Classifying Antibiotic Resistant Proteins with Prot-Bert

Andrew Thomas Costa

October 2023

Sommario

Antimicrobial resistance (AR) occurs when germs like bacteria or fungi develop the ability to withstand the drugs (antibiotics) designed to eradicate them. Bacteria then continue to grow and resistant infections arise, becoming impossible to treat. From the latest report conducted by the CDC in 2019, antimicrobial resistance alone killed 1.27 million people and was associated with 5 million deaths. Specifically, resistance arises due to the encoding of antibiotic resistance genes (ARGs), which can be easily transmitted between bacteria and viruses that are then targeted with antibiotics. The classification of these antibiotic resistance genes is therefore imperative to slowdown its spread, where this classification is the central premise of this paper. With the use of Hugging Face's 'ProtBert', this paper classifies amino acid sequences to detect the presence of resistance sequences.

Indice

Elenco delle figure

Elenco delle tabelle

1 Introduction

Antibiotics are consumed globally and are prevalent in industries like agriculture, medicine and veterinary. Within the first decade of the 21st century, their consumption had increased by 35%. The second-order implications impact the wide economy as a whole, damaging the food trade and recovery from bacterial infections (source: https://www.cdc.gov/drugresistance/about.html). The cumulative cost of healthcare and reduced productivity could reach $100 trillion by 2050. In Europe alone, antibiotic-resistant infections cause an estimated 30,000 deaths annually, with Greece and Italy producing the largest number of cases. Despite these severely consequential limitations, global antibiotic-use increases productivity and economic efficiency, increasing the production of livestock, chicken, pigs and turkey.

Resistance to antibiotics occurs at the cellular level. Pathogenic bacteria acquire ARGs through plasmid exchange at the gene level and develop strong resistance to antibiotics.

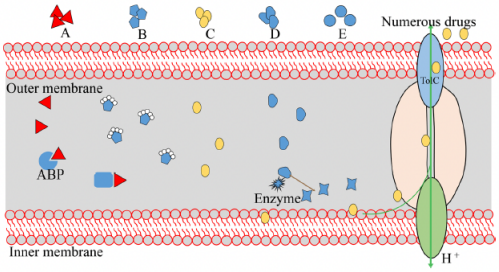

Figura 1: The Mechanism of Bacterial Resistance.

In Figure 1, A, B, C, D and E represent different antibiotics and ABP represents an Antibiotic Binding Protein. Resistance develops by altering the structure of the ABP so that they cannot bind to antibiotics.

This paper aims to implement Hugging Face's 'ProtBert' language model to create embeddings of amino-acid strains, which will then be classified based on whether they are resistant to antibiotics. ProtBert was trained on 106 million proteins, which represents the entire known protein space. Protein sequences were encoded as sequences of integer tokens with 26 unique tokens representing the 20 standard amino acids, selenocysteine (U), an undefined amino-acid (X), another amino acid (OTHER), and 3 additional tokens (START, END and PAD).

2 Exploratory Data Analysis

The dataset was sourced from a hackathon hosted on a platform called Zindi. The dataset is composed of train and test CSV files. The train dataset is composed of 11,846 amino-acid sequences with their respective labels (resistant or not) and the test includes X—X amino-acids.

Figure 2 captures the distribution of the sequence lengths for the amino-acid chains in the train dataset. As one can see, the large majority of the sequences fall between the ranges of 200 and 300 characters, with a small spike of chains between 600 and 700 characters. The test dataset visibly follows the same distribution as the train set. This distribution of sequence lengths might pose several difficulties when training the model, one such being the memory limitations. There are methods to circumvent this, like parallelising the model training or Automatic Mixed Precision.

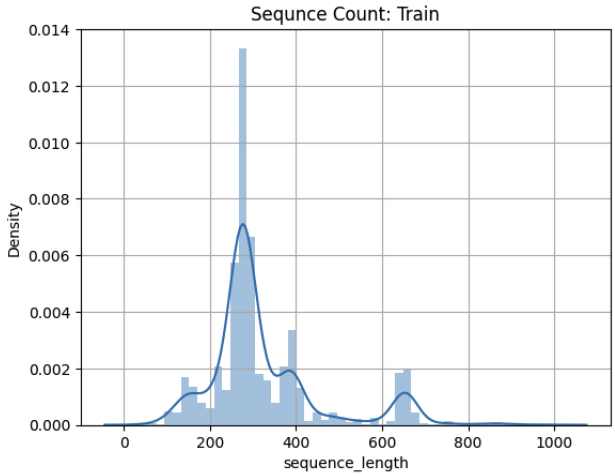

Figura 2: Distribution of Amino-Acid Sequence Lengths for Train Dataset

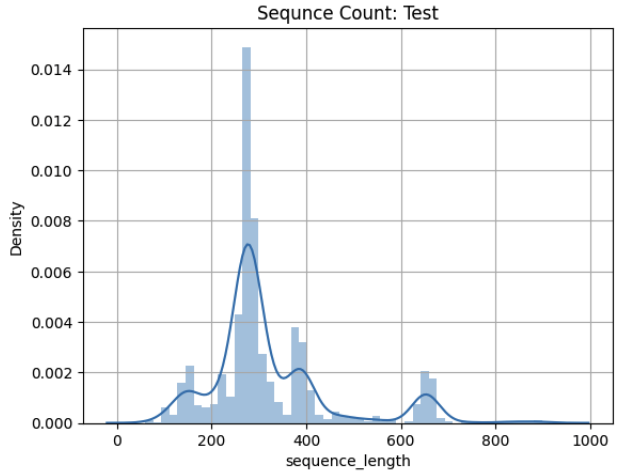

Figura 3: Distribution of Amino-Acid Sequence Lengths for Test Dataset

Figures 4 and 5 show the frequencies for each of the 20 amino acids plus a rare amino acid for the train and test datasets respectively. It is important to note that the distributions are the same, which will ensure that the model will be tested on the same frequency distribution as the train set.

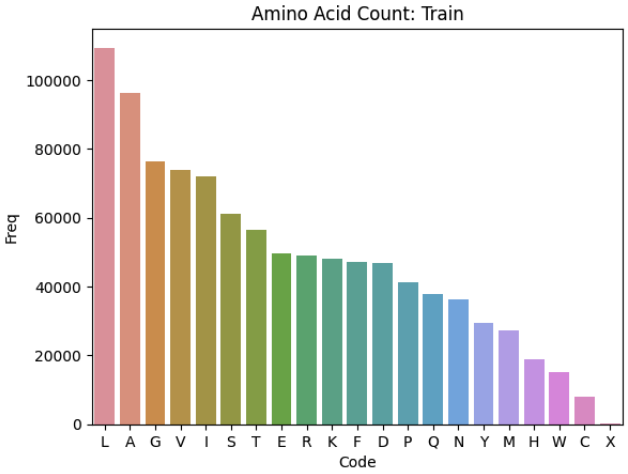

Figura 4: Distribution of Amino-Acid Types in Train Dataset

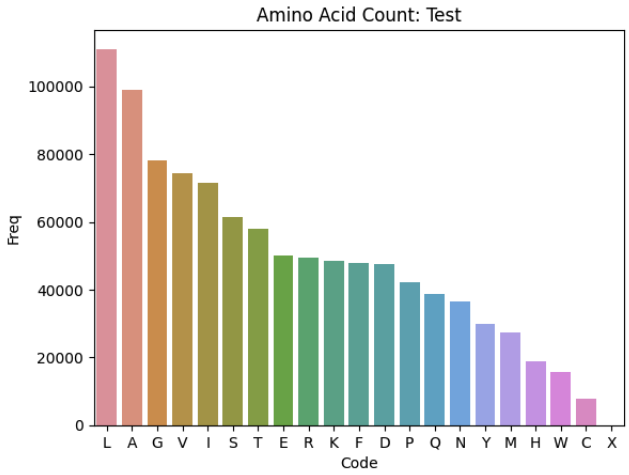

Figura 5: Distribution of Amino-Acid Types in Test Dataset

The bar chart of class counts shows that the dataset is balanced, since there are approximately a relatively equal number of amino-acid sequences for each class. This implies that the model will not bias one class more than another in the training process. Furthermore, imbalanced datasets can fall the Accuracy Paradox, whose underlying premise suggests that it is not beneficial for models to rely on accuracy as a measure for predicting the correct outcomes.

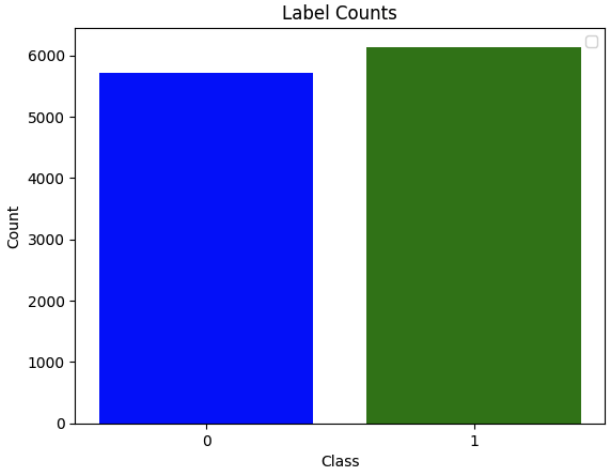

Figura 6: Distribution of Class labels

3 Data Pre-processing

For pre-processing amino-acid sequences, convention dictates to first remove any rare amino-acid and include spacing between them. By implementing spacing between each amino-acid, one maintains positional information and original sequence alignment, which is critical for protein structure prediction. Furthermore, the removal of rare amino-acids simplifies the analysis process through reducing the complexity of the dataset and hence reducing the amount of noise present.
A train and test set are created to ensure the model is not being tested on the same dataset.

The sizes of the created train and test sets are the following:

- The train dataset contains 11,846 sequences for training the model
- The test dataset contains 5,078 sequences for testing the model

The maximum sequence length of the amino acids in the training set is 993. This is too large to train the model and would be computationally heavy for the purpose of this task given hardware limitations. A maximum length of 300 was arbitrarily chosen for this research as a way to alleviate the the aforementioned constraints.

4 Model Selection

There exists sequence-based language models, like BERT, that have been specifically built for natural languages, however this does not make them optimal for protein tasks as they do not have clear-cut multi-letter building blocks. Additionally, proteins show much more variation in their sequence length and more interactions between distant positions.

ProteinBert (ProtBert) was introduced as an improvement to the transformer/BERT architecture to overcome these aforementioned tasks. However before discussing ProtBert in greater depth, it is necessary to expand on what a transformer is and what makes it perform well for language tasks. It is a deep-learning model designed specifically for protein sequences that is an improvement on the transformer/BERT architecture.

A transformer is a neural network that learns context and meaning by tracking the relationships in sequential data like the words in this sentence.

Transformers use self-attention/attention to detect - in subtle ways - that even distance data elements in a series influence and depend upon each other. Attention dynamically highlights and uses the salient parts of the information at hand much like the brain does. The attention mechanism reads the raw data and converts them into distributed representations with one feature vector associated with each word position.

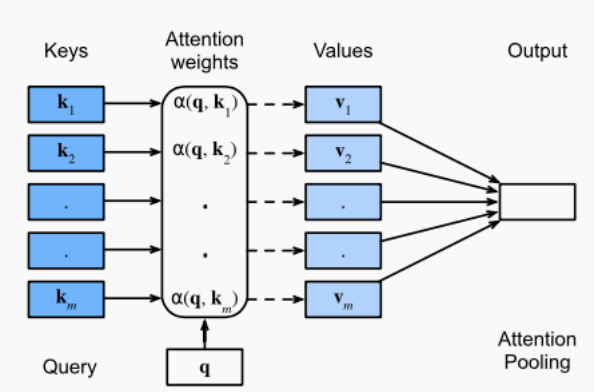

Figura 7: Attention Architecture

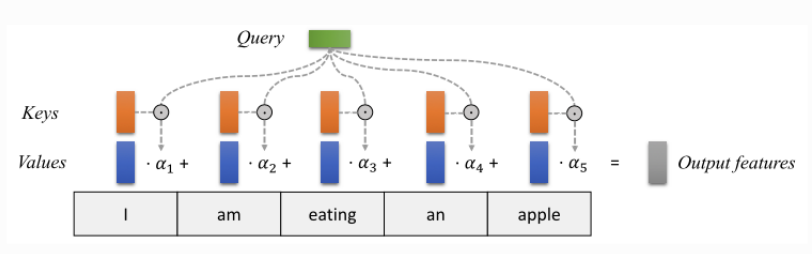

Figura 8: Key-Value Diagram

Each word/character has one key and one value vector and a query is compared to all the keys with a score function to determine the weights. The value vectors of all words are then averaged using the attention weights.

Attention mechanism is composed of large encoder and decoder blocks that have the specific task of of processing the data. Positional encoders tag data elements coming into and out of the network. The attention mechanism works such that it can then follow these tags in order to calculate an algebraic map of how each data element relates to each other. These attention queries can then be computed in parallel by calculating a matrix of equations, which is a process also known as multi-head attention. Scaled dot-product attention is the core behind self-attention. The attention value from element i to j is based on its similarity of the query Q_i and key K_j, using the dot product as the similarity metric. In math, the dot product attention is calculated as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Figura 9: Equation for Attention; Q is query, K is key and V is value

The matrix multiplication QK^T performs the dot product for every possible pair of queries and keys, resulting in a matrix of the shape T×T. Each row represents the attention logits for a specific element i to all other elements in the sequence, which is then applied to a softmax and multiplied with the value vector to obtain a weighted mean (the weights being determined by the attention). The scaling factor of $\frac{1}{\sqrt{d_k}}$ ensures that an appropriate level of variance for the attention values is maintained after initialisation. It is important that at the initialisation period, the level of variance is maintained equal throughout the model, which will imply that Q and K might also have a variance close to one.

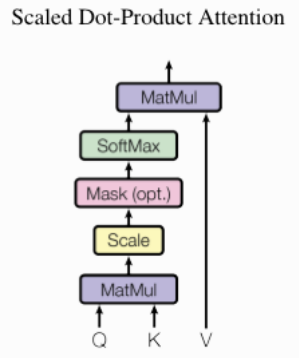

Figura 10: Scaled Dot Product Attention

The difficulty in encoding proteins is that they have different structural properties than human languages. Proteins are more variable in length than sentences and show interactions between distant positions; Protein-Bert, however, helps overcome these difficulties. Protein-Bert separates itself from classical transformers as it is able to distinguish between local (character level) and global (whole protein level) characteristics, allowing itself to multitask on both local and global tasks.

5 Method

In order for the model to be able to interpret the data and learn, we must begin by tokenizing it. What this does is break words into sub-words and then indexes these sub-words with a numerical value, known as an input index, as the model can only interpret such data.

After tokenizing the data, the input tokens are then mapped to sequences of vectors (word embeddings) via an embedding layer into a vector space. These embeddings, or output layers, are then classified to a sequence of tokens, which in turn are then decoded back to their original input format (text), with positional information about each token also being encoded into the embeddings.

Following the embeddings, data is then encoded in batches, returning a dictionary of values instead of just a list of values as this returns input IDs, attention masks and labels. Input IDs are the unique identifiers of the tokens in a sentence and attention masks are used to batch the input sequence together and indicate whether the tokens should be attended to by the model or not. For example, tokens with attention masks with a value of zero will be ignored by the model, whereas attention masks with a value of one will be taken for further processing.

6 Model

The architecture of the implemented neural network, encapsulated within the 'ABRClassifier' class, exhibits a carefully designed structure. Protein-Bert forms the backbone of the model, benefitting from its pretraining on vast protein sequence datasets, thereby inheriting knowledge about biological sequences. The configuration of the model, informed by the 'AutoConfig' module from the Hugging Face Transformers library, ensures compatibility and coherent behavior. Furthermore, essential components of the architecture include a dropout layer, incorporated to mitigate overfitting, and a fully connected (linear) layer, which adapts the model's output to the specific classification task by tailoring the number of output units.

A dropout rate of 0.3, as indicated in the code, signifies that during the training of a neural network, approximately 30% of the neurons in the dropout layer will be "dropped out" or randomly set to zero during each training iteration. Dropout is a regularization technique used in neural networks to prevent overfitting. When dropout is applied with a rate of 0.3, it means that, on average, 30% of the neurons in the dropout layer will be temporarily deactivated or "dropped out" during each forward and backward pass through the network. The primary purpose of dropout is to improve the generalization of the neural network. By randomly dropping out neurons, it reduces the network's reliance on any specific neurons and forces it to learn more robust and generalized features from the data. This, in turn, helps prevent the network from fitting noise in the training data and makes it more likely to perform well on unseen or validation data.

Adam is a popular optimization algorithm that combines the benefits of two other widely used optimizers: AdaGrad and RMSprop. It is known for its ability to adaptively adjust learning rates for each parameter in the model, which can be especially beneficial when dealing with complex, high-dimensional data. The algorithm maintains moving averages of both the gradients (first moment) and the squared gradients (second moment) of the parameters, which helps in adjusting the learning rates based on the historical behavior of each parameter. Adam often converges faster than traditional optimization methods like stochastic gradient descent (SGD) and requires less manual tuning of hyperparameters.

The learning rate determines the step size at which the optimizer updates the model's parameters during training. A learning rate of 0.01 (1e-2), as selected for this paper is a moderate value as it's neither too high nor too low. A higher learning rate might lead to faster convergence but risks overshooting the optimal solution or causing divergence. Conversely, a lower learning rate might be more stable but slower to converge.

7 Results

After running the model for ten epochs, the the evaluation of the model was determined by the number of correct predictions over the total number of predictions. The accuracy of the model was 51.69%, which indicates that the model is as accurate as flipping a coin in order to determine the outcome.

The confusion matrix also confirms the results above, that when the model makes a prediction, it obtains the correct result around 50% of the time. There are a number of strategies that can be implemented to improve the model's performance, like changing the model architecture, tuning the hyperparameters and further engineering the features - however, due to hardware limitations, these were not possible.

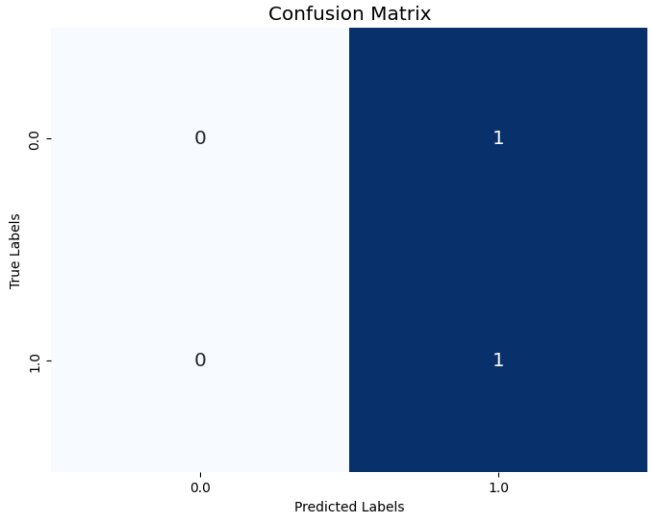

Figura 11: Confusion Matrix for Final Output

8 Conclusion

In summary, the escalating utilization of antibiotics on a global scale across sectors including agriculture, healthcare, and veterinary practice has become a subject of paramount concern in the 21st century. Notably, the first decade of this century witnessed a marked increase of 35% in antibiotic consumption. This surge has spawned a cascade of repercussions that extend beyond the immediate domains of healthcare and agriculture, permeating the broader global economy and casting a shadow over the food trade sector.

The ramifications of antibiotic resistance are profound, with projections indicating that the cumulative healthcare expenditure and resultant productivity losses could reach a staggering 100 trillion USD by 2050. In Europe, the toll of antibiotic-resistant infections is stark, with an estimated annual mortality rate of around 30,000 individuals, with Greece and Italy experiencing particularly pronounced impacts. Paradoxically, antibiotics remain instrumental in bolstering productivity, particularly in the context of livestock and poultry farming. At the heart of antibiotic resistance lies a microscopic battleground, as pathogenic bacteria acquire Antibiotic Resistance Genes (ARGs) through plasmid transfer, conferring immunity against antibiotics.

This paper introduces a novel method employing Hugging Face's 'ProtBert' model to assess amino-acid sequences and discern their susceptibility to antibiotic resistance. This model has been trained on an extensive dataset, encompassing approximately 106 million proteins. Distinguishing itself from conventional language models, ProtBert adeptly navigates the unique structural attributes of proteins.

However, the outcomes of the model's ten-epoch implementation reveal an accuracy rate of a mere 51.69%, a level akin to random chance. This assessment is further substantiated by the confusion matrix, indicating that the model's predictions align with randomness, hovering around the 50% mark. Enhancing the model's performance remains a valid aspiration. Nevertheless, the pursuit of improvement faces pragmatic limitations, such as hardware constraints. Potential avenues for enhancement encompass architectural modifications, hyperparameter fine-tuning, and more sophisticated feature engineering. In the context of the pressing concern of antibiotic resistance, addressing these challenges and augmenting the model's accuracy assume paramount importance for the development of robust tools for the classification of antibiotic-resistant proteins.

We declare that this material, which I/We now submit for assessment, is entirely my/our own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my/our work. I/We understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. This assignment, or any part of it, has not been previously submitted by me/us or any other person for assessment on this or any other course of study.