EC Number Prediction Using Pre-Trained Language Models: A Promising Tool for Protein Research

Ugur Dura*
Informatics Department
Technical University of Munich
Munich, Germany
ugur.dura@tum.de

***Abstract*—Enzyme Commission (EC) numbers play a vital role in accurately understanding enzyme functions and their impact on cellular metabolism. In this paper, we explore the potential of utilizing pre-trained language models for EC number prediction, with a specific focus on protein sequences as a rich source of information. Our objective is to leverage deep learning models, particularly the transformer BERT model, to uncover patterns corresponding to the topological sites of each enzyme that may lead to predict their functionalities. Through comprehensive experimentation, we demonstrate the successful utilization of the transformer BERT model for identifying and classifying de novo enzyme functions based on protein sequences retrieved from UniProt. By treating proteins as linguistic entities, we harness the power of language models to enhance EC number prediction with a high level of accuracy. While our results showcase the effectiveness of this approach, further research is required to improve the classification accuracy for all four digits of the EC number. Future work will focus on refining the training process and incorporating additional contextual information to further enhance the model's performance. In conclusion, our study presents a promising tool for EC number prediction using pre-trained language models. By bridging the gap between protein sequences and language processing, this approach opens up new possibilities for advancements in protein research, including drug discovery, protein engineering, and biomedical applications.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Enzymes, a vital class of proteins within the human body, play a crucial role in catalyzing reactions and regulating various biological processes. Understanding their functions is essential for numerous applications, such as metagenomics, industrial biotechnology, and diagnosing enzyme-related diseases. However, traditional experimental techniques like enzymatic assays have become increasingly time-consuming and challenging to keep up with the rapidly growing number of newly discovered enzymes.

In recent years, computational methods have emerged as invaluable tools for predicting enzyme function and guiding experimental validation. The widely used Enzyme Commission (EC) system employs four-digit codes to specify enzyme functions, enabling computational methods to accurately annotate enzyme activities.

To address this issue, computational methods have emerged as valuable tools for predicting enzyme function and guiding experimental validation. Previous research has explored three main approaches for enzyme EC number prediction: (i) structure-based prediction, which involves first predicting enzyme structures and then assigning EC numbers based on similarities to known templates; (ii) sequence similarity-based methods, which predict enzyme function based on the assumption that enzymes with high sequence similarity tend to have similar functionalities; and (iii) machine learning-based methods, which extract features from enzyme sequences and use machine learning algorithms for classification.

Despite progress made, existing approaches encounter challenges such as homology requirements, feature design limitations, and feature dimensionality non-uniformity. Homology requirements can limit the prediction capability of sequence similarity-based methods when encountering sequences with limited homologies in existing databases. Moreover, feature design often relies on manually crafted or pre-defined features, which may not be optimal or sustainable in the omic era, with the rapid expansion of known enzyme sequences.

In contrast, recent advances in deep neural networks have shown promise as generative and discriminative models for protein science and engineering. Transformer-based language models, particularly, have demonstrated impressive capabilities in various domains, including natural language processing. To address these challenges, this research proposes a novel approach based on deep learning, specifically utilizing Prot-BERT and transfer learning, for predicting first-order EC numbers of enzymes.

In conclusion, this research aims to develop a robust and efficient deep learning-based predictive model for enzyme EC number classification. By combining the power of Prot-BERT with fine-tuning on a large dataset of enzyme sequences, this approach holds the potential to significantly advance our understanding of enzyme functions and aid in various applications related to enzymology.

II. Materials and Methods

A. Data Collection

The dataset used in this study was obtained from SwissProt, a manually curated section of UniProtKB, which contains a vast collection of annotated protein entries. Initially, the dataset comprised 565,254 protein sequences. To focus specifically on enzymes and their respective functions, a rigorous filtering process was applied, resulting in a refined dataset consisting of 271,464 entries. Only entries with at least one assigned EC number were retained for further analysis. In cases where an entry contained multiple EC numbers, the first one listed was used for subsequent prediction tasks.

B. Data Analysis and Preprocessing

To prepare our dataset for the deep learning model, we considered amino acids as the fundamental building blocks, much like an alphabet, used to train the Masked Language Model (MLM). Each amino acid was denoted by a shorthand one-letter symbol, such as 'M' for methionine, 'R' for arginine, and 'X' for unknown amino acids. This concise representation enabled us to effectively handle the diverse range of amino acid sequences present in the dataset. The distribution of amino acid letters in the dataset can be seen in Fig. 1. Notably, leucine emerged as the most common amino acid, and our dataset did not contain any unknown amino acids.

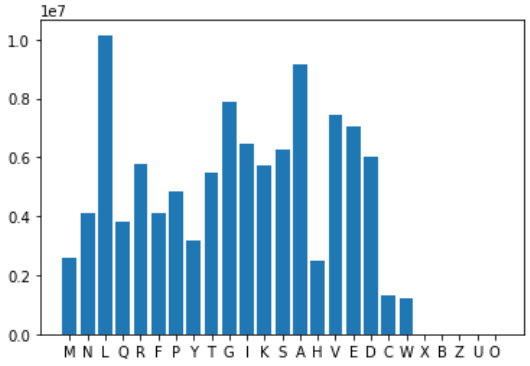
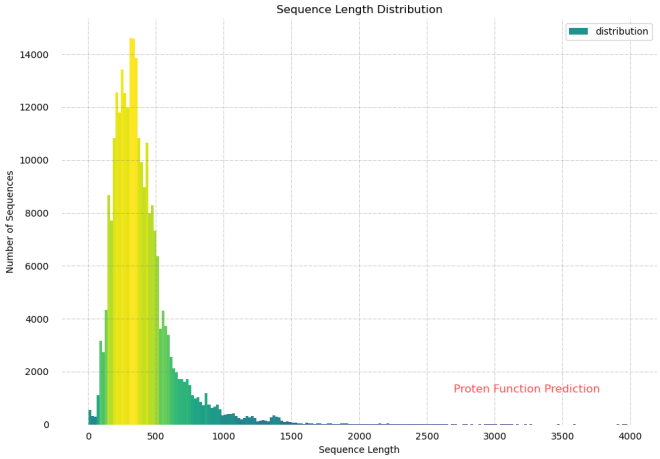


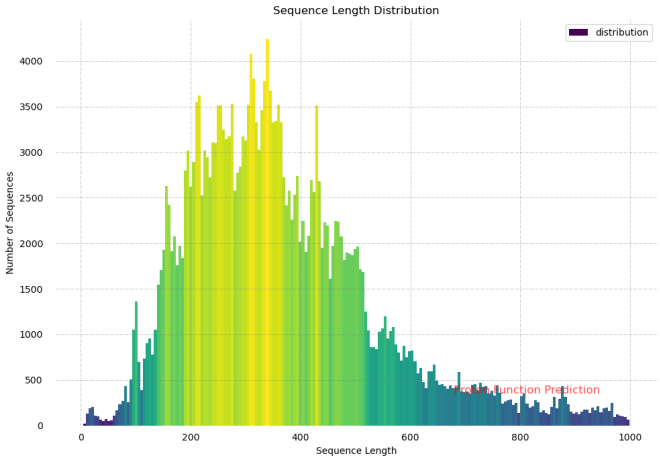
Fig. 1: The plot illustrates the amino acid letter density distribution within the collected dataset. The y-axis represents the number of occurrences, while the x-axis displays the corresponding amino acid letters.

During the data analysis process, one crucial aspect that needed consideration was the variation in sequence lengths. To tokenize and train our model effectively, we needed to decide on a fixed array size for the sequences. Our analysis revealed a significant distribution of sequence lengths, with the majority falling within the range of 200 to 500 amino acids as shown in Fig. 2. This finding presented a challenge, as we had to strike a balance between capturing relevant information and computational efficiency(Fig. 3). Ultimately, an array size of 512 was chosen to accommodate the majority of sequences within this length range.

At the end of this data preprocessing, we employed one-hot encoding to effectively serve the amino acid sequences into our deep learning model. One-hot encoding is a technique



(a)



(b)

Fig. 2: The sequence length histogram of the dataset. The X-axis shows the sequence length and the Y-axis shows the number of sequences accumulated within that range. Left: Sequence length distribution histogram on the whole dataset. Right: Sequence length distribution histogram with range 0-1000.

that converts categorical data, such as the shorthand one-letter symbols representing amino acids, into binary vectors. Each amino acid symbol was transformed into a binary vector of zeros and ones, with a value of one at the position corresponding to the specific amino acid and zeros elsewhere. Let A be a set containing specific amino acid letters, and $\mathbb{R}_A(x)$ be the indicator function. For each amino acid x, $\mathbb{R}_A(x)$ returns 1 if x is in the set A, and 0 if x is not in the set A (eq. 1).This encoding allowed the model to process the amino acid sequences as numerical inputs, facilitating the training and prediction processes.

$$A = \{a_1, a_2, a_3, \ldots, 21\} \tag{1a}$$

$$\mathbb{R}_A(x) = \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{if } x \notin A. \end{cases} \tag{1b}$$

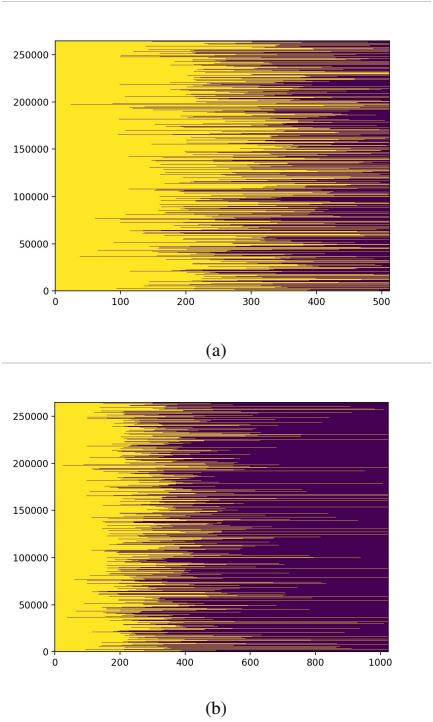


Fig. 3: Representation of the padding and truncation rate of the sequence during tokenization. The X-axis represents the length of the sequence (for (a) max length=512, for (b) max length= 1024), the Y-axis represents the data-set index(262K entry). Yellow shows the tokenized sequence and purple shows the padded area.

By utilizing one-hot encoding, we successfully converted the diverse range of amino acid representations in the dataset into a format suitable for input to the Masked Language Model. By understanding the characteristics of our dataset, we were able to lay a strong foundation for the subsequent development and fine-tuning of our deep learning model.

C. Model Construction

In our research, we utilized the main Bert Layer from the pre-trained ProtBERT-BFD model as the embedding layer. The ”training” parameter of the layer was set to False during fine-tuning to maintain the integrity of the pre-trained embedding mechanism. We employed the mandatory ProtBERT-BFD tokenizer to tokenize our dataset. Following the embedding layer, we used pooling, normalization, two dense layers (with sizes 128 and 32), a dropout layer, and a final dense layer with softmax activation function. The tokenizer outputs, including the tokenized sequence and mask, were fed into the BERT layer. The model's output went through subsequent layers until the final output layer, which provided probability scores for each class. ReLU activation function was used for hidden layers to prevent the vanishing gradient problem and enhance model performance.

III. Results

References

Please number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first . . .”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [4]. Papers that have been accepted for publication should be cited as “in press” [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

References

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.
[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
[4] K. Elissa, “Title of paper if known,” unpublished.
[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

Appendix

Plots

sdfsdfsdf

Tables

sdfsdf