Leveraging NLP Methods for Improved Phage-Bacteria Interaction Predictions Using DNA and Protein Data

Blake J Bleier

University of California, Berkeley

Abstract

As antimicrobial resistance (AMR) grows worldwide, traditional antibiotic treatments are increasingly insufficient, highlighting the need for novel treatment options. Bacteriophages offer a promising alternative to conventional treatments. This research focuses on identifying bacteriophages for treatment by utilizing machine learning models that integrate both DNA and protein sequence information to predict phage-bacteria interactions (PBIs). By leveraging concepts from Natural Language Processing, we embed this biological sequence data using DNABERT-2 and ESM-2 transformer models to make a classification prediction. We explore architectural variations in model design to improve prediction when excess hosts are bucketed as "Other", finding that a dual-headed classifier significantly increases performance over a single-headed classifier model. Our results also indicate that a multimodal combination of DNA and protein data does not significantly enhance the model relative to protein data alone, but that additional investigation with more advanced DNA models may yield better results.

1 Introduction

Antimicrobial resistance occurs when bacteria evolve the capacity to resist antibiotic drugs, largely driven by the overuse of antibiotics in our current healthcare system [1]. This presents a serious threat to global health, resulting in almost 5 million AMR-related deaths in 2019 alone [2]. An actively researched alternative to traditional antibiotic treatments is phage therapy, which utilizes bacteriophages (or viruses) as treatment. Bacteriophages are known to interact with specific lines of bacterial cells and reproduce until the cell bursts via cell lysis [3]. The ability to accurately predict these phage-bacteria interactions offers a novel approach to combat bacterial infections by harnessing naturally occurring phages as treatment tools.

Traditional methods to determine Phage-Bacteria Interactions (PBIs) involve isolating a bacteriophage and experimentally validating bacterial interactions in the lab [4]. However, this is an extremely time and resource intensive process. The need for a scalable, efficient solution has led the field to explore computational models that can predict interactions between bacteriophages and their bacterial hosts.

In this work, we use transfer-learning on two data modalities (protein and DNA sequence information), along with DNABERT-2 and ESM-2 transformer based models, to capture the intricate relationships within sequences of DNA and proteins and enhance the detection of potential PBIs beyond traditional bioinformatics tools.

2 Background

In recent years, machine learning has been explored as a means to enhance efficiency by predicting PBIs in silico. Several studies have explored various machine learning models for this purpose, including CNNs [5,6], gradient boosted forests [7], and transformer embeddings [8,9]. These models have been architected to create different types of predictions. Some use a binary classifier to predict whether a given phage and bacteria will interact [6], others classify which bacterial host a given phage will interact with [5,7,9,10], while others rank potential phage candidates for further in vitro validation [8]. In this work we designed a model to predict whether a given phage and bacteria will interact, and used Boaeckaerts2021 as a baseline, which achieved 89% accuracy among 7 host classes.

These models generally use one major modality as input – this can be phage genetic sequences [5, 10], phage receptor binding protein (RCP) sequences [8,9], or biological features [6,7,11]. Our work takes a novel approach and leverages multiple modalities of phage information - genetic sequences and binding protein sequences – simultaneously to improve accuracy in PBI prediction. We utilize tokenization, embedding, and attention based concepts from Natural Language Processing (NLP) to capture the intricate patterns of interactions within the gene and protein sequences. This approach integrates biological understanding directly into the feature representation, potentially increasing both the accuracy and biological relevance of the predictions.

3 Methods

3.1 Data Collection and Preprocessing

Phage-bacteria interactions were collected from VirusHostDB, a public database that organizes virus/host relationships based on their National Center for Biotechnology Information (NCBI) taxonomy ID's, with 17,639 phage-bacteria pairs in the dataset. Bacteria with fewer than 20 identified interactions were removed to reduce data imbalance, resulting in 13,756 phage-bacteria pairs. Full phage genome sequence and tail protein amino acid sequences were gathered from the NCBI database via API. Phages containing genetic information but no protein information were removed from the dataset.

Additional preprocessing discarded protein sequences with fewer than 200 amino acids [8, 12] and longer than 1024 amino acids (due to the ESM-2 context window limit). Phages with tokenized DNA sequences longer than 51,000 tokens (∼200k nucleotides) were removed due to difficulty processing through the DNABERT-2 transformer model. The final dataset contained 10,652 gene sequences and 53,092 protein sequences across 132 bacterial hosts.

3.2 Handling Long Context DNA:

DNA nucleotide sequences were converted to tokens via DNABERT-2 k-mer based tokenization [13]. The context window for DNABERT-2 is only 512 tokens, but the DNA strands are on average 15k tokens. To handle this long context issue, tokenized genomes were broken into 100 chunks, with overlap added between neighboring chunks to improve context retention. If a tokenized DNA strand did not fill up the full 100 chunks, padding chunks were added to maintain dimensionality. CLS and EOS tokens were added at the beginning and end of each chunk, respectively, and the matrix was passed through a pre-trained DNABERT-2 model to create distinct chunk embeddings for use downstream.

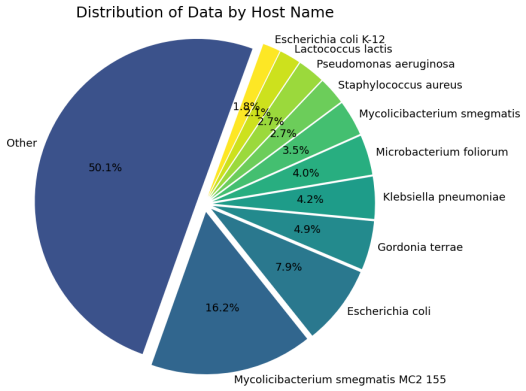

Figure 1: Distribution of host data across the 132 bacterial hosts in the dataset. Specifically shows distribution for the top 10 hosts, bucketing all additional hosts into the "Other" category

3.3 Upsampling to Correct Data Imbalance

The data distribution among bacterial hosts is imbalanced and heavily weighted towards the top few bacterial candidates. Figure 1 shows a pie chart of data percentage for the top 10 hosts, with the remainder of hosts bucketed into the "Other" category. To address class imbalance, a weighted random sampling strategy was utilized during the data loading phase, resulting in a probabilistically even distribution among classes in the training data exposed to the model. Validation and testing data remained unweighted to preserve a realistic assessment of the model.

3.4 Model Architecture

Figure 2 shows the full model architecture developed for this work. Both modalities (DNA and protein) were processed in separate streams, distilled to a single vector each, and concatenated for classification with a fully connected classifier. As mentioned previously, the DNA tokens were chunked into 100 chunks and passed through a pre-trained DNABERT-2 embedding model. The output CLS chunk embeddings were gathered and passed to a second, trainable copy of DNABERT-2. This custom DNABERT-2 was repurposed to directly accept chunk embeddings as opposed to tokens, and to output a new CLS embedding that represents the entire DNA strand in a single vector.

For proteins, up to 10 protein sequences were considered per phage. If a phage had more than 10 associated proteins, the 10 selected proteins were chosen at random. 5.0% of phages were affected by this reduction. Similar to DNA, if a phage did not have 10 proteins, padding proteins were added to retain dimensionality. This matrix was passed to a pre-trained ESM-2 (t33_650M_UR50D) [14] protein transformer model to convert each protein sequence to an embedding. The protein embeddings were then combined via some function f. The two functions evaluated in this work were a mean across the 10 proteins for each embedding dimension, or an absolute value maximum for each embedding dimension, which reduced the protein embeddings into a single embedding vector.

Finally, the DNA and protein embedding vectors were concatenated and passed to one of two feed forward classifiers as shown in the red box of Figure 2. In the simple classifier (1), the concatenated embeddings were passed through hidden layers to a final classification layer, where a softmax was performed to identify the probability of each bacterial host. In the dual-headed classifier (2), the concatenated vector was similarly passed through hidden layers, but split at the end into a binary classifier that predicts "main class" or "other class", and a multi-class classifier that predicts which host is most likely among the main hosts ("other" not included as an option).

Trainable layers in the model are the fully connected classifier layers and the custom DNABERT-2 transformer model. The initial DNABERT-2 and the ESM-2 models were frozen during training.

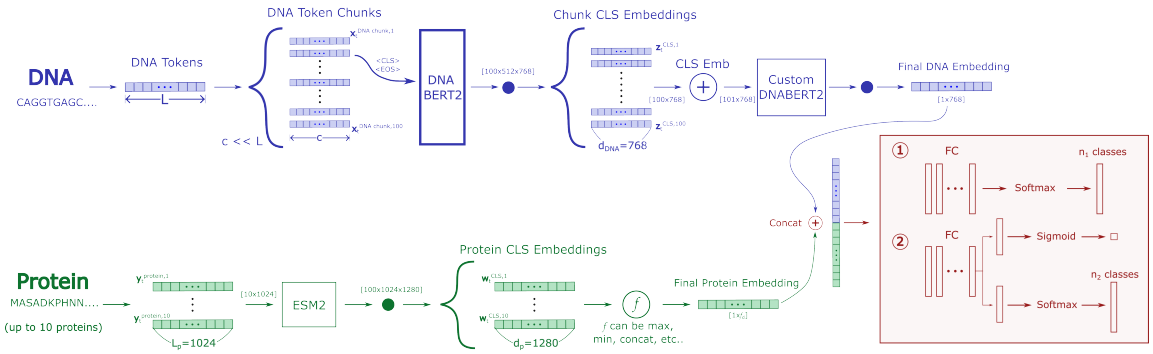

Figure 2: Model architecture for Phage-Bacteria Interaction prediction. Details the processing of DNA and protein sequences through DNABERT-2 and ESM-2, respectively, to generate embedding vectors. Concatenated DNA + protein vector is passed to either (1) a single-headed classification head or (2) a dual-headed classifier

4 Results

4.1 Hyperparameter Optimization

The Optuna parameter optimization library was used to identify the optimal hyperparameters of the model. The parameters varied were learning rate, dropout rate, hidden layer depth and width, protein combination (mean or max), weight initialization, and batch size. 30 experiments were performed for prediction among the top 5 hosts, and the best performing parameters were identified and used for all subsequent experiments. See the Appendix for more information about this optimization study.

4.2 Protein-Only vs Multimodal DNA/Protein Model

A model that only used protein information was compared head-to-head against a multimodal model that included both protein and DNA information. The protein-only model passed the final protein embedding in Figure 2 directly into the fully-connected layers, while the multimodal model concatenated the protein and DNA embeddings before passing the vector to the fully connected layers. Figure 3 shows the training and validation loss curves and the accuracy curves for these model configurations, predicting

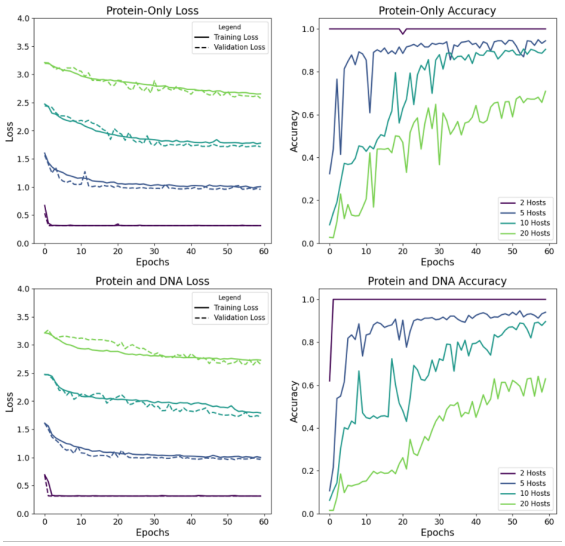

Figure 3: Loss and accuracy results for the protein-only model (a and b) and the protein and DNA model (c and d) across a range of top-n host experiments

among the top 2, 5, 10, or 20 hosts, for 60 epochs. Figure 3a and b correspond to results from the protein-only version of the model while Figure 3c and d correspond to results from the multimodal model. Data not among the top n-hosts for a given run were not used in that training or validation set.

The protein-only model shows good performance across the range of top n-hosts,

with validation loss tracking training loss throughout the experiment, indicating limited overfitting. The loss increases as the number of hosts increases because the prediction weight becomes spread out among more classes, but the accuracy results in Figure 3b still show good performance. With 2 hosts, the validation accuracy is 100% (majority class of 67%) and with 20 hosts, validation accuracy is 71% (majority class of 29%). Baseline from Boaeckaerts2021, who predicted among 7 host classes, was was 89% for reference.

Conversely, the model that incorporates both DNA and protein (Figure 3c and d) shows no meaningful improvement in accuracy compared to the protein-only model. With two hosts, validation accuracy for the multimodal model is 100%, and with 20 hosts validation accuracy is 63%. This was a surprising result as the genetic information was expected to enhance prediction capability.

4.3 Including "Other" Class

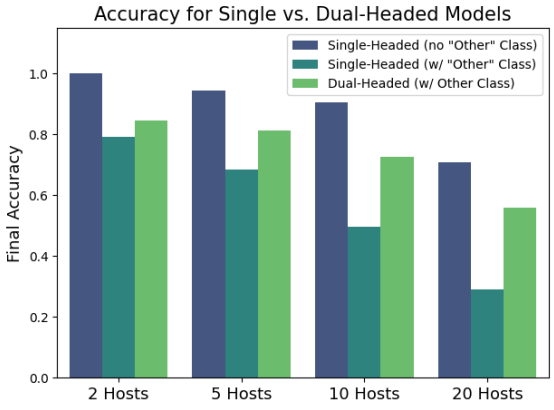

Figure 4: Accuracy results for the single-headed classifier with and without "Other class" as well as the dual-headed classifier with "Other" class. All models were based on the protein-only version.

While predicting which host a phage will interact with among a group of preselected host options is valuable, a more realistic scenario must consider the option that the phage may not interact with any of the preselected choices. To operationalize this idea, we incorporated an additional class named "Other" for data that is not in the top n-hosts. This allows the classifier to predict cases where the interaction falls outside the preselected top-n options.

The single-headed classifier takes the simplistic route of adding an additional class to the list of predictable classes. In the dual-headed approach, a binary classifier first predicts if a sample belongs in the "Other" or "Main Class" categories. If "Main Class" is selected, the sample progresses to a multi-class predictor without "Other" as an option. This splits the decision-making process by first resolving whether the sample is "Other" and then identifying the most likely host if the host is not categorized as "Other".

Figure 4 displays the final validation accuracy values after 60 epochs of training for both the single-headed classifier, with and without the "Other" class, and the dual-headed classifier, which includes the "Other" class. Introducing the "Other" class to the single-headed model significantly reduces accuracy. Accuracy when predicting among the top 2 hosts drops from 100% to 79%, and among the top 20 hosts, it falls from 71% to 29%. By switching the architecture from single-headed to dual-headed, the accuracy shows moderate improvement, especially as host number increases. For instance, in the 20 host experiment, the dual-headed architecture improves the accuracy from 29% to 56%. This demonstrates that by converting the model head to dual-headed, the model is able to recapture much of the performance loss caused by introducing the "Other" category.

5 Discussion

It was anticipated that incorporating DNA information to the model would enhance the

performance relative to the protein-only version. Protein data is only as good as the correctness of the identified receptor binding proteins (RCPs). Although some RCPs are identified through in vitro validation, a significant number are identified by functional prediction tools and are liable to inaccurate predictions. Thus, some RCPs passed to the model as crucial proteins may have no real impact, while other truly critical RCP data may be missing.

In contrast, DNA, by definition, contains the protein codon sequence for any and all RCPs, ensuring that all necessary biological information is present. However, this information must be captured by model's embeddings. A major limitation with DNA data is that the relevant RCP codon sequences are buried within a large amount of unhelpful genetic information, diluting the signal with noise. The results from Figure 3 suggest that despite the codon sequences being present within DNA, the model embeddings were unable to usefully capture that information. Future work could involve using more advanced DNA encoder models as well as trying alternative methods to synthesize the chunked embeddings.

As discussed earlier, including "Other" as a class allows the model to consider scenarios where none of the top n-hosts are likely hosts. However, this addition significantly reduces model performance as shown in Figure 4. Introducing the "Other" class effectively collapses all alternative host options into a single category. Therefore, with the single classifier architecture, we're asking the model to discern whether a phage will interact with 100+ different hosts, or whether it will interact with a specific selected host. The model must be both broad and highly specific simultaneously. Alternatively, the dual-headed approach splits this decision process. For the binary head, while all alternative hosts are collapsed into one class, the main hosts are collapsed into a second class. In the first decision, the model need only predict which clustering is most likely. The granular decision of a which specific host is deferred until after the model selects "Main Class" from the binary head. This splitting of the decision process is hypothesized to be the reason why the dual-headed architecture shows significant improvement over the single-headed classifier. Additional exploration of this hypothesis is available in the Appendix section.

6 Conclusion

This paper demonstrates the potential of machine learning to predict phage-bacteria interactions (PBIs) to combat antimicrobial resistance. Our findings show that the protein-only model exhibited excellent prediction accuracies when predicting among the top n-bacterial hosts. For an n value of 2, 5, 10, and 20, the model achieved an accuracy of 100%, 95%, 90%, and 71%, respectively. However, introducing the "Other" category, which includes data not among the top n-hosts, initially reduced model performance. By converting the model architecture from a single-headed to dual-headed classifier, we were able to mitigate the majority of this performance loss.

Despite the promising concept of incorporating DNA data, the multimodal DNA/protein model did not significantly improve over the protein-only model. The integration of DNA data did not yield the anticipated benefits, likely due to the complexity and volume of genetic data, which may have diluted the model's ability to capture meaningful biological signals. This highlights the need for future work to build more advanced DNA models or to handle the DNA chunk embeddings more effectively.

References

[1] Lee Ventola. The antibiotic resistance crisis. *P & T*, 40:277–283, 2015.

[2] Christopher Murray, Kevin Ikuta, Fablina Sharara, and Lucient Swetschiniski. Global burden of bacterial antimicrobial resistance in 2019: a systematic analysis. *The Lancet*, 399:629–655, jan 2022.

[3] Zhabiz Golkar, Omar Bagasra, and Donald Gene Pace. Bacteriophage therapy: a potential solution for the antibiotic resistance crisis. *JIDC*, 8:129–136, 2014.

[4] Mathias Middelboe and Amy Chan. Isolation and life-cycle characterization of lytic viruses infecting heterotrophic bacteria and cyanobacteria. *Manual of Aquatic Viral Ecology*, 13:118–133, 2010.

[5] Wang Ruohan, Zhang Xianglilan, Wang Jianping, and Shuai Cheng. Deephost: phage host prediction with convolutional neural network. *Briefings in Bioinformatics*, 23, 2022.

[6] Yanan Wang, Fuyi Li, Yun Zhao, Mengya Liu, Sijia Zhang, Yannan Bin, Ian Smith, and Geoffrey Webb. A deep learning-based method for identification of bacteriophage-host interaction. *IEEE*, 18:1801–1810, 2021.

[7] Dimitri Boeckaerts, Michiel Stock, Bjorn Criel, Hans Gerstmans, Bernard De Baets, and Yves Briers. Predicting bacteriophage hosts based on sequences of annotated receptor-binding proteins. *Nature Scientific Reports*, 11:1467, 2021.

[8] Dimitri Boaeckaerts, Michiel Stock, Celia Ferriol-Gonzalex, Jesus Oteo-Iglesias, Rafael Sanjuan, Pilar Domingo-Calap, Bernard De Baets, and Yves Briers. Prediction of klebsiella phage-host specificity at the strain level. *Nature Communications*, 15:4355, 2024.

[9] Mark Edward Gonzales, Jennifer Ureta, and Anish Shrestha. Protein embeddings improve phage-host interaction prediction. *PLoS ONE*, 18, 2023.

[10] Carlos Moyano Gravalos. Deep learning on genomics using nlp-oriented algorithms. Master's thesis, Universitat Politècnica de Catalunya (UPC) - BarcelonaTech, Facultat d'Informàtica de Barcelona (FIB), Barcelona, 6 2023. Thesis supervisor: Carlos Peña (HEIG-VD), Tutor: Carlos Escolano Peinado (Department of Computer Science).

[11] Diogo Leite, Juan Lopez, Xavier Brochet, Miguel Barreto-Sanz, Yok-Ai Que, Gregory Resch, and Carlos Pena-Reyes. Exploration of multiclass and one-class learning methods for prediction of phage-bacteria interaction at strain level. *IEEE*, pages 1818–1825, 2018.

[12] Agnieszka Latka, Petr Leiman, Zuzanna Drulis-Kawa, and Yves Briers. Modeling the architecture of depolymerase-containing receptor binding proteins in klebsiella phages. *Front. Microbiol*, 10:2649, 2019.

[13] Zhihan Zhou, Yanrong Ji, Weijian Li, Pratik Dutta, Ramana Davuluri, and Han Liu. Dnabert-2: Efficient foundation model and benchmark for multi-species genome. *Arxiv*, 2023.

[14] Zeming Lin, Halil Akin, Roshan Rao, Biran Hie, and Alexander Rives. Evolutionary-scale prediction of atomic level protein structure with a language model. *Science*, 379:1123–1130, 2023.

7 Appendix

7.1 Optuna Optimization

The Optuna Optimization process, mentioned in the Results section, was completed to identify the optimal hyperparameters for learning rate, dropout rate, hidden layer depth and width, protein combination (mean or max), weight initialization, and batch size. Table 1 lists these variables along with their potential values for selection by the Optuna algorithm. All variables were categorical except for learning rate, which was selected as a continuous numerical value between listed values.

Over the course of the study, the Optuna package performed 30 experiments, for 30 epochs each, utilizing a Tree-structured Parzen Estimator (TPE) optimization algorithm to select the best parameters. The final configurations were identified as follows: learning rate = 0.0003, dropout rate = 0.3, hidden layer sizes = [512, 256, 128, 64], protein combination = 'mean', weight initialization = 'none', and batch size = 32.

7.2 Loss and Accuracy Curves of Single vs Dual-Headed Architectures

Loss and accuracy curves are provided in Figure 5 for the single and dual-headed classifier architectures as comparison to the model without inclusion of "Other", shown in Figure **??**. The single-headed classifier plots are on top and the multi-headed classifier plots are on the bottom. The single headed classifier showed significant overfitting on training data and had lower accuracy across the board. Most experiments reached an asymptote after only 10-20 epochs. Conversely the dual-headed architecture showed consistent training and validation loss curves and achieved higher accuries by the 20 epoch mark. Loss values are lower as the loss function was a weighted combination of the binary head loss and the multi-head loss.

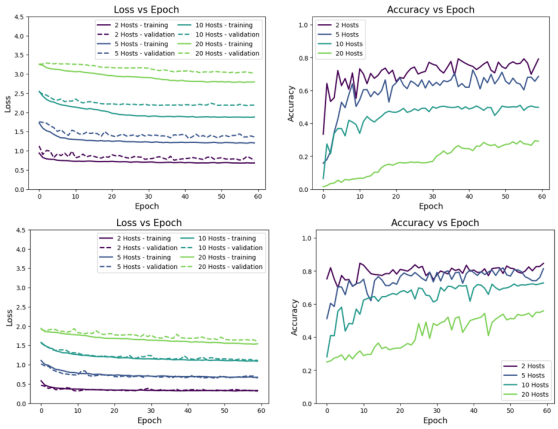

Figure 5: Loss and accuracy results for the protein-only model with a single-headed classifier (top) and a multi-headed classifier (bottom) over 60 epochs, across a range of top-n host experiments.

7.3 Confusion Matrices

7.3.1 Protein Only, no "Other"

The confusion matrix for the protein-only, single-headed model without considering "Other" as an class is shown in Figure 6. This model had 91% validation accuracy, correctly predicting the majority of interactions. The main misclassified prediction is that the model predicted Klebsiella Pneumoniae was Escherichia Coli. The taxonomy classification for bacteria goes domain → kingdom → phylum → class → order → Family → Genus. While these two bacteria are of a different genus, they are of the same family, and may have many similarities between them. This is likely the reason for model confusion in this experiment.

7.3.2 Protein and DNA, no "Other"

Figure 7 shows a confusion matrix for the DNA and protein model as opposed to the protein-only version. The same issue occurs here as well, with bacteria of the same family being confused by the model, but the remainder of the bacteria are well categorized.

Table 1: Parameter Settings for Model Optimization

Parameter	Values
Learning Rate	(1e-5, 3e-2)
Dropout Rate	0.1, 0.3, 0.5
Hidden Layer Sizes	[1024, 512, 256, 64], [512, 256, 128, 64], [512, 256, 128], [512, 256], [256, 128, 64], [128, 64, 32]
Protein Combination	mean, max
Weight Initialization	Kaiming normal, Kaiming uniform, None
Batch Size	8, 16, 32, 64

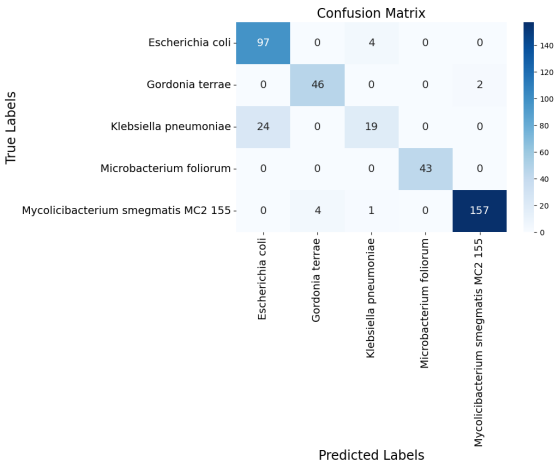

Figure 6: Confusion matrix results the protein-only model without "Other" class as an option. Experiment performed on top-5 hosts.

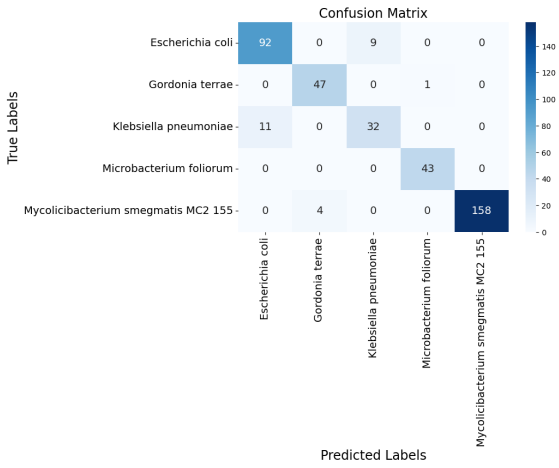

Figure 7: Confusion matrix results the DNA and protein model without "Other" class as an option. Experiment performed on top-5 hosts.

7.3.3 Single-Headed

For this experiment, "Other" was included as an additional category alongside the top 5 candidate hosts. As observed in Figure 8, the majority of classification errors occurred when the model incorrectly labeled an "Other" sample as one of the main class samples. Importantly, when the top n-hosts is set to 5 hosts, the data is comprised of ¿60% "Other" class data. But as mentioned in the Methods section, the data distribution was upsampled during training to create an even balance across all classes. This likely caused an under-sampling of "Other" class data relative to its variety, introducing confusion in the model's predictions. However, if more "Other" class data is proportionally passed to the model, the model learns to mainly pick "Other" to be correct the majority of the time, resulting in different challenges where the model overlooks significant categories.

7.3.4 Dual-Headed

The dual-headed architecture requires a confusion matrix for each classifier head, as shown in Figure 9, with the multi-class head shown on the left and the binary head on the right. The binary classifier achieves a 78% accuracy rate in correctly identifying "Other" samples, a significant improvement over the 45% accuracy of the single-

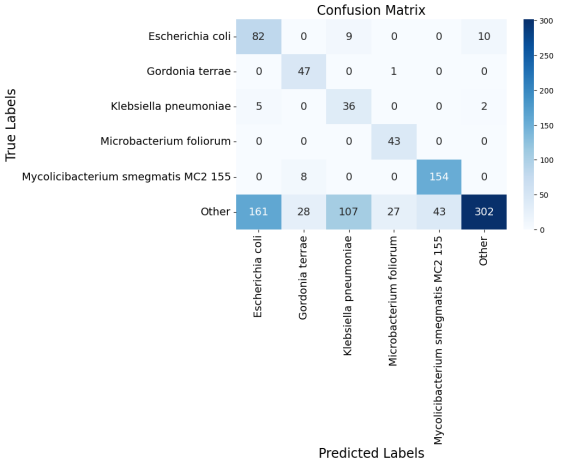

Figure 8: Confusion matrix results the protein-only, single-headed classifier model with "Other" class as an option. Experiment performed on top-5 hosts.

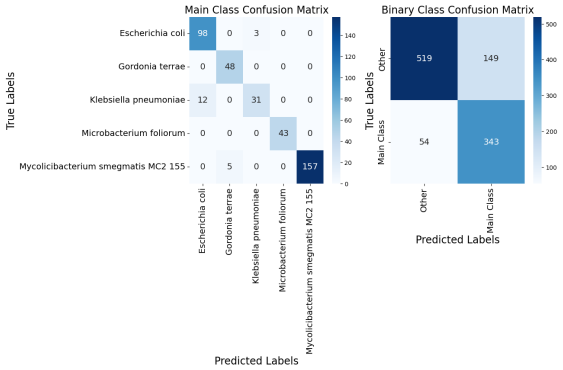

Figure 9: Confusion matrix results the protein-only, dual-headed classifier model with "Other" class as an option. Experiment performed on top-5 hosts.

headed classifier from Figure 8. This large improvement is the main driver of the overall accuracy improvement from employing the dual-headed architecture. Downstream, once the "Main Class" samples are passed to the multi-class head, this classifier does an excellent job predicting among the top 5 classes, suggesting that the main class classifier is not meaningfully contributing to the model's confusion. This result further confirms that the improvement lies in the dual-headed model's capability of discerning which samples are "Other" and which samples are "Main Class", relative to the single-headed model.

The purpose of the present research was to improve a model's ability to classify social media posts about depression vs. anxiety. Baseline models derived from Murarka et al.'s (2021) research were compared to two sets of experimental models that were enhanced in the following ways: 1) adding new data from a different source, and 2) applying an enhanced labeling technique based on validated anxiety and depression inventories used to measure symptoms. While previous research has focused on classification of anxiety and depression using mixed data from multiple sources (e.g., Zeberga et al., 2022),