An Interpretable Classification Method for Predicting Drug Resistance in *M. Tuberculosis*

Hooman Zabeti
School of Computing Science, Simon Fraser University, Burnaby, Canada
hooman_zabeti@sfu.ca

Nick Dexter
Department of Mathematics, Simon Fraser University, Burnaby, Canada

Amir Hosein Safari
School of Computing Science, Simon Fraser University, Burnaby, Canada

Nafiseh Sedaghat
School of Computing Science, Simon Fraser University, Burnaby, Canada

Maxwell Libbrecht
School of Computing Science, Simon Fraser University, Burnaby, Canada

Leonid Chindelevitch
School of Computing Science, Simon Fraser University, Burnaby, Canada

Abstract

Motivation: The prediction of drug resistance and the identification of its mechanisms in bacteria such as *Mycobacterium tuberculosis*, the etiological agent of tuberculosis, is a challenging problem. Modern methods based on testing against a catalogue of previously identified mutations often yield poor predictive performance. On the other hand, machine learning techniques have demonstrated high predictive accuracy, but many of them lack interpretability to aid in identifying specific mutations which lead to resistance. We propose a novel technique, inspired by the group testing problem and Boolean compressed sensing, which yields highly accurate predictions and interpretable results at the same time.

Results: We develop a modified version of the Boolean compressed sensing problem for identifying drug resistance, and implement its formulation as an integer linear program. This allows us to characterize the predictive accuracy of the technique and select an appropriate metric to optimize. A simple adaptation of the problem also allows us to quantify the sensitivity-specificity trade-off of our model under different regimes. We test the predictive accuracy of our approach on a variety of commonly used antibiotics in treating tuberculosis and find that it has accuracy comparable to that of standard machine learning models and points to several genes with previously identified association to drug resistance.

2012 ACM Subject Classification Applied computing; Applied computing → Bioinformatics; Applied computing → Molecular sequence analysis

Keywords and phrases Drug resistance, whole-genome sequencing, interpretable machine learning, integer linear programming, rule-based learning

Digital Object Identifier 10.4230/LIPIcs.WABI.2020.2

Supplementary Material `https://github.com/hoomanzabeti/TB_Resistance_RuleBasedClassifier`

Funding This project was funded by the Genome Canada grant BAC283. LC acknowledges additional funding from the CANSSI CRT and NSERC Discovery.

Acknowledgements The authors would like to thank Dr. Cedric Chauve, Dr. Ben Adcock and Matthew Nguyen for helpful discussions.

20th International Workshop on Algorithms in Bioinformatics (WABI 2020).
Editors: Carl Kingsford and Nadia Pisanti; Article No. 2; pp. 2:1–2:18

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Drug resistance is the phenomenon by which an infectious organism (also known as pathogen) develops resistance to one or more drugs that are commonly used in treatment [36]. In this paper we focus our attention on *Mycobacterium tuberculosis*, the etiological agent of tuberculosis, which is the largest infectious killer in the world today, responsible for over 10 million new cases and 2 million deaths every year [37].

The development of resistance to common drugs used in treatment is a serious public health threat, not only in low and middle-income countries, but also in high-income countries where it is particularly problematic in hospital settings [40]. It is estimated that, without the urgent development of novel antimicrobial drugs, the total mortality due to drug resistance will exceed 10 million people a year by 2050, a number exceeding the annual mortality due to cancer today [35].

Existing models for predicting drug resistance from whole-genome sequence (WGS) data broadly fall into two classes. The first, which we refer to as "catalogue methods," involves testing the WGS data of an isolate for the presence of point mutations (typically single-nucleotide polymorphisms, or SNPs) associated with known drug resistance. If one or more such mutations is identified, the isolate is declared to be resistant [46, 14, 4, 21, 15]. While these methods tend to be easy to understand and apply, they often suffer from poor predictive accuracy [43], especially in identifying novel drug resistance mechanisms or screening resistance to untested or rarely-used drugs.

The second class, which we will refer to as "machine learning methods", seeks to infer the drug resistance of an isolate by training complex models directly on WGS and drug susceptibility test (DST) data [48, 11, 2]. Such methods tend to result in highly accurate predictions at the cost of flexibility and interpretability - specifically, they typically do not provide any insights into the drug resistance mechanisms involved and often do not impose explicit limits on the predictive model's complexity. Learning approaches based on deep neural networks are one such example.

In this paper we propose a novel method, based on the group testing problem and Boolean compressed sensing (CS), for the prediction of drug resistance. Compressed sensing is a mathematical technique for sparse signal recovery from under-determined systems of linear equations [16], and has been successfully applied in many application areas including digital signal processing [13, 12], MRI imaging [26], radar detection [19], and computational uncertainty quantification [29, 9]. Under a sparsity assumption on the unknown signal vector, it has been shown that CS techniques enable recovery from far fewer measurements than required by the Nyquist-Shannon sampling theorem [5]. Boolean CS is a slight modification of the CS problem, replacing the matrix vector product with a Boolean OR operator [28], and has been successfully applied to areas such as group testing for infection [3, 1].

Our approach combines some of the flexibility and interpretability of catalogue methods with the accuracy of machine learning methods – specifically, this method is capable of recovering interpretable rules for predicting drug resistance that both result in a high classification accuracy as well as provide insights into the mechanisms of drug resistance. We show that our methods perform comparably to standard machine learning methods on *Mycobacterium tuberculosis* in terms of predicting first-line drug resistance, while accurately recovering many of the known mechanisms of drug resistance, and identifying some potentially novel ones.

2 Methods

Our proposed method is based on the rule-based classification technique introduced in [28], wherein group testing and Boolean CS are combined to determine subsets of infected individuals from large populations. In that setting the linear system encodes the infection status of the population through testing, and the solution, obtained from a suitable decoder, is a $\{0,1\}$-valued vector representing the infection status of the individuals [6]. Since the infected group is assumed to be small, the solution vector is sparse and can be recovered using relatively few measurements with Boolean CS. The result of solving the Boolean CS problem can then be interpreted as a sparse set of rules for determining infections and used for classification on unseen data.

We present our methodology as follows. Section 2.1 introduces the group testing problem, and discusses how group testing can be combined with compressed sensing to deliver an interpretable predictive model. Section 2.2 introduces modifications to the standard setting to produce an accurate and flexible classifier, which can be tuned for specific evaluation metrics and tasks. Section 2.3 describes the tuning process for providing the desired trade-off between sensitivity and specificity in our model's predictions. Finally, Section 2.4 describes an approximation of the AUROC (area under receiver operating characteristic curve), a standard metric in machine learning, that is valid for evaluating the proposed approach.

2.1 Group testing and Boolean compressed sensing

We frame the problem of predicting drug resistance given sequence data as a group testing problem, originally introduced in [10]. This approach for detecting defective members of a set, was motivated by the need to screen large populations for syphilis while drafting citizens into military service for the United States during the World War II. The screening, performed by testing blood samples, was costly due to the low numbers of infected individuals. To make the screening more efficient, Dorfman suggested pooling blood samples into specific groups and testing the groups instead. A positive result for the group would imply the presence of at least one infected member. The problem then becomes to find the subset of individuals whose infected status would explain all of the positive results without invalidating any of the negative ones. By carefully selecting the groups, the total number of required tests m can be drastically reduced, i.e. if n is the population size, it is possible to achieve $m \ll n$.

Mathematically, a group testing problem with m tests can be described in terms of a Boolean matrix $A \in \{0,1\}^{m \times n}$, where A_{ij} indicates the membership status of subject j in the i-th test group, and a Boolean vector $y \in \{0,1\}^m$, where y_i represents the test result of the i-th group. If $w \in \{0,1\}^n$ is a Boolean vector, with w_j representing the infection status of the j-th individual, then the result of all m tests will satisfy

$$y = A \vee w, \tag{1}$$

where $\vee$ is the Boolean inclusive OR operator, so that (1) can also be written

$$y_i = \bigvee_{j=1}^{n} A_{i,j} \wedge w_j \;\forall\, 1 \leq i \leq m.$$

If the vector w satisfying equation (1) is assumed to be sparse (i.e. there are few infected individuals), the problem of finding w is an instance of the sparse Boolean vector recovery problem:

$$\min \|w\|_0 \;\; \textbf{subject to} \;\; y = A \vee w, \tag{2}$$

where $\|w\|_0$ is the number of non-zero entries in the vector w.

Due to the non-convexity of the ℓ_0-norm and the nonlinearity of the Boolean matrix product, the combinatorial optimization problem (2) is well-known to be NP-hard, see, e.g., [16, Section 2.3] or [33]. In [27] a relaxation of (2) via linear programming is proposed, with the ℓ_0-norm replaced by the ℓ_1-norm (much like in basis pursuit for standard compressed sensing), and with the nonlinear Boolean matrix product also replaced with two closely related linear constraints. We recapitulate their equivalent 0-1 linear programming formulation here:

$$\begin{aligned} \min \ & \sum_{j=1}^{n} w_j \\ \text{s.t.} \ & w \in \{0,1\}^n \\ & A_{\mathcal{P}} w \geq 1 \\ & A_{\mathcal{Z}} w = 0, \end{aligned} \tag{3}$$

where $\mathcal{P} = \{i : y_i = 1\}$ and $\mathcal{Z} = \{i : y_i = 0\}$ are the sets of groups that test positive and negative, respectively. However, this problem is also NP-hard, but can be made tractable for linear programming by relaxing the Boolean constraint on w in (3) to $0 \leq w_j \leq 1$ for all $j \in \{1, \ldots, n\}$.

[28] extended this idea for interpretable rule-based classification, meanwhile proving recovery guarantees for the relaxed problem. Because the Boolean CS problem is based on Boolean algebra, the conditions on the Boolean measurement matrices A that guarantee exact recovery of K-sparse vectors via linear programming are quite different from those of standard CS. Specifically, these guarantees require the definition of K-disjunct matrices, i.e., matrices A for which all unions of their columns of size K do not contain any other columns of the original matrix. Constructions exist for matrices with $\mathcal{O}(K^2 \log(n))$ rows which satisfy this property. We also note that by introducing an *approximate disjunctness* property, allowing for matrices for which a fraction $(1-\varepsilon)$ of all $\binom{n}{K}$ possible K-subsets of the columns satisfy the disjunctness condition, it was shown in [30] that there exist constructions of measurement matrices A which allow for recovery from $\mathcal{O}(K^{3/2}\sqrt{\log(n/\varepsilon)})$ rows.

In the standard setting for uniform recovery results for CS, the measurement matrices A are subgaussian random matrices, i.e., having entries $A_{i,j}$ drawn independently according to a subgaussian distribution. Examples include $m \times n$ matrices consisting of Rademacher or Gaussian random variables, for which uniform recovery of K-sparse vectors via ℓ_1-minimization has been shown under the condition m is $\mathcal{O}(K \log(n/K))$, see, e.g. [16, Chapter 9] for more details. While subgaussian matrices have been shown to possess the most desirable recovery guarantees, they are not always applicable for every measurement scheme, in particular the one considered here.

In this work, we only consider the Boolean constrained problem, i.e. $w \in \{0,1\}^n$, though we adopt the slack variables and regularization proposed by [28] to trade off between the sparsity and the discrepancy with the test results of the relaxed problem. With these modifications in the Boolean constrained problem (3), our problem becomes:

$$\min \quad \sum_{j=1}^{n} w_j + \lambda \sum_{i=1}^{m} \xi_i \tag{4a}$$

$$\text{s.t.} \quad w \in \{0,1\}^n \tag{4b}$$

$$0 \leq \xi_i \leq 1, \quad i \in \mathcal{P} \tag{4c}$$

$$0 \leq \xi_i, \quad i \in \mathcal{Z} \tag{4d}$$

$$A_{\mathcal{P}} w + \xi_{\mathcal{P}} \geq 1 \tag{4e}$$

$$A_{\mathcal{Z}} w - \xi_{\mathcal{Z}} = 0, \tag{4f}$$

where $\lambda > 0$ is a regularization parameter. This Boolean constrained problem formulation can be solved via integer linear programming (ILP) techniques, see, e.g., [28].

2.1.1 Generalization to other contexts

The solution to the ILP (4) can be seen as an interpretable rule-based classifier in contexts beyond standard group testing. Given a rule for forming the matrix A, encoding binary attributes of a set of objects through multiple measurements or tests, and test data y, the general problem is to derive a Boolean disjunction that best classifies previously unseen objects from their features. In such a general setting, a context-specific technique for dichotomizing features may be needed [41]. However, in the case of drug resistance prediction, our features are the presence or absence of specific single-nucleotide polymorphisms (SNPs), and therefore no dichotomization is needed.

From now on, we assume that we have a binary labeled dataset $\mathcal{D}=\{(x_1,y_1),\ldots,(x_m,y_m)\}$, where the $x_i \in \mathcal{X} := \{0,1\}^n$ are n-dimensional binary feature vectors and the $y_i \in \{0,1\}$ are the binary labels. The feature matrix A is defined via $A_{i,j} = (x_i)_j$ (the j-th component of the i-th feature vector). If $\hat{w}$ is the solution of ILP (4) for this feature matrix and the label vector $y = (y_i)_{i=1}^m$, we define the classifier $\hat{c} : \mathcal{X} \to \{0,1\}$ as follows:

$$\hat{c}(x) = x \vee \hat{w}. \tag{5}$$

2.2 Our approach

The formulation of the ILP (4) is designed to provide a trade-off between the sparsity of a disjunctive rule and the total slack, a quantity that resembles (but does not equal) the training error. Unmodified, these conditions are not ideal for machine learning tasks: *i)* they do not allow for accurate expression of this error, and *ii)* they lack the ability to assign different weights to different components of the error. Such a weighting can play a large role in settings where the data is highly unbalanced, or when the cost of a false positive differs greatly from that of a false negative. We now describe an approach that provides more flexibility in the training process and performs better on specific tasks such as ours.

Recall that the regularization parameter λ in equation (4) provides control over the trade-off between the total slack and the sparsity of the solution. It is straightforward to generalize this term to provide useful information about the classifier's false positive and false negative rates. To obtain this information, we modify the ILP (4) in two ways.

For clarity, in the following section we assume that $\hat{c}$ is a binary classifier trained on a sample y with corresponding Boolean feature matrix A. In addition, unless otherwise stated, we refer to the misclassification of a training sample as a false negative if it has label 1 (is in $\mathcal{P}$), and as a false positive if it has label 0 (is in $\mathcal{Z}$). For instance, in the case of drug resistance, a false negative would mean that we incorrectly predict a drug-resistant isolate as sensitive, while a false positive would mean that we predict a drug-sensitive isolate as resistant.

First, note that in ILP (4), $\xi_{\mathcal{P}}$ corresponds to the training error of $\hat{c}$ on the positively labeled subset of the data, while $\xi_{\mathcal{Z}}$ does not correspond to its training error on the negatively labeled subset. This follows from the fact that A is a binary matrix and w is a binary vector, so $\xi_{\mathcal{P}}$ is also a binary vector, with

$$\sum_{i\in\mathcal{P}} \xi_i = 1^T \xi_{\mathcal{P}} = \mathrm{FN}, \tag{6}$$

the number of false negatives. On the other hand, to obtain the number of false positives (FP) we need to modify the constraints (4d) and (4f) by setting

$$\xi_i \in \{0,1\}, \quad i \in \mathcal{Z} \tag{7}$$

and replacing $A_{\mathcal{Z}}w - \xi_{\mathcal{Z}} = 0$ with the inequalities:

$$A_{\mathcal{Z}}w - \xi_{\mathcal{Z}} \geq 0, \tag{8a}$$
$$\alpha_i\xi_i - A_iw \geq 0 \; \forall \; i \in \mathcal{Z}, \tag{8b}$$

where $\alpha_i = \sum_{j=1}^{n} A_{i,j}$ and A_i represent ith row of A. Note that the motivation behind this replacement is to count the number of non-zero elements of $A_{\mathcal{Z}}w$ by $\xi_{\mathcal{Z}}$. Therefore, we can observer that eq.(8a) ensure that $\xi_i = 0$ if $A_iw = 0$ and eq.(8b) ensures that $\xi_i = 1$ if $A_iw > 0$. However, eq.(8a) can be eliminated in those settings where the $\xi_{\mathcal{Z}}$ enter the objective function to be minimized with a positive coefficient. We will see similar situations in the following section.

After these modifications, we obtain

$$\sum_{i\in\mathcal{Z}} \xi_i = 1^T\xi_{\mathcal{Z}} = \text{FP}. \tag{9}$$

To provide the desired flexibility, we further split the regularization term into two terms corresponding to the positive class $\mathcal{P}$ and the negative class $\mathcal{Z}$:

$$\lambda_{\mathcal{P}} \sum_{i\in\mathcal{P}} \xi_i + \lambda_{\mathcal{Z}} \sum_{k\in\mathcal{Z}} \xi_k. \tag{10}$$

The general form of the new ILP is now as follows:

$$\begin{aligned}
\min \quad & \sum_{j=1}^{n} w_j + \lambda_{\mathcal{P}} \sum_{i\in\mathcal{P}} \xi_i + \lambda_{\mathcal{Z}} \sum_{k\in\mathcal{Z}} \xi_k \\
\text{s.t.} \quad & w \in \{0,1\}^n \\
& 0 \leq \xi_i \leq 1, \quad i \in \mathcal{P} \\
& \xi_i \in \{0,1\}, \quad i \in \mathcal{Z} \\
& A_{\mathcal{P}}w + \xi_{\mathcal{P}} \geq 1 \\
& \alpha_i\xi_i - A_iw \geq 0 \; \forall \; i \in \mathcal{Z}
\end{aligned} \tag{11}$$

In this new formulation, $\lambda_{\mathcal{P}}$ and $\lambda_{\mathcal{Z}}$ control the trade-off between the false positives and the false negatives, and jointly influence the sparsity of the rule. This formulation can be further tailored to optimize specific evaluation metrics. In the following section we demonstrate this for sensitivity and specificity, as an example.

2.3 Optimizing sensitivity and specificity

Since the ILP formulation in (11) provides us with direct access to the two components of the training error, we may modify the classifier to optimize a specific evaluation metric. For instance, assume that we would like to train the classifier $\hat{c}$ to maximize the sensitivity at a given specificity threshold $\bar{t}$. First, recall that

$$\text{Specificity} = \frac{\text{TN}}{\text{TN+FP}} = 1 - \frac{\text{FP}}{\text{N}}, \tag{12}$$
$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP+FN}} = 1 - \frac{\text{FN}}{\text{P}}. \tag{13}$$

From equation (10), equation (12) and the definition of $\mathcal{Z}$, we get the constraint

$$\bar{t} \leq 1 - \frac{1^T\xi_{\mathcal{Z}}}{|\mathcal{Z}|} \iff 1^T\xi_{\mathcal{Z}} \leq (1-\bar{t})|\mathcal{Z}|. \tag{14}$$

Our objective is to maximize sensitivity, which is equivalent to minimizing $\sum_{i\in\mathcal{P}}\xi_i$ by equations (13) and (6). Hence, the ILP (11) can be modified as follows:

$$\begin{aligned}
\min \quad & \sum_{j=1}^{n} w_j + \lambda_{\mathcal{P}} \sum_{i\in\mathcal{P}} \xi_i \\
\text{s.t.} \quad & w \in \{0,1\}^n \\
& 0 \le \xi_i \le 1, \quad i \in \mathcal{P} \\
& \xi_i \in \{0,1\}, \quad i \in \mathcal{Z} \\
& A_{\mathcal{P}} w + \xi_{\mathcal{P}} \ge 1 \\
& \alpha_i \xi_i - A_i w \ge 0 \; \forall \, i \in \mathcal{Z} \\
& 1^T \xi_{\mathcal{Z}} \le (1-\bar{t})|\mathcal{Z}|.
\end{aligned} \tag{15}$$

The maximum specificity at given sensitivity can be found analogously.

2.4 Approximating the AUROC

In this section we compute an analog of the AUROC[1] of our classifier given a limit on rule size. Recall that the ROC is a plot demonstrating the performance of a score-producing classifier at different score thresholds, created by plotting the true positive rate (TPR) against the false positive rate (FPR). However, since the rule-based classifier produced by ILP (11) is a discrete classifier, it cannot produce a ROC curve in the usual way. To create a ROC curve for this classifier, we compute the true positive rate (TPR) for different values of the false positive rate (FPR). In addition, we set a limit on the rule size (sparsity) of the classifier.

More precisely, we create the ROC curve by incrementally changing the FPR and computing the optimum value of the TPR. To do so, we put varying upper bounds on the FPR and proceed analogously to the previous section. For instance, assume that we would like to get the best TPR value when the FPR is at most $\hat{t}$, where $0 \le \hat{t} \le 1$, meaning that

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{N}} \le \hat{t}. \tag{16}$$

From equations (10), (16) and the definition of $\mathcal{Z}$ we get

$$\frac{1^T \xi_{\mathcal{Z}}}{|\mathcal{Z}|} \le \hat{t} \iff 1^T \xi_{\mathcal{Z}} \le \hat{t}|\mathcal{Z}|. \tag{17}$$

Assuming further that the limit on rule size is equal to $\hat{s}$, we have the following constraint:

$$1^T w \le \hat{s}. \tag{18}$$

[1] the Area Under the Receiver Operating Characteristic Curve

Therefore, the modified version of the ILP (11) suitable for computing an AUROC is:

$$\begin{aligned} \min \quad & \sum_{i \in \mathcal{P}} \xi_i \\ \text{s.t.} \quad & w \in \{0,1\}^n \\ & 0 \leq \xi_i \leq 1, \quad i \in \mathcal{P} \\ & \xi_i \in \{0,1\}, \quad i \in \mathcal{Z} \\ & A_{\mathcal{P}} w + \xi_{\mathcal{P}} \geq 1 \\ & \alpha_i \xi_i - A_i w \geq 0 \; \forall \; i \in \mathcal{Z} \\ & 1^T w \leq \hat{s} \\ & 1^T \xi_{\mathcal{Z}} \leq \hat{t} |\mathcal{Z}|. \end{aligned} \tag{19}$$

We utilize the CPLEX optimizer [20] to solve the ILP in (19).

3 Implementation

All the methods in this paper are implemented in the Python programming language. We use a Scikit-learn [38] implementation for the machine learning models and the CPLEX optimizer version 12.10.0 [20], together with its Python API, for our method.

3.1 Data

To obtain a dataset to train and evaluate our method on, we combine data from the Pathosystems Resource Integration Center (PATRIC)[47] and the Relational Sequencing TB Data Platform (ReSeqTB)[45]. This results in 8000 isolates together with their resistant/susceptible status (label) for seven drugs, including five first-line drugs (rifampicin, isoniazid, pyrazinamide, ethambutol, and streptomycin) and two second-line drugs (kanamycin and ofloxacin) [34, 8]. The short-read whole genome sequences of these 8000 isolates are downloaded from the European Nucleotide Archive [23] and the Sequence Read Archive [24]. The accession numbers used to obtain the data in our study were: ERP[000192, 006989, 008667, 010209, 013054, 000520], PRJEB[10385, 10950, 14199, 2358, 2794, 5162, 9680], PRJNA[183624, 235615, 296471], and SRP[018402, 051584, 061066].

In order to map the raw sequence data to the reference genome, we use a method similar to that used in previous work [7, 8]. We use the BWA software [25], specifically, the bwa-mem program. We then call the single-nucleotide polymorphisms (SNPs) of each isolate with two different pipelines, SAMtools [18] and GATK [39], and take the intersection of their calls to ensure reliability. The final dataset, which includes the position as well as the reference and alternative allele for each SNP [8], is used as the input to our classifier.

Starting from this input we create a binary feature matrix, where each row represents an isolate and each column indicates the presence or absence of a particular SNP. For each drug, we group all the SNPs with identical presence/absence patterns into a single column, since at most one SNP in a group would ever be selected to be part of a rule. The number of labeled and resistant isolates and of SNPs and SNP groups for each drug is stated in Table 1.

3.2 Train-Test split

To evaluate our classifier we use a stratified train-test split, where the training set contains 80% and the testing set contains 20% of data.

Table 1 Summary of number of isolates in our data.

Drug	Number of isolates	Number of resistant isolates	Number of SNPs	Number of SNP groups
Ethambutol	6,096	1,407	666,349	55,164
Isoniazid	7,734	3,445	666,349	65,090
Kanamycin	2,436	697	666,349	21,513
Ofloxacin	2,911	800	666,349	23,905
Pyrazinamide	3,858	754	666,349	33,942
Rifampicin	7,715	2,968	666,349	65,379
Streptomycin	5,125	2,104	666,349	45,037

3.3 AUROC comparison

The AUROC of our model was computed for two purposes: first, to investigate the effect of the classifier's sparsity (rule size) on its performance, and second, to compare this performance to that of other machine learning methods. We calculated the AUROC of classifiers with various limits on rule size, selected from $\{1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200\}$. For each rule size, we use the formulation in subsection 2.4, increasing the FPR upper bound from 0 to 1 in increments of 0.1. We then train a classifier by using the ILP (19), and compute the effective FPR and TPR. Lastly, we create the ROC curve by plotting the TPRs against the FPRs, and compute the AUROC.

To compare the performance of our model with other machine learning models, we also compute the AUROC of the Random Forest (RF) and ℓ_1-regularized Logistic Regression (LR) models. For these models, we first perform hyper-parameter tuning using grid search with three-fold cross validation, and then select the model with the highest AUROC.

3.4 Sensitivity at a fixed specificity

As another evaluation criteria we compute the sensitivity of our model at a desired specificity level (i.e. $\beta\%$ specificity). To do so, we use the ILP (15). In this formulation, the $\lambda_{\mathcal{P}}$ parameter can be tuned to provide the desired trade-off between the sparsity of the classifier (i.e., rule size) and the number of false negatives. However, in order to make a consistent comparison between the trained models for different drugs, we set a specific limit on rule size and use ILP (19) with the last constraint replaced by the last constraint of ILP (15), i.e. with (17) replaced with (14).

4 Results

Evaluating the performance of an interpretable predictive model can be challenging. While most evaluation methods focus on predictive accuracy, it is essential to assess the model's interpretability. Even though there is no consensus on the definition of interpretability, the "Predictive, Descriptive, Relevant" (PDR) framework introduced by [32] provides general insights into interpretable models, by emphasizing the balance between these characteristics. In this section, we use the PDR framework to evaluate our models in the following ways.

First, in Section 4.1, we assess our method's predictive accuracy by comparing it with RF and LR. At this step we do not have any specific restriction on the rule size, and we report the best AUROC that our model can achieve based on the settings in Section 3.3.

Second, in Section 4.2, we compare the AUROC produced by our method for different limits on rule size. This comparison between the method at different parameter values helps us evaluate its ability to produce a simple model (i.e. a model with a fairly small rule size) with a high AUROC. The simpler models are easier to understand for human users. In

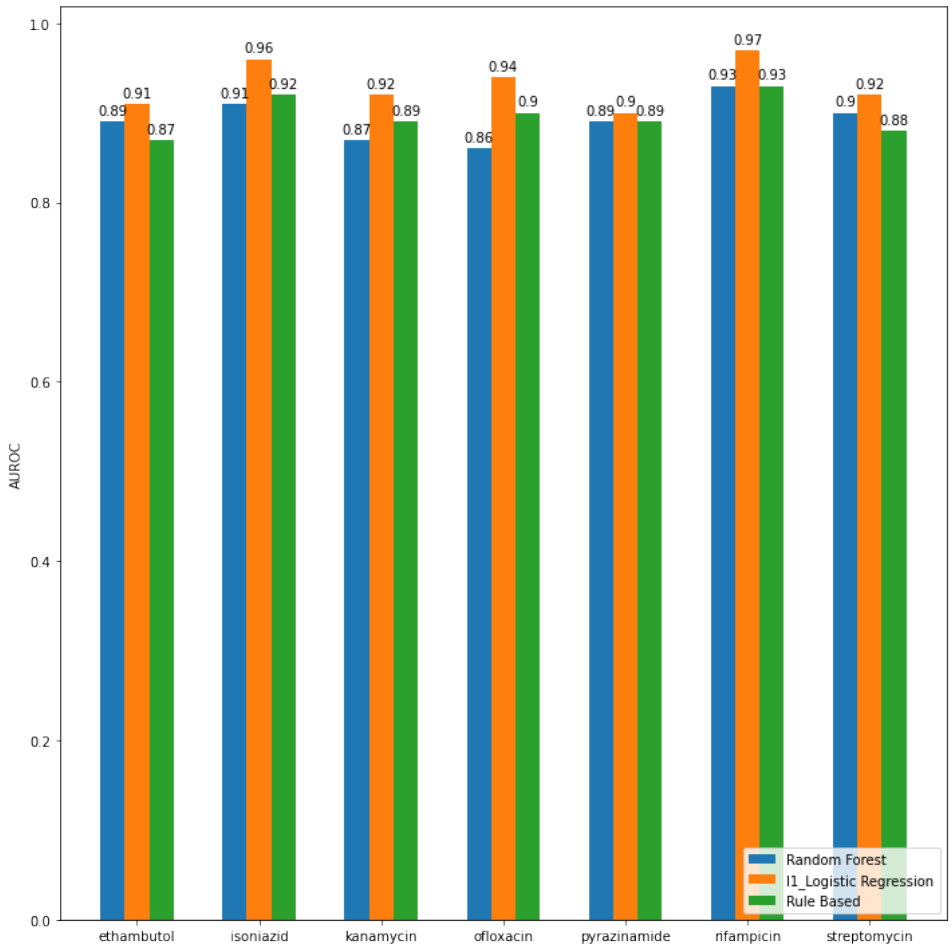

Figure 1 Comparison between the test AUROC of our rule-based model (with no limit imposed on the rule size), ℓ_1-regularized logistic regression and Random Forest.

this paper, we define the descriptiveness of a model by its simplicity (its rule size, i.e., the number of SNPs needed to define it). In addition, we evaluate our method's sensitivity by comparing it with LR and RF. To do so, we compute and compare the sensitivity of these three models at a specificity near 90%. More specifically, this comparison uses the specificity level achieved by the rule-based model that is closest to 90% (in practice, this is always between 88% and 92% for this dataset), since the rule-based model does not achieve every possible specificity level when given a limit on rule size. For this evaluation, we limit model complexity by setting a limit of 20 on the rule size.

Finally, in Section 4.3, we assess the relevance of the model produced by our method by observing the fraction of SNPs used by the model that are located in genes previously reported to be associated with drug resistance. Note that, unlike the approach in [48], we do not limit the genes *a priori* to those with known associations with drug resistance.

4.1 Our models produce competitive AUROCs

Figure 1 illustrates the results of comparing our model to LR and RF. In this figure, we can see that LR provides a higher AUROC for all 7 drugs, but our model produces slightly higher AUROCs than RF for 3 of the drugs, identical AUROCs for 2 other drugs and slightly lower ones for the remaining 2.

Table 2 Comparison between AUROCs of models produced by our method with different rule size limits. We observe that even small rule sizes produce models with a high AUROC.

Drug	Rule size ≤ 10	Rule size ≤ 20	Rule size ≤ 30	Rule size ≤ 40	Max AUROC
Ethambutol	0.86	0.86	0.85	**0.86**	0.87
Isoniazid	0.88	0.89	0.90	**0.91**	0.92
Kanamycin	0.88	0.89	**0.89**	0.88	0.89
Ofloxacin	0.90	0.87	**0.90**	0.88	0.90
Pyrazinamide	0.88	0.88	0.88	**0.89**	0.89
Rifampicin	0.90	0.92	0.92	**0.93**	0.93
Streptomycin	0.84	0.86	0.85	**0.87**	0.88

4.2 Our approach is able to produce simple models with high AUROC

Figure 2 demonstrates the change in AUROC as we increase the limit on the rule size. Our results show that as the limit on the rule size increases, we get higher AUROC on the training set. However, on the test set, we see that the AUROC increases more slowly after a rule size limit of 10, and eventually starts to decrease.

As shown in Figure 2 and Table 2, the AUROC does not increase significantly beyond a rule size limit of 10. Thus, our method is capable of producing models with a rule sizes small enough to keep the model simple yet keep the AUROC within 1% of the maximum.

Table 3 shows the same trend for the ℓ_1-regularized logistic regression. We see that, at the low rule-size limits (such as 10 and 20), our approach produces a comparable performance to that of ℓ_1-regularized logistic regression, while it is slightly worse for larger rule-size limits. At the same time, as we show in Figures 4a and 4b below, our approach results in the recovery of a lot more genes known to be associated with drug resistance than logistic regression.

4.3 Our model uses genes previously associated to drug resistance

Our results show that the models produced by our method contains many SNPs in genes previously associated with drug resistance in *Mycobacterium tuberculosis*. Due to the large size of SNP groups (SNPs in perfect linkage disequilibrium), the causality of specific SNPs remains difficult to determine. However, many of the genes known to be relevant to resistance mechanisms appear among the possible variants that are pointed to by the selected groups of duplicated SNPs.

In Figure 4a we show the number of SNPs within different classes of genes found by our approach with rule size ≤ 20 and specificity $\geq 90\%$, where each gene is classified according to whether it has a known association to drug resistance ("known") or not ("unknown"), with

Table 3 Comparison between AUROCs of models produced by ℓ_1-regularized logistic regression with different numbers of non-zero regression coefficients.

Drug	Non-zero coef. ≤ 10	Non-zero coef. ≤ 20	Non-zero coef. ≤ 30	Non-zero coef. ≤ 40	Max AUROC
Ethambutol	0.87	0.87	0.88	0.89	0.91
Isoniazid	0.90	0.91	0.92	0.93	0.96
Kanamycin	0.90	0.91	0.91	0.92	0.92
Ofloxacin	0.86	0.90	0.94	0.94	0.94
Pyrazinamide	0.81	0.87	0.89	0.89	0.90
Rifampicin	0.92	0.92	0.94	0.94	0.97
Streptomycin	0.88	0.88	0.89	0.90	0.92

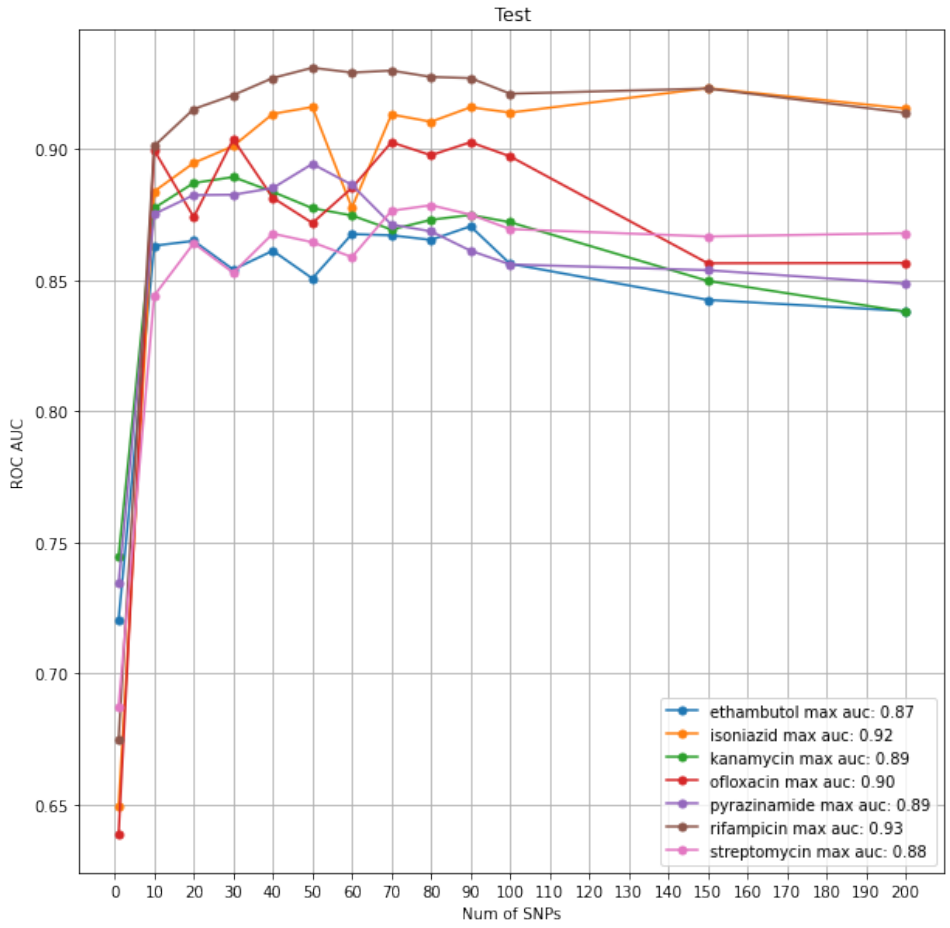

Figure 2 Test AUROC for models trained on each drug with various rule size limits. Beyond a certain rule size, which varies with the drug, the AUROC of the predictive model no longer improves.

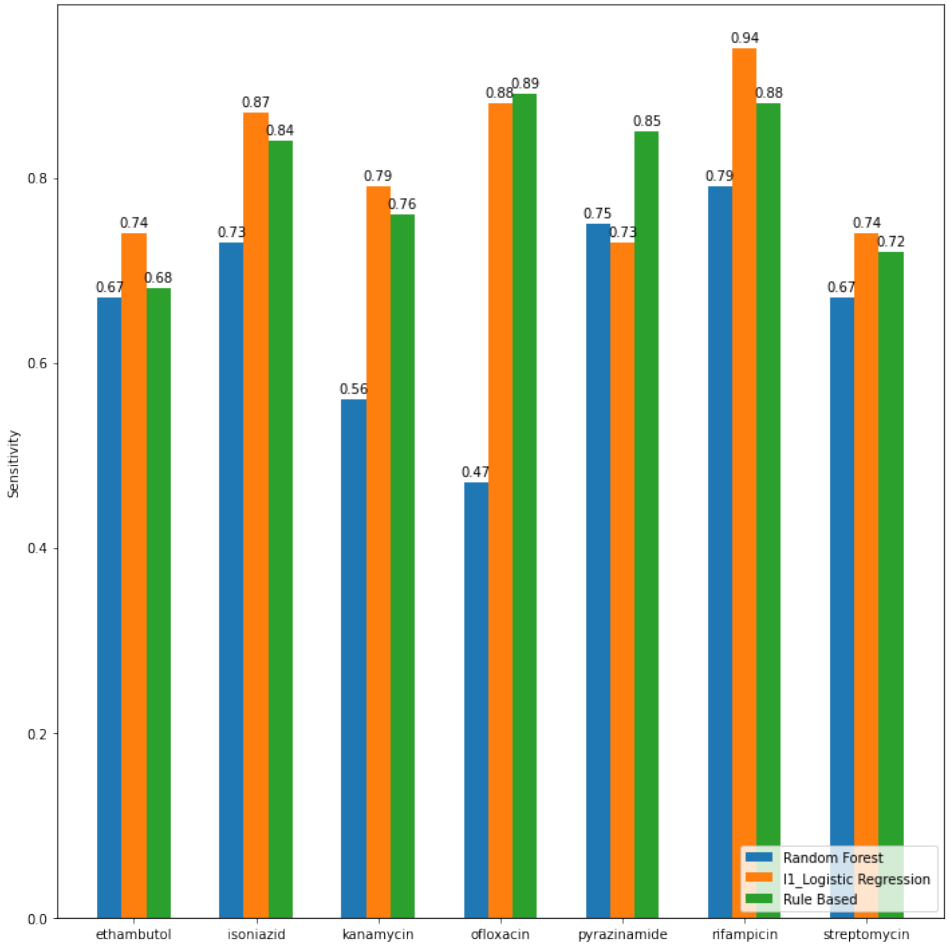

Figure 3 Comparison between the sensitivity of our rule-based method with the rule size limit set to 20, ℓ_1-Logistic regression and Random Forest at around 90% specificity on the testing data.

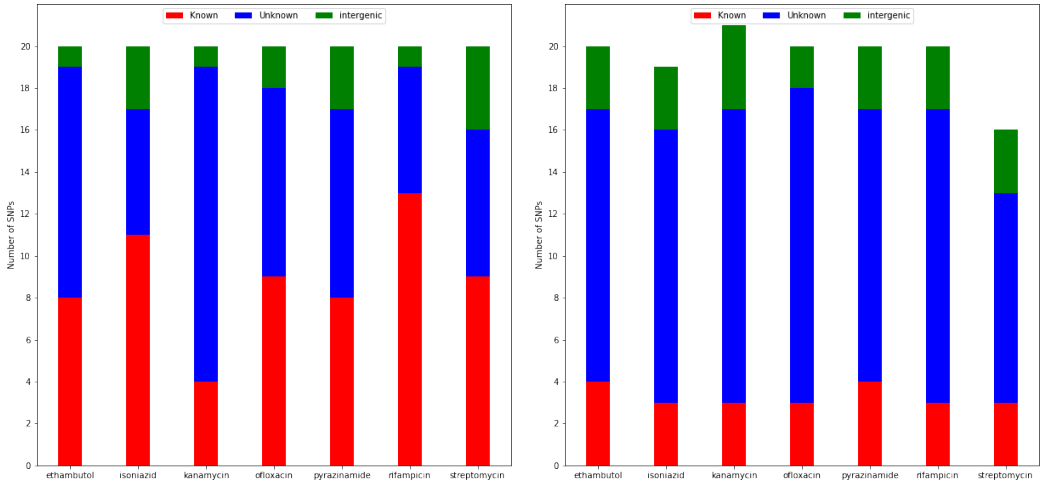

Figure 4 (a) The number of SNPs in genes with known association to drug resistance, genes without such an association, and intergenic regions, in our models with at most 20 SNPs and a specificity of $\geq 90\%$. (b) The same numbers for ℓ_1-Logistic regression models with as close as possible to 20 non-zero regression coefficients.

an additional class for SNPs in intergenic regions. We show these numbers for ℓ_1-Logistic regression models with as close as possible to 20 non-zero regression coefficients in Figure 4b. A comparison between these figures suggests that when both approaches are restricted to a small number of features, our approach detects more relevant SNPs than ℓ_1-logistic regression. The list of "known" genes, selected through a data-driven and consensus-driven process by a panel of experts, is the one in [31], containing 183 out of over 4,000 *M. tuberculosis* genes. We note that in both cases, a group of SNPs in perfect linkage disequilibrium was coded as "known" if at least one of the SNPs was contained in a known gene, "intergenic" if none of them appeared in a gene, and as "unknown" otherwise.

4.4 Running time

We run our code on a cluster node with 2 CPU sockets, each with an 8-core 2.60 GHz Intel Xeon E5-2640 v3 with 32 threads. The training of a single model with fixed hyper-parameters takes between 1 and 8 minutes. This suggests that once a suitable value is chosen for the hyper-parameters, the optimization used to determine the optimal rule can be performed efficiently. Overall, producing the ROC curve for each drug takes between 3 and 18 hours, depending on the number of labeled isolates available for each drug.

5 Conclusion

In this paper, we introduced a new approach for creating rule-based classifiers. Our method utilizes the group testing problem and Boolean compressed sensing. It can produce interpretable, highly accurate, flexible classifiers which can be optimized for particular evaluation metrics.

We used our method to produce classifiers for predicting drug resistance in *Mycobacterium tuberculosis*. The classifiers' predictive accuracy was tested on a variety of antibiotics commonly used for treating tuberculosis, including five first-line and two second-line drugs.

We show that our method could produce classifiers with a high AUROC, slightly less than that of unrestricted ℓ_1-Logistic regression, and comparable to Random Forest, as well as ℓ_1-Logistic regression restricted to a comparably small number of selected features for interpretability. In addition, we show that our method is capable of producing accurate models with a rule size small enough to keep the model understandable for human users. Finally, we show that our approach can provide useful insights into its input data - in this case, it could help identify genes associated with drug resistance.

We note that the presence of SNPs with identical presence/absence patterns, which would be referred to as being in perfect linkage disequilibrium (LD) in genetics [42], is common in bacteria such as *Mycobacterium tuberculosis* whose evolution is primarily clonal [17]. For this reason, while the grouping of such SNPs together substantially greatly simplifies the computational task at hand, it is challenging to ascertain the exact representative of each group that should be selected to determine the drug resistance status of an isolate. Determining this representative would likely require larger sample sizes or a built-in prior knowledge of the functional effects of individual SNPs.

We also note that the genes we define as having a known association to drug resistance are not specific to the drug being tested, i.e. some of them may have been found to be associated with the resistance to a drug other than the one being predicted. This is to be expected, however, as the distinct resistance mechanisms are generally less numerous than antibiotics [44]. It will be interesting to see whether methods such as ours are able to detect specific, for instance, by testing it on data for newly developed antibiotics such as bedaquiline and delamanid [22].

Our goal in this paper was to introduce a novel method for producing interpretable models and explore its accuracy, descriptive ability, and relevance in detecting drug resistance in *Mycobacterium tuberculosis* isolates. In this study, the focus was mostly on the predictive accuracy, and we will explore the similarities and differences between our model and other interpretable techniques (both model-based and *post-hoc* ones) in future work.

References

1 Matthew Aldridge, Oliver Johnson, Jonathan Scarlett, et al. Group testing: an information theory perspective. *Foundations and Trends® in Communications and Information Theory*, 15(3-4):196–392, 2019.

2 Gustavo Arango-Argoty, Emily Garner, Amy Pruden, Lenwood S Heath, Peter Vikesland, and Liqing Zhang. DeepARG: a deep learning approach for predicting antibiotic resistance genes from metagenomic data. *Microbiome*, 6(1):1–15, 2018.

3 G. K. Atia and V. Saligrama. Boolean compressed sensing and noisy group testing. *IEEE Transactions on Information Theory*, 58(3):1880–1901, 2012.

4 P. Bradley, N. Gordon, T Walker, et al. Rapid antibiotic-resistance predictions from genome sequence data for Staphylococcus aureus and Mycobacterium tuberculosis. *Nature Communications*, 6, 2015.

5 E. J. Candes and M. B. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, 2008.

6 Albert Cohen, Wolfgang Dahmen, and Ronald DeVore. Compressed sensing and best k-term approximation. *Journal of the American mathematical society*, 22(1):211–231, 2009.

7 Francesc Coll, Ruth McNerney, José Afonso Guerra-Assunção, Judith R. Glynn, João Perdigão, Miguel Viveiros, Isabel Portugal, Arnab Pain, Nigel Martin, and Taane G. Clark. A robust snp barcode for typing mycobacterium tuberculosis complex strains. *Nature Communications*, 2014. doi:10.1038/ncomms5812.

8 Wouter Deelder, Sofia Christakoudi, Jody Phelan, Ernest Diez Benavente, Susana Campino, Ruth McNerney, Luigi Palla, and Taane Gregory Clark. Machine learning predicts accurately Mycobacterium tuberculosis drug resistance from whole genome sequencing data. *Frontiers in Genetics*, 10:922, 2019.

9 Alireza Doostan and Houman Owhadi. A non-adapted sparse approximation of PDEs with stochastic inputs. *Journal of Computational Physics*, 230(8):3015–3034, 2011.

10 Robert Dorfman. The detection of defective members of large populations. *The Annals of Mathematical Statistics*, 14(4):436–440, 1943.

11 Sorin Drăghici and R Brian Potter. Predicting HIV drug resistance with neural networks. *Bioinformatics*, 19(1):98–107, 2003.

12 M. F. Duarte and Y. C. Eldar. Structured compressed sensing: From theory to applications. *IEEE Transactions on Signal Processing*, 59(9):4053–4085, 2011.

13 Y.C. Eldar and G. Kutyniok. *Compressed Sensing: Theory and Applications*. Cambridge University Press, 2012. URL: `https://books.google.ca/books?id=9ccLAQAAQBAJ`.

14 Coll F, McNerney R, Preston MD, et al. Rapid determination of anti-tuberculosis drug resistance from whole-genome sequences. *Genome Med.*, 7:51, 2015.

15 Silke Feuerriegel, Viola Schleusener, Patrick Beckert, Thomas A. Kohl, Paolo Miotto, Daniela M. Cirillo, Andrea M. Cabibbe, Stefan Niemann, and Kurt Fellenberg. PhyResSE: a web tool delineating Mycobacterium tuberculosis antibiotic resistance and lineage from whole-genome sequencing data. *Journal of Clinical Microbiology*, 53(6):1908–1914, 2015.

16 S. Foucart and H. Rauhut. *A Mathematical Introduction to Compressive Sensing*. Applied and Numerical Harmonic Analysis. Springer New York, 2013. URL: `https://books.google.ca/books?id=zb28BAAAQBAJ`.

17 Sebastien Gagneux. Ecology and evolution of Mycobacterium tuberculosis. *Nat Rev Microbiol*, 16:202–213, 2018.

18 Li H1, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R, and 1000 Genome Project Data Processing Subgroup. The sequence alignment/map format and samtools. *Bioinformatics*, 25(16):2078–2079, 2009.

19 Matthew A Herman and Thomas Strohmer. High-resolution radar via compressed sensing. *IEEE transactions on signal processing*, 57(6):2275–2284, 2009.

20 IBM. IBM ILOG CPLEX Optimization Studio V12.10.0 documentation. `https://www.ibm.com/support/knowledgecenter/SSSA5P`, 2020.

21 H. Iwai, M. Kato-Miyazawa, T Kirikae, and T. Miyoshi-Akiyama. CASTB (the comprehensive analysis server for the Mycobacterium tuberculosis complex): A publicly accessible web server for epidemiological analyses, drug-resistance prediction and phylogenetic comparison of clinical isolates. *Tuberculosis*, pages 843–844, 2015.

22 Suha Kadura, Nicholas King, Maria Nakhoul, Hongya Zhu, Grant Theron, Claudio U Köser, and Maha Farhat. Systematic review of mutations associated with resistance to the new and repurposed Mycobacterium tuberculosis drugs bedaquiline, clofazimine, linezolid, delamanid and pretomanid. *Journal of Antimicrobial Chemotherapy*, May 2020. dkaa136.

23 Rasko Leinonen, Ruth Akhtar, Ewan Birney, Lawrence Bower, Ana Cerdeno-Tárraga, et al. The European Nucleotide Archive. *Nucleic Acids Research*, 39:D28–31, 2011.

24 Rasko Leinonen, Hideaki Sugawara, Martin Shumway, and International Nucleotide Sequence Database Collaboration. The sequence read archive. *Nucleic acids research*, 39(suppl_1):D19–D21, 2010.

25 H Li. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv*, 2013.

26 Michael Lustig, David Donoho, and John M Pauly. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007.

27 D. Malioutov and M. Malyutov. Boolean compressed sensing: LP relaxation for group testing. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3305–3308, 2012.

28 Dmitry Malioutov and Kush Varshney. Exact rule learning via Boolean compressed sensing. In *International Conference on Machine Learning*, pages 765–773, 2013.

29 L Mathelin and KA Gallivan. A compressed sensing approach for partial differential equations with random input data. *Communications in computational physics*, 12(4):919–954, 2012.

30 Arya Mazumdar. On almost disjunct matrices for group testing. In Kun-Mao Chao, Tsan-sheng Hsu, and Der-Tsai Lee, editors, *Algorithms and Computation*, pages 649–658, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

31 Paolo Miotto, Belay Tessema, Elisa Tagliani, Leonid Chindelevitch, et al. A standardised method for interpreting the association between mutations and phenotypic drug-resistance in *Mycobacterium tuberculosis*. *European Respiratory Journal*, 50(6), 2017.

32 W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Interpretable machine learning: definitions, methods, and applications. *arXiv*, 2019.

33 Balas Kausik Natarajan. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234, 1995.

34 Tra-My Ngo and Yik-Ying Teo. Genomic prediction of tuberculosis drug-resistance: benchmarking existing databases and prediction algorithms. *BMC Bioinformatics*, 20(1):68, 2019.

35 Jim O'Neill. Antimicrobial resistance: Tackling a crisis for the health and wealth of nations. Technical report, Review on Antimicrobial Resistance, 2014. URL: `https://amr-review.org/Publications.html`.

36 World Health Organization. Antimicrobial resistance: global report on surveillance. Technical report, WHO, 2014.

37 World Health Organization. Global tuberculosis report 2019. Technical report, WHO, 2019.

38 F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

39 Ryan Poplin, Valentin Ruano-Rubio, Mark A. DePristo, Tim J. Fennell, Mauricio O. Carneiro, Geraldine A. Van der Auwera, David E. Kling, Laura D. Gauthier, Ami Levy-Moonshine, David Roazen, Khalid Shakir, Joel Thibault, Sheila Chandran, Chris Whelan, Monkol Lek, Stacey Gabriel, Mark J Daly, Ben Neale, Daniel G. MacArthur, and Eric Banks. Scaling accurate genetic variant discovery to tens of thousands of samples. *bioRxiv*, 2017.

40 Mario C Raviglione and Ian M Smith. XDR tuberculosis—implications for global public health. *New England Journal of Medicine*, 356(7):656–659, 2007.

41 Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144, 2016.

42 James Emmanuel San, Shakuntala Baichoo, Aquillah Kanzi, Yumna Moosa, Richard Lessells, Vagner Fonseca, John Mogaka, Robert Power, and Tulio de Oliveira. Current affairs of microbial genome-wide association studies: Approaches, bottlenecks and analytical pitfalls. *Frontiers in Microbiology*, 10:3119, 2020. URL: `https://www.frontiersin.org/article/10.3389/fmicb.2019.03119`.

43 V. Schleusener, C. Köser, P. Beckert, et al. Mycobacterium tuberculosis resistance prediction and lineage classification from genome sequencing: comparison of automated analysis tools. *Scientific Reports*, 7, 2017.

44 Almeida Da Silva, Pedro Eduardo, Palomino, and Juan Carlos. Molecular basis and mechanisms of drug resistance in Mycobacterium tuberculosis: classical and new drugs. *Journal of Antimicrobial Chemotherapy*, 66(7):1417–1430, May 2011.

45 Angela M Starks, Enrique Avilés, Daniela M Cirillo, Claudia M Denkinger, David L Dolinger, Claudia Emerson, Jim Gallarda, Debra Hanna, Peter S Kim, Richard Liwski, et al. Collaborative effort for a centralized worldwide tuberculosis relational sequencing data platform. *Clinical Infectious Diseases*, 61(suppl_3):S141–S146, 2015.

46 A Steiner, D Stucki, M Coscolla, S Borrell, and S Gagneux. KvarQ: targeted and direct variant calling from fastq reads of bacterial genomes. *BMC Genomics*, 15, 2014.

47 Alice R Wattam, David Abraham, Oral Dalay, Terry L Disz, Timothy Driscoll, Joseph L Gabbard, Joseph J Gillespie, Roger Gough, Deborah Hix, Ronald Kenyon, et al. PATRIC, the bacterial bioinformatics database and analysis resource. *Nucleic acids research*, 42(D1):D581–D591, 2014.

48 Yang Yang, Katherine E Niehaus, Timothy M Walker, Zamin Iqbal, A Sarah Walker, Daniel J Wilson, Tim EA Peto, Derrick W Crook, E Grace Smith, Tingting Zhu, et al. Machine learning for classifying tuberculosis drug-resistance from DNA sequencing data. *Bioinformatics*, 34(10):1666–1671, 2018.