**Classification of Antibiotic Resistance Gene Sequences using Nucleotide Transformer**

**Prasanth Kumar Thuthika**

**Yesasvi Sai Nandigam**

**Transformers**

The Transformer model, first introduced by Vaswani et al. (2017), represents a groundbreaking development in natural language processing (NLP) and artificial intelligence. Unlike traditional sequence-based models such as recurrent neural networks (RNNs) and long short-term memory (LSTM) networks, which process inputs sequentially, Transformers employ an innovative mechanism called self-attention. This mechanism enables the model to analyse all elements of a sequence simultaneously, allowing for the efficient capture of contextual relationships across both short and long distances. By eliminating the sequential nature of processing, the Transformer architecture accelerates training and improves scalability for large datasets. The design incorporates an encoder-decoder structure, where the encoder processes input data and the decoder generates output sequences, both relying heavily on self-attention and feed-forward neural layers. Moreover, the use of positional encodings helps the model preserve the order of sequence elements, addressing its otherwise order-agnostic nature. These advancements have made Transformers a foundational component in state-of-the-art applications, inspiring influential models like BERT (Devlin et al., 2019) and GPT (Brown et al., 2020). The Transformer’s versatility has extended beyond NLP into other areas, including computer vision, bioinformatics, and protein structure prediction.

A Transformer model processes input using an encoder-decoder structure driven by the self-attention mechanism. The input sequence is first tokenized and converted into embeddings, which are then enriched with positional encodings to retain the order of tokens. In the encoder, self-attention calculates relationships between all tokens, highlighting relevant parts of the sequence for each token. These attention scores are combined with feed-forward neural networks to produce a set of encoded representations. The decoder takes these encoded representations and, along with the previously generated output tokens, applies a similar self-attention mechanism and cross-attention to align the input and output sequences. Finally, the decoder generates the output token by token, passing through a SoftMax layer to predict the next word in the sequence. This process ensures efficient, parallelized computation while capturing both local and global dependencies.

A diagram of a software algorithm

AI-generated content may be incorrect.

**Fig-1: Nucleotide Transformer Model Architecture**

**Nucleotide Transformer**

The Nucleotide Transformer model marks a significant advancement in bioinformatics by applying transformer architectures to DNA and RNA sequences. Derived from the original Transformer model (Vaswani et al., 2017), which was primarily designed for natural language processing (NLP), the Nucleotide Transformer adapts this approach to process biological sequences. These models are trained to understand the “language” of nucleotides, drawing parallels between the sequential structure of genetic data and the linguistic patterns in text. By leveraging self-attention mechanisms, Nucleotide Transformers capture the complex and long-range dependencies within nucleotide sequences, which are critical for understanding regulatory elements, mutations, and functional domains in genomes.

Three advanced Nucleotide Transformer models highlight the growing capabilities of transformer architectures in bioinformatics. The first model, trained on over 500 million nucleotide sequences from multiple species, demonstrates robust performance across diverse genomic tasks such as transcription factor binding site prediction and motif discovery. The second model, with 3.5 billion parameters, significantly expands the scope of analysis, enabling deep insights into complex biological patterns like sequence evolution, enhancer activity prediction, and cross-species genomic comparisons. A third, even larger-scale model, with over 10 billion parameters, was trained on an extensive dataset comprising both DNA and RNA sequences. This model excels in tasks such as de novo sequence generation, long-range dependency detection, and fine-grained functional annotation, proving invaluable for high-resolution genomic studies. Together, these Nucleotide Transformers exemplify the potential of large-scale, multi-species training in revolutionizing genomic research and precision medicine.

Several versions of Nucleotide Transformers have been developed, each tailored to specific use cases and datasets. For example, DNABERT (Ji et al., 2021) was trained on a large corpus of genomic sequences and focuses on tasks such as sequence classification, mutation impact prediction, and motif discovery. Another model, Genome Transformer, is optimized for whole-genome data analysis, leveraging extensive datasets from public repositories like GenBank and ENCODE. These models often use k-mer tokenization strategies, which segment sequences into overlapping substrings of fixed lengths, enabling the transformer to process biological sequences effectively. The datasets used to train Nucleotide Transformers are typically vast and diverse, including reference genomes, transcriptomic data, and epigenomic datasets. For instance, DNABERT was trained on sequences from human and other species’ genomes, incorporating diverse genomic regions to enhance its ability to generalize across different tasks. The training data is pre-processed to ensure uniformity, with careful consideration of sequence length, encoding, and biological relevance.

The use cases for Nucleotide Transformers are diverse and impactful. These models excel in predicting transcription factor binding sites, identifying enhancer elements, and classifying diseases based on genomic mutations. They are also increasingly used in drug discovery, where they help identify therapeutic targets by analysing genomic and transcriptomic data. Additionally, Nucleotide Transformers have been applied in microbiome analysis, enabling the characterization of microbial communities and their functional potential. By improving the interpretability and accuracy of sequence-based analyses, Nucleotide Transformers are transforming how researchers study complex biological systems.

**A diagram of a process

AI-generated content may be incorrect.**

**Fig-2a: Application for downstream genomics prediction tasks through fine-tuning**

**Fig-2b: Downstream task prediction through probing is similar but without the rescaling weights in the N.T**

**Objective**

Antibiotic resistance poses a critical global health challenge, driven largely by the spread of antibiotic resistance genes (ARGs) in bacteria, which reduce the efficacy of antibiotics in treating infections (Partridge et al., 2018). Machine learning (ML) and deep learning (DL) methods have shown promising applications in the classification and prediction of ARGs, enabling researchers to analyse large-scale genomic data and identify resistance-associated patterns with high accuracy (Yang et al., 2020). Current approaches utilize convolutional neural networks (CNNs), recurrent neural networks (RNNs), and pre-trained transformer-based models for sequence classification and resistance prediction. These methods have significantly improved the speed and precision of ARG detection compared to traditional bioinformatics tools.

The objective of this project is to utilize the Nucleotide Transformer model, a cutting-edge transformer-based architecture, and fine-tune it specifically for the classification of antibiotic resistance gene (ARG) sequences. By training the model on curated, labelled datasets containing resistant, non-resistant gene sequences, synthetic non-resistant gene sequences the goal is to enable the accurate identification and classification of ARGs. The fine-tuned model will be tested and validated using independent datasets to evaluate its performance, robustness, and generalizability across a diverse range of bacterial species.

**Data**

The data for this project is sourced from the CARD database <https://card.mcmaster.ca> and consists of 22,000 antibiotic-resistant gene sequences. From this dataset, 10,000 sequences were randomly selected. Additionally, 5,000 sequences were extracted from the rest of the data and modified by introducing mutations and shuffling nucleotide positions to create synthetic non-resistant gene sequences. Non-resistant gene sequences were downloaded from the RefSeq database <https://www.ncbi.nlm.nih.gov/refseq/> . To ensure there were no repeated sequences between the non-resistant gene sequences and the resistant sequences from CARD, BLAST was used to filter out any duplicates. The data was then labeled as follows: **1** for antibiotic-resistant genes and **0** for antibiotic non-resistant and synthetic non-resistant genes. The final dataset comprises 20,000 sequences, which were divided into training, testing, and validation datasets.

**Methodology**

Download the model from   
hugging face

Data

Validation

Train data / Test data

Training

500B Parameters

3 B Parameters

Save the model

Cross validation with other species

Validation

Results

Finetune

**Training Setup**

We trained and evaluated transformer-based models for sequence classification, specifically focusing on the 500M multi-species model and the 3.5B parameter model. Given the extensive computational requirements of these models, careful consideration was given to optimizing resource utilization and training efficiency.

The transformer models were sourced from Hugging Face, a widely used repository for state-of-the-art deep learning models. We selected the InstaDeepAI/Nucleotide-Transformer-v2-500m-multi-species model as the primary model for training. This model, containing 500 million parameters, serves as an efficient baseline for sequence classification tasks. Additionally, we also used the 3.5 billion parameter model, which requires significantly greater computational resources. To ensure seamless access and avoid dependency on real-time internet connections during training, we downloaded the complete model repository from Hugging Face and stored it locally on our High-Performance Computing (HPC) cluster.

The training was conducted on an HPC cluster with Slurm job scheduling. Initially, our training workflow was designed to leverage GPU acceleration, given the high parameter count of the selected models. However, due to ongoing GPU maintenance and limited availability of high-memory GPU nodes, we adapted our training pipeline to run efficiently on CPUs.

To optimize the training of the 500M multi-species model on CPUs, several adjustments were made to improve efficiency and handle computational limitations. The dataset was pre-processed and tokenized using Hugging Face Datasets with batched tokenization to leverage multi-threading, while padding and truncation were optimized to reduce unnecessary computation. Since CPUs have limited parallel processing compared to GPUs, the training batch size was reduced to 8, along with an evaluation batch size of 8 to balance memory usage. Mixed precision training, typically used on GPUs, was disabled since CPUs do not support FP16, and instead, dynamic loss scaling was applied to prevent underflow issues. Given the longer training time on CPUs, checkpointing was enabled in Hugging Face’s Trainer by using resume\_from\_checkpoint=True, allowing the job to resume from the last checkpoint in case of interruptions. The model was trained using Slurm with one node, 8 CPUs per task, 100GB of memory, and a maximum runtime of 48 hours. The training script was executed within a dedicated Conda environment (ml) with all dependencies pre-installed, ensuring a smooth workflow despite the limited computational resources.

The training process began by loading the locally stored model and initializing the Hugging Face Trainer API. The progress of training is continuously logged using Slurm output logs, allowing for real-time monitoring of performance. As the training progresses, we observe decreasing iteration times, indicating improved efficiency in batch processing. Despite running on CPUs, our optimizations help maintain reasonable training times while ensuring stable performance.

**Expected Results**

The primary expected outcome of this project is the successful development of a fine-tuned Nucleotide Transformer model capable of accurately classifying antibiotic resistance genes (ARGs) from nucleotide sequences. By leveraging the pretrained contextual representations learned from large-scale genomic data, the fine-tuning process on carefully labeled ARG datasets is anticipated to significantly enhance the model’s ability to detect subtle sequence motifs, mutations, and discriminative features that differentiate resistant genes from non-resistant and synthetic variants.

The model is anticipated to attain elevated performance metrics namely accuracy, precision, recall, and F1-score on both validation and test sets, signifying its efficacy in generalizing over varied sequence inputs. Furthermore, we expect that the model's attention-based architecture will enable it to concentrate on biologically significant areas within sequences, possibly exposing illuminating patterns associated with resistance mechanisms.

The model is anticipated to exhibit strong performance when assessed on synthetic non-resistant sequences, showcasing its capacity to differentiate genuine resistance-conferring patterns from artificially modified or ambiguous ones. These findings will ultimately confirm the suitability of transformer-based models for classifying genomic sequences and demonstrate their potential for application in the development of diagnostic tools, surveillance of antibiotic resistance, and more general bioinformatics research.

**References**

1. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., … & Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems, 33*, 1877–1901. <https://doi.org/10.48550/arXiv.2005.14165>
2. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186. https://doi.org/10.18653/v1/N19-1423
3. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*, 5998–6008. <https://doi.org/10.48550/arXiv.1706.03762>
4. Ji, Y., Zhou, Z., Liu, H., & Davuluri, R. V. (2021). DNABERT: Pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. *Bioinformatics, 37*(15), 2112–2120. <https://doi.org/10.1093/bioinformatics/btab083>
5. ENCODE Project Consortium. (2020). Expanded encyclopaedias of DNA elements in the human and mouse genomes. *Nature, 583*(7818), 699–710. <https://doi.org/10.1038/s41586-020-2493-4>
6. Partridge, S. R., Kwong, S. M., Firth, N., & Jensen, S. O. (2018). Mobile genetic elements associated with antimicrobial resistance. *Clinical Microbiology Reviews, 31*(4), e00088-17. <https://doi.org/10.1128/CMR.00088-17>
7. Yang, Y., Niehaus, K. E., Walker, T. M., Iqbal, Z., & Pečerska, J. (2020). Machine learning for detecting antimicrobial resistance genes in bacterial genomes. *Frontiers in Microbiology, 11*, 550. <https://doi.org/10.3389/fmicb.2020.00550>
8. Partridge, S. R., Kwong, S. M., Firth, N., & Jensen, S. O. (2018). Mobile genetic elements associated with antimicrobial resistance. *Clinical Microbiology Reviews, 31*(4), e00088-17. <https://doi.org/10.1128/CMR.00088-17>
9. Yang, Y., Niehaus, K. E., Walker, T. M., Iqbal, Z., & Pečerska, J. (2020). Machine learning for detecting antimicrobial resistance genes in bacterial genomes. *Frontiers in Microbiology, 11*, 550. <https://doi.org/10.3389/fmicb.2020.00550>