BenchmarkDR: A modular and expandable benchmarking pipeline for machine learning based antimicrobial resistance prediction

Niklas Stotzem[1], Fernando Guntoro[1], and Leonid Chindelevitch[1]

Imperial College London, United Kingdom
{n.stotzem20, f.guntoro20, l.chindelevitch}@imperial.ac.uk

Abstract. The access to Next Generation Sequencing data has raised interest in the application and development of machine learning methods for antimicrobial resistance (AMR) prediction. The diversity of algorithms as well as possible representations of the genome in terms of different features leaves researchers with the issue of comparing new methods to existing ones or choosing the appropriate method for their data. To give them a helpful tool, we have developed BenchmarkDR (https://github.com/WGS-TB/BenchmarkDR), a modular and easily extendable end-to-end pipeline to benchmark the prediction performance of the variety of available methods. Currently, BenchmarkDR supports the preprocessing of raw genomic sequencing input data into three different representations and the training and evaluation of 16 binary classification methods for categorical predictions and 8 regression methods for MIC predictions. Its modular design makes it easily extendable with other preprocessing approaches and prediction methods. We believe it represents a valuable addition to the AMR prediction toolkit and will provide valuable insights into the methods' relative strengths and weaknesses on a variety of bacterial datasets.

Keywords: Benchmarking · Antimicrobial Resistance Prediction · Machine Learning.

1 Introduction

The increasing number of drug resistant (DR) bacteria is quickly becoming one of the biggest threats in public health [19]. Leveraging machine learning (ML) methods to predict drug resistance from Next Generation Sequencing (NGS) data shows promising results [2, 16, 24] and could also help identify or confirm resistance mechanisms when using interpretable methods [8].

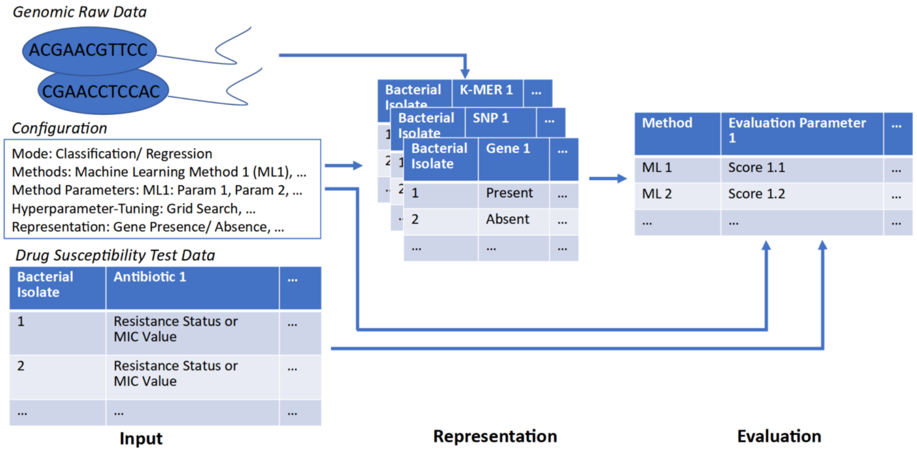
The application of ML methods to NGS data requires preprocessing of the raw data and allows different modelling approaches in terms of the extracted features. Commonly used representations are K-mers [5, 14, 8], single nucleotide polymorphisms (SNPs) [6, 24, 18] and genes [4, 18]. K-mers represent consecutive substrings of nucleotides of length K occurring in the genome, while SNPs are point mutations where individual base-pairs differ with respect to a reference genome. The combination of different representations with different ML methods leads to a variety of approaches to the prediction problem, creating the need for a systematic approach to compare them.

2 Results

To help shed light on the different ML approaches to DR prediction, we built BenchmarkDR (https://github.com/WGS-TB/BenchmarkDR), an end-to-end pipeline that enables the benchmarking of different prediction methods for classification (working with binary susceptible/resistant labels) and regression (working with minimum inhibitory concentration (MIC) values) by automatically creating the desired feature representations and training and evaluating a variety of ML methods. It is implemented in Python 3 and based on the workflow manager Snakemake v6.3 [12]. The pipeline is reproducible and can scale from a single local machine to a large computational cluster. Due to its rule-based design, where each step of the pipeline is defined as an individual rule with its own input and output, it provides modularity and can be easily extended with other preprocessing approaches or ML methods.

2.1 Workflow

BenchmarkDR's workflow (Figure 1) is split into two major parts. Based on an individual configuration, genome sequencing data is processed into different feature representations. These representations are then used to train different machine learning models, and evaluate their accuracy if true labels are available.


Genomic Raw Data
ACGAACGTTCC
CGAACCTCCAC
Configuration
Drug Susceptibility Test Data
Bacterial Isolate | K-MER 1 | ...
Bacterial Isolate | SNP 1 | ...
Input
Representation
Evaluation

Mode: Classification/ Regression
Methods: Machine Learning Method 1 (ML1), ...
Method Parameters: ML1: Param 1, Param 2, ...
Hyperparameter-Tuning: Grid Search, ...
Representation: Gene Presence/ Absence, ...

Bacterial Isolate	Antibiotic 1	...
1	Resistance Status or MIC Value	...
2	Resistance Status or MIC Value	...
...	...	...

Bacterial Isolate	Gene 1	...
1	Present	...
2	Absent	...
...	...	...

Method	Evaluation Parameter 1	...
ML 1	Score 1.1	...
ML 2	Score 1.2	...
...	...	...

Fig. 1. Overview of BenchmarkDR's workflow

Input Data The input data required from the user can be split into three categories: the genomic data (genotype), the drug resistance data (phenotype), and the configuration (setting). The genotype must comprise paired-end short reads of bacterial isolates in FASTQ format, which is a standard format for NGS. Furthermore, an assembled reference genome in FASTA format is required. Additionally, the drug resistance labels for the individual isolates, which are either binary (0 = susceptible and 1 = resistant) or quantitative (the MIC), have to be provided. The user also has to configure the pipeline according to their needs, e.g. paths to the genotypes and phenotypes, the representations and methods to be used, and the method parameters. Meaningful defaults are available for all of these except the genotype and the phenotype paths.

Representations BenchmarkDR allows to train the machine learning models on three different most common representations of the genomic data (genes, k-mers, SNPs). For each bacterial isolate, a binary vector is created, representing whether a certain feature, e.g. a gene, is present (1) or absent (0) in comparison to the specified reference genome. The choice of tools used to create the representations was based on benchmark papers, popularity, and ease of installation. Once the representation for each isolate is determined, all the results are pooled together in a table. The tools selected for each preprocessing will be explained in the following.

Gene Presence Absence One representation relies on the presence or absence of identified genes in each isolate. To assemble the fragmented DNA from the input data into contiguous sequences, SPAdes v3.15.2 [3] is used. The choice was based on the benchmarking results of Heydari et al. [9]. The '–isolates' flag is used if the coverage-depth of the respective isolates is greater than 100. The coverage depth is automatically determined based on the formula NL/G [22]. Here, L is the read length and N the number of reads, both of which are obtained from the FASTQ files, while G is the genome length, which is obtained from the reference strain. Once the genomes are assembled, the genes are annotated by Prokka v1.13.4 [21].

SNPs To determine the SNPs, the approach is similar to Yoshimura et al. [23]. In a first step, the reads are aligned using BWA version 0.1.17 [13], meaning that their likely position within the genome based on the reference genome is determined. In further processing, Samtools 1.12 [7] is used to correct errors and to sort the alignments according to their position. Picard v2.25.6 [1] then removes duplicates. Eventually, a pileup file, combining the data of the reference genome and the sorted fragments, is created using Samtools, which is eventually used by VarScan v2.4.4 [10] to determine the SNPs.

K-mers Based on the benchmark by Manekar et al. [17], KMC v3.1.2rc1 [11] is the method of choice to count the K-mers in each isolate. In the consolidation step where the aggregated table is created, the count is converted into a binary

presence/absence representation. In contrast to the other representations, k-mers have the advantage that they do not rely on a reference genome and require only one rule within the pipeline to be determined.

Training and Evaluation of Machine Learning Methods

Machine Learning Methods The pipeline offers a variety of ML methods (Table 1) to be trained on the different representations to predict drug susceptibility. A variety of standard models from Scikit-Learn v0.24 [20] are available, as well as the inherently interpretable method INGOT-DR [24].

Table 1. Overview of ML methods currently available in BenchmarkDR; the $*$ marks the methods for which the L_1, L_2, or Elastic Net penalty can be further selected.

Binary Classification	Regression
Logistic Regression*	Linear Regression*
Support Vector Machine Classification	Support Vector Machine Regression
Decision Trees	Decision Tree Regressor
Random Forests	Random Forest Regressor
Extremely Randomized Trees	Gradient Boosted Trees Regressor
AdaBoost Decision Tree Classifier	AdaBoost Decision Tree Regressor
Gradient Boosted Decision Trees	
Stochastic Gradient Descent Classifier*	
K-Nearest Neighbours	
Gaussian/ Complement Naive Bayes	
INterpretable GrOup Testing for Drug Resistance (INGOT-DR) [24]	

A configuration file allows the user to choose the parameters for the methods. Furthermore, hyperparameter tuning can be conducted via grid search or randomized search and cross-validation.

Output The pipeline's output provides a range of performance metrics. In addition to measuring the training time needed by each method, it provides different indicators to measure the prediction performance. For the binary classification task, accuracy, balanced accuracy, F1-score, AUC, as well as sensitivity and specificity are evaluated. For the regression task using MIC data, the metrics of mean squared error, mean squared log error and coefficient of determination $\left(R^2\right)$ are provided.

3 Conclusion & Future Work

With BenchmarkDR, we have built an end-to-end pipeline allowing a user-friendly and systematic benchmarking of a variety of ML methods on different genomic representations. It eliminates tedious manual preprocessing and allows to easily compare methods on the user's own data. Nevertheless, already preprocessed data can also be integrated. Furthermore, its modular design facilitates the further addition of genomic representations and ML methods. These additions, favourably driven by the developers of new methods themselves, will increase the tool's community value in the future.

The provision of a comprehensive dataset, already preprocessed into different representations, will further improve the benchmarking aspect, and is a future direction we plan to explore. Lastly, the pure benchmarking purpose of choosing the best performing method can be expanded by adding explainability methods for the available ML methods, e.g. SHAP [15], to further contribute to advancing our knowledge about genetic drug resistance mechanisms in bacterial pathogens.

References

1. Picard toolkit. http://broadinstitute.github.io/picard/ (2018)
2. Anahtar, M.N., Yang, J.H., Kanjilal, S.: Applications of machine learning to the problem of antimicrobial resistance: an emerging model for translational research. Journal of Clinical Microbiology **59**(7) (jun 2021). https://doi.org/10.1128/jcm.01260-20
3. Bankevich, A., et al.: SPAdes: A new genome assembly algorithm and its applications to single-cell sequencing. Journal of Computational Biology **19**(5), 455–477 (may 2012). https://doi.org/10.1089/cmb.2012.0021
4. Benkwitz-Bedford, S., et al.: Machine learning prediction of resistance to subinhibitory antimicrobial concentrations from escherichia coli genomes (mar 2021). https://doi.org/10.1101/2021.03.26.437296
5. Břinda, K., et al.: Rapid inference of antibiotic resistance and susceptibility by genomic neighbour typing. Nature Microbiology **5**(3), 455–464 (feb 2020). https://doi.org/10.1038/s41564-019-0656-6
6. Chen, M.L., et al.: Beyond multidrug resistance: Leveraging rare variants with machine and statistical learning models in mycobacterium tuberculosis resistance prediction. EBioMedicine **43**, 356–369 (may 2019). https://doi.org/10.1016/j.ebiom.2019.04.016
7. Danecek, P., et al.: Twelve years of SAMtools and BCFtools. GigaScience **10**(2) (jan 2021). https://doi.org/10.1093/gigascience/giab008
8. Drouin, A., Letarte, G., Raymond, F., Marchand, M., Corbeil, J., Laviolette, F.: Interpretable genotype-to-phenotype classifiers with performance guarantees. Scientific Reports **9**(1) (mar 2019). https://doi.org/10.1038/s41598-019-40561-2
9. Heydari, M., Miclotte, G., Demeester, P., de Peer, Y.V., Fostier, J.: Evaluation of the impact of illumina error correction tools on de novo genome assembly. BMC Bioinformatics **18**(1) (aug 2017). https://doi.org/10.1186/s12859-017-1784-8
10. Koboldt, D.C., et al.: VarScan 2: Somatic mutation and copy number alteration discovery in cancer by exome sequencing. Genome Research **22**(3), 568–576 (feb 2012). https://doi.org/10.1101/gr.129684.111
11. Kokot, M., Długosz, M., Deorowicz, S.: KMC 3: counting and manipulating k-mer statistics. Bioinformatics **33**(17), 2759–2761 (may 2017). https://doi.org/10.1093/bioinformatics/btx304

12. Koster, J., Rahmann, S.: Snakemake - a scalable bioinformatics workflow engine. Bioinformatics **28**(19), 2520–2522 (aug 2012). https://doi.org/10.1093/bioinformatics/bts480
13. Li, H.: Aligning sequence reads, clone sequences and assembly contigs with bwa-mem. ArXiv **1303** (03 2013)
14. Liu, Z., et al.: Evaluation of machine learning models for predicting antimicrobial resistance of actinobacillus pleuropneumoniae from whole genome sequences. Frontiers in Microbiology **11** (feb 2020). https://doi.org/10.3389/fmicb.2020.00048
15. Lundberg, S., Lee, S.I.: A unified approach to interpreting model predictions
16. Mahé, P., Tournoud, M.: Predicting bacterial resistance from whole-genome sequences using k-mers and stability selection. BMC Bioinformatics **19**(1) (oct 2018). https://doi.org/10.1186/s12859-018-2403-z
17. Manekar, S.C., Sathe, S.R.: A benchmark study of k-mer counting methods for high-throughput sequencing. GigaScience (oct 2018). https://doi.org/10.1093/gigascience/giy125
18. Moradigaravand, D., Palm, M., Farewell, A., Mustonen, V., Warringer, J., Parts, L.: Prediction of antibiotic resistance in escherichia coli from large-scale pan-genome data. PLOS Computational Biology **14**(12), e1006258 (dec 2018). https://doi.org/10.1371/journal.pcbi.1006258
19. Murray, C.J., et al.: Global burden of bacterial antimicrobial resistance in 2019: a systematic analysis. The Lancet **399**(10325), 629–655 (Feb 2022). https://doi.org/10.1016/s0140-6736(21)02724-0, https://doi.org/10.1016/s0140-6736(21)02724-0
20. Pedregosa, F., et al.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
21. Seemann, T.: Prokka: rapid prokaryotic genome annotation. Bioinformatics **30**(14), 2068–2069 (mar 2014). https://doi.org/10.1093/bioinformatics/btu153
22. Sims, D., Sudbery, I., Ilott, N.E., Heger, A., Ponting, C.P.: Sequencing depth and coverage: key considerations in genomic analyses. Nature Reviews Genetics **15**(2), 121–132 (jan 2014). https://doi.org/10.1038/nrg3642
23. Yoshimura, D., et al.: Evaluation of SNP calling methods for closely related bacterial isolates and a novel high-accuracy pipeline: BactSNP. Microbial Genomics **5**(5) (may 2019). https://doi.org/10.1099/mgen.0.000261
24. Zabeti, H., Dexter, N., Safari, A.H., Sedaghat, N., Libbrecht, M., Chindelevitch, L.: INGOT-DR: an interpretable classifier for predicting drug resistance in m. tuberculosis (may 2020). https://doi.org/10.1101/2020.05.31.115741