hidden an amino acid residue is within a protein structure.

11. **alpha-CH chemical shifts** - It is a measure of resonant frequencies of C (carbon) and H (hydrogen) bonds.
12. **Residue volume** - It is the spatial size or volume occupied by an amino acid in the 3-D structure of a protein.
13. **Relative mutability** - As defined by Dayhoff, relative mutability is directly proportional to the ratio of the number of changes to the number of occurrences.
14. **Localized electrical effect** - This defines the influence of a charged group on the electron distribution of the amino acid.
15. **Steric parameter** - It refers to obstruction caused by atoms or a group of atoms that prevents interaction with other groups of atoms or molecules. This impacts how protein-protein interaction and ligand-protein interaction work.

Each amino acid can be represented by a vector of 15 elements, where each element of the vector stands for one of the previously mentioned qualities. In other words, if a protein sequence is of length 200 then it is represented using a 2D embedding of size [200,15]. But to ensure that all protein sequences have the same length we add padding which changes the size to [1000, 15]. This is done based on our assumption that the maximum length of protein sequences is fixed to 1000 amino acids. The process to generate embeddings is defined in Figure 3.7

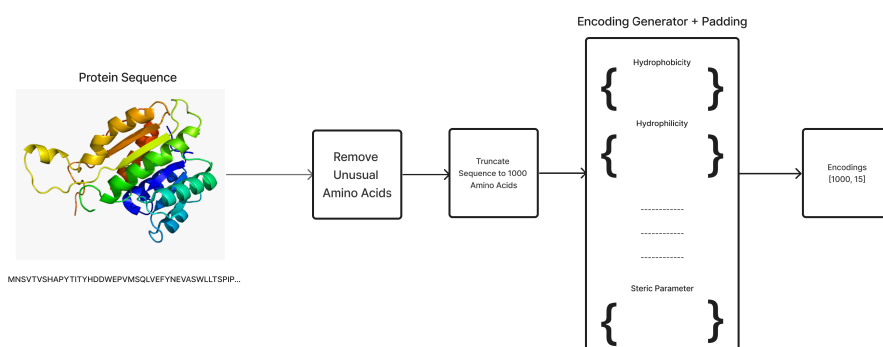


Figure 3.7: From Protein Sequences to Protein Embedding

3.7 Model Architecture

The structure of a protein is determined by the sequence of amino acids which further determines the functionality of the protein. The protein sequence embeddings are created by using numerical representations of amino acids as highlighted in the method described in section 3.6. These embeddings are fed to a Long Short-Term Memory (LSTM) neural network, a special class of recurrent neural network (RNN).

Traditional networks do not use information about previous events in sequence to inform the events that occur later in the sequence. This is where a Recurrent Neural Network (RNN) comes into the picture. It has the intrinsic ability to recognize patterns and characteristics in sequential data. All RNNs have a repeating module which forms a chain-like structure. Figure 3.8 shows how RNN forms a chain that allows information to persist where 'A' represents a very simple structure that uses a logistic sigmoid or tanh layer.

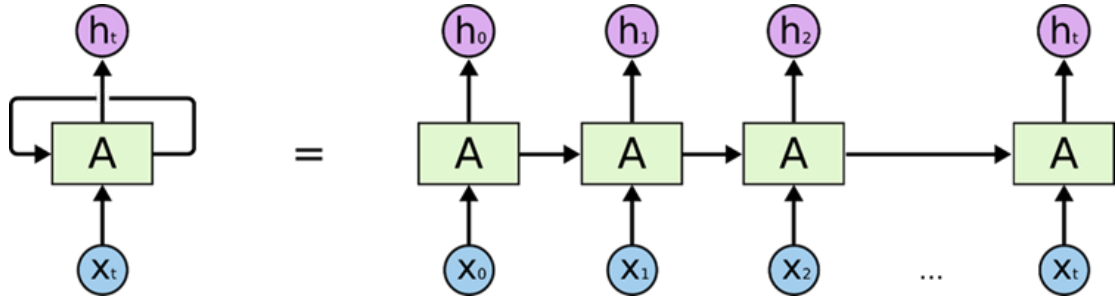


Figure 3.8: A simple RNN structure

Source: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

The RNN leverages Backpropagation through time (BPTT) to perform a backward pass to adjust the model's parameters. BPTT is different from the usual backpropagation as BPTT adds up errors across each timestep because RNN shares the same parameters across each layer. Due to this, the RNN faces the problem of vanishing/ exploding gradient where the gradient values become either too small or too large respectively.

Theoretically, RNNs are capable of handling “long-term” dependencies. This means that RNN should be able to handle scenarios where a substantial temporal gap exists between the timesteps where the information is available and where the information is required. In reality, RNNs cannot remember long-term dependencies due to the vanishing gradient problem.

To resolve this Hochreiter and Schmidhuber, (1997) proposed the idea of Long Short-Term Memory, a special class of RNN. The LSTM networks are structurally quite different from the basic RNN where the repeating module is just a simple activation function. The repeating structure in LSTM is composed of 4 distinct components as shown in Figure 3.9.

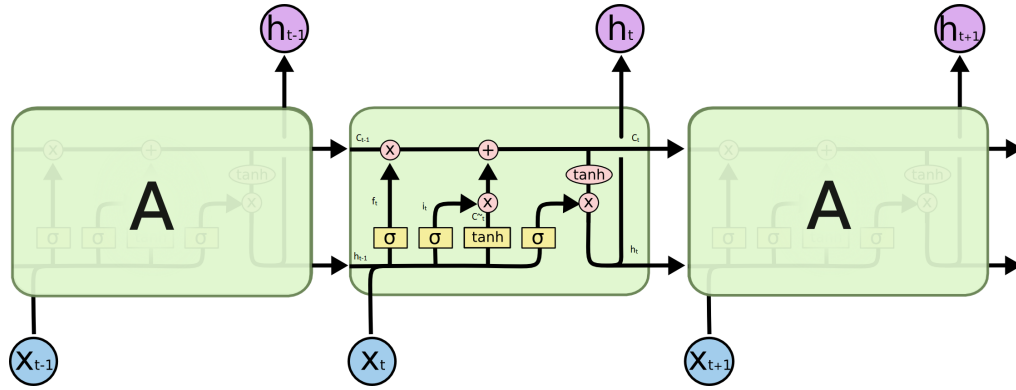


Figure 3.9: A simple LSTM structure

Source: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

$$f_t = \sigma (W_f . [h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma (W_i . [h_{t-1}, x_t] + b_i)$$

$$\hat{C}_t = \sigma (W_C . [h_{t-1}, x_t] + b_C)$$

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t$$

Figure 3.10: Equations describing key LSTM components

LSTM units are very similar to RNN but structurally they use gates to resolve the problem of long-term dependencies. These gates control how the information propagates through the network. Forget gate (f_t) is used to decide if the information from previous timesteps is required or not. If the information is not relevant, then it is forgotten. The output of this gate is between 0 to 1. The input gate (i_t) defines the importance of the new information and how much of this new information is required. The output of this gate is between 0 to 1. As we can see in Figure 3.10, the cell state at timestep t (C_t) is defined using two terms. First, multiply the output of the forget gate with the cell state of the previous timestep. Second, multiply the value of the input gate with the new information at timestep t .

$$C_t = (f_t * C_{t-1}) + (i_t * \hat{C}_t)$$

The equation above makes it quite clear that the cell state can be updated through a controlled flow of information and does not get attenuated over time. The output gate is very similar to the other gates as the value of this gate is defined between 0 to 1. To calculate the output of the current unit, multiply o_t with $\tanh(C_t)$.

$$H_t = o_t * \tanh(C_t).$$

Based on the equations above and in Figure 3.10, it is evident that using cell state (C_t / C_{t-1}), forget gate (f_t), and candidate value (\hat{C}_t) the LSTM networks can address the problem of long-term dependencies. The inherent ability of LSTM networks to handle variable input length and long-term dependencies via the use of gating mechanisms makes it an ideal candidate for protein function prediction.

As the functionality of a protein depends on the specific sequence in which the amino acids are arranged, therefore, it is reasonable to assert that the nature of the underlying data is sequential. This further highlights that the LSTM network is a good choice for solving the Automated Function Prediction problem.

Therefore, in this research, we propose the use of a 3-layer LSTM network with 100 hidden units in each layer. The input to this network is AAindex1 encodings of the amino acids present in a protein sequence. The output of the last hidden state in the final layer, which is a vector of size 100 is treated as features extracted from the protein sequence.

To use the features extracted from protein embeddings by LSTM, dense layers are added for the prediction of GO term annotations. The feature vectors generated by LSTM are passed as input to a fully connected network of 3 dense linear layers with the ReLU activation function and the last layer (4th layer) that uses the sigmoid activation function. The network takes the input tensor of size 100 to return a tensor of size 1500 where each element represents the probability of the corresponding label being annotated as GO term for that sequence. Figure 3.11 illustrates the entire process.

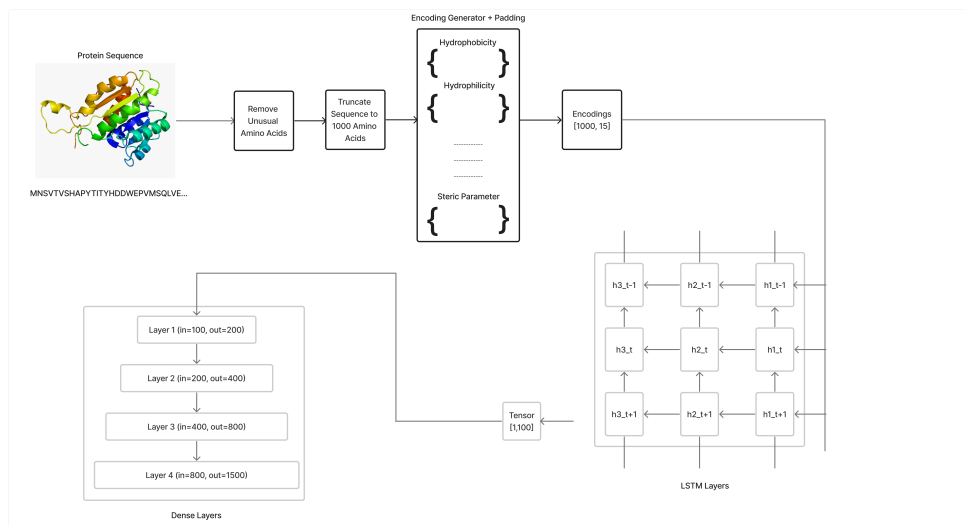


Figure 3.11: shows the end-to-end pipeline

Chapter 4: Results

In this chapter, we will evaluate our models and analyze the results. But before that, let's first understand the experimental setup.

4.1 Experimental Setup

A set of experiments was devised where the model (as defined in the previous chapter) is trained using varying AAindex1 properties. Because of the time and resource constraints under which this study was performed, only 15 AAindex1 properties were considered. The experiment began by utilizing only 5 AAindex1 properties, followed by the incorporation of 5 additional properties. The final experiment uses the complete set of 15 AAindex1 properties to train the model. This means that in each experiment, the size of the vector representing the amino acid increases by 5.

This approach allows us to systematically measure the impact of increasing the number of AAindex1 properties on the model's performance. The groups described below define the properties used in each experiment to encode amino acids:

Group 1: Hydrophobicity, Hydrophilicity, Flexibility parameter for no rigid neighbours, Polarizability parameter, Normalized van der Waals volume

Group 2: Hydrophobicity, Hydrophilicity, Flexibility parameter for no rigid neighbours, Polarizability parameter, Normalized van der Waals volume, Polarity, Isoelectric Point, Bulkiness, Molecular Weight, and Average Accessible Surface Area

Group 3: Hydrophobicity, Hydrophilicity, Flexibility Parameter for no Rigid Neighbors, Polarizability Parameter, Normalized van der Waals volume, Polarity, Isoelectric Point, Bulkiness, Molecular Weight, Average Accessible Surface Area, alpha-CH chemical Shifts, Residue Volume, Relative Mutability, Localized Electrical Effect, and Steric Parameter

In each experiment, the same end-to-end pipeline is used as described in Figure 3.7. The only difference is the number of AAindex1 properties used for encoding the protein sequences. The models are evaluated on the test dataset, for which the F_{max} score is reported in Table 4.1.

4.2 Results

The results of the experiment are described in Table 4.1

Table 4.1: F_{max} score and properties used in each experiment.

Experiment	Properties	F_{max}
1	Group 1	0.2537
2	Group 2	0.2653
3	Group 3	0.2710

The first experiment is to check if only Group 1 properties are sufficient to annotate the proteins, but an F_{max} score of 0.25 proves that the model performed poorly. This poor performance can be attributed to the fact that only 5 features were used to capture the complex patterns in the dataset.

In the next experiment, Group 2 is used, which means that each amino acid is now represented using an array of size 10. The outcome of this experiment is a model with a marginally improved F_{max} score of 0.2653. Though there is a minuscule improvement in the F_{max} score, it is not acceptable to have such a low F_{max} score.

Finally, we use all 15 properties (Group 3) to encode the amino acids in a protein sequence. The expectation was that this model would give improved performance, but the model had a F_{max} score of 0.271, which shows a marginal improvement in the F_{max} score compared to the previous experiment.

It can be observed from Table 4.1 that all three experiments generated a model with an F_{max} score in the range of 0.25-0.28. While there is a slight improvement in the F_{max} score as the number of properties used for encoding protein sequence increases, the overall performance of models is slightly questionable across all three experiments.

4.3 Discussion

From Table 4.1 it can be inferred that the experiments did not yield a reliable model when compared to some of the top-performing models that participated in the CAFA-5 challenge. The poor performance of the models across all three experiments raises questions about the quality of the protein encodings. It proves that using only 15 AAindex1 properties that represent specific physicochemical and biochemical properties is not enough to capture the nuances of Automated Function Prediction (AFP). Looking at the table above it can be observed that increasing the number of properties to encode amino acids improves the F_{max} score slightly. There are more than 550 properties in the AAindex1 database and adding more of these properties could help improve the F_{max} score. However, adding new properties to encode amino acids will not improve the F_{max} continuously but this improvement in F_{max} will stagnate at some point. The selection of physicochemical and biochemical properties requires comprehensive knowledge of proteins and computational biology.

Another contributing factor to the low F_{max} score is the nature of the problem. The CAFA challenge is a multi-label classification problem where the labels are related to each other such that they can be represented using a Directed Acyclic Graph (DAG). Due to the hierarchical nature of this representation, a large majority of the proteins are annotated with labels present at the upper levels of the DAG. This creates a class imbalance problem which leads to bias towards dominant labels. This bias leads to a lower F_{max} score.

Chapter 5: Conclusion

5.1 Conclusion

As part of this study, we proposed the use of the AAindex1 database to create encodings for protein sequences. These encodings are then fed to a 3-layer LSTM network, with each layer having 100 hidden units. The features extracted by the LSTM network are passed through 4 fully connected layers. The results were quite underwhelming, as the model using 15 properties from the AAindex1 database to encode each amino acid was the top-performing model with a F_{max} score of 0.271. The F_{max} emphasises the fact that using only 15 properties may not be sufficient. Although increasing the length of amino acid encoding by using additional AAindex1 properties could help improve model performance, but only to a certain extent. Therefore, the hypothesis that encoding amino acids in protein sequences using only selected 15 AAindex1 properties will provide enough information for the model to annotate new protein sequences accurately does not hold up as is evident from low F_{max} scores in all experiments.

This approach has its limitations. There is no information available to the model to interpret that the relationship between labels is hierarchical in nature. Additionally, there could be a potential issue with the assumption that all unorthodox amino acids that are part of the sequence should be removed. This could result in a loss of information.

5.2 Future Scope

Overall, there is still a lot of scope for improvement, even with this approach. One of the most important steps is the selection of AAindex1 properties that are relevant to protein function prediction because any physicochemical property that does not play any role in determining protein function will act as noise.

Currently, the model focuses on sequential dependencies in protein sequences. Another option is to create a hybrid model by using single- and multi-headed attention layers. Such a hybrid model can capture different levels of relationships and recognize complex relationships and dependencies in the input sequence.

Though this research does not yield a model that can successfully predict all the relevant GO terms, it can still be used as a stepping-stone to create a hybrid model that can not only recognize the complex patterns in the underlying data but also recognize the hierarchical relationship between the labels.

Bibliography

- [1] S. Altschul. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, September 1997.
- [2] Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, October 1990.
- [3] Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, J. Michael Cherry, Allan P. Davis, Kara Dolinski, Selina S. Dwight, Janan T. Eppig, Midori A. Harris, David P. Hill, Laurie Issel-Tarver, Andrew Kasarskis, Suzanna Lewis, John C. Matese, Joel E. Richardson, Martin Ringwald, Gerald M. Rubin, and Gavin Sherlock. Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, May 2000.
- [4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. 2014. Publisher: arXiv Version Number: 7.
- [5] Renzhi Cao, Colton Freitas, Leong Chan, Miao Sun, Haiqing Jiang, and Zhangxin Chen. ProLanGO: Protein Function Prediction Using Neural Machine Translation Based on a Recurrent Neural Network. *Molecules*, 22(10):1732, October 2017.
- [6] Wyatt T. Clark and Predrag Radivojac. Information-theoretic evaluation of predicted ontological annotations. *Bioinformatics*, 29(13):i53–i61, July 2013.
- [7] Domenico Cozzetto, Federico Minneci, Hannah Curren, and David T. Jones. FFPred 3: feature-based function prediction for all Gene Ontology domains. *Scientific Reports*, 6(1):31865, August 2016.
- [8] Pascale Gaudet, Nives Škunca, James C. Hu, and Christophe Dessimoz. Primer on the Gene Ontology. In Christophe Dessimoz and Nives Škunca, editors, *The Gene Ontology Handbook*, volume 1446, pages 25–37. Springer New York, New York, NY, 2017. Series Title: Methods in Molecular Biology.
- [9] Qingtian Gong, Wei Ning, and Weidong Tian. GoFDR: A sequence alignment based method for predicting protein functions. *Methods*, 93:3–14, January 2016.
- [10] Troy Hawkins, Meghana Chitale, Stanislav Luban, and Daisuke Kihara. PFP: Automated prediction of gene ontology functional annotations with confidence scores using protein sequence data. *Proteins: Structure, Function, and Bioinformatics*, 74(3):566–582, February 2009.
- [11] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, November 1997.
- [12] Yuxiang Jiang, Tal Ronnen Oron, Wyatt T. Clark, Asma R. Bankapur, et al. An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biology*, 17(1):184, December 2016.

- [13] S. Kawashima, H. Ogata, and M. Kanehisa. AAindex: Amino Acid Index Database. *Nucleic Acids Research*, 27(1):368–369, January 1999.
- [14] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. 2016. Publisher: arXiv Version Number: 4.
- [15] Maxat Kulmanov and Robert Hoehndorf. DeepGOPlus: improved protein function prediction from sequence. *Bioinformatics*, 36(2):422–429, January 2020.
- [16] Maxat Kulmanov, Mohammed Asif Khan, and Robert Hoehndorf. DeepGO: predicting protein functions from sequence and interactions using a deep ontology-aware classifier. *Bioinformatics*, 34(4):660–668, February 2018.
- [17] David Ma Martin, Matthew Berriman, and Geoffrey J Barton. GOtcha: a new method for prediction of protein function assessed by the annotation of seven genomes. *BMC Bioinformatics*, 5(1):178, 2004.
- [18] Kenta Nakai, Akinori Kidera, and Minoru Kanehisa. Cluster analysis of amino acid indices for prediction of protein structure and function. *"Protein Engineering, Design and Selection"*, 2(2):93–100, 1988.
- [19] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. DeepWalk: online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, New York New York USA, August 2014. ACM.
- [20] Predrag Radivojac, Wyatt T Clark, Tal Ronnen Oron, Schnoes, et al. A large-scale evaluation of computational protein function prediction. *Nature Methods*, 10(3):221–227, March 2013.
- [21] Ashish Ranjan, Md Shah Fahad, David Fernandez-Baca, Akshay Deepak, and Sudhakar Tripathi. Deep Robust Framework for Protein Function Prediction using Variable-Length Protein Sequences. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pages 1–1, 2019.
- [22] Ashish Ranjan, Archana Tiwari, and Akshay Deepak. A Sub-Sequence Based Approach to Protein Function Prediction via Multi-Attention Based Multi-Aspect Network. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pages 1–1, 2021.
- [23] The Gene Ontology Consortium. The Gene Ontology: enhancements for 2011. *Nucleic Acids Research*, 40(D1):D559–D564, January 2012.
- [24] Weidong Tian, Adrian K. Arakaki, and Jeffrey Skolnick. EFICAz: a comprehensive approach for accurate genome-scale enzyme function inference. *Nucleic Acids Research*, 32(21):6226–6239, 2004.
- [25] Thi Thuy Duong Vu and Jaehee Jung. Protein function prediction with gene ontology: from traditional to deep learning models. *PeerJ*, 9:e12019, August 2021.
- [26] Mark N. Wass and Michael J. E. Sternberg. ConFunc—functional annotation in the twilight zone. *Bioinformatics*, 24(6):798–806, March 2008.
- [27] Ronghui You, Xiaodi Huang, and Shanfeng Zhu. DeepText2GO: Improving large-scale protein function prediction with deep semantic text representation. *Methods*, 145:82–90, August 2018.

- [28] Ronghui You, Shuwei Yao, Hiroshi Mamitsuka, and Shanfeng Zhu. DeepGraphGO: graph neural network for large-scale, multispecies protein function prediction. *Bioinformatics*, 37(Supplement_1):i262–i271, August 2021.
- [29] Ronghui You, Zihan Zhang, Yi Xiong, Fengzhu Sun, Hiroshi Mamitsuka, and Shanfeng Zhu. GOLabeler: improving sequence-based large-scale protein function prediction by learning to rank. *Bioinformatics*, 34(14):2465–2473, July 2018.
- [30] Fuhao Zhang, Hong Song, Min Zeng, Fang-Xiang Wu, Yaohang Li, Yi Pan, and Min Li. A Deep Learning Framework for Gene Ontology Annotations With Sequence- and Network-Based Information. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 18(6):2208–2217, November 2021.
- [31] Chenguang Zhao, Tong Liu, and Zheng Wang. PANDA2: protein function prediction using graph neural networks. *NAR Genomics and Bioinformatics*, 4(1):lqac004, January 2022.