SOFTWARE

# DeepMRG: a multi-label deep learning classifier for predicting bacterial metal resistance genes

**Muhit Islam Emon**[*], **Liqing Zhang**[*]

Department of Computer Science, Virginia Tech, Blacksburg, Virginia 24060, USA

[*]muhitemon@vt.edu (MIE); lqzhang@cs.vt.edu (LZ)

## Abstract

The widespread misuse of antibiotics has escalated antibiotic resistance into a critical global public health concern. Beyond antibiotics, metals function as antibacterial agents. Metal resistance genes (MRGs) enable bacteria to tolerate metal-based antibacterials and may also foster antibiotic resistance within bacterial communities through co-selection. Thus, predicting bacterial MRGs is vital for elucidating their involvement in antibiotic resistance and metal tolerance mechanisms. Currently, the "best hit" approaches are utilized to identify and annotate MRGs. This method is sensitive to cutoff values and produces a high false negative rate. To address such limitations, we introduce DeepMRG, a deep learning-based multi-label classifier, to predict bacterial MRGs. Because a bacterial MRG can confer resistance to multiple metals, DeepMRG is designed as a multi-label classifier capable of predicting multiple metal labels associated with an MRG. It leverages bit score-based similarity distribution of sequences with experimentally verified MRGs. To ensure unbiased model evaluation, we employed a clustering method to partition our dataset into six subsets, five for cross-validation and one for testing, with non-homologous sequences, mitigating the impact of sequence homology. DeepMRG consistently achieved high overall F1-scores and significantly reduced false negative rates across a wide range of datasets. It can be used to predict bacterial MRGs in metagenomic or isolate assemblies. The web server of DeepMRG can be accessed at `https://deepmrg.cs.vt.edu/deepmrg` and the source code is available at `https://github.com/muhit-emon/DeepMRG` under the MIT license.

## Introduction

Antibiotic resistance poses a significant threat to global human health and is increasingly becoming a silent pandemic due to the widespread and inappropriate use of antibiotics [1, 2]. In response, there has been an increased reliance on metal-based antibacterial agents. However, it is important to note that bacteria can develop resistance to metals through exposure [3]. Moreover, the use of these antibacterial metals can contribute to the emergence and persistence of antibiotic resistance in bacterial populations through co-selection [4, 5]. Therefore, it is crucial to systematically and comprehensively detect and annotate bacterial metal resistance genes (MRGs) to gain

insights into their role in developing antibiotic resistance and to understand the key mechanisms behind bacterial tolerance to metals.

Currently, the prediction of an MRG is conducted using the "best hit" method, which involves comparing the query gene sequence to existing reference databases using programs such as BLAST [6] and DIAMOND [7] and annotating the gene's function based on the reference sequence it shows the highest similarity to [8, 9]. However, the best hit method requires setting identity cutoff scores (and/or alignment lengths) and is sensitive to these cutoff values, making it challenging to decide on an appropriate threshold. Generally, a high identity cutoff is applied when using the best hit method to predict bacterial MRGs. For instance, the authors in [10] and [11] employed the best hit method with an identity greater than 80% to reference sequences to predict MRGs. While the best hit method with a high cutoff value generally exhibits a low false positive rate [12], it can result in a high false negative rate [8, 13].

To overcome the limitations of the best hit approach, we introduce DeepMRG, a multi-label classifier that utilizes deep learning to predict bacterial MRGs. We designed DeepMRG as a multi-label classifier since an MRG can confer resistance to multiple metals. DeepMRG aligns a query gene sequence with experimentally confirmed MRGs, extracting alignment bit scores. These alignment bit scores are then used to derive the similarity distribution of the query sequence with 66 types of experimentally verified MRGs. This bit score-based similarity distribution serves as the feature for the deep neural networks. To minimize the effect of sequence homology on model evaluation, we created the training, validation, and test datasets with a clustered split method. DeepMRG demonstrated good predictive performance for MRGs during both 5-fold cross-validation and on the test dataset. Furthermore, we assessed DeepMRG's ability to identify and classify sequences with low similarity to experimentally confirmed MRGs. DeepMRG was also validated using an independent set of heavy metal resistance genes and *in silico* spike-in experiment. DeepMRG excelled in precision, recall, and F1-score in all the conducted experiments, with notably lower false negative rates.

DeepMRG is implemented as an easy-to-use web server available via `https://deepmrg.cs.vt.edu/deepmrg` and as a command line tool freely available at `https://github.com/muhit-emon/DeepMRG`. DeepMRG is fully documented in S1 File.

# Design and implementation

## Data collection and processing

We collected antibacterial biocide and metal resistance genes from BacMet [5]. BacMet contains 753 gene sequences in a database named BacMet EXP DB where the genes have been experimentally verified to confer resistance to metals and/or antibacterial biocides. Additionally, it provides BacMet Predicted DB, a database containing 155,512 potential resistance genes compiled from public sequence repositories based on sequence homology to genes with experimentally verified resistance functions.

We focus on predicting bacterial MRGs in this paper. Therefore, we extracted MRGs from BacMet databases by searching for metal names in gene metadata, excluding genes that confer resistance only to antibacterial biocides. It left us 485 sequences in

BacMet EXP MRG DB and 93,367 sequences in BacMet Predicted MRG DB, as shown in Fig 1A. BacMet EXP MRG is used throughout the paper to denote a gene in BacMet EXP MRG DB, and BacMet Predicted MRG is used to indicate a gene sequence belonging to BacMet Predicted MRG DB.

We validated the annotations of BacMet Predicted MRGs by taking their best hit to BacMet EXP MRG DB. We employed DIAMOND [7], a tool similar to BLAST [6] but considerably faster, to align the BacMet Predicted MRGs against BacMet EXP MRG DB and identified their best hit. Based on the sequence identity, e-value, and coverage of the best hit alignment, the BacMet Predicted MRGs were divided into the following two databases, as shown in Fig 1B:

1. **BacMet Predicted MRG DB partition 1**: A BacMet Predicted MRG is included in the BacMet Predicted MRG DB partition 1 if its best hit to a BacMet EXP MRG has $\geq$ 50% sequence identity, e-value $\leq$ 1e-10, alignment coverage $\geq$ 80%, and it has the same metal resistance annotation to the BacMet EXP MRG. 78,012 genes passed these constraints and were added to the BacMet Predicted MRG DB partition 1. We set the cutoffs of identity and e-value following [14] where these thresholds were used to identify high and mid quality antibiotic resistance genes from public databases. The alignment coverage cutoff was selected following [5].

2. **BacMet Predicted MRG DB partition 2**: A BacMet Predicted MRG is added to the BacMet Predicted MRG DB partition 2 if its best hit to a BacMet EXP MRG has $<$ 50% sequence identity, e-value $\leq$ 1e-10, alignment coverage $\geq$ 80%, and possesses identical metal resistance annotation as the BacMet EXP MRG. After satisfying all these constraints, 5885 genes were placed in the BacMet Predicted MRG DB partition 2.

We used the sequences in BacMet Predicted MRG DB partition 1 to construct the training, validation, and test datasets for our DeepMRG model. We held out the sequences in BacMet Predicted MRG DB partition 2 and the test set sequences for evaluating and comparing the performance of DeepMRG with the BLAST best hit method.

## Construction of training, validation, and test datasets using clustered split

We created the training, validation, and test datasets from the BacMet Predicted MRG DB partition 1 following the non-homologous database split technique employed in [15]. The sequences in BacMet Predicted MRG DB partition 1 are categorized into 63 different types based on the metal labels they confer resistance to (types and the number of sequences within each type are presented in S2 File). We utilized MMseqs2 [16] to cluster the sequences in each type using 40% sequence identity and 50% coverage thresholds. Then, the clusters in each type were randomly split into six sets, where an entire cluster was placed in one of the six sets. Subsequently, we combined the corresponding sets from all 63 types and obtained six datasets (D1-D5 and TEST). Datasets D1 to D5 were used for 5-fold cross-validation, and the dataset TEST was for testing, as shown in Fig 2. Constructing training, validation, and test datasets utilizing this clustered split approach reduces the impact of sequence homology on evaluating the deep learning model.

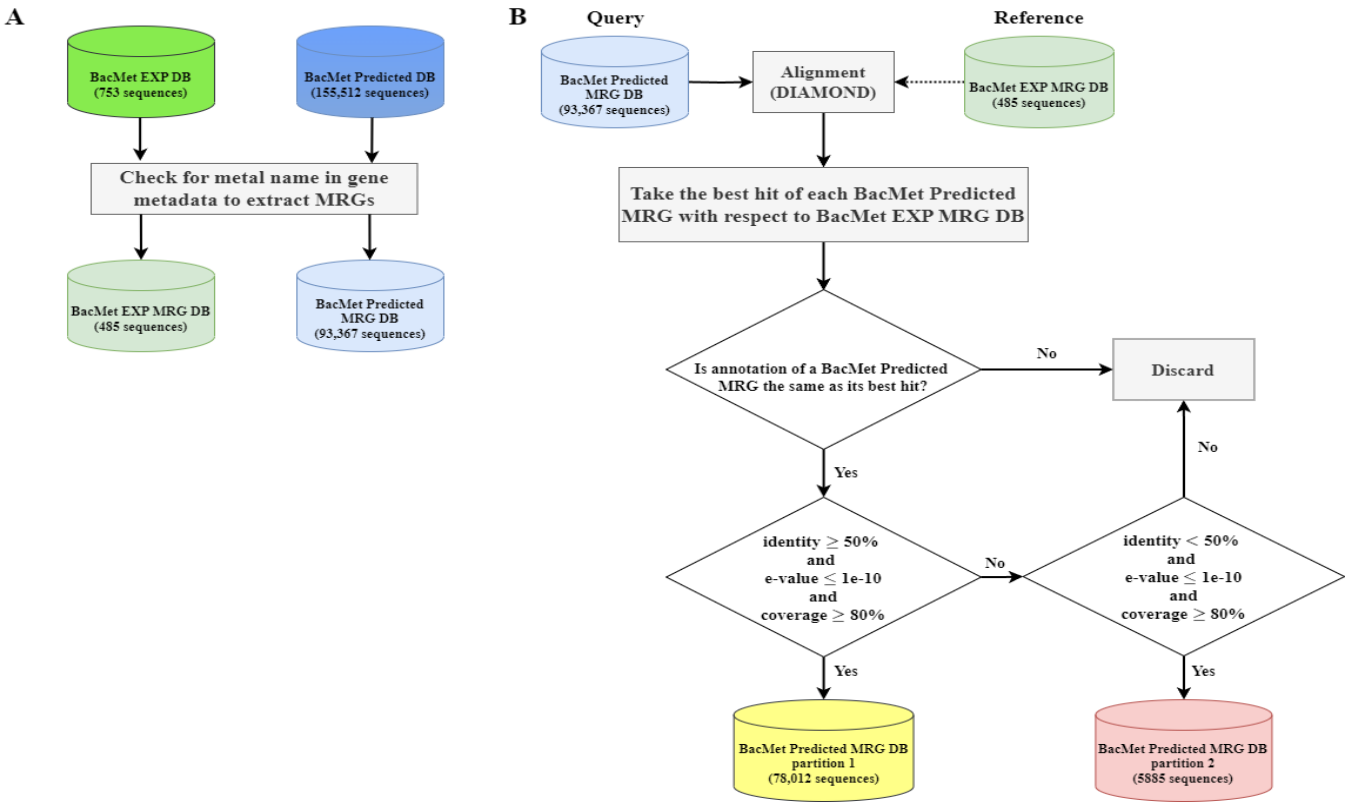


**Fig 1. Collection and processing of data.** A: Extraction of MRGs from BacMet [5] databases by searching metal names in gene metadata. B: Partition of BacMet Predicted MRG DB sequences into two databases. BacMet Predicted MRGs were aligned against the experimentally verified MRGs using DIAMOND [7]. The best hit was selected for each BacMet Predicted MRG and a set of filters were applied to create BacMet Predicted MRG DB partitions 1 and 2.

## Feature extraction

Deep learning models require a sequence to be represented as a vector of numerical values called features for prediction or classification tasks. In this paper, we adapted the concept of bit score-based similarity distribution used in [14] and [17]. The features are the alignment bit scores between full-length gene sequences and 66 types of experimentally verified MRGs available in BacMet EXP MRG DB. We used the bit score as the indicator of sequence similarity because it is independent of the database size, unlike the e-value [18]. The process for computing features for a full-length gene sequence is outlined as follows (refer to Fig 3):

1. The query gene sequence (QS) is aligned to the experimentally confirmed 485 MRGs in BacMet EXP MRG DB using DIAMOND under the 'very sensitive' parameter with the alignment constraints: a minimum sequence identity of 20%, an e-value lower than 1e-7, and a minimum alignment coverage of 60%. If QS does not have alignment with any of the 485 BacMet EXP MRGs, it is filtered out and not considered for further prediction. Otherwise, we go to the following steps to calculate the feature vector of QS. The alignment step works as a filter and

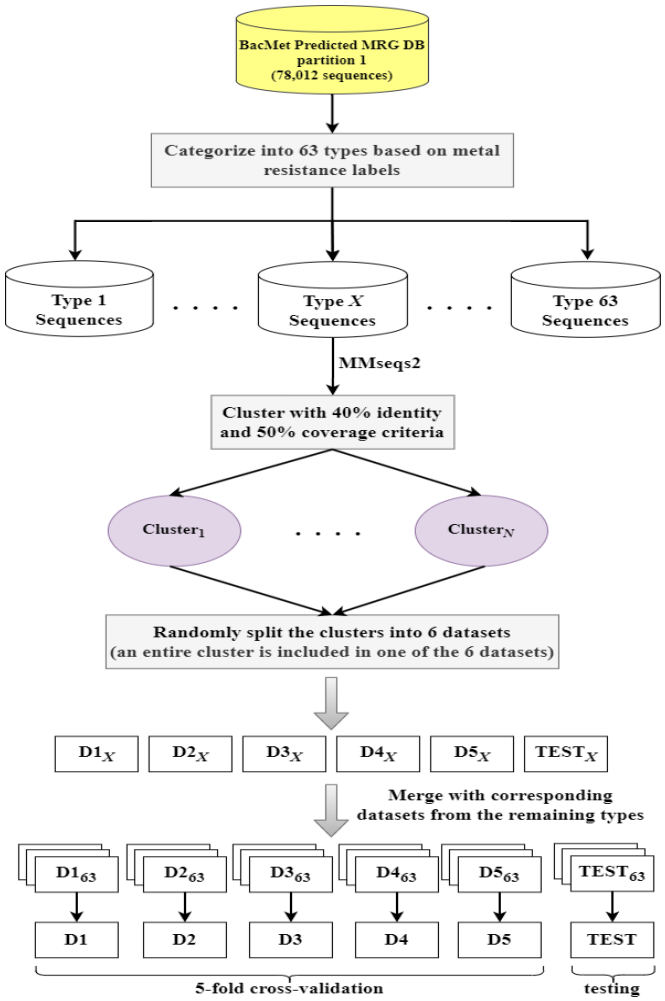


**Fig 2. Clustered split.** The sequences in BacMet Predicted MRG DB partition 1 are categorized into 63 types based on their metal resistance labels. The sequences in each type (type $X$ in this figure where $1 \leq X \leq 63$) were grouped into clusters using MMseqs2 [16] at 40% identity and 50% coverage. The clusters were randomly partitioned in six sets ($D1_X$-$D5_X$ and $TEST_X$) where an entire cluster was included in one of the six sets. Finally, the corresponding datasets from the remaining types were combined to make the datasets used for 5-fold cross-validation and testing (D1-D5 and TEST) of our deep learning model.

only passes MRG-like sequences to our deep learning model for prediction.

2. BacMet EXP MRGs are categorized into 66 types according to their metal resistance labels (types and the number of sequences within each type are presented in S3 File).

3. Here, we use the type $X$ ($1 \leq X \leq 66$) to demonstrate how we compute the feature vector for QS. Assume that there are $N$ MRGs in type $X$. The alignment bit scores between QS and these $N$ MRGs are obtained from step 1. Subsequently, we take the maximum among these $N$ bit scores, yielding bit $\text{score}_X$, which serves as the

representative bit score between QS and type $X$ of experimentally verified MRGs.

Similarly to the previous procedure, we get 66 bit scores which are then normalized to the $[0, 1]$ interval to represent the similarity distribution of QS with 66 types of experimentally confirmed bacterial MRGs. Thus, the feature vector of QS for DeepMRG is $V_{QS} \in \mathbb{R}^{66}$ where $V_{QS} = [\text{bit score}_1, \ldots, \text{bit score}_{33}, \ldots, \text{bit score}_{66}]$.

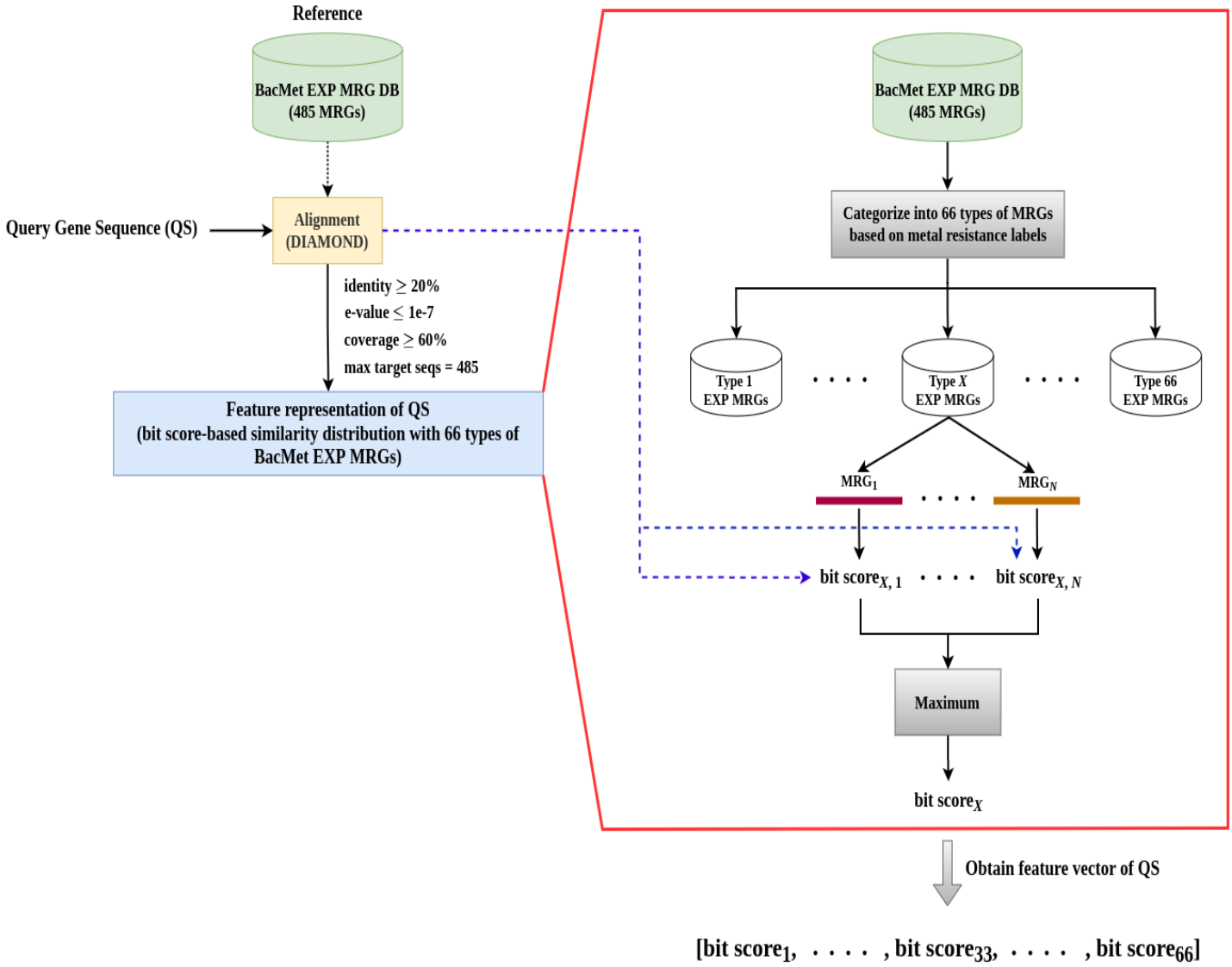


**Fig 3. The feature vector construction process of a query gene sequence (QS).** QS is aligned with the experimentally verified bacterial MRGs using DIAMOND [7]. BacMet EXP MRGs are categorized into 66 types according to their metal resistance labels. We use type $X$ ($1 \leq X \leq 66$) in this figure to show the feature extraction process for QS. We assume that $N$ experimentally verified MRGs are in type $X$. We get the bit scores between QS and these $N$ MRGs from the alignment step. Selecting the highest bit score among these $N$ scores yields bit score$_X$, representing the similarity between QS and type $X$ of BacMet EXP MRGs. Similarly, we get 66 such bit scores for each of the 66 types. The feature vector $V_{QS} \in \mathbb{R}^{66}$ of QS contains these bit scores as feature values.

## Deep learning model architecture and training

Subsequently, we trained DeepMRG, a multi-label deep neural network model, for classifying a gene sequence into one or multiple metal resistance categories by taking into account the similarity distribution of the sequence to 66 types of experimentally verified bacterial MRGs. Deep learning models can discern relevant features without human interference, which is one of its key advantages [19]. The DeepMRG model consists of an input layer, three hidden dense layers with 55, 45, and 35 nodes respectively, and an output layer. Each hidden dense layer utilizes the ELU activation function. To prevent overfitting, we applied dropout regularization with a dropout rate of 10% after each hidden layer. The Adam optimizer was used to update the weights, and binary cross entropy served as the loss function. The output layer consists of 23 nodes (one per metal). We employed the sigmoid activation function in the output layer, which computes the likelihood scores for each metal category. The model architecture is shown in Fig 4. The model was implemented using Python 3.6 with Scikit-learn 0.24.2 and Keras 2.2.0 with Tensorflow 2.2.0 backend.

We performed 5-fold cross-validation using sets D1 to D5 (see Fig 2), training five deep neural networks, each sequentially using one set for validation and the remaining four for training.

In 5-fold cross-validation, we trained five deep neural networks. For an input gene sequence, these five neural networks produce likelihood scores for each metal category. Then, we sum the scores from the five neural networks for each metal (the maximum achievable score is 5 per metal). The input sequence is then classified into the metal categories with the corresponding score of more than a threshold (default 3.5).

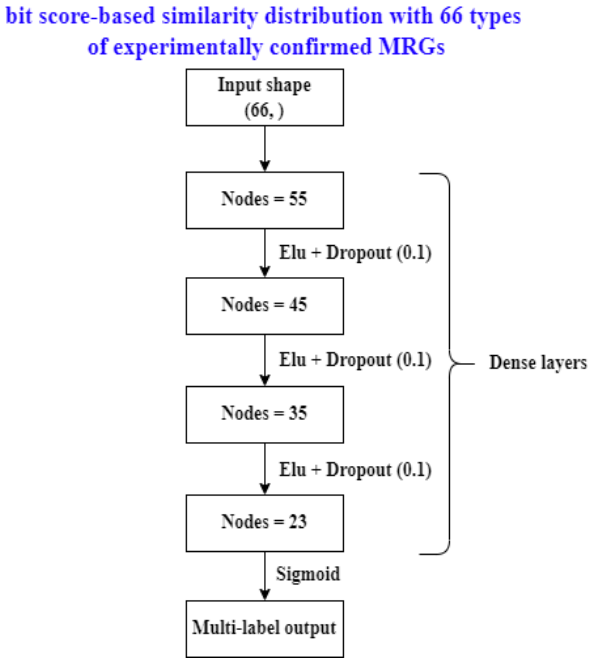


**Fig 4. DeepMRG model architecture.**

### BLAST best hit as the competing method

We used the BLAST best hit approaches as baseline comparison methods. We aligned the query sequences against the BacMet experimentally confirmed MRG database using BLASTp [6] and took the best hit to annotate them. As representatives of the best hit approach, we utilized various sequence identity cutoffs and ran the BLASTp program with the option '-max_target_seqs 1'.

## Results

### Performance evaluation metrics

We evaluated the performance of DeepMRG using both label-based and sample-based metrics. Label-based metrics assess the model's performance based on the prediction of class labels. We used precision, recall, and F1-score as label-based metrics. We also reported both macro-average and weighted-average results. The macro-average calculates the average performance of all classes, treating each class equally. On the other hand, the weighted average takes into account the number of sequences in each class, providing a performance measure that is influenced by the class distribution. The weighted-average is particularly useful when the class sizes are imbalanced.

Additionally, we utilized sample-based metrics to evaluate the performance of our model. Sample-based metrics assess the model's performance on an individual sample level within the dataset. As sample-based metrics, we reported samples-average precision, samples-average recall, and samples-average F1-score. A detailed description of the performance evaluation metrics is provided in S4 File.

### Performance of DeepMRG under 5-fold cross-validation

We evaluated the performance of DeepMRG through 5-fold cross-validation using clustered splits (see Fig 2). The overall F1-scores, including their mean and standard deviation, obtained from all five cross-validation experiments, are presented in Table 1. DeepMRG consistently achieved high F1-scores, with an average macro-average of 98.2%, an average weighted-average of 98.6%, and an average samples-average of 98.6%. These results indicate that DeepMRG can accurately predict MRGs and exhibits good generalization capabilities as a deep learning model. Furthermore, the standard deviations of 1.3% for macro-average F1-scores and 0.9% for weighted-average and samples-average F1-scores across five independent cross-validation experiments highlight DeepMRG's stability. Detailed classification reports for each metal, including precision, recall, and F1-score per fold, can be found in S5 File.

### Performance on the test dataset

We compared DeepMRG with BLAST best hit approaches on the test dataset comprising 11,447 sequences created using a clustered split method, as shown in Fig 2. BLAST best hit with an identity cutoff at 80% (BLAST-80%), a common choice in prior studies [10, 11] to predict bacterial MRGs, achieved very high precision ($\geq 0.99$) for all metal categories on our test dataset (see Fig 5). However, its recall was

**Table 1. The 5-fold cross-validation results of DeepMRG to predict MRGs.**

| Fold | Macro-average F1-score | Weighted-average F1-score | Samples-average F1-score |
|---|---|---|---|
| 1st | 0.99 | 0.98 | 0.98 |
| 2nd | 0.99 | 0.99 | 0.99 |
| 3rd | 0.98 | 0.98 | 0.98 |
| 4th | 0.99 | 1 | 1 |
| 5th | 0.96 | 0.98 | 0.98 |
| Mean $\pm$ SD | $0.982 \pm 0.013$ | $0.986 \pm 0.009$ | $0.986 \pm 0.009$ |

Performance assessment of DeepMRG through 5-fold cross-validation. Here, SD stands for standard deviation.

compromised for many metals, impacting the overall F1-scores, as presented in Table 2. BLAST best hit with an identity cutoff of 60% (BLAST-60%) maintained the same precision levels as BLAST-80% while significantly improving recall across all metal categories, resulting in a higher overall F1-scores on the test dataset. DeepMRG also exhibited very high precision ($\geq$ 0.98) for all metal labels, akin to BLAST-80% and BLAST-60%. It demonstrated similar or better recall rates than the best hit methods across all metals except copper (Cu). Overall, DeepMRG outperformed BLAST-80% in terms of F1-scores and achieved a similar weighted-average F1-score and samples-average F1-score as BLAST-60%, along with a slightly better macro-average F1-score than BLAST-60%.

## Performance on the sequences in BacMet Predicted MRG DB partition 2

We employed DeepMRG to classify the 5885 sequences from BacMet Predicted MRG DB Partition 2 and conducted a performance comparison with BLAST best hit approaches. The construction process of this database is detailed earlier in the "data collection and processing" section. None of these 5885 sequences were utilized for training DeepMRG. Notably, 75% of these sequences exhibit less than 60% sequence identity with our training set, as shown in S1 Fig. Because the sequences in BacMet Predicted MRG DB Partition 2 have less than 50% sequence identity with the BacMet EXP MRGs (see Fig 1), using a 50% or greater identity cutoff in the BLAST best hit method would result in these potential MRGs not being detected. Therefore, in this section, we employed 40% and 30% identity cutoff values for the BLAST best hit method (namely BLAST-40% and BLAST-30%) to compare with DeepMRG.

**Table 2. Performance on test dataset.**

| Methods | Macro-average F1-score | Weighted-average F1-score | Samples-average F1-score |
|---|---|---|---|
| DeepMRG | 0.99 | 0.98 | 0.95 |
| BLAST best hit (80%, 1e-7, 60%) | 0.87 | 0.84 | 0.71 |
| BLAST best hit (60%, 1e-7, 60%) | 0.98 | 0.98 | 0.95 |

Performance of DeepMRG and BLAST best hit methods under different parameters on test dataset. Here, BLAST best hit (x%, y, z%) refers to the best alignment hit with identity $\geq$ x%, e-value $\leq$ y, and coverage $\geq$ z%.

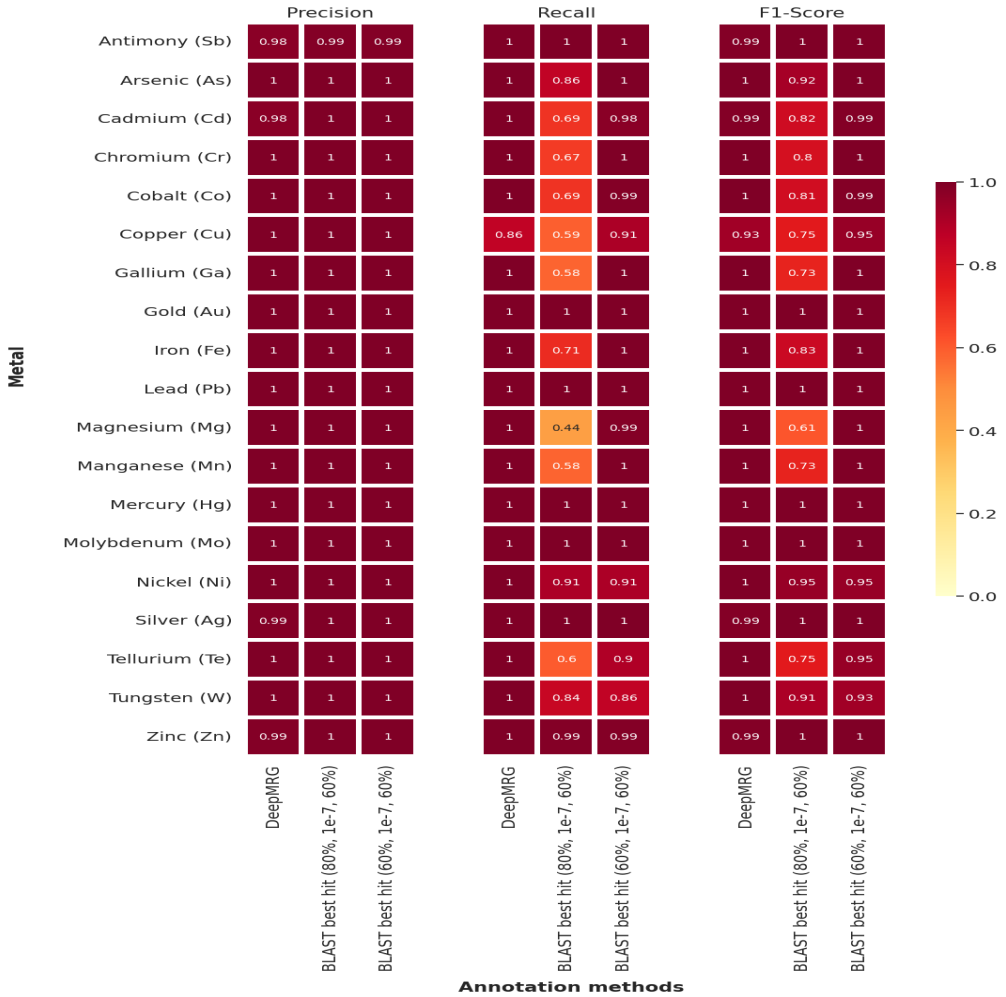


**Fig 5. Precision, recall, and F1-score comparison between DeepMRG and BLAST best hit approaches with different parameter settings for individual metal categories in the test dataset.** BLAST best hit (x%, y, z%) in this figure refers to the best alignment hit with identity $\geq$ x%, e-value $\leq$ y, and coverage $\geq$ z%. As a result of the clustered split, not all 23 metal labels are present in the test dataset.

BLAST-40% achieved similar or better precision compared to BLAST-30% on individual metal labels, while BLAST-30% exhibited higher recall rates (see S6 File). Table 3 shows the overall F1-scores of DeepMRG, BLAST-40%, and BLAST-30%. DeepMRG outperformed BLAST-40% and achieved overall F1-scores highly comparable to BLAST-30%. DeepMRG also demonstrated higher F1-scores than BLAST-40% and F1-scores on par with BLAST-30% for most metal categories, except silver (Ag).

This comparison underscores the best hit approach's sensitivity to sequence identity cutoff values. In contrast, DeepMRG eliminates the need for these threshold settings, delivering consistent and excellent performance in predicting bacterial MRGs across diverse datasets.

**Table 3. Performance on BacMet Predicted MRG DB partition 2.**

| Methods | Macro-average F1-score | Weighted-average F1-score | Samples-average F1-score |
|---|---|---|---|
| DeepMRG | 0.96 | 0.99 | 0.98 |
| BLAST best hit (40%, 1e-7, 60%) | 0.90 | 0.87 | 0.80 |
| BLAST best hit (30%, 1e-7, 60%) | 0.99 | 0.99 | 0.99 |

Performance of DeepMRG and BLAST best hit methods with different parameter settings on the sequences belonging to BacMet Predicted MRG DB partition 2. Here, BLAST best hit (x%, y, z%) refers to the best alignment hit with identity $\geq$ x%, e-value $\leq$ y, and coverage $\geq$ z%.

## Validation through an independent set of heavy metal resistane genes

Based on an independent study by Klonowska *et al*. [20], we obtained 53 heavy metal resistance (HMR) genes from heavy metal-tolerant *Cupriavidus* strain STM 6070 to evaluate DeepMRG's ability to predict novel bacterial MRGs. In this study, the authors conducted wet lab experiments to demonstrate that STM 6070 exhibits significantly higher tolerance to $Ni^{2+}$and $Zn^{2+}$concentrations compared to *Cupriavidus taiwanensis* strains. Moreover, computational and comparative genomics approaches were used to identify HMR genes in the STM 6070 genome, potentially involved in arsenic, cadmium, chromium, cobalt, copper, nickel, silver, and zinc resistance.

Employing a sequence identity threshold of 50% or more in the best hit approach is a common practice in bioinformatics [21]. Therefore, we set the identity cutoff parameter to 50% for the BLAST best hit method (BLAST-50%) and compared its results with those of DeepMRG for the independent dataset. The prediction results of DeepMRG and BLAST-50% in terms of F1-scores for the HMR genes in the independent dataset are shown in Table 4. BLAST best hits of these HMR genes against the experimentally validated MRGs, along with details on sequence identity and alignment coverage, are presented in S7 File. While BLAST-50% had better precision for nickel (Ni) and silver (Ag), DeepMRG consistently achieved higher or equal recall for all eight metal labels in the independent set (Table 1 of S8 File). DeepMRG also yielded higher F1-scores for all metals except nickel (Ni), surpassing BLAST-50% in overall F1-scores. It is noteworthy that STM 6070 displays higher tolerance to $Ni^{2+}$and $Zn^{2+}$, and DeepMRG demonstrated high recall ($\geq$ 0.75) for both metals.

We found that out of these 53 HMR genes, six were initially filtered out by our alignment step due to high e-value and low coverage. Adjusting the e-value and coverage cutoffs to 1e-3 and 40%, respectively, following [5, 22], we re-evaluated DeepMRG on the independent dataset. Although two genes were still filtered out during the alignment step, DeepMRG demonstrated improved performance as shown in Tables 2 and 3 of S8 File.

## Validation of DeepMRG through an *in silico* spike-in experiment

MRGs might constitute only a minor portion of the genes in real-world microbial datasets. Thus, assessing how well DeepMRG performs when non-target genes are prevalent is essential. To evaluate DeepMRG's ability to predict a small number of MRGs within a majority of non-target genes, we constructed a negative microbial

**Table 4. Performance on the independent dataset.**

| Methods | Macro-average F1-score | Weighted-average F1-score | Samples-average F1-score |
|---|---|---|---|
| DeepMRG | 0.72 | 0.75 | 0.71 |
| BLAST best hit (50%, 1e-7, 60%) | 0.58 | 0.56 | 0.47 |

Performance of DeepMRG and BLAST best hit method with 50% identity cutoff (BLAST-50%) on the independent set of *Cupriavidus* strain STM 6070 HMR genes. BLAST best hit (x%, y, z%) refers to the best alignment hit with identity $\geq$ x%, e-value $\leq$ y, and coverage $\geq$ z%.

dataset simulating a spike-in metagenomic experiment. In this section, our primary emphasis is on the binary classification performance of DeepMRG, specifically in distinguishing positive samples (MRGs) from negative samples. To create negative samples, we collected 27,036 bacterial housekeeping genes from UniProt [23], using 184 different Gene Ontology (GO) terms obtained from [24]. The associated GO terms and biological functions for these genes can be found in S9 File. Employing MMseqs2 at 30% identity and 50% coverage, we conducted clustering of the housekeeping genes together with the experimentally validated MRGs. Those housekeeping genes sharing a cluster with an experimentally confirmed MRG were excluded, resulting in a final set of 26,377 genes as negative samples. Next, we selected the 53 HMR genes from the independent dataset to serve as positive samples. The resulting spike-in dataset comprised a total of 26,430 genes, with the positive samples accounting for $(53/26,430)\% = 0.2\%$. DeepMRG achieved an 85% recall rate, identifying 45 out of 53 HMR genes, with a very low false positive rate ($< 1\%$). This highlights DeepMRG's efficacy in predicting MRGs within a large pool of negative samples, mirroring real-world scenarios where MRGs account for a small fraction of the total genes.

### Application of DeepMRG on metagenomic or isolate assembly data

DeepMRG can be applied to metagenomic or isolate assemblies to predict bacterial MRGs. We implemented a pipeline, as shown in Fig 6, for predicting bacterial MRGs from metagenomic or isolate assembled contigs. This pipeline takes assembled contigs as input and employs Prodigal [25] to predict genes from the contigs. These predicted genes are then aligned to the experimentally confirmed 485 MRGs in BacMet EXP MRG DB using DIAMOND with the alignment parameters: sequence identity $\geq$ 20%, e-value $\leq$ 1e-7, and alignment coverage $\geq$ 60% as discussed earlier in the "feature extraction" section. Then, using our feature computation approach, each predicted gene is represented by a vector $V \in \mathbb{R}^{66}$ referring to the bit score-based similarity distribution with 66 types of BacMet EXP MRGs. Finally, DeepMRG is employed to identify and annotate MRGs. The entire pipeline was built using Nextflow [26], with parallel computation for the gene prediction step by Prodigal.

## Availability and future directions

DeepMRG can be accessed through our web server at `https://deepmrg.cs.vt.edu/deepmrg`, providing a user-friendly interface for

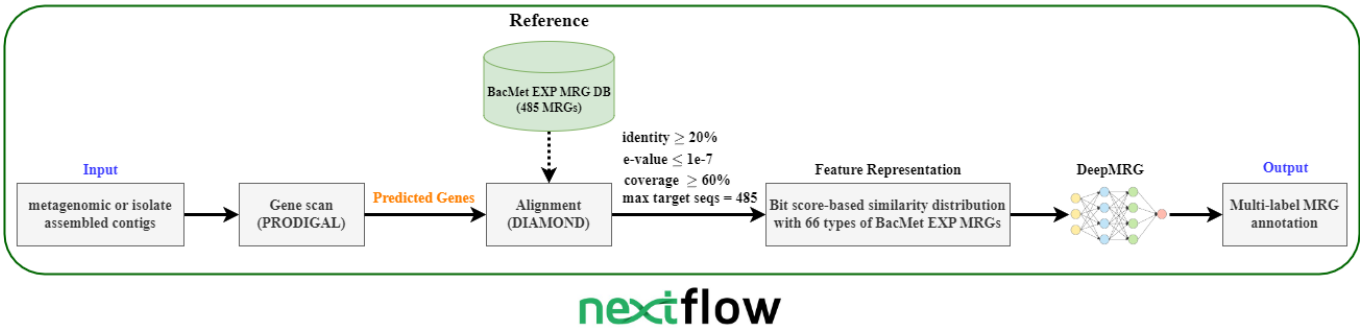


**Fig 6. Pipeline for predicting bacterial MRGs from assembled contigs using DeepMRG.**

bacterial MRG prediction tasks. Alternatively, it can be installed locally and run as a command-line tool. The source code of DeepMRG, along with detailed instructions for installation and execution, can be found on our GitHub repository at https://github.com/muhit-emon/DeepMRG. All the datasets used in this paper are available from Zenodo (https://doi.org/10.5281/zenodo.10070602).

DeepMRG can process either full-length gene sequences or pre-assembled contigs for MRG prediction. In the case of contigs, the first step involves predicting open reading frames (ORFs), and subsequently, the resulting protein sequences are fed into DeepMRG for MRG prediction. However, the current pipeline of DeepMRG has a limitation in that it cannot directly handle short reads without the need for assembly. To address this limitation, our future plans involve expanding DeepMRG's capabilities to enable direct prediction from short reads, eliminating the need for assembly. This development would significantly enhance DeepMRG's applicability in genomics or metagenomics research and enable more efficient analysis of short read data.

As DeepMRG's initial step involves aligning query gene sequences with experimentally validated MRGs, it might face some drawbacks typically associated with alignment-based methods. For example, DIAMOND may fail to identify MRGs very diverged from the experimentally validated MRGs, leading to false negatives. Comparatively, protein structures are more conserved than sequences and, therefore, can be leveraged to improve the performance of the prediction models. Consequently, we aim to incorporate protein 3D structures and protein language model (PLM) generated embeddings into DeepMRG. This integration of structural data and PLM is expected to significantly enhance the accuracy and reliability of the prediction model, providing deeper insights into the functional properties and mechanisms underlying metal resistance. By leveraging such information, DeepMRG will be better equipped to make precise predictions and contribute to a more comprehensive understanding of MRGs.

## Supporting information

**S1 File. DeepMRG documentation.** Complete documentation for using DeepMRG. (PDF)

**S2 File. Types of the sequences in BacMet Predicted MRG DB partition 1.** Types of the sequences belonging to BacMet Predicted MRG DB partition 1 and the number of sequences within each type.
(PDF)

**S3 File. Types of BacMet EXP MRG DB sequences.** Types of the sequences in BacMet EXP MRG DB and the number of sequences within each type.
(PDF)

**S4 File. Performance evaluation metrics.** Details of the performance evaluation metrics used in our paper.
(PDF)

**S5 File. Detailed results of DeepMRG under 5-fold cross-validation.** Individual metal category-wise classification reports of DeepMRG for each fold under the 5-fold cross-validation experiment.
(PDF)

**S6 File. Detailed results of DeepMRG and BLAST best hit approaches on BacMet Predicted MRG DB partition 2.** Individual metal category-wise classification reports of DeepMRG and BLAST best hit methods (under different parameters) on the sequences in BacMet Predicted MRG DB partition 2.
(PDF)

**S7 File. BLAST best hit sequence identity of *Cupriavidus* strain STM 6070 heavy metal resistance (HMR) genes against the experimentally confirmed MRGs.** BLAST best hits of all 53 STM 6070 HMR genes against the BacMet experimentally confirmed MRG database, including sequence identity and alignment coverage information.
(PDF)

**S8 File. Detailed results of DeepMRG and BLAST best hit with 50% sequence identity cutoff (BLAST-50%) on the independent set.** (Table 1) Individual metal category-wise classification reports of DeepMRG and BLAST-50% on the HMR genes in the independent dataset. (Table 2) The overall F1-scores of DeepMRG (with the initial DIAMOND alignment parameters: identity $\geq$ 20%, e-value $\leq$ 1e-3, and alignment coverage $\geq$ 40%) on the independent dataset. (Table 3) Individual metal category-wise classification results of DeepMRG (with the initial DIAMOND alignment parameters: identity $\geq$ 20%, e-value $\leq$ 1e-3, and coverage $\geq$ 40%) on the independent set.
(PDF)

**S9 File. Gene Ontology (GO) terms of bacterial housekeeping genes.** Biological functions and GO terms associated with the bacterial housekeeping genes used to

construct the negative microbial dataset.
(PDF)

**S1 Fig. Sequence identity histogram of BacMet Predicted MRG DB partition 2 against our training dataset.** Histogram of the best hit identity of the sequences in BacMet Predicted MRG DB partition 2 with our training dataset.
(TIF)


## Acknowledgments

We thank Professor Xiaofang Li from the Institute of Genetics and Developmental Biology of the Chinese Academy of Sciences for insightful suggestions.


## Author contributions

**Conceptualization:** Muhit Islam Emon, Liqing Zhang.

**Data curation:** Muhit Islam Emon.

**Formal analysis:** Muhit Islam Emon.

**Funding acquisition:** Liqing Zhang.

**Methodology:** Muhit Islam Emon.

**Project administration:** Liqing Zhang.

**Software & web server:** Muhit Islam Emon.

**Supervision:** Liqing Zhang.

**Validation:** Muhit Islam Emon.

**Writing – original draft:** Muhit Islam Emon.

**Writing – review & editing:** Muhit Islam Emon, Liqing Zhang.

## References


1. Mendelson M, Sharland M, Mpundu M. Antibiotic resistance: calling time on the ‘silent pandemic’. JAC-antimicrobial resistance. 2022 Apr;4(2):dlac016.

2. Wright GD. The antibiotic resistome. Expert opinion on drug discovery. 2010 Aug 1;5(8):779-88.

3. Turner RJ, Huang LN, Viti C, Mengoni A. Metal-Resistance in Bacteria: Why Care?. Genes. 2020 Dec 8;11(12):1470.

4. Baker-Austin C, Wright MS, Stepanauskas R, McArthur JV. Co-selection of antibiotic and metal resistance. Trends in microbiology. 2006 Apr 1;14(4):176-82.

5. Pal C, Bengtsson-Palme J, Rensing C, Kristiansson E, Larsson DJ. BacMet: antibacterial biocide and metal resistance genes database. Nucleic acids research. 2014 Jan 1;42(D1):D737-43.

6. Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic acids research. 1997 Sep 1;25(17):3389-402.

7. Buchfink B, Xie C, Huson DH. Fast and sensitive protein alignment using DIAMOND. Nature methods. 2015 Jan;12(1):59-60

8. Yang Y, Jiang X, Chai B, Ma L, Li B, Zhang A, Cole JR, Tiedje JM, Zhang T. ARGs-OAP: online analysis pipeline for antibiotic resistance genes detection from metagenomic data using an integrated structured ARG-database. Bioinformatics. 2016 Aug 1;32(15):2346-51.

9. Davis JJ, Boisvert S, Brettin T, Kenyon RW, Mao C, Olson R, Overbeek R, Santerre J, Shukla M, Wattam AR, Will R. Antimicrobial resistance prediction in PATRIC and RAST. Scientific reports. 2016 Jun 14;6(1):1-2.

10. Zhou L, Zhao Z, Shao L, Fang S, Li T, Gan L, Guo C. Predicting the abundance of metal resistance genes in subtropical estuaries using amplicon sequencing and machine learning. Ecotoxicology and Environmental Safety. 2022 Aug 1;241:113844.

11. Li LG, Xia Y, Zhang T. Co-occurrence of antibiotic and metal resistance genes revealed in complete genome collection. The ISME journal. 2017 Mar;11(3):651-62.

12. Forsberg KJ, Patel S, Gibson MK, Lauber CL, Knight R, Fierer N, Dantas G. Bacterial phylogeny structures soil resistomes across habitats. Nature. 2014 May 29;509(7502):612-6.

13. McArthur AG, Tsang KK. Antimicrobial resistance surveillance in the genomic age. Annals of the New York Academy of Sciences. 2017 Jan;1388(1):78-91.

14. Arango-Argoty G, Garner E, Pruden A, Heath LS, Vikesland P, Zhang L. DeepARG: a deep learning approach for predicting antibiotic resistance genes from metagenomic data. Microbiome. 2018 Dec;6:1-5.

15. Cantu VA, Salamon P, Seguritan V, Redfield J, Salamon D, Edwards RA, Segall AM. PhANNs, a fast and accurate tool and web server to classify phage structural proteins. PLoS computational biology. 2020 Nov 2;16(11):e1007845.

16. Steinegger M, Söding J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. Nature biotechnology. 2017 Nov;35(11):1026-8.

17. Ji B, Pi W, Liu W, Liu Y, Cui Y, Zhang X, Peng S. HyperVR: a hybrid deep ensemble learning approach for simultaneously predicting virulence factors and antibiotic resistance genes. NAR Genomics and Bioinformatics. 2023 Mar 1;5(1):lqad012.

18. Pearson WR. An introduction to sequence similarity (“homology”) searching. Current protocols in bioinformatics. 2013 Jun;42(1):3-1.

19. Min S, Lee B, Yoon S. Deep learning in bioinformatics. Briefings in bioinformatics. 2017 Sep 1;18(5):851-69.

20. Klonowska A, Moulin L, Ardley JK, Braun F, Gollagher MM, Zandberg JD, Marinova DV, Huntemann M, Reddy TB, Varghese NJ, Woyke T. Novel heavy metal resistance gene clusters are present in the genome of Cupriavidus neocaledonicus STM 6070, a new species of Mimosa pudica microsymbiont isolated from heavy-metal-rich mining site soil. BMC genomics. 2020 Dec;21(1):1-8.

21. Bengtsson-Palme J, Larsson DJ, Kristiansson E. Using metagenomics to investigate human and environmental resistomes. Journal of Antimicrobial Chemotherapy. 2017 Oct 1;72(10):2690-703.

22. Arango-Argoty GA, Dai D, Pruden A, Vikesland P, Heath LS, Zhang L. NanoARG: a web service for detecting and contextualizing antimicrobial resistance genes from nanopore-derived metagenomes. Microbiome. 2019 Dec;7:1-8.

23. UniProt Consortium. UniProt: a worldwide hub of protein knowledge. Nucleic acids research. 2019 Jan 8;47(D1):D506-15.

24. Moumi NA, Brown CL, Vikesland PJ, Pruden A, Zhang L. Protein-Protein Interaction Network Analysis Reveals Distinct Patterns of Antibiotic Resistance Genes. In2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) 2022 Dec 6 (pp. 73-76). IEEE.

25. Hyatt D, Chen GL, LoCascio PF, Land ML, Larimer FW, Hauser LJ. Prodigal: prokaryotic gene recognition and translation initiation site identification. BMC bioinformatics. 2010 Dec;11:1-1.

26. Di Tommaso P, Chatzou M, Floden EW, Barja PP, Palumbo E, Notredame C. Nextflow enables reproducible computational workflows. Nature biotechnology. 2017 Apr;35(4):316-9.