# Final challenge: Cellular Component Ontology Prediction

Imane Elbacha & Rajae Sebai

January 22, 2023

## Abstract

*This project aims to explore and apply various machine learning and deep learning techniques to a classification problem in the field of bioinformatics. The focus of the study is on protein engineering, a rapidly growing area of research that aims to understand and manipulate the structure and function of proteins. Proteins are large biomolecules that play a vital role in the functioning of all living organisms. They are composed of long chains of amino acids, and their 3D structure, determined by the sequence of these amino acids, dictates their specific chemical functionality. The challenge at hand is to classify a dataset of 6111 protein sequences and their corresponding graph representations, which depict the Euclidean distance between residues, their order in the sequence, and chemical interactions, into 18 different classes based on their cellular location. The goal is to use the sequence and structure of these proteins to accurately predict their functional location, as determined by the Cellular Component ontology. This project aims to evaluate the performance of various machine learning models in achieving this goal and gain insights into the relationship between protein sequence and structure. In this report, we present the topic, models and their performances and areas of improvement*

## 1. Introduction

Proteins play a crucial role in all living organisms, from catalyzing chemical reactions to providing structural support and playing a key role in the immune system. However, understanding the exact details of how the sequence of amino acids in a protein determines its specific chemical functionality is still not fully understood. This challenge presents a dataset of 6111 protein sequences and their graph representation of structure, with the goal of using this information to classify the proteins into 18 different classes, each representing a characteristic of the location where the protein performs its function.

One limitation of this project is the complexity of protein structure and function. Proteins are composed of long chains of amino acids, and can contain up to 4,500 of these amino acids. Additionally, the process by which the sequence of amino acids folds into a 3D shape and determines its chemical functionality is not fully understood. This makes it difficult to develop models that can accurately predict protein function based on sequence and structure alone.

To address this challenge, we applied several machine learning models including logistic regression, support vector machines, GNNs and GATs. However, the results of these models were not as good as we expected. To overcome this limitation, we decided to use a transfer learning method. We used the pre-trained model ESM-2 [1] to finetune the model and improve the results.

In fact, ESM-2 [1] is a state-of-the-art protein language model developed by the Meta Fundamental AI Research Protein Team (FAIR) at Facebook. It is a general-purpose model that can predict various properties of proteins such as structure and function directly from individual sequences. ESM-2 outperforms all other tested single-sequence protein language models and has been used in structure prediction tasks. The team also developed ESMFold, a variation of ESM-2 that can generate accurate structure predictions from the sequence of a protein. ESM-2 was introduced in the preprint ”Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences” (Rives et al., 2019) and was officially released in the paper by Lin et al. in 2022 [1].

In the following section, we present the models and their performances while focusing on ESM-2 which gave the best results.

## 2. Dataset

The dataset for this project contains 6,111 proteins in total. These proteins are represented by their sequences, edges and attributes, graph indicators, and labels. The sequences of each protein can be found in the ’sequences.txt’ file, with each line representing the sequence of one protein. The edges of the proteins are listed in the ’edgelist.txt’ file, with each line corresponding to an edge represented by the ids of its endpoints. It is important to note that these proteins are modeled as undirected graphs and there are a total of 15,213,222 edges for all 6,111 proteins.

The ’edge_attributes.txt’ file provides attributes for the edges of the 6,111 proteins. Each line stores the attributes of an edge, with 5 attributes in total separated by the comma character. These attributes include the distance between the two connected nodes and binary indicators for the different types of edges such as distance-based, peptide bond, k-NN, and hydrogen bond edges.

The 'node_attributes.txt' file contains the attributes of the nodes, or amino acids, of the 6,111 graphs. Each line stores the attributes of a node, with 86 attributes in total separated by the comma character. These attributes include 3D coordinates of the node, one-hot encoding of the amino acid type, hydrogen bond acceptor and donor status, and amino acid features derived from the EXPASY protein scale. There are a total of 1,572,264 nodes in the 6,111 proteins.

Lastly, the 'graph_indicator.txt' file contains the graph identifiers for all nodes of all 6,111 graphs. The value in each line denotes the graph to which the node with that specific id belongs. The 'graph labels.txt' file contains the names of all proteins along with the class labels of those proteins that belong to the training set. The proteins for which the class label is not available belong to the test set and the goal of this project is to predict the class label of each one of those proteins.

# 3. Models and results

In this section, we present the main models that were tested in this project and discuss their performance highlighting the strength and limitations.

## 3.1. Sequence baseline

The task of classifying the proteins in this dataset was approached using logistic regression as a baseline model. The results obtained using this model show a loss of 1.67. This loss value serves as a reference point for the performance of all the other models that were tested in this study. It is important to note that while logistic regression is a simple and widely used method, it may not be the best suited for this specific task due to the complexity of the protein sequences and the structural information provided. Additionally, logistic regression is only able to classify based on the sequence of the protein, not taking into account the structural information provided in the dataset. Therefore, it is likely that more complex models will be able to achieve better performance on this task.

## 3.2. Support vector classification

The Support Vector Classification (SVC) model is a popular algorithm for classification problems, which separates the data by finding the best boundary or hyperplane that maximizes the margin between different classes. In this project, we applied SVC on the sequence of the proteins, and by optimizing the parameters using a gridsearch method, we were able to achieve a loss of 1.11. This is a significant improvement compared to the logistic regression baseline, which had a loss of 1.67. The SVC model was able to classify the proteins more accurately by identifying the key features in the sequence that separated them into different classes.

One of the reasons that SVC performed well on this task is that it is a powerful algorithm that can handle non-linearly separable data by using the kernel trick. Also, it's robust to outliers, so it's suitable for high-dimensional data, like the one in this dataset. However, there are some limitations to this approach. The SVC model is sensitive to the choice of kernel and the regularization parameter. Therefore, a proper selection of these parameters is crucial for achieving the best performance. Also, SVC requires a large amount of data to work well, which may not be always available in bioinformatics applications.

## 3.3. Structure Baseline

Graph Neural Networks (GNNs) are a class of neural network models that are designed to process graph-structured data. They are particularly useful for tasks such as node classification, link prediction, and graph classification. GNNs are able to capture the underlying structure of the graph by propagating information between nodes through their edges. This allows them to learn representations of the graph that are useful for the task at hand.

In the case of this project, GNNs were used to classify proteins based on their structure represented as graphs. The data was loaded and processed to create the necessary input for the GNN, which consisted of the node attributes and graph adjacency matrix. The GNN used in this project is a GAT that has 2 message passing layers and a sum aggregation function. The results of this model had a loss of 1.86 which was not as good as the baseline logistic regression and SVC on sequence data. This limitation is likely due to the complexity of the protein structure and the difficulty of capturing all of the relevant information with the GNN architecture and parameters used in this project.

One area for improvement in this project could be to incorporate more deep GAT or GCN architectures, in order to better capture the structural information present in the protein graphs. Additionally, incorporating additional information such as edge attributes or external information like protein-ligand could also improve the performance of the model. Another potential avenue for improvement could be to fine-tune the pre-trained transformer language models on a larger dataset of protein sequences, in order to better capture the nuances of the specific protein classes being analyzed in this project. Furthermore, it could be valuable to explore other feature representations and feature selection techniques to extract more informative features from the data. Additionally, the use of ensemble methods and further hyperparameter tuning could also potentially improve the performance of the models.

In the next sections, we are going to explore some of these improvement areas and present the results and limitations

## 3.4. Graph Attention Networks and Graph Convolutional Networks

Graph Attention Networks (GAT) and Graph Convolutional Networks (GCN) are both state-of-the-art graph neural network architectures that have been designed to improve the performance of GNNs on graph-structured data. GATs use self-attention mechanisms to assign different weights to the different edges in the graph, allowing the model to focus on the most important edges when passing messages between nodes. GCNs, on the other hand, use a convolutional architecture to aggregate information from the neighborhood of each node in the graph.

GATs and GCNs have been shown to outperform traditional GNNs on a wide range of graph-based tasks, including node classification, link prediction, and graph classification. However, they also come with their own set of limitations. GATs, for example, tend to be computationally expensive and memory-intensive, making them difficult to train on large graphs. GCNs, on the other hand, are sensitive to the sparsity and the scale of the graph, which can lead to oversmoothing and loss of information.

In our project, we implemented both GAT and GCN models as a way to improve the results obtained from our baseline GNN model. However, despite their advantages over a simple GNN such as the ability to handle graph-structured data, we were not able to achieve better results with these models. This could be due to the complexity of the problem and the small size of the dataset we were working with. Additionally, the graph representation of the proteins requires preprocessing before being used as input for GAT and GCN models which may have affected the results. In conclusion, while GAT and GCN models have the potential to provide better results for this problem, it is important to consider the limitations of the dataset and the preprocessing requirements before implementing them.

## 3.5. Language models on sequential data

Transfer learning is a method of using pre-trained models on one task to improve performance on a different but related task. In this project, we utilized transfer learning techniques to improve the performance of our models on the protein classification task. We applied pre-trained language models on sequential data, such as **ESM-2** and **ProtBert**, to extract high-level features from the protein sequences, and then fine-tuned them for the specific task of classification. This method allowed us to leverage the vast amount of data and computational resources used to train the pre-trained models, resulting in improved performance compared to training models from scratch. The benefits of transfer learning include faster convergence, better performance and requiring less data.

### 3.5.1 The pre-trained models

In order to improve the performance of our protein classification task we employ pre-trained models. One of the models we used is ESM-2 [1], developed by the Meta Fundamental AI Research Protein Team (FAIR). This model is a state-of-the-art protein language model that can be used to predict structure, function, and other protein properties directly from individual sequences. We used both the 35 million parameter version and the 150 million parameter version of this model in our experiments.

One of the key benefits of using pre-trained models like ESM-2 is that they have been trained on a large amount of data and with sophisticated architectures, allowing them to learn robust and generalizable representations of protein sequences. The ESM-2 language models are pre-trained on millions of protein sequences using the masked language modeling objective. This objective trains the model to predict the identity of randomly selected amino acids in a protein sequence by observing their context in the rest of the sequence. This allows the model to learn dependencies between the amino acids and internalize sequence patterns across evolution.

Another pre-trained model we used is ProtBert, which is based on the Bert model and was pretrained on a large corpus of protein sequences in a self-supervised fashion. This means it was pretrained on the raw protein sequences only, with no humans labelling them in any way. The model uses an automatic process to generate inputs and labels from those protein sequences, and differs from the original Bert version in that it treats sequences as separate documents. The masking in this model follows the original Bert training by randomly masking 15% of the amino acids in the input.

In conclusion, the use of pre-trained models such as ESM-2 and ProtBert allowed us to improve the performance of our protein classification task. Both models were trained on large corpus of protein sequences, allowing them to learn robust and generalizable representations of protein sequences. We found that ESM-2 with 150 million parameters gave the best results, likely due to its ability to learn a deeper understanding of sequence patterns across evolution. Overall, the use of pre-trained models proved to be a valuable approach in minimizing the loss of our protein classification task.

#### 3.5.2 Data representation

The tokenizer used by the ESM-2 model is an important component that is responsible for preprocessing the input data before it is fed into the model. In the context of protein sequence classification, the tokenizer converts the sequences of amino acids into a format that can be understood by the model. This typically involves breaking down the sequences into smaller units, called tokens, and then encoding them into numerical form. The tokenizer used by ESM-2 is specifically designed to handle the unique characteristics of protein sequences, such as the fact that they are composed of a fixed set of 20 different amino acids. It may also handle any specific preprocessing steps for this ESM model to work efficiently on this task.

The tokenizer used by ESM-2 is a key element in the preprocessing step, it enables the model to understand the input data by breaking down the sequences of amino acids into tokens, and then encoding them into numerical form. This process allows the model to identify the different amino acids and their position in the sequence. By doing this, the tokenizer allows the model to learn the dependencies between the different amino acids, which is crucial for protein sequence classification. Additionally, the tokenizer can be fine-tuned on the specific task of protein sequence classification to improve performance. This allows the tokenizer to better understand the unique characteristics of the input data, leading to better performance on the task of protein sequence classification. Overall, the tokenizer plays a crucial role in the overall performance of the model, as it ensures that the input data is properly formatted and encoded for the model to effectively learn from it.

#### 3.5.3 Performance analysis

Multiple machine learning and deep learning algorithms were used in this project in order to perform the classfication task on protein data. Both sequential (amino-acids text sequences) and structural (graph representation of the protein) were used during this project. The following Table summarizes the overall performances obtained by the methods presented in this paper.

| Method | log loss |
|---|---|
| **Structure** | |
| GAT (node features) | 1.85 |
| **Sequence** | |
| Logistic Regression | 1.67 |
| SVC | 1.11 |
| Finetuned ProtBert | 2.22 |
| Finetuned ESM-2 (35M parameters) | 1.37 |
| Finetuned ESM-2 (150M parameters) | **1.06** |

Table 1. Protein classification performances using sequential and structural protein information. The error metric is the multi-class log loss.

## 4. Conclusion

In this project, we investigated various machine learning techniques for protein classification, including traditional methods such as logistic regression and support vector machines, as well as graph neural networks (GNNs) like GAT and GCN. Additionally, we utilized a transfer learning approach with the pre-trained model ESM-2.

Our results indicated that traditional machine learning algorithms performed the best when applied to sequence data. However, when utilizing the transfer learning approach with ESM-2, we achieved the highest level of accuracy. This can be attributed to the pre-trained model's ability to learn robust and generalizable representations of protein sequences through its exposure to large amounts of data and computational resources.

On the other hand, the performance of GNN models was not as promising as traditional machine learning algorithms or transfer learning. This can be attributed to the limited amount of labeled data available for training these models and the complexity of protein structure, which makes it challenging to represent it in a graph.

In summary, GNNs have potential as a method for protein classification, but their performance may not be as strong when dealing with limited labeled data. Further research with larger labeled datasets and advanced GNN architectures is necessary to achieve improved results. It's worth noting that Transfer learning is a method in which a model developed for a task is reused as the starting point for a model on a second task. The idea is to transfer knowledge from the first task to the second task. The perks of transfer learning are that it can be used to improve the performance of a model in a new task by using the knowledge gained from a previous task.

## References

[1] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, and Alexander Rives. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022. 1, 3

[2] ORCID ProfileChristian Dallago Ghalia Rehawi Yu Wang Llion Jones Tom Gibbs Tamas Feher Christoph Angerer Martin Steinegger ORCID ProfileDebsindhu Bhowmik Burkhard Rost Ahmed Elnaggar, Michael Heinzinger. Prottrans: Towards cracking the language of life's code through self-supervised learning, 2021.

[3] Roshan Rao Brian Hie Zhongkai Zhu Wenting Lu Nikita Smetanin Robert Verkuil Ori Kabeli Yaniv Shmueli Allan dos Santos Costa Maryam Fazel-Zarandi Tom Sercu Sal-

vatore Candido Alexander Rives Zeming Lin, Halil Akin. Evolutionary-scale prediction of atomic level protein structure with a language model, 2022.