

¹ An interpretable classification method for ² predicting drug resistance in *M. tuberculosis*

³ **Hooman Zabeti**

⁴ School of Computing Science, Simon Fraser University, Burnaby, BC, V5A 1S6, Canada

⁵ **Nick Dexter**

⁶ Department of Mathematics, Simon Fraser University, Burnaby, BC, V5A1S6, Canada

⁷ **Amir Hosein Safari**

⁸ School of Computing Science, Simon Fraser University, Burnaby, BC, V5A 1S6, Canada

⁹ **Nafiseh Sedaghat**

¹⁰ School of Computing Science, Simon Fraser University, Burnaby, BC, V5A 1S6, Canada

¹¹ **Maxwell Libbrecht**

¹² School of Computing Science, Simon Fraser University, Burnaby, BC, V5A 1S6, Canada

¹³ **Leonid Chindelevitch**

¹⁴ School of Computing Science, Simon Fraser University, Burnaby, BC, V5A 1S6, Canada

¹⁵ — **Abstract** —

¹⁶ **Motivation:** The prediction of drug resistance and the identification of its mechanisms in bacteria
¹⁷ such as *Mycobacterium tuberculosis*, the etiological agent of tuberculosis, is a challenging problem.
¹⁸ Modern methods based on testing against a catalogue of previously identified mutations often yield
¹⁹ poor predictive performance. On the other hand, machine learning techniques have demonstrated
²⁰ high predictive accuracy, but many of them lack interpretability to aid in identifying specific
²¹ mutations which lead to resistance. We propose a novel technique, inspired by the group testing
²² problem and Boolean compressed sensing, which yields highly accurate predictions and interpretable
²³ results at the same time.

²⁴ **Results:** We develop a modified version of the Boolean compressed sensing problem for identifying
²⁵ drug resistance, and implement its formulation as an integer linear program. This allows us to
²⁶ characterize the predictive accuracy of the technique and select an appropriate metric to optimize.
²⁷ A simple adaptation of the problem also allows us to quantify the sensitivity-specificity trade-off of
²⁸ our model under different regimes. We test the predictive accuracy of our approach on a variety
²⁹ of commonly used antibiotics in treating tuberculosis and find that it has accuracy comparable to
³⁰ that of standard machine learning models and points to several genes with previously identified
³¹ association to drug resistance.

³² **Availability:** https://github.com/hoomanzabeti/TB_Resistance_RuleBasedClassifier

³³ **Contact:** hooman_zabeti@sfu.ca

³⁴

³⁵ **2012 ACM Subject Classification** Applied computing - Life and medical sciences - Computational
³⁶ biology - Molecular sequence analysis

³⁷ **Keywords and phrases** Drug resistance; whole-genome sequencing; interpretable machine learning;
³⁸ integer linear programming; rule-based learning

³⁹ **Digital Object Identifier** 10.4230/LIPIcs.WABI.2020.2



© Hooman Zabeti, Nick Dexter, Amir Hosein Safari, Nafiseh Sedaghat, Maxwell Libbrecht, Leonid Chindelevitch;

licensed under Creative Commons License CC-BY

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

LIPIcs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

2:2 An interpretable classification method for drug resistance

40 **Acknowledgements** The authors would like to thank Dr. Cedric Chauve, Dr. Ben Adcock and
41 Matthew Nguyen for helpful discussions. This project was funded by the Genome Canada grant
42 BAC283. LC acknowledges additional funding from the CANSSI CRT and NSERC Discovery.

43 1 Introduction

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

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

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

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

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

80 Our approach combines some of the flexibility and interpretability of catalogue methods with
81 the accuracy of machine learning methods—specifically, this method is capable of recovering
82 interpretable rules for predicting drug resistance that both result in a high classification
83 accuracy as well as provide insights into the mechanisms of drug resistance. We show that

84 our methods perform comparably to standard machine learning methods on *Mycobacterium*
85 *tuberculosis* in terms of predicting first-line drug resistance, while accurately recovering
86 many of the known mechanisms of drug resistance, and identifying some potentially novel
87 ones.

88 **2 Methods**

89 Our proposed method is based on the rule-based classification technique introduced in
90 [28], wherein group testing and Boolean CS are combined to determine subsets of infected
91 individuals from large populations. In that setting the linear system encodes the infection
92 status of the population through testing, and the solution, obtained from a suitable decoder,
93 is a $\{0, 1\}$ -valued vector representing the infection status of the individuals [6]. Since the
94 infected group is assumed to be small, the solution vector is sparse and can be recovered
95 using relatively few measurements with Boolean CS. The result of solving the Boolean CS
96 problem can then be interpreted as a sparse set of rules for determining infections and used
97 for classification on unseen data.

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

106 **2.1 Group testing and Boolean compressed sensing**

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

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

$$123 \quad y = A \vee w, \quad (1)$$

2:4 An interpretable classification method for drug resistance

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

$$125 \quad y_i = \bigvee_{j=1}^n A_{i,j} \wedge w_j \quad \forall 1 \leq i \leq m.$$

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

$$129 \quad \min \|w\|_0 \text{ subject to } y = A \vee w, \quad (2)$$

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

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

137

$$138 \quad \begin{aligned} & \min \sum_{j=1}^n w_j \\ & \text{s.t. } w \in \{0, 1\}^n \\ & \quad A_{\mathcal{P}} w \geq 1 \\ & \quad A_{\mathcal{Z}} w = 0, \end{aligned} \quad (3)$$

139 where $\mathcal{P} = \{i : y_i = 1\}$ and $\mathcal{Z} = \{i : y_i = 0\}$ are the sets of groups that test positive and
140 negative, respectively. However, this problem is also NP-hard, but can be made tractable
141 for linear programming by relaxing the Boolean constraint on w in (3) to $0 \leq w_j \leq 1$ for all
142 $j \in \{1, \dots, n\}$.

143 [28] extended this idea for interpretable rule-based classification, meanwhile proving recovery
144 guarantees for the relaxed problem. Because the Boolean CS problem is based on Boolean
145 algebra, the conditions on the Boolean measurement matrices A that guarantee exact recovery
146 of K -sparse vectors via linear programming are quite different from those of standard CS.
147 Specifically, these guarantees require the definition of K -disjunct matrices, i.e., matrices A
148 for which all unions of their columns of size K do not contain any other columns of the
149 original matrix. Constructions exist for matrices with $\mathcal{O}(K^2 \log(n))$ rows which satisfy this
150 property. We also note that by introducing an *approximate disjunctness* property, allowing
151 for matrices for which a fraction $(1 - \varepsilon)$ of all $\binom{n}{K}$ possible K -subsets of the columns satisfy
152 the disjunctness condition, it was shown in [30] that there exist constructions of measurement
153 matrices A which allow for recovery from $\mathcal{O}(K^{3/2} \sqrt{\log(n/\varepsilon)})$ rows.

154 In the standard setting for uniform recovery results for CS, the measurement matrices A
155 are subgaussian random matrices, i.e., having entries $A_{i,j}$ drawn independently according
156 to a subgaussian distribution. Examples include $m \times n$ matrices consisting of Rademacher
157 or Gaussian random variables, for which uniform recovery of K -sparse vectors via ℓ_1 -
158 minimization has been shown under the condition m is $\mathcal{O}(K \log(n/K))$, see, e.g. [16, Chapter
159 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:

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

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

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

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

$$A_{\mathcal{P}} w + \xi_{\mathcal{P}} \geq 1 \quad (4e)$$

$$A_{\mathcal{Z}} w - \xi_{\mathcal{Z}} = 0, \quad (4f)$$

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 $\mathcal{D} = \{(x_1, y_1), \dots, (x_m, y_m)\}$, 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 $\hat{c} : \mathcal{X} \rightarrow \{0, 1\}$ as follows:

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

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

2:6 An interpretable classification method for drug resistance

201 this term to provide useful information about the classifier's false positive and false negative
 202 rates. To obtain this information, we modify the ILP (4) in two ways.

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

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

$$213 \quad \sum_{i \in \mathcal{P}} \xi_i = 1^T \xi_{\mathcal{P}} = \text{FN}, \quad (6)$$

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

$$216 \quad \xi_i \in \{0, 1\}, \quad i \in \mathcal{Z} \quad (7)$$

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

$$218 \quad A_{\mathcal{Z}}w - \xi_{\mathcal{Z}} \geq 0, \quad (8a)$$

$$219 \quad \alpha_i \xi_i - A_i w \geq 0 \quad \forall i \in \mathcal{Z}, \quad (8b)$$

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

227 After these modifications, we obtain

$$228 \quad \sum_{i \in \mathcal{Z}} \xi_i = 1^T \xi_{\mathcal{Z}} = \text{FP}. \quad (9)$$

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

$$231 \quad \lambda_{\mathcal{P}} \sum_{i \in \mathcal{P}} \xi_i + \lambda_{\mathcal{Z}} \sum_{k \in \mathcal{Z}} \xi_k. \quad (10)$$

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

$$233 \quad \begin{aligned} \min \quad & \sum_{j=1}^n w_j + \lambda_{\mathcal{P}} \sum_{i \in \mathcal{P}} \xi_i + \lambda_{\mathcal{Z}} \sum_{k \in \mathcal{Z}} \xi_k \\ \text{s.t.} \quad & w \in \{0, 1\}^n \\ & 0 \leq \xi_i \leq 1, \quad i \in \mathcal{P} \\ & \xi_i \in \{0, 1\}, \quad i \in \mathcal{Z} \\ & A_{\mathcal{P}}w + \xi_{\mathcal{P}} \geq 1 \\ & \alpha_i \xi_i - A_i w \geq 0 \quad \forall i \in \mathcal{Z} \end{aligned} \quad (11)$$

²³⁴ In this new formulation, $\lambda_{\mathcal{P}}$ and $\lambda_{\mathcal{Z}}$ control the trade-off between the false positives and the
²³⁵ false negatives, and jointly influence the sparsity of the rule. This formulation can be further
²³⁶ tailored to optimize specific evaluation metrics. In the following section we demonstrate this
²³⁷ for sensitivity and specificity, as an example.

²³⁸ **2.3 Optimizing sensitivity and specificity**

²³⁹ Since the ILP formulation in (11) provides us with direct access to the two components of
²⁴⁰ the training error, we may modify the classifier to optimize a specific evaluation metric. For
²⁴¹ instance, assume that we would like to train the classifier \hat{c} to maximize the sensitivity at a
²⁴² given specificity threshold \bar{t} . First, recall that

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} = 1 - \frac{\text{FP}}{\text{N}}, \quad (12)$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} = 1 - \frac{\text{FN}}{\text{P}}. \quad (13)$$

²⁴³ From equation (10), equation (12) and the definition of \mathcal{Z} , we get the constraint

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

²⁴⁴ Our objective is to maximize sensitivity, which is equivalent to minimizing $\sum_{i \in \mathcal{P}} \xi_i$ by
²⁴⁵ equations (13) and (6). Hence, the ILP (11) can be modified as follows:

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

²⁵² The maximum specificity at given sensitivity can be found analogously.

²⁵³ **2.4 Approximating the AUROC**

²⁵⁴ In this section we compute an analog of the AUROC¹ of our classifier given a limit on rule
²⁵⁵ size. Recall that the ROC is a plot demonstrating the performance of a score-producing
²⁵⁶ classifier at different score thresholds, created by plotting the true positive rate (TPR) against
²⁵⁷ the false positive rate (FPR). However, since the rule-based classifier produced by ILP (11)
²⁵⁸ is a discrete classifier, it cannot produce a ROC curve in the usual way. To create a ROC
²⁵⁹ curve for this classifier, we compute the true positive rate (TPR) for different values of
²⁶⁰ the false positive rate (FPR). In addition, we set a limit on the rule size (sparsity) of the
²⁶¹ classifier.

²⁶² More precisely, we create the ROC curve by incrementally changing the FPR and computing
²⁶³ the optimum value of the TPR. To do so, we put varying upper bounds on the FPR and

¹ the Area Under the Receiver Operating Characteristic Curve

2:8 An interpretable classification method for drug resistance

264 proceed analogously to the previous section. For instance, assume that we would like to get
265 the best TPR value when the FPR is at most \hat{t} , where $0 \leq \hat{t} \leq 1$, meaning that

$$266 \quad \text{FPR} = \frac{\text{FP}}{\text{N}} \leq \hat{t}. \quad (16)$$

267

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

$$269 \quad \frac{1^T \xi_{\mathcal{Z}}}{|\mathcal{Z}|} \leq \hat{t} \iff 1^T \xi_{\mathcal{Z}} \leq \hat{t} |\mathcal{Z}|. \quad (17)$$

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

271

$$272 \quad 1^T w \leq \hat{s}. \quad (18)$$

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

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

274

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

276 3 Implementation

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

280 3.1 Data

281 To obtain a dataset to train and evaluate our method on, we combine data from the Pathosystems
282 Resource Integration Center (PATRIC)[47] and the Relational Sequencing TB Data
283 Platform (ReSeqTB)[45]. This results in 8000 isolates together with their resistant/sus-
284 ceptible status (label) for seven drugs, including five first-line drugs (rifampicin, isoniazid,
285 pyrazinamide, ethambutol, and streptomycin) and two second-line drugs (kanamycin and
286 ofloxacin) [34]. The short-read whole genome sequences of these 8000 isolates are down-
287 loaded from the European Nucleotide Archive [23] and the Sequence Read Archive [24].
288 The accession numbers used to obtain the data in our study were: ERP[000192, 006989,
289 008667, 010209, 013054, 000520], PRJEB[10385, 10950, 14199, 2358, 2794, 5162, 9680],
290 PRJNA[183624, 235615, 296471], and SRP[018402, 051584, 061066].

291 In order to map the raw sequence data to the reference genome, we use a method similar to
292 that used in previous work [7, 8]. We use the BWA software [25], specifically, the bwa-mem

293 program. We then call the single-nucleotide polymorphisms (SNPs) of each isolate with two
294 different pipelines, SAMtools [18] and GATK [39], and take the intersection of their calls to
295 ensure reliability. The final dataset, which includes the position as well as the reference and
296 alternative allele for each SNP [8], is used as the input to our classifier.

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

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
Oflloxacin	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

Table 1 Summary of number of isolates in our data

302 **3.2 Train-Test split**

303 To evaluate our classifier we use a stratified train-test split, where the training set contains
304 80% and the testing set contains 20% of data.

305 **3.3 AUROC comparison**

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

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

318 **3.4 Sensitivity at a fixed specificity**

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

2:10 An interpretable classification method for drug resistance

326 4 Results

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

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

336 Second, in Section 4.2, we compare the AUROC produced by our method for different limits
337 on rule size. This comparison between the method at different parameter values helps us
338 evaluate its ability to produce a simple model (i.e. a model with a fairly small rule size) with
339 a high AUROC. The simpler models are easier to understand for human users. In this paper,
340 we define the descriptiveness of a model by its simplicity (its rule size, i.e., the number of
341 SNPs needed to define it). In addition, we evaluate our method's sensitivity by comparing it
342 with LR and RF. To do so, we compute and compare the sensitivity of these three models at
343 a specificity near 90%. More specifically, this comparison uses the specificity level achieved
344 by the rule-based model that is closest to 90% (in practice, this is always between 88% and
345 92% for this dataset), since the rule-based model does not achieve every possible specificity
346 level when given a limit on rule size. For this evaluation, we limit model complexity by
347 setting a limit of 20 on the rule size.

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

352 4.1 Our models produce competitive AUROCs

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

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

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

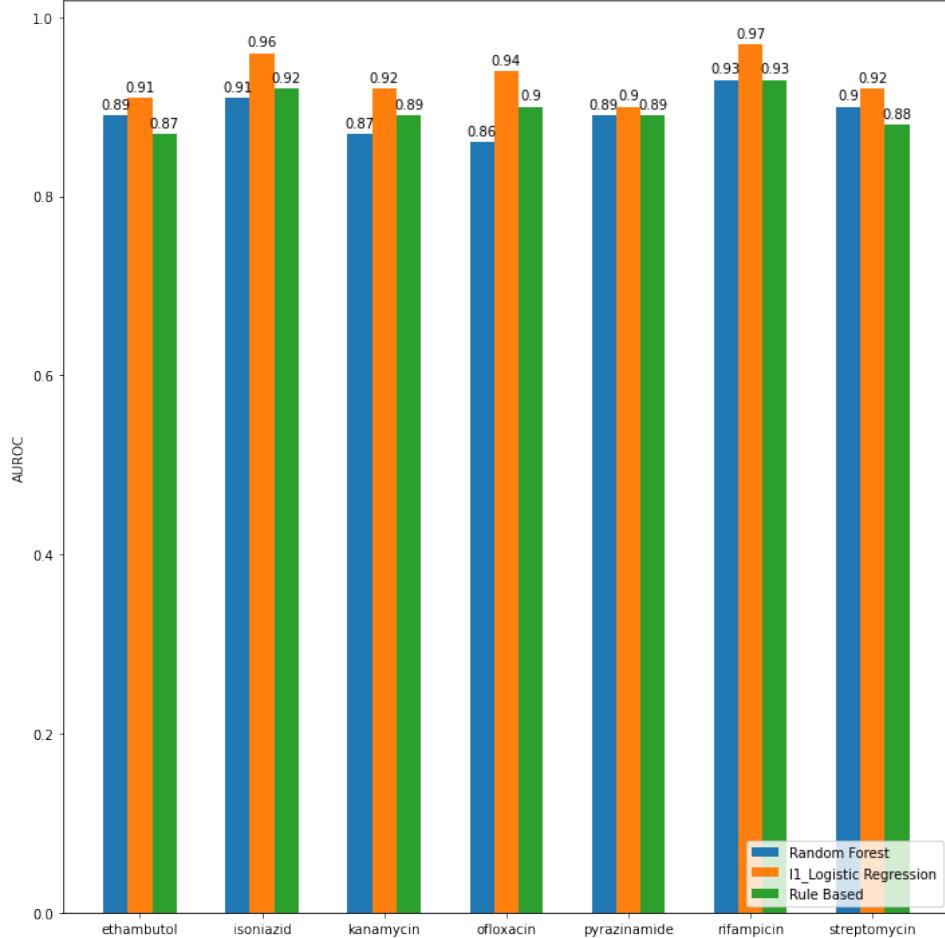
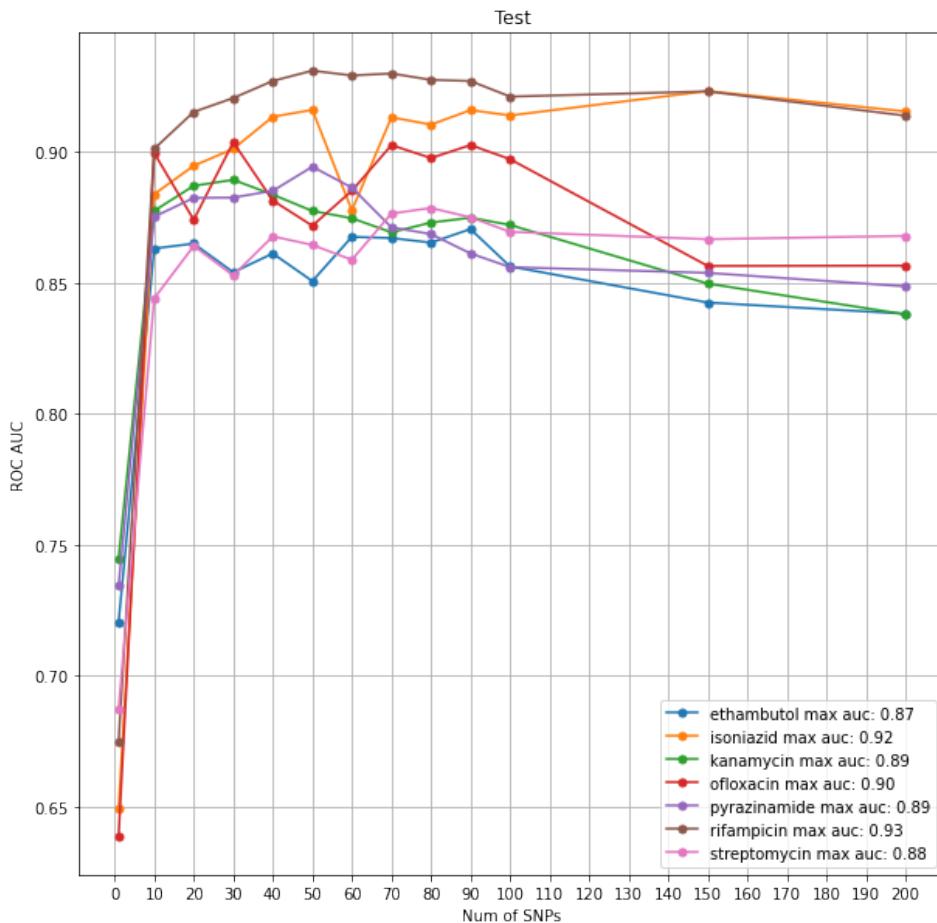


Figure 1 Comparison between the test AUROC of our rule-based model (with no limit imposed on the rule size), ℓ_1 -regularized logistic regression and Random Forest.

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

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

2:12 An interpretable classification method for drug resistance



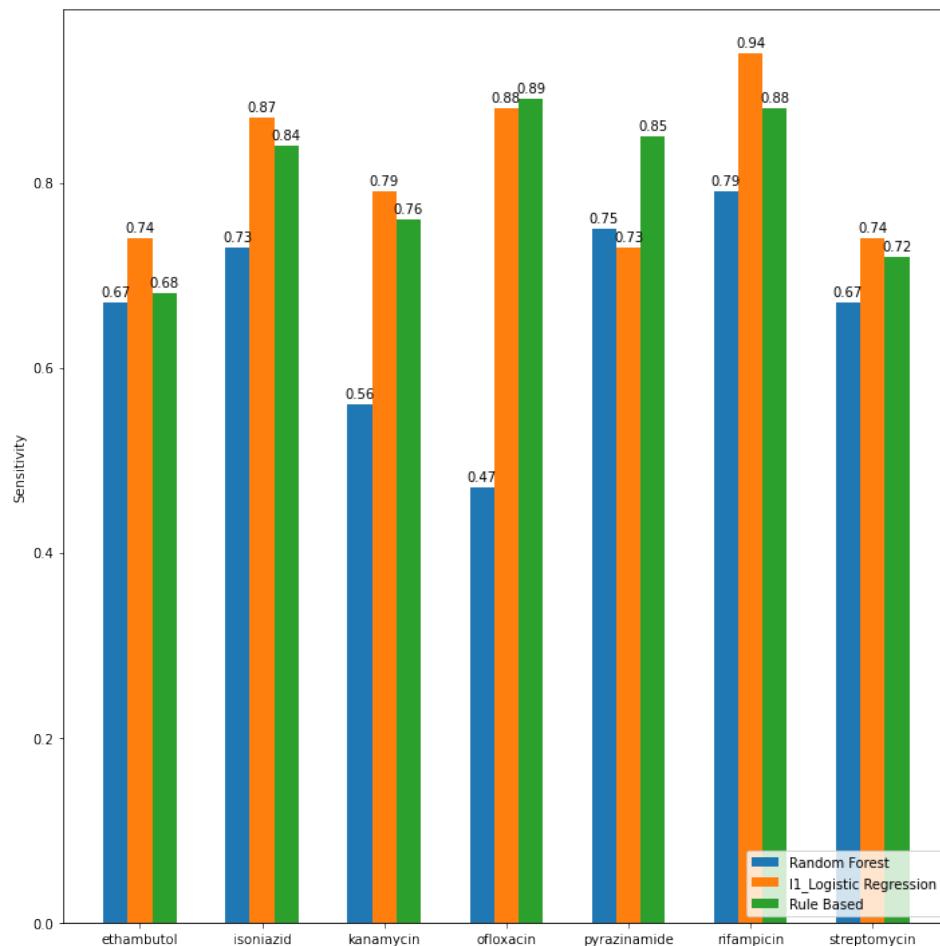
■ **Figure 2** Test AUROC for models trained on each drug with various rule size limits. Beyond a certain rule size, which varies with the drug, the AUROC of the predictive model no longer improves.

357 4.2 Our approach is able to produce simple models with high 358 AUROC

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

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

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



■ **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.

2:14 An interpretable classification method for drug resistance

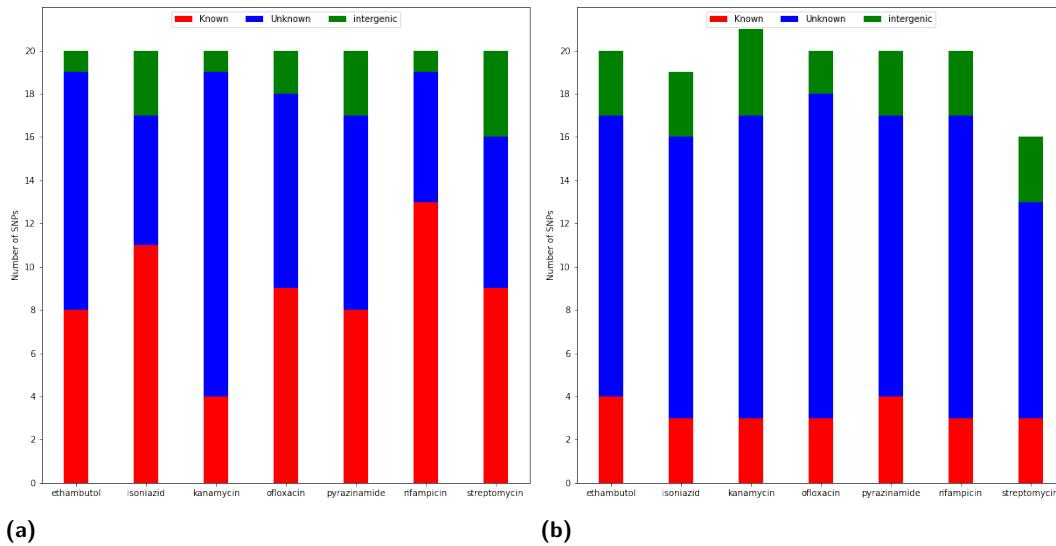


Figure 4 (a) The number of SNPs in genes with known association to drug resistance, genes without such an association, and intergenic regions, in our models with at most 20 SNPs and a specificity of $\geq 90\%$. (b) The same numbers for ℓ_1 -Logistic regression models with as close as possible to 20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

371 4.3 Our model uses genes previously associated to drug resistance

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

378 In Figure 4a we show the number of SNPs within different classes of genes found by our
379 approach with rule size ≤ 20 and specificity $\geq 90\%$, where each gene is classified according to
380 whether it has a known association to drug resistance (“known”) or not (“unknown”), with
381 an additional class for SNPs in intergenic regions. We show these numbers for ℓ_1 -Logistic
382 regression models with as close as possible to 20 non-zero regression coefficients in Figure 4b.
383 A comparison between these figures suggests that when both approaches are restricted to a
384 small number of features, our approach detects more relevant SNPs than ℓ_1 -logistic regression.
385 The list of “known” genes, selected through a data-driven and consensus-driven process by a
386 panel of experts, is the one in [31], containing 183 out of over 4,000 *M. tuberculosis* genes.
387 We note that in both cases, a group of SNPs in perfect linkage disequilibrium was coded as
388 “known” if at least one of the SNPs was contained in a known gene, “intergenic” if none of
389 them appeared in a gene, and as “unknown” otherwise.

390 4.4 Running time

391 We run our code on a cluster node with 2 CPU sockets, each with an 8-core 2.60 GHz Intel
392 Xeon E5-2640 v3 with 32 threads. The training of a single model with fixed hyper-parameters
393 takes between 1 and 8 minutes. This suggests that once a suitable value is chosen for the
394 hyper-parameters, the optimization used to determine the optimal rule can be performed

395 efficiently. Overall, producing the ROC curve for each drug takes between 3 and 18 hours,
396 depending on the number of labeled isolates available for each drug.

397 **5 Conclusion**

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

402 We used our method to produce classifiers for predicting drug resistance in *Mycobacterium*
403 *tuberculosis*. The classifiers' predictive accuracy was tested on a variety of antibiotics
404 commonly used for treating tuberculosis, including five first-line and two second-line drugs.
405 We show that our method could produce classifiers with a high AUROC, slightly less than that
406 of unrestricted ℓ_1 -Logistic regression, and comparable to Random Forest, as well as ℓ_1 -Logistic
407 regression restricted to a comparably small number of selected features for interpretability.
408 In addition, we show that our method is capable of producing accurate models with a rule
409 size small enough to keep the model understandable for human users. Finally, we show that
410 our approach can provide useful insights into its input data - in this case, it could help
411 identify genes associated with drug resistance.

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

420 We also note that the genes we define as having a known association to drug resistance are
421 not specific to the drug being tested, i.e. some of them may have been found to be associated
422 with the resistance to a drug other than the one being predicted. This is to be expected,
423 however, as the distinct resistance mechanisms are generally less numerous than antibiotics
424 [44]. It will be interesting to see whether methods such as ours are able to detect specific,
425 for instance, by testing it on data for newly developed antibiotics such as bedaquiline and
426 delamanid [22].

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

432 **References**

433 1 Matthew Aldridge, Oliver Johnson, Jonathan Scarlett, et al. Group testing: an information
434 theory perspective. *Foundations and Trends® in Communications and Information Theory*,
435 15(3-4):196–392, 2019.

2:16 An interpretable classification method for drug resistance

436 2 Gustavo Arango-Argoty, Emily Garner, Amy Pruden, Lenwood S Heath, Peter Vikesland, and
437 Liqing Zhang. DeepARG: a deep learning approach for predicting antibiotic resistance genes
438 from metagenomic data. *Microbiome*, 6(1):1–15, 2018.

439 3 G. K. Atia and V. Saligrama. Boolean compressed sensing and noisy group testing. *IEEE*
440 *Transactions on Information Theory*, 58(3):1880–1901, 2012.

441 4 P. Bradley, N. Gordon, T Walker, et al. Rapid antibiotic-resistance predictions from genome
442 sequence data for *Staphylococcus aureus* and *Mycobacterium tuberculosis*. *Nature Communications*,
443 6, 2015.

444 5 E. J. Candes and M. B. Wakin. An introduction to compressive sampling. *IEEE Signal*
445 *Processing Magazine*, 25(2):21–30, 2008.

446 6 Albert Cohen, Wolfgang Dahmen, and Ronald DeVore. Compressed sensing and best k -term
447 approximation. *Journal of the American mathematical society*, 22(1):211–231, 2009.

448 7 Francesc Coll, Ruth McNerney, José Afonso Guerra-Assunção, Judith R. Glynn, João Perdigão,
449 Miguel Viveiros, Isabel Portugal, Arnab Pain, Nigel Martin, and Taane G. Clark. A robust
450 snp barcode for typing *mycobacterium tuberculosis* complex strains. *Nature Communications*,
451 2014. URL: <https://doi.org/10.1038/ncomms5812>.

452 8 Wouter Deelder, Sofia Christakoudi, Jody Phelan, Ernest Diez Benavente, Susana Campino,
453 Ruth McNerney, Luigi Palla, and Taane Gregory Clark. Machine learning predicts accurately
454 *Mycobacterium tuberculosis* drug resistance from whole genome sequencing data. *Frontiers in*
455 *Genetics*, 10:922, 2019.

456 9 Alireza Doostan and Houman Owhadi. A non-adapted sparse approximation of PDEs with
457 stochastic inputs. *Journal of Computational Physics*, 230(8):3015–3034, 2011.

458 10 Robert Dorfman. The detection of defective members of large populations. *The Annals of*
459 *Mathematical Statistics*, 14(4):436–440, 1943.

460 11 Sorin Drăghici and R Brian Potter. Predicting HIV drug resistance with neural networks.
461 *Bioinformatics*, 19(1):98–107, 2003.

462 12 M. F. Duarte and Y. C. Eldar. Structured compressed sensing: From theory to applications.
463 *IEEE Transactions on Signal Processing*, 59(9):4053–4085, 2011.

464 13 Y.C. Eldar and G. Kutyniok. *Compressed Sensing: Theory and Applications*. Cambridge
465 University Press, 2012. URL: <https://books.google.ca/books?id=9ccLAQAAQBAJ>.

466 14 Coll F, McNerney R, Preston MD, et al. Rapid determination of anti-tuberculosis drug
467 resistance from whole-genome sequences. *Genome Med.*, 7:51, 2015.

468 15 Silke Feuerriegel, Viola Schleusener, Patrick Beckert, Thomas A. Kohl, Paolo Miotto, Daniela M.
469 Cirillo, Andrea M. Cabibbe, Stefan Niemann, and Kurt Fellenberg. PhyResSE: a web tool
470 delineating *Mycobacterium tuberculosis* antibiotic resistance and lineage from whole-genome
471 sequencing data. *Journal of Clinical Microbiology*, 53(6):1908–1914, 2015.

472 16 S. Foucart and H. Rauhut. *A Mathematical Introduction to Compressive Sensing*. Applied
473 and Numerical Harmonic Analysis. Springer New York, 2013. URL: <https://books.google.ca/books?id=zb28BAAQBAJ>.

475 17 Sébastien Gagneux. Ecology and evolution of *Mycobacterium tuberculosis*. *Nat Rev Microbiol*,
476 16:202–213, 2018.

477 18 Li H1, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin
478 R, and 1000 Genome Project Data Processing Subgroup. The sequence alignment/map format
479 and samtools. *Bioinformatics*, 25(16):2078–2079, 2009.

480 19 Matthew A Herman and Thomas Strohmer. High-resolution radar via compressed sensing.
481 *IEEE transactions on signal processing*, 57(6):2275–2284, 2009.

482 20 IBM. IBM ILOG CPLEX Optimization Studio V12.10.0 documentation. <https://www.ibm.com/support/knowledgecenter/SSSA5P>, 2020.

484 21 H. Iwai, M. Kato-Miyazawa, T Kirikae, and T. Miyoshi-Akiyama. CASTB (the comprehensive
485 analysis server for the *Mycobacterium tuberculosis* complex): A publicly accessible web server
486 for epidemiological analyses, drug-resistance prediction and phylogenetic comparison of clinical
487 isolates. *Tuberculosis*, pages 843–844, 2015.

488 22 Suha Kadura, Nicholas King, Maria Nakhoul, Hongya Zhu, Grant Theron, Claudio U Köser,
489 and Maha Farhat. Systematic review of mutations associated with resistance to the new and
490 repurposed *Mycobacterium tuberculosis* drugs bedaquiline, clofazimine, linezolid, delamanid
491 and pretomanid. *Journal of Antimicrobial Chemotherapy*, 05 2020. dkaa136.

492 23 Rasko Leinonen, Ruth Akhtar, Ewan Birney, Lawrence Bower, Ana Cerdeno-Tárraga, et al.
493 The European Nucleotide Archive. *Nucleic Acids Research*, 39:D28–31, 2011.

494 24 Rasko Leinonen, Hideaki Sugawara, Martin Shumway, and International Nucleotide Se-
495 quence Database Collaboration. The sequence read archive. *Nucleic acids research*,
496 39(suppl_1):D19–D21, 2010.

497 25 H Li. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv*,
498 2013.

499 26 Michael Lustig, David Donoho, and John M Pauly. Sparse MRI: The application of compressed
500 sensing for rapid MR imaging. *Magnetic Resonance in Medicine: An Official Journal of the
501 International Society for Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007.

502 27 D. Malioutov and M. Malyutov. Boolean compressed sensing: LP relaxation for group testing.
503 In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*,
504 pages 3305–3308, 2012.

505 28 Dmitry Malioutov and Kush Varshney. Exact rule learning via Boolean compressed sensing.
506 In *International Conference on Machine Learning*, pages 765–773, 2013.

507 29 L Mathelin and KA Gallivan. A compressed sensing approach for partial differential equations
508 with random input data. *Communications in computational physics*, 12(4):919–954, 2012.

509 30 Arya Mazumdar. On almost disjunct matrices for group testing. In Kun-Mao Chao, Tsan-
510 sheng Hsu, and Der-Tsai Lee, editors, *Algorithms and Computation*, pages 649–658, Berlin,
511 Heidelberg, 2012. Springer Berlin Heidelberg.

512 31 Paolo Miotto, Belay Tessema, Elisa Tagliani, Leonid Chindelevitch, et al. A standardised
513 method for interpreting the association between mutations and phenotypic drug-resistance in
514 *Mycobacterium tuberculosis*. *European Respiratory Journal*, 50(6), 2017.

515 32 W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Interpretable
516 machine learning: definitions, methods, and applications. *arXiv*, 2019.

517 33 Balas Kausik Natarajan. Sparse approximate solutions to linear systems. *SIAM journal on
518 computing*, 24(2):227–234, 1995.

519 34 Tra-My Ngo and Yik-Ying Teo. Genomic prediction of tuberculosis drug-resistance: bench-
520 marking existing databases and prediction algorithms. *BMC Bioinformatics*, 20(1):68, 2019.

521 35 Jim O'Neill. Antimicrobial resistance: Tackling a crisis for the health and wealth of nations.
522 Technical report, Review on Antimicrobial Resistance, 2014. URL: <https://amr-review.org/Publications.html>.

2:18 An interpretable classification method for drug resistance

524 36 World Health Organization. Antimicrobial resistance: global report on surveillance. Technical
525 report, WHO, 2014.

526 37 World Health Organization. Global tuberculosis report 2019. Technical report, WHO, 2019.

527 38 F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel,
528 P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher,
529 M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine
530 Learning Research*, 12:2825–2830, 2011.

531 39 Ryan Poplin, Valentín Ruano-Rubio, Mark A. DePristo, Tim J. Fennell, Mauricio O. Carneiro,
532 Geraldine A. Van der Auwera, David E. Kling, Laura D. Gauthier, Ami Levy-Moonshine,
533 David Roazen, Khalid Shakir, Joel Thibault, Sheila Chandran, Chris Whelan, Monkol Lek,
534 Stacey Gabriel, Mark J Daly, Ben Neale, Daniel G. MacArthur, and Eric Banks. Scaling
535 accurate genetic variant discovery to tens of thousands of samples. *bioRxiv*, 2017.

536 40 Mario C Raviglione and Ian M Smith. XDR tuberculosis—implications for global public health.
537 *New England Journal of Medicine*, 356(7):656–659, 2007.

538 41 Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should i trust you?" Explaining
539 the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International
540 Conference on Knowledge Discovery and Data Mining*, pages 1135–1144, 2016.

541 42 James Emmanuel San, Shakuntala Baichoo, Aquillah Kanzi, Yumna Moosa, Richard Lessells,
542 Vagner Fonseca, John Mogaka, Robert Power, and Tulio de Oliveira. Current affairs of
543 microbial genome-wide association studies: Approaches, bottlenecks and analytical pitfalls.
544 *Frontiers in Microbiology*, 10:3119, 2020. URL: <https://www.frontiersin.org/article/10.3389/fmicb.2019.03119>.

546 43 V. Schleusener, C. Köser, P. Beckert, et al. Mycobacterium tuberculosis resistance prediction
547 and lineage classification from genome sequencing: comparison of automated analysis tools.
548 *Scientific Reports*, 7, 2017.

549 44 Almeida Da Silva, Pedro Eduardo, Palomino, and Juan Carlos. Molecular basis and mechanisms
550 of drug resistance in Mycobacterium tuberculosis: classical and new drugs. *Journal of
551 Antimicrobial Chemotherapy*, 66(7):1417–1430, 05 2011.

552 45 Angela M Starks, Enrique Avilés, Daniela M Cirillo, Claudia M Denkinger, David L Dolinger,
553 Claudia Emerson, Jim Gallarda, Debra Hanna, Peter S Kim, Richard Liwski, et al. Collaborative
554 effort for a centralized worldwide tuberculosis relational sequencing data platform.
555 *Clinical Infectious Diseases*, 61(suppl_3):S141–S146, 2015.

556 46 A Steiner, D Stucki, M Coscolla, S Borrell, and S Gagneux. KvarQ: targeted and direct variant
557 calling from fastq reads of bacterial genomes. *BMC Genomics*, 15, 2014.

558 47 Alice R Wattam, David Abraham, Oral Dalay, Terry L Disz, Timothy Driscoll, Joseph L
559 Gabbard, Joseph J Gillespie, Roger Gough, Deborah Hix, Ronald Kenyon, et al. PATRIC, the
560 bacterial bioinformatics database and analysis resource. *Nucleic acids research*, 42(D1):D581–
561 D591, 2014.

562 48 Yang Yang, Katherine E Niehaus, Timothy M Walker, Zamin Iqbal, A Sarah Walker, Daniel J
563 Wilson, Tim EA Peto, Derrick W Crook, E Grace Smith, Tingting Zhu, et al. Machine
564 learning for classifying tuberculosis drug-resistance from DNA sequencing data. *Bioinformatics*,
565 34(10):1666–1671, 2018.