Predictive Computational Phenotyping and Biomarker Discovery Using Reference-Free Genome Comparisons

Alexandre Drouin[1,*], **Sébastien Giguère**[2], **Maxime Déraspe**[3], **Mario Marchand**[1,4], **Michael Tyers**[2], **Vivian G. Loo**[5], **Anne-Marie Bourgault**[5], **François Laviolette**[1,4,†], **Jacques Corbeil**[3,4,†]

[1] Department of Computer Science and Software Engineering, Université Laval, Québec, Canada
[2] Institute for Research in Immunology and Cancer, Université de Montréal, Montréal, Canada
[3] Department of Molecular Medicine, Université Laval, Québec, Canada
[4] Big Data Research Centre, Université Laval, Québec, Canada
[5] Department of Microbiology, McGill University Health Centre, Montréal, Canada

Abstract

The identification of genomic biomarkers is a key step towards improving diagnostic tests and therapies. We present a new reference-free method for this task that relies on a k-mer representation of genomes and a machine learning algorithm that produces intelligible models. The method is computationally scalable and well-suited for whole genome sequencing studies. The method was validated by generating models that predict the antibiotic resistance of *C. difficile*, *M. tuberculosis*, *P. aeruginosa* and *S. pneumoniae*. We show that the obtained models are accurate and that they highlight biologically relevant biomarkers, while providing insight into the process of antibiotic resistance acquisition.

Background

Despite an era of supercomputing and increasingly precise instrumentation, many biological phenomena remain misunderstood. For example, phenomena such as the development of some cancers, or the lack of efficiency of a treatment on an individual, still puzzle researchers. One approach to understanding such events is the elaboration of *case-control* studies, where a group of individuals that exhibit a given biological state (phenotype) is compared to a group of individuals that do not. In this setting, one seeks biological characteristics (biomarkers), that are predictive of the phenotype of interest. Such biomarkers can serve as the basis for diagnostic tests, or they can guide the development of new therapies or drug treatments by providing insight on the biological processes that underlie a phenotype (Azuaje, 2011; Koboldt et al., 2013; Mbianda et al., 2015; Simon, 2011). With the help of computational tools, such studies can be conducted at a much larger scale and thus, produce more significant results.

In this work, we focus on the identification of genomic biomarkers. These include any genomic variation, from single nucleotide substitutions and indels, to large scale genomic rearrangements. With the increasing throughput and decreasing cost of DNA sequencing, it is now possible to search for such biomarkers in the whole genomes of a large set of individuals (Koboldt et al., 2013; van Dijk et al., 2014). This motivates the need for computational tools that can cope with large amounts of genomic data and identify the subtle variations that are biomarkers of a phenotype.

Genomic biomarker discovery relies on multiple genome comparisons. Genomes are usually compared based on a set of single nucleotide polymorphisms (SNP) (Brookes, 1999; Koboldt et al., 2013; Nielsen et al., 2011). A SNP exists at a single base pair location in the genome when a variation occurs within a population. The identification of SNPs relies on multiple sequence alignment, which is computationally expensive and can produce inaccurate results in the presence of large-scale genomic rearrangements, such as gene insertions, deletions, duplications, inversions, or translocations (Bonham-Carter et al., 2014; Leimeister et al., 2014; Song et al., 2014; Vinga & Almeida, 2003; Vinga, 2007).

Recently, methods for genome comparison that alleviate the need for multiple sequence alignment, i.e., reference-free genome comparison, have been investigated (Bonham-Carter et al., 2014; Leimeister et al., 2014; Song et al., 2014; Vinga & Almeida, 2003; Vinga, 2007). In this work,

* Correspondence: alexandre.drouin.8@ulaval.ca
† Shared senior authorship

we use such an approach, by comparing genomes based on the k-mers, i.e., sequences of k nucleotides, that they contain. The main advantage of this method is that it is robust to genomic rearrangements. Moreover, it provides a fully unbiased way to compare genomic samples and identify genomic variations that are associated with a phenotype. However, this representation of the genomes is far less compact than a set of SNPs and thus poses additional computational challenges.

Our objective is thus to find the most concise set of genomic features (k-mers) that allows to accurately predict the phenotype (Azuaje, 2011). Including uninformative or redundant biomarkers in this set would lead to additional validation costs and could mislead researchers. In this work, we favor an approach based on machine learning, where we seek a computational model of the phenotype that is accurate and sparse, i.e. that relies on the fewest genomic features. Learning such models from a large data representations, such as the k-mer representation, is a challenging problem (Hastie et al., 2013). Indeed, in this setting, there are much more genomic features than genomes, which increases the danger of overfitting, i.e., learning random noise patterns that lead to poor generalization performance. In addition, the majority of the k-mers are uninformative and cannot be used to predict the phenotype. Finally, due to the structure of genomes, for example, genes and chromosomes, many k-mers occur simultaneously and are thus highly correlated.

Previous work in the field of biomarker discovery has mostly combined feature selection and predictive modeling methods (Azuaje, 2011; Saeys et al., 2007). Feature selection serves to identify features that are associated with the phenotype. These features are then used to construct a predictive model with the hope that it can accurately predict the phenotype. The most widespread approach consists in measuring the association between the features and the phenotype with a statistical test, such as the χ^2 test or a t-test. Then, features that are not deemed associated are discarded and the rest are passed down to a modeling algorithm. In the machine learning literature, such methods are referred to as *filter methods* (Guyon & Elisseeff, 2003; Hastie et al., 2013).

When considering millions of features, it is not possible to efficiently perform multivariate statistical tests. Hence, filter methods are limited to univariate statistical tests. While univariate filters are highly scalable, they discard multivariate patterns in the data, that is, combinations of features that are, together, predictive of the phenotype. Moreover, the feature selection is performed independently of the modeling, which can lead to a suboptimal choice of features. To address these limitations, *embedded methods* integrate the feature selection as part of the learning algorithm (Guyon & Elisseeff, 2003; Saeys et al., 2007). These methods select features based on their ability to compose an accurate predictive model of the phenotype. Moreover, some of these methods, such as the Set Covering Machine (Marchand & Shawe-Taylor, 2002), can consider multivariate interactions between features.

In this study, we propose to apply the Set Covering Machine (SCM) algorithm to genomic biomarker discovery. We devise extensions to this algorithm that make it well suited for learning from extremely large sets of genomic features. We combine this algorithm with the k-mer representation of genomes and show that the method produces uncharacteristically sparse models, which explicitly highlight the relationship between genomic variations and the phenotype of interest. We present statistical guarantees on the accuracy of the models obtained using this approach. Moreover, we propose an efficient implementation of the method, which can readily scale to large genomic datasets containing thousands of individuals and hundreds of millions of k-mers.

We used our method to model the antibiotic resistance of four common human pathogens, including Gram-negative and Gram-positive bacteria. Antibiotic resistance is a growing public health concern, as many multidrug-resistant bacterial strains are starting to emerge. This compromises our ability to treat common infections, which increases mortality and health care costs (World Health Organization, 2014). Better computational methodologies to assess resistance phenotypes will assist in tracking epidemics, improve diagnosis, enhance treatment, and facilitate the development of new drugs. Our study highlights that, with whole genome sequencing and machine learning algorithms, such as the SCM, we can readily zero in on the genes, mutations, and processes responsible for antibiotic resistance and other phenotypes of interest.

Machine Learning for Biomarker Discovery

The problem of distinguishing two groups of living organisms based on their genomes can be formalized as a *supervised learning* problem. In this setting, we assume that we are given a data sample $\mathcal{S}$ that contains *learning examples*. These examples are pairs $(\mathbf{x}, y)$, where $\mathbf{x}$ is a genome and y is a label that corresponds to one of two possible phenotypes. More specifically, we assume that $\mathbf{x} \in \{A, T, G, C\}^*$, which corresponds to the set of all possible strings of nucleotides and that $y \in \{0, 1\}$. In this work, we assign the label $y = 1$ to the case group and $y = 0$ to the control group. The examples in $\mathcal{S}$ are assumed to be drawn independently from an unknown, but fixed, data generating distribution D. Hence $\mathcal{S} \stackrel{\text{def}}{=} \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_m, y_m)\} \sim D^m$.

Most learning algorithms are designed to learn from a vector representation of the data. To learn from genomes, we must define a function $\boldsymbol{\phi} : \{A, T, G, C\}^* \rightarrow \mathbb{R}^d$, that takes a genome as input and maps it to some d dimensional vector space known as the feature space. We choose to represent each genome by the presence or absence of every possible k-mer. This representation is detailed in the *Methods* section.

Subsequently, a learning algorithm can be applied to the set $S' \stackrel{\text{def}}{=} \{(\boldsymbol{\phi}(\mathbf{x}_1), y_1), ..., (\boldsymbol{\phi}(\mathbf{x}_m), y_m)\}$ to obtain a model $h : \mathbb{R}^d \rightarrow \{0, 1\}$. The model is a function that, given the feature representation of a genome, estimates the associated phenotype. The objective is to obtain a model h that has a good generalization performance, i.e., that minimizes the probability $R(h)$ of making a prediction error for any example drawn according to the distribution D, where

$$R(h) \stackrel{\text{def}}{=} \underset{(\mathbf{x},y)\sim D}{Pr}[h(\boldsymbol{\phi}(\mathbf{x})) \neq y]. \quad (1)$$

Application Specific Constraints

Biomarker discovery leads to two additional constraints on the model h. These are justified by the cost of applying the model in practice and on the ease of interpretation of the model by domain experts.

First, we strive for a model that is sparse, i.e., that uses a minimal set of features to predict the phenotype. This property is important, as it can greatly reduce the cost of applying the model in practice. For example, if the model relies on a sufficiently small number of features, these can be measured by using alternative methods, e.g., PCR amplification, rather than sequencing entire genomes.

In addition, the model must be easily interpretable by domain experts. This is essential for extracting useful biological information from the data, to facilitate comprehension, and is critical for adoption by the scientific community. We make two observations in an attempt to obtain a clear definition of interpretability. The first is that the structure of a model can affect its interpretability. For example, rule-based models, such as decision trees (Breiman et al., 1984), are naturally understood as their predictions consist in answering a series of questions; effectively following a path in the tree. In contrast, linear models, such as those obtained with Support Vector Machines (Cortes & Vapnik, 1995) or Neural Networks (Cheng & Titterington, 1994), are complex to interpret, as their predictions consist in computing linear combinations of features. The second observation is that, regardless of the structure of the model, sparsity is also essential to its interpretability. Indeed, models with many rules will inevitably be harder to interpret.

The Set Covering Machine

The SCM (Marchand & Shawe-Taylor, 2002) is a greedy learning algorithm that produces uncharacteristically sparse rule-based models. In our case, these rules are individual units that detect the presence or the absence of a k-mer in a genome. These rules are boolean-valued, i.e., they can either output true or false. The models learnt by the SCM are logical combinations of such rule, which can be conjunctions (logical-AND) or disjunctions (logical-OR). To make a prediction, each rule in the model is evaluated against the genome of interest. Then, the results are aggregated to obtain the model's prediction. A conjunction model assigns the positive class to a genome if *all* the rules output true, whereas a disjunction model does the same if *at least one* rule outputs true.

The time required for learning a model with the SCM algorithm grows linearly with the number of genomes in the dataset and with the number of k-mers under consideration. This makes it particularly well suited for learning from large genomic datasets. Moreover, as it will be discussed later, we have developed an efficient implementation of the SCM, which can readily scale to hundreds of millions of k-mers and thousands of genomes, while requiring a few gigabytes of memory. This is achieved by keeping the data on external storage, e.g., a hard drive, and accessing it in small contiguous blocks. This is in sharp contrast with other learning algorithms, which require that the entire dataset be stored in the computer's memory.

The SCM algorithm is detailed in the Appendix. In the *Methods* section, we propose algorithmic and theoretical extensions to the SCM algorithm that make it well suited genomic biomarker discovery.

Results

Data

We obtained antibiotic resistance datasets for four bacteria: *Clostridium difficile*, *Mycobacterium tuberculosis*, *Pseudomonas aeruginosa* and *Streptococcus pneumoniae*. These datasets consisted in whole genome sequencing reads, as well as susceptibility data for multiple antibiotics. The *P. aeruginosa*, *M. tuberculosis* and *S. pneumoniae* datasets were respectively obtained from Kos et al. (Kos et al., 2015), Merker et al. (Merker et al., 2015) and Croucher et al. (Croucher et al., 2013). The *C. difficile* dataset was obtained from Dr. Vivian G. Loo and Dr. Anne-Marie Bourgault. The *C. difficile* genomes were submitted to the European Nucleotide Archive [EMBL:PRJEB11776] and the related antibiotic resistance data are provided as supplementary material.

All genomes were assembled and subsequently split into

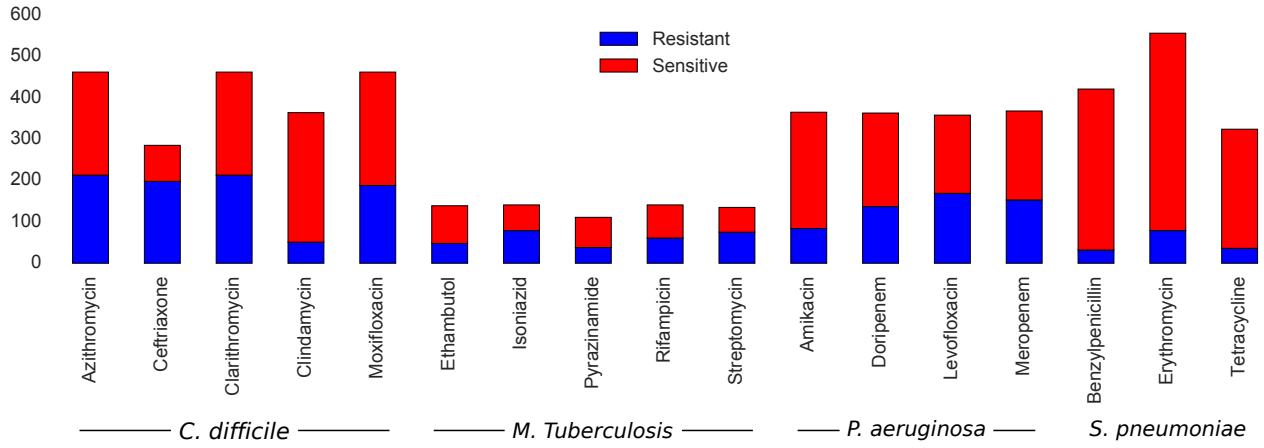

Figure 1. Distribution of resistant and sensitive isolates in each dataset.

k-mers. Motivation for working with assembled genomes rather than with raw reads, is given in the *Methods* section. We used k-mers of length 31, as this value is often used for bacterial genome assembly (Boisvert et al., 2012). Moreover, we investigated different k-mer lengths, ranging from $k = 11$ to $k = 91$, and found no significant variation in the accuracy of the obtained models (Appendix, Table S5).

We considered each (pathogen, antibiotic) combination individually, yielding 17 datasets in which the number of examples (m) ranged from 111 to 556 and the number of k-mers ($|\mathcal{K}|$) ranged from 10 to 123 millions. Figure 1 shows the distribution of resistant and sensitive isolates in each dataset. Further details about the datasets are provided as supporting information (Appendix, Table S1).

The SCM Models are Sparse and Accurate

We empirically compared the generalization performance and the sparsity of the models obtained using the SCM, Linear Support Vector Machines (SVM) (Cortes & Vapnik, 1995) and the CART decision tree algorithm (Breiman et al., 1984). CART and SVM are state-of-the-art machine learning algorithms and have been abundantly used in biological applications (Kingsford & Salzberg, 2008; Noble, 2006). For the SVM, we considered both L_1 regularization (L1SVM), which is known to yield sparse models (Hastie et al., 2013), and L_2 regularization (L2SVM). We used the SVM implementation from LIBLINEAR (Fan et al., 2008) and the CART implementation from Scikit-learn (Pedregosa et al., 2011), as these implementations are highly optimized and widely used among the machine learning community.

Our implementation of the SCM was able to efficiently process all the k-mers from the genomes, that is, the entire feature space. The time required for one training of the algorithm varied between 33 seconds and 2 hours depending on the bacterium, and the memory requirements were always inferior to 8 gigabytes. However, for SVM and CART, the entire dataset had to be placed in the computer's memory, which generated massive memory requirements. Hence, the dimensionality of the feature space had to be reduced prior to applying these learning algorithms. For these algorithms, we filtered the features using a univariate filter (Azuaje, 2011; Guyon & Elisseeff, 2003; Saeys et al., 2007). More specifically, we measured the association between each feature and the phenotype using a χ^2 test of independence. Then, we retained the 1 000 000 features that were most associated with the phenotype.

For each antibiotic resistance dataset, we randomly split the data into a training set $\mathcal{S}$ ($2/3$ of the data) and a testing set $\mathcal{T}$ ($1/3$ of the data). Then, we trained each algorithm on $\mathcal{S}$ and evaluated its generalization performance on $\mathcal{T}$. The hyperparameter values, i.e., tunable parameters of the algorithm, were set by 5-fold cross-validation on $\mathcal{S}$ (see (Hastie et al., 2013)). We repeated this procedure 10 times, each time on a different random split, $\mathcal{S}$ and $\mathcal{T}$, of the data.

For each bacterium/antibiotic combination, the average testing set error rate and number of rules in the model over the 10 repetitions are shown in Table 1. The standard deviations, as well as the sensitivities and specificities are provided as supporting information (Appending, Tables S2, S3 and S4). The algorithms were also compared to a baseline method, which always predicts the most abundant class in the training set (resistant or sensitive).

The SCM tends to learn sparser models than the other algorithms (CART: $p = 0.003$, L1SVM: $p = 0.0003$, L2SVM: $p = 0.0003$)[1]. In terms of error rate, all the algorithms surpass the baseline method. This means that relevant information about antibiotic resistance was found in the

[1]All the p-values reported in this study were obtained by applying a Wilcoxon signed-rank test (Wilcoxon, 1945).

Table 1. Comparison of the Set Covering Machine (SCM), the CART algorithm (CART), L_1 and L_2 regularized Support Vector Machines (L1SVM, L2SVM) and the baseline, which predicts the most abundant class in the dataset. The prefix χ^2 indicates that a univariate filter used. The values are average error rates and number of k-mers in the model (in parenthesis) for 10 repetitions of the experiment. The best error rates are in bold.

Dataset	SCM	χ^2 + CART	χ^2 + L1SVM	χ^2 + L2SVM	Baseline
C. difficile					
Azithromycin	**0.030** (3.3)	0.086 (7.2)	0.064 (20326.0)	0.056 (10^6)	0.406
Ceftriaxone	**0.073** (2.6)	0.117 (6.8)	0.087 (8114.1)	0.102 (10^6)	0.250
Clarithromycin	**0.011** (3.0)	0.070 (8.0)	0.062 (36686.1)	0.059 (10^6)	0.406
Clindamycin	0.021 (1.4)	0.011 (2.0)	**0.009** (598.2)	0.021 (10^6)	0.123
Moxifloxacin	**0.020** (1.0)	**0.020** (1.3)	**0.020** (25.6)	0.048 (10^6)	0.374
M. tuberculosis					
Ethambutol	0.179 (1.4)	0.185 (1.9)	**0.153** (201.3)	0.221 (10^6)	0.298
Isoniazid	0.021 (1.0)	0.021 (1.1)	**0.017** (104.7)	0.125 (10^6)	0.354
Pyrazinamide	**0.318** (3.1)	0.371 (4.4)	0.353 (481.2)	0.342 (10^6)	0.395
Rifampicin	**0.031** (1.4)	**0.031** (1.5)	**0.031** (130.0)	0.196 (10^6)	0.458
Streptomycin	0.050 (1.0)	0.052 (1.6)	**0.043** (98.8)	0.137 (10^6)	0.435
P. aeruginosa					
Amikacin	0.175 (4.9)	0.206 (14.1)	0.187 (11514.6)	**0.164** (10^6)	0.172
Doripenem	0.270 (1.4)	**0.261** (1.9)	**0.261** (950.0)	0.275 (10^6)	0.311
Levofloxacin	**0.072** (1.2)	0.076 (1.0)	0.085 (148.9)	0.212 (10^6)	0.417
Meropenem	0.267 (1.6)	**0.261** (1.0)	0.328 (5368.5)	0.327 (10^6)	0.374
S. pneumoniae					
Benzylpenicillin	0.013 (1.1)	0.012 (2.3)	**0.011** (124.9)	0.013 (10^6)	0.064
Erythromycin	**0.037** (2.0)	0.047 (3.8)	0.041 (328.8)	0.042 (10^6)	0.118
Tetracycline	0.031 (1.1)	**0.029** (1.2)	0.032 (1108.5)	0.037 (10^6)	0.101

genomes. The error rate of the SCM is smaller or equal to the one of CART on $12/17$ dataset ($p = 0.074$), L1SVM on $11/17$ datasets ($p = 0.179$) and L2SVM on $16/17$ datasets ($p = 0.001$). These results suggest that, in addition to producing extremely sparse models, the SCM compares favorably, in terms of error rate, to these state-of-the-art learning algorithms.

Multivariate Genomic Patterns Are Essential

To demonstrate the importance of considering multivariate interactions between k-mers, we compared the SCM to a variant which uses a univariate filter as a preprocessing step (χ^2 + SCM). The same experimental protocol was used. The results are detailed in Table 2. In terms of sparsity, the SCM learns sparser models than the χ^2 + SCM ($p = 0.001$). This suggests that the χ^2 + SCM adds more rules to its models in order to compensate for the removal of multivariate patterns. In terms of generalization performance, the SCM produces models with smaller error rates than the χ^2 + SCM ($p = 0.054$). This suggests that the use of univariate hypothesis testing to select a subset of k-mers that are potentially associated with the phenotype is detrimental to the accuracy and sparsity of the models. Thus, being able to consider all the features without the need for filtering is an important property of the SCM algorithm.

Table 2. Comparison of the Set Covering Machine (SCM) and a variant which performs univariate feature selection prior to learning (χ^2 + SCM). The values are average error rates and number of k-mers in the model (in parenthesis) for 10 repetitions of the experiment. The best error rates are in bold.

Dataset	SCM	χ^2 + SCM
C. difficile		
Azithromycin	**0.030** (3.3)	0.075 (3.0)
Ceftriaxone	**0.073** (2.6)	0.111 (3.2)
Clarithromycin	**0.011** (3.0)	0.069 (3.5)
Clindamycin	0.021 (1.4)	**0.008** (2.3)
Moxifloxacin	**0.020** (1.0)	0.021 (1.1)
M. tuberculosis		
Ethambutol	0.179 (1.4)	**0.174** (3.2)
Isoniazid	**0.021** (1.0)	**0.021** (1.2)
Pyrazinamide	**0.318** (3.1)	0.366 (5.8)
Rifampicin	0.031 (1.4)	**0.029** (1.3)
Streptomycin	**0.050** (1.0)	**0.050** (2.1)
P. aeruginosa		
Amikacin	0.175 (4.9)	**0.164** (9.7)
Doripenem	**0.270** (1.4)	0.307 (8.5)
Levofloxacin	**0.072** (1.2)	0.083 (3.5)
Meropenem	**0.267** (1.6)	0.331 (9.1)
S. pneumoniae		
Benzylpenicillin	**0.013** (1.1)	**0.013** (1.3)
Erythromycin	**0.037** (2.0)	0.041 (5.1)
Tetracycline	**0.031** (1.1)	0.033 (2.2)

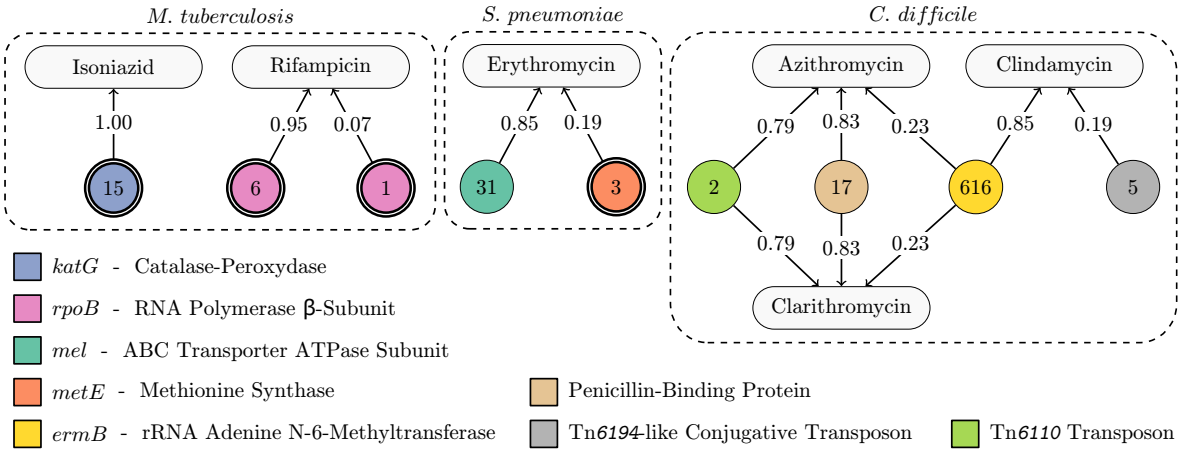

Figure 2. Antibiotic resistance models: Six antibiotic resistance models, which are all disjunctions (logical-OR). The rounded rectangles correspond to antibiotics. The circular nodes correspond to k-mer rules. A single border indicates a presence rule and a double border indicates an absence rule. The numbers in the circles show to the number of equivalent rules. A rule is connected to an antibiotic if it was included in its model. The weight of the edges gives the importance of each rule as defined by Equations (2) and (3).

The SCM Models are Biologically Relevant

We investigated the biological relevance of the obtained models. To achieve this, we retrained the algorithm on each dataset, using all the available data for training. This yielded a single phenotypic model for each dataset. For each model, we extracted the k-mer sequences associated to each rule, as well as its equivalent rules, i.e., the rules that the SCM found to be equally predictive of the phenotype (*Methods*).

The equivalent rules are not used for prediction, but they can be used to obtain insight on the type of genomic variation that was identified by the algorithm. For example, a small number of rules targeting k-mers that only share a single or few nucleotides, suggests a point mutation. Alternatively, a large number of rules, that target k-mers which can be assembled to form a long sequence, suggests a large-scale genomic variation, such as a gene acquisition or deletion.

Then, we annotated each k-mer sequence by using Nucleotide Blast (Altschul et al., 1990) to search it against a set of annotated genomes. The annotated models for each dataset are illustrated in the Appendix, Figure S1. Below, we interpret and discuss a subset of these models, which are illustrated in Figure 2. For each genomic variation identified by the algorithm, a thorough literature review was performed, with the objective of finding known, and validated, associations to antibiotic resistance.

For *M. tuberculosis*, the model for isoniazid resistance contains a single rule, which targets the catalase-peroxidase enzyme encoded by the *katG* gene. This enzyme is responsible for activating isoniazid, a prodrug, into its toxic form. Mutations S315T, S315G, S315I and S315N in the substrate access channel are all known to alter its catalytic properties (Cade et al., 2010), resulting in resistance to isoniazid (Da Silva & Palomino, 2011). As illustrated in Figure 3, the k-mer sequences of the rule found by the SCM, as well as its 14 equivalent rules, all overlap codon 315, indicating that mutations in this region are associated with resistance. This demonstrates that, even if k-mers do not include positional information, our method can pinpoint, *de novo*, the location of point mutations out of millions of base pairs. Interestingly, due to the presence of multiple mutations at this position, the algorithm opted for rules capturing the absence of the wild type sequence. This effectively includes the presence of any mutation at this position and leads to a simpler and more robust model than a disjunction containing a rule for all possible mutation at this site.

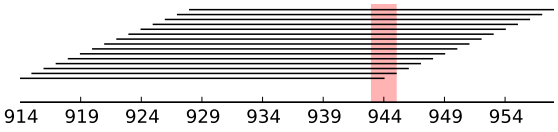

Figure 3. Patterns in equivalent k-mers suggest the type of genomic variation: Base pairs of the *katG* gene that are covered by the rule in the isoniazid model and its equivalent rules. The black lines indicate the positions of the k-mers. The red box indicates the position of codon 315, an isoniazid resistance determinant.

The model for rifampicin resistance contains two rules, which target the rifampicin resistance-determining region (RRDR) of the *rpoB* gene. This gene, which encodes the β-subunit of the RNA polymerase, is the target of rifampicin.

The antibiotic binds to RpoB, which inhibits the elongation of messenger RNA. Mutations in the RRDR are known to cause conformational changes that result in poor binding of the drug and cause resistance (Da Silva & Palomino, 2011). Furthermore, one of the rules has a much greater importance than the other. This suggests the existence of two clusters of rifampicin resistant strains, one being predominant, both harboring mutations in different regions of the RRDR.

For *S. pneumoniae*, the first and most important rule of the erythromycin resistance model targets the *mel* gene. The *mel* gene is part of the macrolide efflux genetic assembly (MEGA) and is known to confer resistance to erythromycin (Daly et al., 2004; Ambrose et al., 2005). Of note, this gene is found on an operon with either the *mefA* or the *mefE* gene, which are also part of the MEGA and confer resistance to erythromycin (Daly et al., 2004). It is likely that the algorithm targeted the *mel* gene to obtain a concise model that includes all of these resistance determinants. The second rule in the model is an absence rule that targets the wild-type version of the *metE* gene. This gene is involved in the synthesis of methionine (Basavanna et al., 2013). Alterations in this gene could lead to a lack of methionine in the cell and impact the ribosomal machinery, which is the drug's target. However, further validation is required to confirm this resistance determinant.

For *C. difficile*, the resistance models for azithromycin and clarithromycin, two macrolide antibiotics, share a rule with the resistance model for clindamycin, a lincosamide antibiotic. These three antibiotics function by binding the 50S subunit of the ribosome and interfering with bacterial protein synthesis (Tenson et al., 2003). Cross-resistance between macrolide and lincosamide antibiotics is caused by the presence of the *ermB* gene that encodes rRNA adenine N-6-methyltransferase, an enzyme that methylates position 2058 of the 23S rRNA within the larger 50S subunit (Farrow et al., 2000; Tenson et al., 2003; Vester & Douthwaite, 2001). The shared rule for the macrolide and the lincosamide models rightly targets the *ermB* gene. This rule has 616 equivalent rules, all of the *presence* type, targeting *ermB*. Arguably, the algorithm correctly found the presence of this gene to be a cross-resistance determinant, in agreement with the literature (Farrow et al., 2000; Tenson et al., 2003; Vester & Douthwaite, 2001).

Azithromycin and clarithromycin have similar mechanisms of action and, as expected, their resistance models are identical. They contain a presence rule that targets a region of the Tn*6110* transposon, characterized in *C. difficile* strain QCD-6626 (Brouwer et al., 2011). This region is located 136 base pairs downstream of a 23S rRNA methyltransferase, which is a gene known to be associated with macrolide resistance (Kaminska et al., 2010). The next rule in the models targets the presence of the penicillin-binding protein that is associated with resistance to β-lactam antibiotics, such as ceftriaxone (Waxman & Strominger, 1983). Among the azithromycin-resistant isolates in our dataset, 92.7% are also resistant to ceftriaxone. Similarly, 92.2% of the clarithromycin-resistant isolates are resistant to ceftriaxone. Hence, this rule was likely selected due to these strong correlations.

Finally, the model for clindamycin resistance contains a rule targeting a Tn*6194*-like conjugative transposon. This transposon contains the *ermB* gene, which is associated with resistance to this antibiotic (Wasels et al., 2013). Moreover, it is rarely found in clinical isolates, which could explain its smaller importance.

Spurious Correlations can be Overcome

One limitation of statistical approaches that derive models from data is their inability to distinguish causal variables from those that are highly correlated with them. To our knowledge, it is very difficult to prevent this pitfall, however we can leverage the interpretability and the sparsity of our models to identify and overcome spurious correlations. In our data, one notable example of such situation is the correlation between resistance to some antibiotics. This is illustrated in Figure 4 A. As a consequence, the model obtained for streptomycin is identical to the one obtained for isoniazid (Appendix, Figure S1). However, it is well known that these two antibiotics have different mechanisms of action and thus, different resistance mechanisms.

We propose the following procedure to close in on the causal genomic variations:

1. Learn a model using the SCM.
2. Validate the rules of the model (mutations, indels or structural variations) by searching the literature or with laboratory experiments.
3. If a rule is not associated truly with the phenotype, remove the k-mers corresponding to the rule and its equivalent rules from the data.
4. Repeat until a good association is found.

We validated this procedure by applying it to the streptomycin dataset. Six rounds were required in order to converge to the true resistance mechanism, i.e., the *rpsL* gene (Nair et al., 1993). The models obtained throughout the iterations are illustrated in Figure 4 B. These models contain rules targeting the *katG* and the *rpoB* genes, which are respectively isoniazid and rifampicin resistance determinants (Cade et al., 2010; Da Silva & Palomino, 2011). Again, this is explained by the strong correlations between

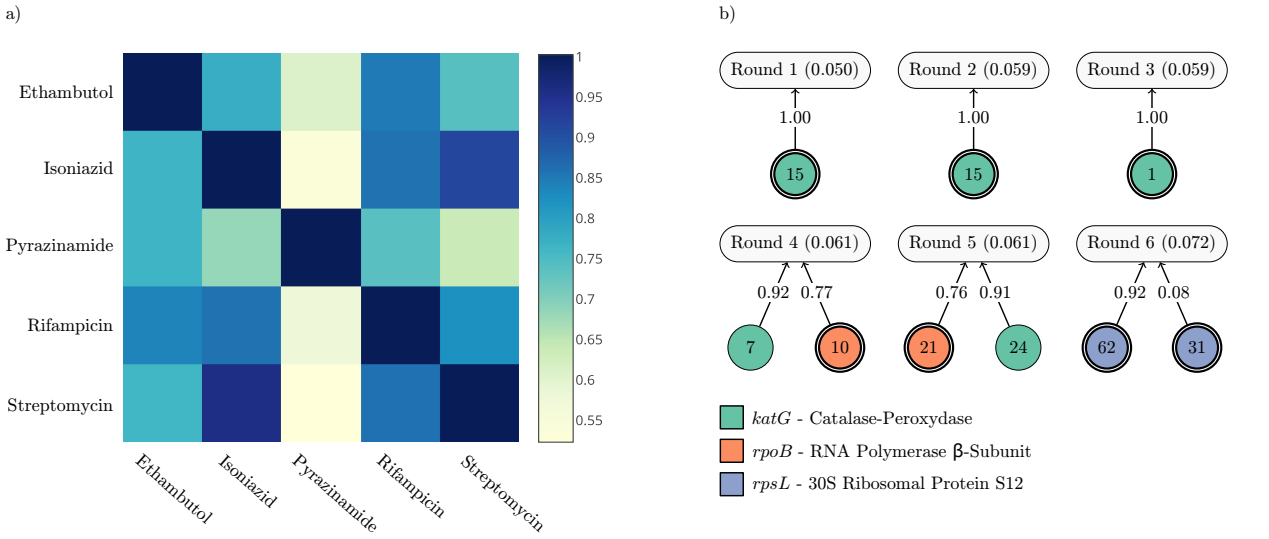

Figure 4. Overcoming spurious correlations: This figures shows how spurious correlations in the data can affect the models produced by the Set Covering Machine. a) For each antibiotic, the corresponding line shows the proportion of isolates that have the same label (sensitive or resistant) in the dataset of each other antibiotic. b) The antibiotic resistance models learnt by the SCM at each iteration of the correlation removal procedure. Each model is represented by a rounded rectangle identified by the round number and the estimated error rate. All the models are disjunctions (logical-OR). The circular nodes correspond to k-mer rules. A single border indicates a presence rule and a double border indicates an absence rule. The numbers in the circles show to the number of equivalent rules. A rule is connected to an antibiotic if it was included in its model. The weight of the edges gives the importance of each rule.

the resistance to streptomycin and isoniazid (95.6% identical), and streptomycin and rifampicin (85.9% identical).

It is important to point out that this procedure was applicable due to the sparse and interpretable nature of the models generated by the SCM and illustrates the risks associated with the clinical use of uninterpretable models.

The SCM can Predict the Level of Resistance

To further demonstrate how our method can be used to explore the relationships between genotypes and phenotypes, we used it to predict the level of benzylpenicillin resistance in *S. pneumoniae*. For this bacterium, penicillin resistance is often mediated by alterations that reduce the affinity of penicillin-binding proteins (Fani et al., 2014). Moderate-level resistance is due to alterations in PBP2b and PBP2x, whereas high-level resistance is due to additional alterations in PBP1a. Based on antibiotic susceptibility data described in the Appendix, Table S1, we defined three levels of antibiotic resistance, which were used to group the isolates: high-level resistance (R), moderate-level resistance (I) and sensitive (S). We then attempted to discriminate highly resistant isolates from sensitive isolates and moderately resistant isolates from sensitive isolates. The same protocol as in the previous sections was used.

For discriminating highly resistant and sensitive isolates, we obtained an error rate of 0.013. The obtained model rightly targeted the *pbp1a* gene. Based on the protocol presented in the Appendix, we removed all the k-mer located in this gene and repeated the experiment. The obtained model targeted the *pbp2b* gene and the error rate was 0.017. These results are consistent with the literature, since they indicate that alterations in both genes are equally predictive of a high-level of resistance and thus, that they occur simultaneously in isolates that are highly resistant to penicillin (Fani et al., 2014).

For discriminating moderately resistant and sensitive isolates, we obtained an error rate of 0.064. The obtained model rightly targeted the *pbp2b* gene. Again, we removed all the k-mer located in this gene and repeated the experiment. The obtained model targeted the *pbp2x* gene and the error rate was 0.072. In accordance with the literature, this indicates that alterations in both genes are predictive of moderate-level resistance. However, from our results, it seems that alterations in *pbp2b* are slightly more predictive of this phenotype.

Discussion

We have addressed the problem of learning computational phenotyping models from whole genome sequences. We sought a method that produces accurate models that are interpretable by domain experts, while relying on a minimal set of biomarkers. Our results for predicting antibiotic resistance demonstrate that we have achieved this goal.

In terms of accuracy, our method was shown to produce models that are more accurate than those obtained with other biomarker discovery methods. For most datasets, the error rates are well below 10%. Given the inherent noise in the antibiotic susceptibility measurements, it is likely that these error rates are near optimal. Moreover, for some datasets, none of the methods produced highly accurate models. We hypothesize that this is due to either the quality of the sequencing data, an insufficient number of learning examples, or extra-genomic factors that make the phenotype hard to infer from the genotype. For example, epigenetic modifications have been shown alter gene expression in bacteria and play a role in their virulence (Adam et al., 2008; Casadesús & Low, 2006). Assuming the availability of the data, future work could explore extensions to jointly learn models from genetic and epigenetic data.

In terms of sparsity, the SCM was shown to produce the sparsest models. Notably, this was achieved without negatively impacting the prediction accuracy of the models. We hypothesize that this is due to the small number of genomic variations that drive most genome-related phenotypes.

Hence, we presented empirical evidence that the SCM outperforms Linear Support Vector Machines and CART decision trees, paired with statistical hypothesis testing, in the context of genomic biomarker discovery. This suggest that the conjunctions and disjunctions produced by the SCM, in addition to being intuitively understandable, are more suitable for this task. In the next section, we derive very tight statistical guarantees on the accuracy of the models obtained using our approach. Such theoretical results are uncommon for this type of tool and, together with the sparsity property of the SCM, support the suitability of this algorithm for genomic biomarker discovery.

The algorithmic and theoretical extensions of the Set Covering Machine algorithm proposed in this study have enabled its application to genomic biomarker discovery. Biologically relevant insight was acquired for antibiotic resistance, a biological phenomenon that remains misunderstood. Indeed, within hours of computation, we have retrieved antibiotic resistance mechanisms that have been reported over the past decades. Hence, this method could be used to rapidly gain insight on the causes of resistance to new antibiotics, for which the mechanism of action might not be fully understood. Furthermore, as our results suggest, our method could be used to discover resistance mechanisms that are shared by multiple antibiotics, which would allow the development of more effective combination therapies.

Finally, note that no assumption was made on the type of organism under study and therefore, we are confident that this method will easily transpose to other organisms and even to metagenomic studies. This method is broadly applicable and is not limited to predicting drug response. The efficiency and the comprehensiveness of the models obtained using our method could guide biological efforts for understanding a plethora of phenotypes. To this end, we provide Kover, an implementation of our method that efficiently combines the modeling power of the Set Covering Machine with the versatility of the k-mer representation. Kover is readily applicable to other organisms and phenotypes. It is open-source and is available at `http://github.com/aldro61/kover`.

Methods

Genome Assembly and k-merization

All genomes were assembled using the SPAdes genome assembler (Bankevich et al., 2012) and were split into k-mers using the Ray Surveyor tool, which is part of the Ray *de novo* genome assembler (Boisvert et al., 2010; 2012). Note that genome assembly is not mandatory for applying our method, one could use the reads instead of k-mers. Nevertheless, it can help to homogenize data obtained under different conditions, e.g., different sequencing technologies, and to attenuate the noise resulting from sequencing errors.

Applying the Set Covering Machine to Genomes

We represent each genome by the presence or absence of each possible k-mer. There are 4^k possible k-mers and hence, for $k = 31$, we consider $4^{31} \approx 4 \cdot 10^{18}$ k-mers. Let $\mathcal{K}$ be the set of all, possibly overlapping, k-mers present in at least one genome of the training set $\mathcal{S}$. Observe that $\mathcal{K}$ omits k-mers that are absent in $\mathcal{S}$ and thus non-discriminatory, which allows the SCM to efficiently "work" in this enormous feature space. Then, for each genome $\mathbf{x}$, let $\boldsymbol{\phi}(\mathbf{x}) \in \{0,1\}^{|\mathcal{K}|}$ be a $|\mathcal{K}|$ dimensional vector, such that its component $\phi_i(\mathbf{x}) = 1$ if the i-th k-mer of $\mathcal{K}$ is present in $\mathbf{x}$ and 0 otherwise. An example of this representation is given in Figure 5. We consider two types of boolean-valued rules: presence rules and absence rules, which rely on the vectors $\boldsymbol{\phi}(\mathbf{x})$ to determine their outcome. For each k-mer $k_i \in \mathcal{K}$, we define a presence rule as $p_{k_i}(\boldsymbol{\phi}(\mathbf{x})) \stackrel{\text{def}}{=} I[\phi_i(\mathbf{x}) = 1]$ and an absence rule as $a_{k_i}(\boldsymbol{\phi}(\mathbf{x})) \stackrel{\text{def}}{=} I[\phi_i(\mathbf{x}) = 0]$, where $I[a] = 1$ if a is true and $I[a] = 0$ otherwise. The SCM, which is detailed in the Appendix, can then be applied by using $\{(\boldsymbol{\phi}(\mathbf{x}_1), y_1), ..., \boldsymbol{\phi}(\mathbf{x}_m), y_m)\}$ as the set $\mathcal{S}$ of learning examples and by using the set of presence/absence rules defined above as the set $\mathcal{R}$ of boolean-valued rules. This yields a phenotypic model which explicitly highlights the importance of a small set of k-mers. In addition, this model has a form which is simple to interpret, since its predictions are the result of a simple logical operation.

$$\mathcal{K} = \left\{ \begin{array}{cccc} \text{CAGATA} & \text{GATAGA} & \text{GAACAG} & \text{CGATGA} \\ \text{AGATAG} & \text{AGAACA} & \text{ATAGAA} & \text{CCGGCT} \\ \text{AACAGC} & \text{TAGAAC} & \text{TTTCGG} & \text{AAATAC} \end{array} \right\}$$

$$\mathbf{x} = \text{CAGATAGAACAGC}$$

$$\boldsymbol{\phi}(\mathbf{x}) = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|} \hline 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ \hline \end{array}$$

CAGATA TTTCGG AGATAG GATAGA CGATGA AACAGC ATAGAA CCGGCT TAGAAC GAACAG AGAACA AAATAC

Figure 5. The k-mer representation: An example of the k-mer representation. Given the set of observed k-mers $\mathcal{K}$ and a genome $\mathbf{x}$, the corresponding vector representation is given by $\boldsymbol{\phi}(\mathbf{x})$.

Tiebreaker Function

At each iteration of the SCM algorithm (Marchand & Shawe-Taylor, 2002), the rules are assigned a utility score based on their ability to classify the examples for which the outcome of the model is not settled. The number of such examples decreases at each iteration. Consequently, it is increasingly likely that many rules have an equal utility score. This phenomenon is accentuated when considering much more rules than learning examples, as it is the case for biomarker discovery. We therefore extend the algorithm by introducing a tiebreaker function for rules of equal utility. The latter selects the rule that best classifies all the learning examples, i.e., the one with the smallest empirical error rate. This simple strategy favors rules that are more likely to be associated with the phenotype.

Exploiting Equivalent Rules

When applying the SCM to genomic data, even the tiebreaker function rarely reduces the set of candidate rules to a single rule. This is a consequence of the inherent correlation that occurs between k-mers that overlap, or those that are nearby in the genomic structure. The remaining rules are deemed equivalent. Our goal being to obtain concise models, only one of these rules is included in the model and used for prediction. This rule is selected randomly, but other strategies could be applied. As we have demonstrated in our experiments, these rules provide a unique approach for deciphering, *de novo*, new biological mechanisms without the need for prior information. Indeed, the set of k-mers targetted by these rules can be analyzed to draw conclusions on the type of genomic variation that was identified by the algorithm, e.g., point mutation, indel or structural variation.

Measuring the Importance of Rules

We propose a measure of importance for the rules in a conjunction or disjunction model. Taking rule importance into consideration can facilitate the interpretation of the model. A good measure of importance should measure the impact of each rule on the predictions of the model. Observe that for any example $\mathbf{x}$, a conjunction model predicts $h(\mathbf{x}) = 0$ if at least one of its rules returns 0. Thus, when a rule returns 0, it directly contributes to the outcome of the model. Moreover, a conjunction model predicts $h(\mathbf{x}) = 1$ if and only if exactly all of its rules return 1. Hence, in this case, all the rules contribute equally to the prediction and thus, we do not need to consider this case in the measure of importance. The importance of a rule r in a conjunction model is therefore given by:

$$I_{\wedge}(r) \stackrel{\text{def}}{=} \frac{\sum_{(\mathbf{x},y)\in\mathcal{S}} I[r(\mathbf{x}) = 0 \wedge h(\mathbf{x}) = 0]}{\sum_{(\mathbf{x},y)\in\mathcal{S}} I[h(\mathbf{x}) = 0]}, \qquad (2)$$

where $r(\mathbf{x})$ is the outcome of rule r on example $\mathbf{x}$. In contrast, a disjunction model predicts $h(\mathbf{x}) = 1$ if at least one of its rules return 1. Moreover, it predicts $h(\mathbf{x}) = 0$ if and only if exactly all of its rules returns 0. The importance of a rule in a disjunction model is thus given by:

$$I_{\vee}(r) \stackrel{\text{def}}{=} \frac{\sum_{(\mathbf{x},y)\in\mathcal{S}} I[r(\mathbf{x}) = 1 \wedge h(\mathbf{x}) = 1]}{\sum_{(\mathbf{x},y)\in\mathcal{S}} I[h(\mathbf{x}) = 1]}. \qquad (3)$$

An Upper Bound on the Error Rate

When the number of learning examples is much smaller than the number of features, many machine learning algorithms tend to overfit the training data and thus, have a poor generalization performance (Hastie et al., 2013). Genomic biomarker discovery fits precisely in this regime. Using sample-compression theory (Floyd & Warmuth, 1995; Littlestone & Warmuth, 1986; Marchand & Sokolova, 2005), we obtained an upper bound on the error rate $R(h)$ of any model h learnt using our proposed approach.

Formally, for any distribution D, with probability at least $1-\delta$, over all datasets $\mathcal{S}$ drawn according to D^m, we have that all models h have $R(h) \leq \epsilon$, where

$$\epsilon = 1 - \exp\left(\frac{-1}{m - m_{\mathcal{Z}} - r}\left[\ln\binom{m}{m_{\mathcal{Z}}} + \ln\binom{m - m_{\mathcal{Z}}}{r} + |h|\cdot\ln(2\cdot|\mathcal{Z}|) + \ln\left(\frac{\pi^6(|h|+1)^2(r+1)^2(m_{\mathcal{Z}}+1)^2}{216\cdot\delta}\right)\right]\right), \qquad (4)$$

where m is the number of learning examples, $|h|$ is the number of rules in the model, $\mathcal{Z}$ is a set containing $m_{\mathcal{Z}} \leq |h|$ learning examples (genomes) in which each k-mer in the model can be found, $|\mathcal{Z}|$ is the total number of nucleotides in $\mathcal{Z}$ and r is the number of prediction errors made by h on $\mathcal{S} \setminus \mathcal{Z}$. The steps required to obtain this bound are detailed in the Appendix.

This theoretical result guarantees that our method will achieve good generalization, regardless of the number of possible features under consideration (4^k), provided that we obtain a sparse model (small $|h|$) that makes few errors on the training set (small r). Hence, this sample-compression analysis indicates that we are not in an overfitting situation, even if the number of features is much larger than the number of example. This is counter-intuitive with respect to classical machine learning theory and highlights the benefits of the sample-compression approach. Moreover, this theoretical result is consistent with our empirical results (Appendix, Table S5) which indicate that cross-validating the k-mer length does not lead to more accurate models ($p = 0.551$).

Finally, following the idea of Marchand and Shawe-Taylor (Marchand & Shawe-Taylor, 2002), we attempted to use the bound value as a substitute for 5-fold cross-validation. In this case, the bound value was used to determine the best combination of hyperparameter values (Appendix, Table S6). This led to a 5-fold decrease in the number of SCM trainings required to conduct the experiments and yielded sparser models ($p = 0.014$) with similar accuracies ($p = 0.463$).

Efficient Implementation

The large size of genomic datasets tends to surpass the memory resources of modern computers. Hence, there is a need for algorithms that can process such datasets without solely relying on the computer's memory. *Out-of-core* algorithms achieve this by making efficient use of external storage, such as file systems. Along with this work, we propose *Kover*, an out-of-core implementation of the Set Covering Machine tailored for presence/absence rules of k-mers. Kover implements all the algorithmic extensions proposed in this work. It makes use of the HDF5 library (The HDF Group, 1997-2015) to efficiently store the data and process it in blocks. Moreover, it exploits atomic CPU instructions to accelerate computations. The details are provided in the Appendix. Kover is implemented in the Python and C programming languages, is open-source software and can be obtained free of charge.

Availability of supporting data

The data sets supporting the results of this article are available in the GenBank and EMBL repositories at [EMBL:PRJEB2632, EMBL:PRJEB11776, EMBL:PRJEB7281, GenBank:PRJNA264310].

Kover, the out-of-core implementation of our method, is open-source and available at `http://github.com/aldro61/kover`.

Competing interests

The authors declare that they have no competing interests.

Author contributions

AD, FL, MM and SG designed the algorithmic extensions to the Set Covering Machine algorithm. AD, FL and MM derived the sample compression bound for the Set Covering Machine. AD designed the out-of-core implementation of the Set Covering Machine. AD, FL, JC, MM and SG designed the experimental protocols and AD conducted the experimentations. AD, JC, MD and SG evaluated the biological relevance of the models. MD acquired the data and prepared it for analysis. AMB and VL acquired and provided the C. difficile genomes and the associated antibiotic resistance data. AD, FL, JC, MD, MM, MT and SG wrote the manuscript. All authors have read and approved the final manuscript.

Additional data files

The following additional data are available with the online version of this paper. Additional data file 1 is a table containing the antibiotic resistance classifications for the *Clostridium difficile* genomes of Dr. Loo and Dr. Bourgault [EMBL:PRJEB11776].

Acknowledgements

The authors acknowledge Dr. Éric Audemard, Dr. Sébastien Boisvert, Dr. Sylvain Moineau, Dr. Jean-Louis Plouhinec and Dr. Paul H. Roy for helpful comments and suggestions. Computations were performed on the Colosse supercomputer at Université Laval (resource allocation project: nne-790-ae), under the auspices of Calcul Québec and Compute Canada. AD is recipient of an Alexander Graham Bell Canada Graduate Scholarship Doctoral Award of the National Sciences and Engineering Research Council of Canada (NSERC). This work was supported in part by the NSERC Discovery Grants (FL; 262067, MM; 122405) and an award to MT from the Ministère de l'enseignement supérieur, de la recherche, de la science et de la technologie du Québec through Génome Québec. JC acknowledges the Canada Research Chair in Medical Genomics. AMB and VL acknowledge the Consortium de Recherche sur le Clostridium difficile, which consists of the following partners: Fonds de la Recherche en Santé du Québec, Canadian Institutes of Health Research, Ministre de la Santé et des Services Sociaux du Québec, Institut National de Santé Publique du Québec, Health Canada, Centre Hospitalier de l'Université de Montréal, McGill University Health Centre, CHU de Québec, and CHU de Sherbrooke.

References

Adam, Mike, Murali, Bhuvana, Glenn, Nicole O., and Potter, S. Steven. Epigenetic inheritance based evolution of antibiotic resistance in bacteria. *BMC Evolutionary Biology*, 8(1):1–12, 2008. ISSN 1471-2148. doi: 10.1186/1471-2148-8-52. URL http://dx.doi.org/10.1186/1471-2148-8-52.

Altschul, Stephen F, Gish, Warren, Miller, Webb, Myers, Eugene W, and Lipman, David J. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, October 1990.

Ambrose, Karita D, Nisbet, Rebecca, and Stephens, David S. Macrolide Efflux in Streptococcus pneumoniae Is Mediated by a Dual Efflux Pump (mel and mef) and Is Erythromycin Inducible. *Antimicrobial Agents and Chemotherapy*, 49(10):4203–4209, October 2005.

Azuaje, Francisco. *Bioinformatics and Biomarker Discovery*. "Omic" Data Analysis for Personalized Medicine. John Wiley & Sons, August 2011.

Bankevich, Anton, Nurk, Sergey, Antipov, Dmitry, Gurevich, Alexey A, Dvorkin, Mikhail, Kulikov, Alexander S, Lesin, Valery M, Nikolenko, Sergey I, Pham, Son K, Prjibelski, Andrey D, Pyshkin, Alex, Sirotkin, Alexander, Vyahhi, Nikolay, Tesler, Glenn, Alekseyev, Max A, and Pevzner, Pavel A. SPAdes: A New Genome Assembly Algorithm and Its Applications to Single-Cell Sequencing. *Journal of computational biology*, 19(5):455–477, 2012.

Basavanna, Shilpa, Chimalapati, Suneeta, Maqbool, Abbas, Rubbo, Bruna, Yuste, Jose, Wilson, Robert J, Hosie, Arthur, Ogunniyi, Abiodun D, Paton, James C, Thomas, Gavin, and Brown, Jeremy S. The Effects of Methionine Acquisition and Synthesis on Streptococcus Pneumoniae Growth and Virulence. *PLoS ONE*, 8(1):e49638, January 2013.

Boisvert, Sébastien, Laviolette, François, and Corbeil, Jacques. Ray: simultaneous assembly of reads from a mix of high-throughput sequencing technologies. *Journal of Computational Biology*, 17(11):1519–1533, 2010.

Boisvert, Sébastien, Raymond, Frédéric, Godzaridis, Élénie, Laviolette, François, and Corbeil, Jacques. Ray Meta: scalable de novo metagenome assembly and profiling. *Genome biology*, 13(12):R122, 2012.

Bonham-Carter, Oliver, Steele, Joe, and Bastola, Dhundy. Alignment-free genetic sequence comparisons: a review of recent approaches by word analysis. *Briefings in Bioinformatics*, 15(6):890–905, 2014.

Breiman, Leo, Friedman, Jerome, Stone, Charles J, and Olshen, Richard A. *Classification and regression trees*. CRC press, 1984.

Brookes, Anthony J. The essence of snps. *Gene*, 234(2):177–186, 1999.

Brouwer, Michael S M, Warburton, Philip J, Roberts, Adam P, Mullany, Peter, and Allan, Elaine. Genetic organisation, mobility and predicted functions of genes on integrated, mobile genetic elements in sequenced strains of Clostridium difficile. *PLoS ONE*, 6(8):e23014, 2011.

Cade, Christine E, Dlouhy, Adrienne C, Medzihradszky, Katalin F, Salas-Castillo, Saida Patricia, and Ghiladi, Reza A. Isoniazid-resistance conferring mutations in mycobacterium tuberculosis katg: Catalase, peroxidase, and inh-nadh adduct formation activities. *Protein Science*, 19(3):458–474, 2010.

Casadesús, Josep and Low, David. Epigenetic gene regulation in the bacterial world. *Microbiology and molecular biology reviews*, 70(3):830–856, 2006.

Cheng, B and Titterington, D M. Neural networks: A review from a statistical perspective. *Statistical Science*, 1994.

Cortes, Corinna and Vapnik, Vladimir. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

Croucher, Nicholas J, Finkelstein, Jonathan A, Pelton, Stephen I, Mitchell, Patrick K, Lee, Grace M, Parkhill, Julian, Bentley, Stephen D, Hanage, William P, and Lipsitch, Marc. Population genomics of post-vaccine changes in pneumococcal epidemiology. *Nature genetics*, 45(6):656–663, May 2013.

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

Daly, Melissa M, Doktor, Stella, Flamm, Robert, and Shortridge, Dee. Characterization and prevalence of MefA, MefE, and the associated msr(D) gene in Streptococcus pneumoniae clinical isolates. *Journal of clinical microbiology*, 42(8):3570–3574, August 2004.

Fan, Rong-En, Chang, Kai-Wei, Hsieh, Cho-Jui, Wang, Xiang-Rui, and Lin, Chih-Jen. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research*, 9:1871–1874, June 2008.

Fani, Fereshteh, Leprohon, Philippe, Zhanel, George G, Bergeron, Michel G, and Ouellette, Marc. Genomic

analyses of DNA transformation and penicillin resistance in Streptococcus pneumoniae clinical isolates. *Antimicrobial Agents and Chemotherapy*, 58(3):1397–1403, 2014.

Farrow, K A, Lyras, D, and Rood, J I. The macrolide-lincosamide-streptogramin B resistance determinant from Clostridium difficile 630 contains two erm(B) genes. *Antimicrobial Agents and Chemotherapy*, 44(2): 411–413, February 2000.

Floyd, Sally and Warmuth, Manfred. Sample compression, learnability, and the Vapnik-Chervonenkis dimension. *Machine Learning*, 21(3):269–304, 1995.

Guyon, Isabelle and Elisseeff, André. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, March 2003.

Hastie, Trevor, Tibshirani, Robert, and Friedman, Jerome. *The Elements of Statistical Learning*. Data Mining, Inference, and Prediction. Springer Science & Business Media, November 2013.

Kaminska, Katarzyna H, Purta, Elzbieta, Hansen, Lykke H, Bujnicki, Janusz M, Vester, Birte, and Long, Katherine S. Insights into the structure, function and evolution of the radical-SAM 23S rRNA methyltransferase Cfr that confers antibiotic resistance in bacteria. *Nucleic Acids Research*, 38(5):1652–1663, March 2010.

Kingsford, Carl and Salzberg, Steven L. What are decision trees? *Nature Biotechnology*, 26(9):1011–1013, September 2008.

Koboldt, Daniel C, Steinberg, Karyn Meltz, Larson, David E, Wilson, Richard K, and Mardis, Elaine R. The Next-Generation Sequencing Revolution and Its Impact on Genomics. *Cell*, 155(1):27–38, September 2013.

Kos, Veronica N, Deraspe, Maxime, McLaughlin, Robert E, Whiteaker, James D, Roy, Paul H, Alm, Richard A, Corbeil, Jacques, and Gardner, Humphrey. The resistome of Pseudomonas aeruginosa in relationship to phenotypic susceptibility. *Antimicrobial Agents and Chemotherapy*, 59(1):427–436, January 2015.

Leimeister, Chris-Andre, Boden, Marcus, Horwege, Sebastian, Lindner, Sebastian, and Morgenstern, Burkhard. Fast alignment-free sequence comparison using spaced-word frequencies. *Bioinformatics*, pp. btu177, 2014.

Littlestone, N. and Warmuth, M. Relating data compression and learnability. Technical report, University of California Santa Cruz, Santa Cruz, CA, 1986.

Marchand, Mario and Shawe-Taylor, John. The set covering machine. *The Journal of Machine Learning Research*, 3:723–746, 2002.

Marchand, Mario and Sokolova, Marina. Learning with Decision Lists of Data-Dependent Features. *Journal of Machine Learning Research*, pp. 427–451, 2005.

Mbianda, Christiane, El-Meanawy, Ashraf, and Sorokin, Andrey. Mechanisms of BK virus infection of renal cells and therapeutic implications. *Journal of Clinical Virology*, 71:59–62, October 2015.

Merker, Matthias, Blin, Camille, Mona, Stefano, Duforet-Frebourg, Nicolas, Lecher, Sophie, Willery, Eve, Blum, Michael G B, Rüsch-Gerdes, Sabine, Mokrousov, Igor, Aleksic, Eman, Allix-Béguec, Caroline, Antierens, Annick, Augustynowicz-Kopeć, Ewa, Ballif, Marie, Barletta, Francesca, Beck, Hans Peter, Barry, Clifton E, Bonnet, Maryline, Borroni, Emanuele, Campos-Herrero, Isolina, Cirillo, Daniela, Cox, Helen, Crowe, Suzanne, Crudu, Valeriu, Diel, Roland, Drobniewski, Francis, Fauville-Dufaux, Maryse, Gagneux, Sébastien, Ghebremichael, Solomon, Hanekom, Madeleine, Hoffner, Sven, Jiao, Wei-Wei, Kalon, Stobdan, Kohl, Thomas A, Kontsevaya, Irina, Lillebæk, Troels, Maeda, Shinji, Nikolayevskyy, Vladyslav, Rasmussen, Michael, Rastogi, Nalin, Samper, Sofia, Sanchez-Padilla, Elisabeth, Savic, Branislava, Shamputa, Isdore Chola, Shen, Adong, Sng, Li-Hwei, Stakenas, Petras, Toit, Kadri, Varaine, Francis, Vukovic, Dragana, Wahl, Céline, Warren, Robin, Supply, Philip, Niemann, Stefan, and Wirth, Thierry. Evolutionary history and global spread of the Mycobacterium tuberculosis Beijing lineage. *Nature genetics*, 47(3):242–249, March 2015.

Nair, J, Rouse, D A, Bai, G H, and Morris, S L. The rpsL gene and streptomycin resistance in single and multiple drug-resistant strains of Mycobacterium tuberculosis. *Molecular microbiology*, 10(3):521–527, November 1993.

Nielsen, Rasmus, Paul, Joshua S, Albrechtsen, Anders, and Song, Yun S. Genotype and SNP calling from next-generation sequencing data. *Nature Reviews Genetics*, 12(6):443–451, June 2011.

Noble, William S. What is a support vector machine? *Nature Biotechnology*, 24(12):1565–1567, December 2006.

Pedregosa, Fabian, Varoquaux, Gaël, Gramfort, Alexandre, Michel, Vincent, Thirion, Bertrand, Grisel, Olivier, Blondel, Mathieu, Prettenhofer, Peter, Weiss, Ron, Dubourg, Vincent, Vanderplas, Jake, Passos, Alexandre, Cournapeau, David, Brucher, Matthieu, Perrot, Matthieu, and Duchesnay, Édouard. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, February 2011.

Saeys, Yvan, Inza, Iñaki, and Larrañaga, Pedro. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, October 2007.

Simon, Richard. Genomic biomarkers in predictive medicine: an interim analysis. *EMBO molecular medicine*, 3(8):429–435, August 2011.

Song, Kai, Ren, Jie, Reinert, Gesine, Deng, Minghua, Waterman, Michael S, and Sun, Fengzhu. New developments of alignment-free sequence comparison: measures, statistics and next-generation sequencing. *Briefings in Bioinformatics*, 15(3):343–353, 2014.

Tenson, Tanel, Lovmar, Martin, and Ehrenberg, Måns. The mechanism of action of macrolides, lincosamides and streptogramin B reveals the nascent peptide exit path in the ribosome. *Journal of Molecular Biology*, 330(5): 1005–1014, July 2003.

The HDF Group. Hierarchical Data Format, version 5, 1997-2015. http://www.hdfgroup.org/HDF5/.

van Dijk, Erwin L, Auger, Hélène, Jaszczyszyn, Yan, and Thermes, Claude. Ten years of next-generation sequencing technology. *Trends in Genetics*, 30(9):418–426, 2014.

Vester, B and Douthwaite, S. Macrolide resistance conferred by base substitutions in 23S rRNA. *Antimicrobial Agents and Chemotherapy*, 45(1):1–12, January 2001.

Vinga, S and Almeida, J. Alignment-free sequence comparison-a review. *Bioinformatics*, 19(4):513–523, March 2003.

Vinga, Susana. Biological sequence analysis by vector-valued functions: revisiting alignment-free methodologies for dna and protein classification. *Advanced Computational Methods for Biocomputing and Bioimaging*, pp. 71–107, 2007.

Wasels, François, Spigaglia, Patrizia, Barbanti, Fabrizio, and Mastrantonio, Paola. Clostridium difficile erm(B)-containing elements and the burden on the in vitro fitness. *Journal of medical microbiology*, 62(Pt 9):1461–1467, September 2013.

Waxman, D J and Strominger, J L. Penicillin-binding proteins and the mechanism of action of beta-lactam antibiotics. *Annual Review of Biochemistry*, 52:825–869, 1983.

Wilcoxon, Frank. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80, December 1945.

World Health Organization. *Antimicrobial resistance: global report on surveillance*. World Health Organization, 2014.