

Optimizing Gene-Based Testing for Antibiotic Resistance Prediction

David Hagerman¹, Anna Johnning^{1,2}, Roman Naeem¹,
Fredrik Kahl¹, Erik Kristiansson^{1,2}, Lennart Svensson¹

¹Chalmers University of Technology

²Fraunhofer-Chalmers Centre

{david.hagerman, anna.johnning, nroman}@chalmers.se,

{fredrik.kahl, erik.kristiansson, lennart.svensson}@chalmers.se

Abstract

Antibiotic Resistance (AR) is a critical global health challenge that necessitates the development of cost-effective, efficient, and accurate diagnostic tools. Given the genetic basis of AR, techniques such as Polymerase Chain Reaction (PCR) that target specific resistance genes offer a promising approach for predictive diagnostics using a limited set of key genes. This study introduces GenoARM, a novel framework that integrates reinforcement learning (RL) with transformer-based models to optimize the selection of PCR gene tests and improve AR predictions, leveraging observed metadata for improved accuracy. In our evaluation, we developed several high-performing baselines and compared them using publicly available datasets derived from real-world bacterial samples representing multiple clinically relevant pathogens. The results show that all evaluated methods achieve strong and reliable performance when metadata is not utilized. When metadata is introduced and the number of selected genes increases, GenoARM demonstrates superior performance due to its capacity to approximate rewards for unseen and sparse combinations. Overall, our framework represents a major advancement in optimizing diagnostic tools for AR in clinical settings.

Datasets — <https://www.ncbi.nlm.nih.gov/pathogens/>

Code — <https://github.com/Eiphodos/GenoPhen>

Introduction

Antimicrobial Resistance (AMR) poses an escalating challenge to global health, estimated to result in more than 10 million yearly deaths by 2050 (Laxminarayan et al. 2013; O’Neill 2014, 2016), the overwhelming majority of which are attributed to antibiotic resistance (AR). Pathogens become resistant to antibiotics through mutations in their genome, either by mutations in pre-existing genes or by acquiring antibiotic-resistance genes from other bacterial cells. The rate at which pathogens acquire resistance traits significantly exceeds the pace of new drug development, underscoring the urgent need for increased research efforts in this domain. As a result, there is a strong demand for tools that assist in prescribing the most effective antibiotics against infections. These tools should be capable of estimating antibi-

otic resistance with high precision, enabling rapid identification of resistant strains and targeted treatment, which could reduce antibiotic misuse and slow the spread of resistance.

Recent studies (Tharmakulasingam et al. 2022, 2023; Macesic et al. 2020; Jin et al. 2024; Kuang et al. 2022) have shown that deep learning models using complete pathogen genomic information can accurately predict AR. However, the requirement for full genome sequencing presents significant obstacles in a clinical context, primarily due to its high cost, slow processing times, and overall impracticality for rapid decision-making. An alternative approach involves using tests based on Polymerase Chain Reaction (PCR) amplification, which focuses on analyzing a targeted subset of known resistance mutations and/or genes. Compared to full gene sequencing, PCR tests are faster, more cost-effective, and therefore often more suitable for clinical use.

While PCR tests offer a faster and cheaper alternative to full gene sequencing, they require the careful selection of a specific subset of genes for testing. With more than 5,000 genes and mutations linked to AR (Alcock et al. 2023), each varying in prevalence and diagnostic value, identifying the most informative subset is a significant challenge; naive approaches like brute-force testing or selecting the most common AR genes are costly and often suboptimal.

A method capable of identifying the most informative subset of genes could transform susceptibility testing by targeting the full resistance phenotype, rather than addressing individual resistance types as standard PCR assays do. This would reduce costs and improve efficiency by eliminating redundant measurements, such as co-located resistance genes, and enable laboratories to optimize PCR assays. By consolidating the detection of all resistance types into a single test, this approach simplifies diagnostics while retaining accuracy and practicality for clinical use.

Motivated by the success of reinforcement learning (RL) in solving complex optimization problems, this work explores its application to efficiently identify the most informative gene subsets for PCR and other gene tests. RL’s ability to handle high-dimensional state-action spaces while learning complex relationships between different components makes it a good choice for this problem. Predicting resistance against a wide range of antibiotics from a limited subset of gene test results is a partial set classification problem, requiring an approach that can utilize unordered and

missing data. Additionally, AR can vary greatly depending on metadata data like country and date, thus these factors need to be accounted for when selecting a subset. Finally, labels in AR datasets are commonly missing due to non-standardized testing procedures, and models therefore need to be able to handle missing labels.

In this study, we address the challenges presented by optimizing gene test selection to enhance AR predictions and our main contributions can be summarized as the following:

- We present a model for AR prediction using a novel representation of gene test results.
- We introduce a reinforcement learning-based method for training a gene test selection policy that makes use of a trained AR prediction model to evaluate the reward.
- We develop two high-performing baselines and conduct extensive evaluations to ensure robust comparison across all methods.
- We improve predictive performance by incorporating metadata, demonstrating its significant impact on model accuracy.

These contributions are integrated into a comprehensive framework named GenoARM that delivers a tailored gene test selection policy along with a trained antibiotic resistance prediction model. Extensive experimentation on the large-scale National Center for Biotechnology Information (NCBI) (Sayers et al. 2022) dataset demonstrates the broad applicability and effectiveness of GenoARM across various pathogens.

Related Work

Deep Learning and Antibiotic Resistance

Several recent works (Chakraborty et al. 2022; Kim et al. 2022; Ren et al. 2022), have identified both the potential and the challenges that exist within the intersection of deep learning and AR prediction, and significant effort is already underway in the area.

Many methods (Tharmakulasingam et al. 2022, 2023; Macesic et al. 2020; Jin et al. 2024; Kuang et al. 2022) utilize full gene sequences as input data. These methods either utilize dimensionality reduction on the sequences or convert the sequences to genes through existing algorithms. Gene sequences are rich in information, but they require whole-genome sequencing, which is often impractical in clinical routine.

Some attempts (Her and Wu 2018; Hyun et al. 2020; Kavvas et al. 2018) have been made to identify important AR genes. In (Her and Wu 2018), they use a genetic algorithm to select the best performing genes. However, to predict multiple resistances, their method requires a large subset of genes, not achievable by a single PCR test. Others (Hyun et al. 2020; Kavvas et al. 2018) manage to identify new AR genes along with important existing ones, however, only while utilizing full genome sequences.

Feature Selection Methods

Selecting an optimal subset of genes for testing, based on the predictive performance of a model, can be framed as

a black-box optimization problem, where the prediction model serves as the black box. A wide range (Xue et al. 2016) of methods has been proposed to address such problems, including evolutionary algorithms, search-based techniques and heuristic approaches. For instance, evolutionary algorithms have been explored (Salimans et al. 2017) as alternatives to reinforcement learning for optimizing policy parameters. Several black-box optimization techniques (Pudjihartono et al. 2022) and heuristic methods (Alirezanejad et al. 2020) have been used to identify key gene features. Although adaptable to our problem, exploring these methods fully is beyond the scope of this study.

Active Feature Acquisition

The challenge of selecting the optimal subset of genes also falls under the umbrella of active feature acquisition (AFA) problems. The majority of research (Nam, Fleming, and Brunskill 2021; Li and Oliva 2021; Fahy and Yang 2019; Yin et al. 2020; Shim, Hwang, and Yang 2018) in that area considers the problem as a Markov Decision Process (MDP), where at each state, one can either decide to observe another feature or make a prediction. The INVASE framework (Yoon, Jordon, and Schaar 2018) observes all features before selecting the relevant features, which requires all tests to be performed and eliminates the potential gain of only selecting a subset. ODIN (Zannone et al. 2019) and many other methods (Yu et al. 2023; Yin et al. 2020; Li and Oliva 2021; Nam, Fleming, and Brunskill 2021; Shim, Hwang, and Yang 2018) assume that once an action has been taken, the new state and the result of that action can be observed (often for a cost). This additional information improves subsequent choices but is not applicable to our problem since doctors would not test genes one at a time but instead test for a set of genes all at once.

Methodology

We propose a dual-architecture framework for optimizing PCR test selection in antibiotic resistance prediction, denoted as GenoARM. The framework contains two main modules, an AR prediction model that predicts resistance and a policy network to select the optimal PCR test subset, see Figure 1 for more details.

Problem Formulation

The problem of finding an optimal gene test subset can be formulated as follows. Let $G = \{g_1, g_2, \dots, g_n\}$ be the set of n possible PCR gene tests and denote the result of a test measuring the presence of gene g_k by $e_i(g_k)$ for subject i , defined as

$$e_i(g_k) = \begin{cases} (1, k), & \text{if the } k\text{th gene is present,} \\ (0, k), & \text{otherwise.} \end{cases} \quad (1)$$

The test results for a subset of genes $S \subseteq G$ is then given by

$$e_i(S) = \bigcup_{g \in S} \{e_i(g)\}. \quad (2)$$

Let $A = \{a_1, a_2, \dots, a_{|A|}\}$ be a set of antibiotics and denote the resistance status of subject i to A as $y_i =$

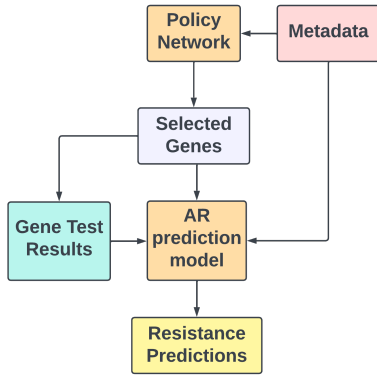


Figure 1: An overview of the full GenoARM framework as used during inference.

$(y_i(a_1), y_i(a_2), \dots, y_i(a_{|A|}))$ where $y_i(a_k) = 0$ if the subject is susceptible, $y_i(a_k) = 1$ if the subject is resistant, and $y_i(a_k) = -1$ if the antibiotic has not been tested. Let $M = \{m_1, m_2, \dots, m_{|M|}\}$ be a set of unique metadata, and let $o_i \in M$ indicate the metadata observed for subject i . Our dataset comes in tuples $\{e_i(G), y_i, o_i\}_{i=1}^N$, providing information on gene presence, antibiotic resistance, and metadata for each sample.

The objective is to find subsets $S_m \subset G$ of gene tests for all $m \in M$ constrained to a maximum size K , such that the resistance profile y_i can be accurately predicted for all subjects i with $o_i \in M$. Let $f(e_i(S_{o_i}), o_i)$ denote a function that predicts the probability of resistance of the antibiotics A using the selected subset of gene tests and metadata, and let $t(f(e_i(S_{o_i}), o_i), y_i)$ denote the mean predictive performance of f across all tested antibiotics in A . The task is then to maximize the average t across all samples i , while limiting the subset size to K :

$$\max_{S_{o_i} \subset G, |S_{o_i}| \leq K} \frac{1}{N} \sum_{i=1}^N t(f(e_i(S_{o_i}), o_i), y_i). \quad (3)$$

Note that each S_m , $m \in M$ can be optimized independently. Put simply, we seek a subset of genes, combined with fixed metadata, that maximizes the average predictive performance of the resistance model across all antibiotics.

AR Prediction Model

The AR prediction model’s task is to determine the resistance profile of a pathogen sample based on a limited set of results from PCR tests. The results of the PCR test include information about the specific genes tested and the prevalence of those genes within the sample. The output, or resistance profile, is a vector whose elements represent the probability that the sampled bacterium exhibits resistance to different antibiotics. An overview of the architecture can be seen in Figure 2.

Formally, we define the AR prediction model as $f_\phi : \{0, 1\}^K \times \mathbb{N}^K, \mathbb{N}^{|M|} \rightarrow [0, 1]^{|A|}$, which maps K gene features, where $K < |G|$, their corresponding existence maps and the observed metadata to $|A|$ Bernoulli probabilities.

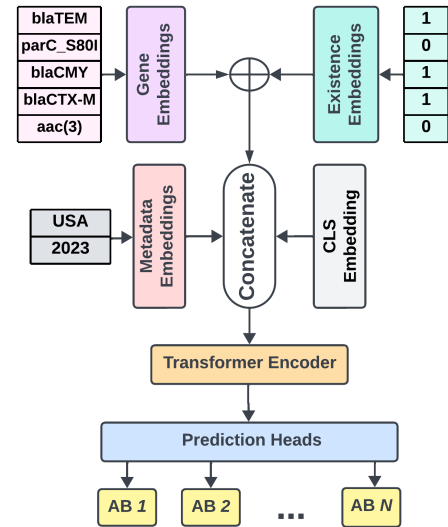


Figure 2: Detailed AR prediction model architecture. Gene test names and metadata is shown un-tokenized but would be tokenized prior to being embedded.

Loss function: The prediction model is parameterized by ϕ and trained by directly minimizing the mean binary cross entropy loss over all antibiotics. As antibiotic labels can be missing from the data set, the loss is ignored in those cases.

Architecture: The architecture of the AR prediction model contains three main components: First, an embedding module that converts the tokenized gene data, existence maps and observed metadata into a feature vector. Next, a stack of transformer-encoder layers refines these feature vectors (Vaswani et al. 2017). Last, A distinct heads, one for each predicted antibiotic. The heads are two-layer multi-layer perceptrons (MLP) using ReLU activations.

The gene tokens and the corresponding existence tokens are processed by separate embedding layers, producing embedding vectors that are combined via addition to create a new feature vector. This method reduces the potential state space from $2 * K^{|G|}$ to $K^{|G|} + 2$, significantly simplifying policy optimization. The resulting gene existence vector is then concatenated with the embedded metadata tokens and a CLS token to form the final feature vector. Positional embeddings are excluded, as genes form an unordered set, and their inclusion would hinder learning by increasing input state complexity.

Policy Network

Given a model that predicts the resistance profile of bacteria based on the outcomes of a subset of gene PCR tests, we need the ability to determine which PCR tests to perform. More specifically, we want to know which subset of tests that gives the best predictive performance for the AR prediction model. In more formal terms, we want to find a policy π that we can follow to select this subset.

While the subset can be selected in a single step, we formulate the problem of finding the policy π as a sequential problem. Given a set of genes already selected, an optimal

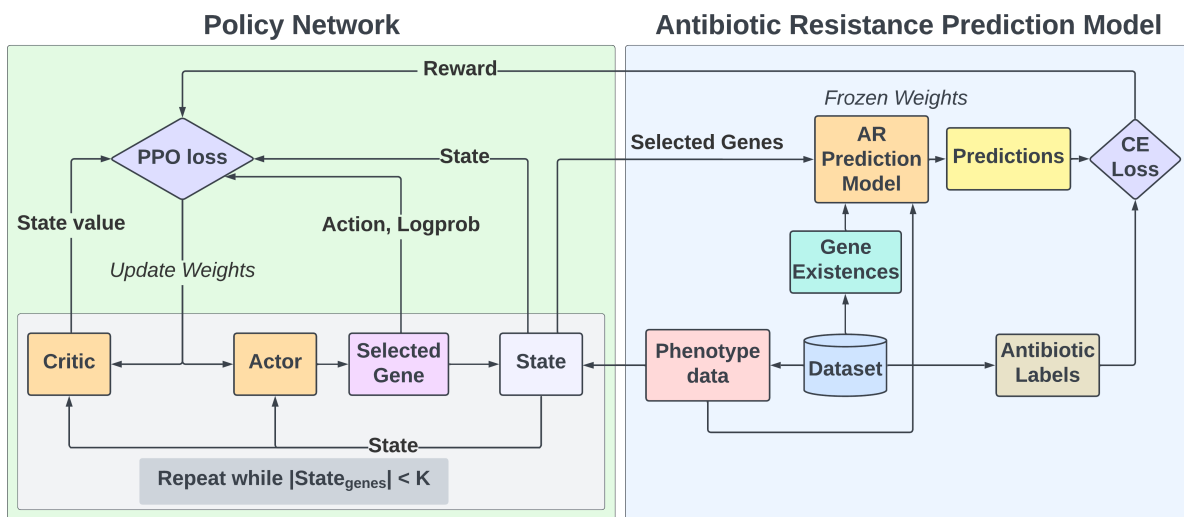


Figure 3: An overview of the training framework for the policy network.

policy should then select the subsequent gene to test such that the final subset of genes is optimal once the episode has ended. This allows us to frame it within the structure of an episodic Markov Decision Process (MDP) and utilize reinforcement learning methods.

In this formulation, we seek a policy $\pi(a|s)$ in which the policy selects an action based on the current state. The state s contains information on the genes that have already been selected and the action a is defined as selecting a specific gene to test. Once the K genes have been selected, the episode is terminated and the genes and their corresponding test results are given to the AR model. The predictions from the AR model and the ground truth labels are then used to compute the reward (r), defined as the negative mean cross-entropy loss over all antibiotics. Since the reward is always zero except at the end of an episode, the return is equal to the reward at the final step with a discount factor of 1.

The objective of the policy network is to maximize the expected return. We optimize the policy through proximal policy optimization (Schulman et al. 2017) and gather data for the policy by allowing it to act in the environment defined by f_ϕ . An overview of the training framework for the policy network can be found in Figure 3.

Architecture: The policy network is an actor-critic architecture using actor and critic agents with non-shared weights. The actor agent learns the policy $\pi_\phi(a|s)$ and is parameterized by ϕ . The critic agent learns the value function $V_w(s)$ and is parameterized by w . Both the actor and the critic are three-layer MLP’s using tanh activations. The state s is constructed by a concatenation of a vector s_g representing the currently selected genes and a vector s_m representing the metadata. The vector s_g is created from the sum of currently selected tokens, embedded with a separate learned embedding layer and an MLP. The use of summation ensures that different permutations of the same genes are encoded into an identical state. It reduces the number of possible states down from $\sum_{j=1}^K \frac{N!}{(N-j)!}$ to $\sum_{j=1}^K \frac{N!}{j!(N-j)!}$ compared to if

each different permutation of selected genes was defined as a unique state. The metadata vector S_m is created in a similar way, as a sum of the embedded tokenized metadata.

Choice of reward: While several metrics such as accuracy or Pareto front optimization (Yu et al. 2023), can be used as rewards, these metric-based rewards tend to be non-smooth. This non-smoothness results in abrupt changes in rewards from minor adjustments in state or input, which can destabilize the learning process. To address this, we utilize the unreduced loss from our prediction model as the reward for selecting a specific set of genes. The unreduced loss serves as an indicator of predictive performance, and provides a more stable and informative reward signal that still aligns with the goal metric.

Weight initialization: While the action space for π_θ is large, many genes are not commonly encountered in pathogens, making them inefficient to test, as their results are likely negative in most samples. Our experiments show that finding a good policy in this space can be challenging. To address this, we initialize the last linear layer in the actor module such that the weights are drawn from a normal distribution with a small variance $\sim \mathcal{N}(0, 10^{-4})$, while the biases are set to the logarithm of the normalized gene prevalence in the training dataset. This initialization scheme ensures that the initial action distribution aligns with the prevalence of genes, improving convergence during training without restricting the action space.

Training Procedure

The two networks are trained sequentially. First, the AR prediction model is trained using a randomly sampled subset S_{random} of gene tests. The size of S_{random} is set to be larger than the maximum cardinality K but much smaller than the full set of gene tests $|G|$. This ensures that the network is exposed to a wide variety of gene combinations while still focusing on predicting AR from a small subset of genes.

Once the AR prediction model has been trained, its

weights are frozen, and the model is used as a static environment. The policy network then interacts with this environment, using its current policy to gather experience. After sufficient experience has been collected, the policy network is updated through proximal policy optimization. This iterative process of gathering experience and updating the policy continues until the policy network has converged.

Results

In this section, we evaluate the effectiveness of our proposed framework in optimizing PCR test selection for AR prediction. We assess the performance of the reinforcement learning model and the transformer-based AR prediction model across several datasets. To demonstrate the generalizability and robustness of our method it was evaluated on five clinically relevant pathogens: *Klebsiella pneumoniae*, *Escherichia coli*, *Salmonella*, *Staphylococcus aureus*, and *Acinetobacter*.

Experimental Setup

The bulk of our experiments are conducted using a subset of $K = 5$ genes as the most commonly used PCR tests in clinical practice target up to 5 genes. However, there are various types of PCR tests and other targeted methods available for assessing gene prevalence in pathogens. Therefore, we also perform experiments to demonstrate the impact of different values of K on performance. For hyperparameter details we refer to our GitHub repository.

We present results for each species as a weighted sum of evaluation metrics across all antibiotics to account for imbalances in resistance rates and missing labels. The accuracy for each antibiotic is weighted by its number of samples relative to the total number of samples. The very major error and major error are weighted by the number of resistant or susceptible samples relative to the total number of resistant or susceptible samples, respectively.

Datasets

The datasets used in this study are from the publicly available National Center for Biotechnology Information (NCBI) pathogen detection system (Sayers et al. 2022). The splits used for cross-validation is shared on our GitHub repository.

The predicted antibiotics for each species were selected based on specific criteria to ensure reliable analysis. Each antibiotic had to be present in at least one-third of all samples, and the ratio between susceptibility and resistance could not exceed 90% in either direction. Based on these criteria, the resulting number of predicted antibiotics was 15 for *K.pneumoniae*, 13 for *E.coli*, 10 for *Salmonella*, 8 for *S.aureus*, and 14 for *Acinetobacter*. The data has been filtered to exclude samples where resistance labels for more than 50% of the chosen antibiotics are missing. The resistance rates for different antibiotics varied significantly, from 10.5% resistance for piperacillin-tazobactam to 46% for ampicillin in the *E.coli* dataset. The gene test name data has been pre-processed to remove unnecessary annotations that describe specific variants, as many resistance gene classes include sequence variations with the same

phenotype, adding no relevant information. For example, “blaCTX-M-124” is shortened to “blaCTX-M”. Not all resistance gene classes have variant-level details, so the pre-processing improves comparability across different types of resistance genes. However, we kept information on amino acid changes in mutated chromosomal genes since they are known to correlate with the induced phenotype.

The metadata used for each sample is the collection date, the isolation source and the geographic location. The collection date contains information regarding both day, month, and year of collection but only the year was used in this study as the focus was to track changes in resistance over longer periods of time. The isolation source, which specifies the origin of the analyzed sample, was pre-processed into broader categories to ensure consistency. For example, “blood_whole” and “blood_culture” were grouped under the general category “blood”. The geographic location metadata was standardized to reflect the country where each sample was collected.

The resulting size of each dataset after pre-processing is 6111 for *E.coli*, 841 for *K.pneumoniae*, 5961 for *Salmonella*, 1050 for *Acinetobacter*, and 607 for *S.aureus*.

Baselines

Since no existing methods directly address the problem of optimizing PCR test selection for antibiotic resistance prediction, we developed several competitive baselines for comparison. The baselines identify gene test subsets and are evaluated against our proposed frameworks, GenoARM and GenoAR (GenoARM without metadata).

RandEvolve is a genetic algorithm designed to optimize gene subsets using a trained AR prediction model. The algorithm begins by randomly selecting initial candidate gene subsets based on gene frequencies in the training set. The performance of these candidates is then evaluated using the accuracy of the AR prediction model. The best-performing candidate undergoes mutation, replacing specific genes, and generating new candidates. These mutated candidates are evaluated and compared with the previously best candidate. The best gene tests are then selected and the process is repeated over multiple generations. When metadata is included, RandEvolve finds the best-performing genes for each unique set of metadata values by evaluating candidates on the subset of training data that matches the metadata.

OptStat constructs a gene test subset and a prediction model by iteratively expanding an initially empty subset V . For each gene test $g_k \in G$ it forms a new subset $S_k = V \cup \{g_k\}$ and evaluates the prediction accuracy on training data based on the gene test results $e_i(S_k)$. The predictions are determined by selecting the most frequent outcome, resistance or susceptibility, across matching samples in the training data. The subset S_k that yields the highest weighted average accuracy across all antibiotics then replaces V . This process repeats until $|V| = K$. When metadata is included, the algorithm generates separate subsets and models for each unique metadata combination, evaluating them on the corresponding training data.

Gene Test Optimization, 5 Gene Tests

We evaluated the performance of GenoARM and all baselines using five-fold cross-validation on the datasets for the five pathogens, optimizing for a subset of $K = 5$ gene tests. The results are summarized in Table 1 and Table 2. We also present qualitative example gene subsets from the first fold for *E.coli* in Table 3.

We observe that all evaluated methods have fairly similar results when no metadata is used, with OptStat slightly outperforming the other methods. We also note that four out of five genes were identical for all three evaluated methods, with *aph(3b)-I* selected in OptStat instead of *sul2*, improving its predictive performance against aminoglycoside antibiotics. Two of the five selected genes, *blaCTX-M* and *blaTEM*, produce resistance against beta-lactam antibiotics. This is a reasonable selection as 6 of the 13 predicted antibiotics are beta-lactam based, including the most difficult, ampicillin, with close to 50% resistant samples.

The results show that metadata improves performance on all species for models using function approximators to do AR prediction, such as RandEvolve-M and GenoARM, while OptStat-M obtained much smaller increases and in some cases even a decrease in performance. This discrepancy can be attributed to the fact that OptStat-M relies on statistical information derived from each unique metadata value. With fewer available samples for each predictor and gene test policy, the resulting estimates become less reliable, leading to inconsistent performance.

RandEvolve-M also faces challenges due to its gene test policy search being restricted to data associated with the corresponding metadata. That is, to select the gene tests for a certain metadata m_l it only evaluates performance on data with $o_i = m_l$. However, it still performs well, likely due to two key factors. First, the neural network used to perform AR prediction allows RandEvolve-M to generalize predictions effectively, even in scenarios with limited data. Second, despite the large number of unique metadata combinations, the data is heavily clustered around a few common combinations. For instance, in the *E.coli* dataset there are 2550 unique combinations but the 25 most common combinations account for 90% of the data. The remaining 2525 combinations either have very few or no samples at all. As a result, relatively high performance can be achieved from just these common combinations, which helps to explain RandEvolve-M’s robustness despite the metadata constraints.

GenoARM generally achieves the best performance across species when metadata is utilized, though the difference compared to RandEvolve-M is relatively small. The difference could perhaps be attributed to GenoARM’s ability to handle data with less frequent metadata more effectively. Unlike methods that rely on predefined gene test policies, GenoARM’s policy network can evaluate unseen states rather than defaulting to a fixed set of gene tests, since the policy relies on a function approximation.

In Table 3 We can observe that the gene test subset found for metadata {USA, 2020, Urine} differs from the general subset found by methods not using the metadata. Specifically, the genes *aac(3)* and *sul2* has been replaced by *tet(A)* and *tet(B)*, both of which confer tetracycline resistance. This

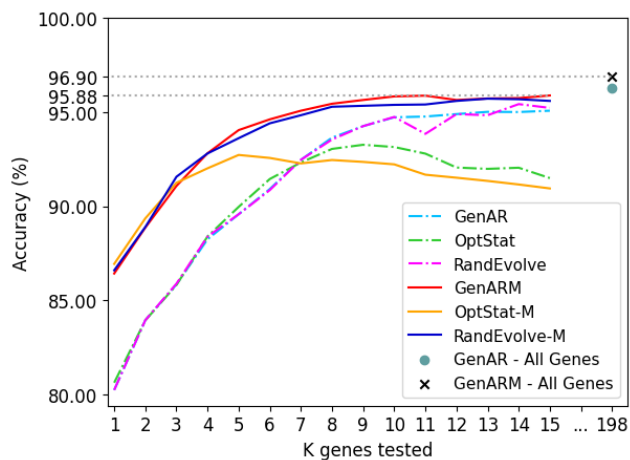


Figure 4: Accuracy on *E.coli* of all methods for increasing number of genes. “GenAR - All genes” and “GenARM - All genes” utilizes all genes in the dataset as inputs.

change might be explained by the significant increase in tetracycline resistance for urine infections in the USA in 2020, where only ampicillin resistance was higher.

Ablation Study on Size of Gene Test Subset

In Figure 4, we examine how performance is influenced by the size of the gene test subset K . Each model is trained and evaluated separately on the first cross-validation fold for *E.coli* using different values of K . Separate AR predictions models have been trained using all AR genes as input. We note that GenAR and RandEvolve perform similarly, with and without metadata. Metadata has a significant impact on performance when K is small, but this effect diminishes as K increases. With a larger gene test budget, the most informative genes can be included, reducing the relative importance of metadata. OptStat’s performance peaks at smaller gene subsets and declines for larger K due to missing data for most gene combinations, leading to default predictions.

Finally, we observe that models with an optimized set of 15 genes nearly match the performance of models trained on all 198 genes. This suggests that the number of truly informative genes is relatively small, and that an accurate resistance profile can be predicted from a small set of carefully selected genes.

Ablation Study on the Effect of Metadata

We analyze the impact of each unique type of metadata on performance by comparing models trained with either a single type of metadata, all types of metadata, or no metadata at all on the first cross-validation fold from *E.coli* data. The results, presented in Table 4, show that each type of metadata enhanced performance compared to the baseline with no metadata, with the most significant improvement coming from the “Source” metadata. Including all types of metadata further enhanced performance, leading to higher accuracy and lower major and very major errors, suggesting that each

Method	<i>E.coli</i>			<i>K.pneumoniae</i>			<i>Acinetobacter</i>		
	Acc ↑	ME ↓	VME ↓	Acc ↑	ME ↓	VME ↓	Acc ↑	ME ↓	VME ↓
RandEvolve	89.84	0.0395	0.2803	84.85	0.2083	0.1203	89.23	0.1379	0.0893
OptStat	89.94	0.0357	0.2873	85.27	0.2002	0.1184	89.80	0.1316	0.0845
GenoAR	89.91	0.0389	0.2797	85.16	0.2316	0.1181	88.84	0.1364	0.0962
RandEvolve-M	93.44	0.0303	0.1670	85.70	0.2217	0.0996	90.15	0.1423	0.0720
OptStat-M	92.48	0.0291	0.2078	82.59	0.2560	0.1292	87.25	0.1950	0.0864
GenoARM	93.77	0.0260	0.1641	86.80	0.2076	0.0906	90.69	0.1368	0.0665

Table 1: Antibiotic resistance prediction performance, averaged across all folds, for all methods on *E.coli*, *K.pneumoniae*, and *Acinetobacter* using a subset of 5 gene tests. GenoAR is GenoARM without metadata.

Method	<i>Salmonella</i>			<i>S.aureus</i>			Average		
	Acc ↑	ME ↓	VME ↓	Acc ↑	ME ↓	VME ↓	Acc ↑	ME ↓	VME ↓
RandEvolve	92.67	0.0448	0.1335	91.54	0.0340	0.2205	89.62	0.0923	0.1688
OptStat	92.63	0.0461	0.1313	91.88	0.0327	0.2117	89.90	0.0892	0.1667
GenoAR	92.48	0.0449	0.1387	91.78	0.0352	0.2076	89.44	0.0974	0.1681
RandEvolve-M	94.54	0.0341	0.0976	91.21	0.0474	0.1970	91.01	0.0952	0.1267
OptStat-M	93.23	0.0312	0.1444	92.19	0.0379	0.1856	89.55	0.1098	0.1507
GenoARM	94.50	0.0375	0.0917	91.07	0.0578	0.1431	91.36	0.0934	0.1112

Table 2: Antibiotic resistance prediction performance, averaged across all folds, for all methods on *Salmonella*, *S.aureus*, and the average over all five evaluated pathogens and all folds using a subset of 5 gene tests. GenoAR is GenoARM without metadata.

Method	Metadata	Genes
RandEvolve	-	{ aac(3), blaCTX-M, blaTEM, parC_S80I, sul2 }
OptStat	-	{ aac(3), blaCTX-M, blaTEM, parC_S80I, aph(3b)-I }
GenoAR	-	{ aac(3), blaCTX-M, blaTEM, parC_S80I, sul2 }
RandEvolve-M	{USA, 2020, Urine}	{ blaCTX-M, blaTEM, parC_S80I, tet(A), tet(B) }
OptStat-M	{USA, 2020, Urine}	{ blaCTX-M, blaTEM, parC_S80I, tet(A), tet(B) }
GenoARM	{USA, 2020, Urine}	{ blaCTX-M, blaTEM, parC_S80I, tet(A), tet(B) }

Table 3: Example gene test subsets found by all methods on *E.coli* data.

Method	<i>E.coli</i>		
	Acc ↑	ME ↓	VME ↓
GenoAR	89.57	0.0420	0.2756
GenoARM (Year)	91.94	0.0338	0.2091
GenoARM (Country)	91.25	0.0305	0.2438
GenoARM (Source)	92.33	0.0309	0.2024
GenoARM (All)	94.04	0.0255	0.1531

Table 4: Ablation study over the different choices of metadata used in GenoARM.

type of metadata provides potentially complementary information.

Limitations

This study has several limitations. Our method relies on known AR genes, necessitating retraining to incorporate new genes, which may limit its adaptability to emerging resistance. Additionally, the data used in the study is predominantly from the United States, potentially biasing our models towards this region’s AR landscape and limiting gener-

alizability. Including more diverse datasets would improve global applicability.

Conclusion

In this study, we developed a novel framework that combines reinforcement learning and transformer-based classification models to optimize PCR test selection and improve antibiotic resistance predictions. The optimized PCR tests, along with our high-performing AR prediction model, show significant potential in guiding the selection of appropriate antibiotic treatments. Our results demonstrate that the GenoARM framework performs well across various pathogens effectively utilizing metadata to enhance predictions. The genetic algorithm RandEvolve also proves to be a competitive alternative. Both methods benefit from our proposed AR prediction model, which efficiently utilizes existence embeddings to represent gene presence. Overall, our framework represents a significant advancement in optimizing diagnostic tools for AR in clinical settings. By enhancing the precision of AR predictions and enabling more targeted treatments, our method has the potential to mitigate the impact of AR on global health.

Acknowledgements

The computations were enabled by resources provided by Chalmers e-Commons at Chalmers and the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725.

References

- Alcock, B. P.; et al. 2023. CARD 2023: expanded curation, support for machine learning, and resistome prediction at the Comprehensive Antibiotic Resistance Database. *Nucleic Acids Research*, 51: D690–D699.
- Alirezanejad, M.; Enayatifar, R.; Motameni, H.; and Nematzadeh, H. 2020. Heuristic filter feature selection methods for medical datasets. *Genomics*, 112(2): 1173–1181.
- Chakraborty, C.; Bhattacharya, M.; Sharma, A. R.; Roy, S. S.; Islam, M. A.; Chakraborty, S.; Nandi, S. S.; and Dhama, K. 2022. Deep learning research should be encouraged for diagnosis and treatment of antibiotic resistance of microbial infections in treatment associated emergencies in hospitals. *International Journal of Surgery*, 105.
- Fahy, C.; and Yang, S. 2019. Dynamic Feature Selection for Clustering High Dimensional Data Streams. *IEEE Access*, 7: 127128–127140. Conference Name: IEEE Access.
- Her, H.-L.; and Wu, Y.-W. 2018. A pan-genome-based machine learning approach for predicting antimicrobial resistance activities of the Escherichia coli strains. *Bioinformatics*, 34(13): i89–i95.
- Hyun, J. C.; Kavvas, E. S.; Monk, J. M.; and Palsson, B. O. 2020. Machine learning with random subspace ensembles identifies antimicrobial resistance determinants from pan-genomes of three pathogens. *PLOS Computational Biology*, 16(3): e1007608. Publisher: Public Library of Science.
- Jin, C.; Jia, C.; Hu, W.; Xu, H.; Shen, Y.; and Yue, M. 2024. Predicting antimicrobial resistance in *E. coli* with discriminative position fused deep learning classifier. *Computational and Structural Biotechnology Journal*, 23: 559–565.
- Kavvas, E. S.; Catoiu, E.; Mih, N.; Yurkovich, J. T.; Seif, Y.; Dillon, N.; Heckmann, D.; Anand, A.; Yang, L.; Nizet, V.; Monk, J. M.; and Palsson, B. O. 2018. Machine learning and structural analysis of Mycobacterium tuberculosis pan-genome identifies genetic signatures of antibiotic resistance. *Nature Communications*, 9: 4306.
- Kim, J. I.; Maguire, F.; Tsang, K. K.; Gouliouris, T.; Peacock, S. J.; McAllister, T. A.; McArthur, A. G.; and Beiko, R. G. 2022. Machine Learning for Antimicrobial Resistance Prediction: Current Practice, Limitations, and Clinical Perspective. *Clinical Microbiology Reviews*, 35(3): e00179–21. Publisher: American Society for Microbiology.
- Kuang, X.; Wang, F.; Hernandez, K. M.; Zhang, Z.; and Grossman, R. L. 2022. Accurate and rapid prediction of tuberculosis drug resistance from genome sequence data using traditional machine learning algorithms and CNN. *Scientific Reports*, 12(1): 2427. Number: 1 Publisher: Nature Publishing Group.
- Laxminarayan, R.; Duse, A.; Wattal, C.; Zaidi, A. K.; Wertheim, H. F.; Sumpradit, N.; Vlieghe, E.; Hara, G. L.; Gould, I. M.; Goossens, H.; et al. 2013. Antibiotic resistance—the need for global solutions. *The Lancet infectious diseases*, 13(12): 1057–1098.
- Li, Y.; and Oliva, J. 2021. Active Feature Acquisition with Generative Surrogate Models. In *Proceedings of the 38th International Conference on Machine Learning*, 6450–6459. PMLR. ISSN: 2640-3498.
- Macesic, N.; Bear Don't Walk, O. J.; Pe'er, I.; Tatonetti, N. P.; Peleg, A. Y.; and Uhlemann, A.-C. 2020. Predicting Phenotypic Polymyxin Resistance in Klebsiella pneumoniae through Machine Learning Analysis of Genomic Data. *mSystems*, 5(3): 10.1128/msystems.00656–19. Publisher: American Society for Microbiology.
- Nam, H. A.; Fleming, S.; and Brunskill, E. 2021. Reinforcement Learning with State Observation Costs in Action-Contingent Noiselessly Observable Markov Decision Processes. In *Advances in Neural Information Processing Systems*, volume 34, 15650–15666. Curran Associates, Inc.
- O'Neill, J. 2014. Antimicrobial resistance: tackling a crisis for the health and wealth of nations. *Rev. Antimicrob. Resist.*
- O'Neill, J. 2016. Tackling drug-resistant infections globally: final report and recommendations.
- Pudjihartono, N.; Fadason, T.; Kempa-Liehr, A. W.; and O'Sullivan, J. M. 2022. A Review of Feature Selection Methods for Machine Learning-Based Disease Risk Prediction. *Frontiers in bioinformatics*, 2.
- Ren, Y.; Chakraborty, T.; Doijad, S.; Falgenhauer, L.; Falgenhauer, J.; Goesmann, A.; Hauschild, A.-C.; Schwengers, O.; and Heider, D. 2022. Prediction of antimicrobial resistance based on whole-genome sequencing and machine learning. *Bioinformatics*, 38(2): 325–334.
- Salimans, T.; Ho, J.; Chen, X.; Sidor, S.; and Sutskever, I. 2017. Evolution Strategies as a Scalable Alternative to Reinforcement Learning. arXiv:1703.03864.
- Sayers, E., W; et al. 2022. Database resources for biotechnology information. *Nucleic acids research*, 50(D1).
- Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347.
- Shim, H.; Hwang, S. J.; and Yang, E. 2018. Joint Active Feature Acquisition and Classification with Variable-Size Set Encoding. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.
- Tharmakulasingam, M.; Gardner, B.; La Ragione, R.; and Fernando, A. 2023. Rectified Classifier Chains for Prediction of Antibiotic Resistance From Multi-Labelled Data With Missing Labels. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 20(1): 625–636. Conference Name: IEEE/ACM Transactions on Computational Biology and Bioinformatics.
- Tharmakulasingam, M.; Gardner, B.; Ragione, R. L.; and Fernando, A. 2022. Explainable Deep Learning Approach for Multilabel Classification of Antimicrobial Resistance

With Missing Labels. *IEEE Access*, 10: 113073–113085.
Conference Name: IEEE Access.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Xue, B.; Zhang, M.; Browne, W. N.; and Yao, X. 2016. A Survey on Evolutionary Computation Approaches to Feature Selection. *IEEE Transactions on Evolutionary Computation*, 20(4): 606–626.

Yin, H.; Li, Y.; Pan, S. J.; Zhang, C.; and Tschitschek, S. 2020. Reinforcement Learning with Efficient Active Feature Acquisition. ArXiv:2011.00825 [cs].

Yoon, J.; Jordon, J.; and Schaar, M. v. d. 2018. INVASE: Instance-wise Variable Selection using Neural Networks. In *International Conference on Learning Representations*.

Yu, Z.; Li, Y.; Kim, J.; Huang, K.; Luo, Y.; and Wang, M. 2023. Deep Reinforcement Learning for Cost-Effective Medical Diagnosis. ArXiv:2302.10261 [cs].

Zannone, S.; Lobato, J. M. H.; Zhang, C.; and Palla, K. 2019. ODIN: Optimal Discovery of High-value INformation Using Model-based Deep Reinforcement Learning. In *Real-world Sequential Decision Making Workshop, ICML*.