Resolving Complexity of Within-host Pathogen Strain Diversity and Antimicrobial Resistance using Optimization and Machine Learning

by

Guo Liang Gan

B.Sc., The University of Hong Kong, 2016

Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in the School of Computing Science Faculty of Applied Sciences

c⃝Guo Liang Gan 2018
SIMON FRASER UNIVERSITY
Fall 2018

Copyright in this work rests with the author. Please ensure that any reproduction or re-use is done in accordance with the relevant national copyright legislation.
Approval

Name: Guo Liang Gan
Degree: MSc Thesis (Computing Science)
Title: Resolving Complexity of Within-host Pathogen Strain Diversity and Antimicrobial Resistance using Optimization and Machine Learning

Examiner Committee:
Chair: Igor Shinkar
   Assistant Professor
   School of Computing Science

Leonid Chindelevitch
   Senior Supervisor
   Assistant Professor
   School of Computing Science

Cedric Chauve
   Supervisor
   Professor
   Department of Mathematics

Maxwell Libbrecht
   Supervisor
   Assistant Professor
   School of Computing Science

Kay C. Wiese
   Examiner
   Associate Professor
   School of Computing Science

Date Defended: September 26, 2018
Abstract

Hosts such as humans and arthropods are often infected by diverse bacterial populations and such heterogeneity poses a great threat in effective treatments. Moreover, the development of antimicrobial resistance further aggravates the efficacy of treatment. In this thesis, we present work towards solving two crucial problems in the realm of epidemiology, the within-host strain diversity problem and the antimicrobial resistance problem.

For the strain diversity problem, we resolved complex infections and co-infections pattern within a sample by developing a two-stage optimization pipeline using integer linear programming. Results on simulated data showed that our methods achieved a precision of at least 92% and a recall of at least 90% for the first-stage of our pipeline, and a soft-precision of at least 68% and a soft-recall of at least 79% for the second-stage. Soft-precision and soft-recall are similar to precision and recall but less stringent. This demonstrates that our methods are able to predict strains which are extremely close to the true strains and are comparable to or better than existing tools.

For the antimicrobial resistance problem, we explored different machine learning models to classify the response of a pathogen to a single drug, where we specifically focused on analyzing the robustness of the models and deriving resistance-related markers from them. The models achieved at least 85% test AUC (Area under the ROC curve) for first-line drugs and at least 76% for second-line drugs. Our results showed that the models are robust given datasets of varying diversity, and the resistance-related markers which were frequently reported in the literature were highly ranked by the models.

Our work on both problems paves the way for robust strain typing in the presence of within-host heterogeneity and effective drug resistance diagnosis, overcoming essential challenges to solving both problems which are currently not addressed by any existing methodology for pathogen genomics.

Keywords: Computational Biology; Combinatorial Optimization; Machine Learning
Dedication

To my dearest family
Acknowledgements

First and foremost, I would like to thank my supervisors Leonid Chindelevitch and Cedric Chauve for giving me the opportunity to study here in this beautiful city, Vancouver, Canada. My sincere thanks to Leonid for constantly pushing me to achieve my best and his invaluable advice on my career and research. My heartfelt thanks to Cedric for his continuous support in my research and educating me on how to perform quality research work. I would also like to thank my supervisor Maxwell Libbrecht for inspiring me with different perspectives on tackling a machine learning problem. Thanks to Kay C. Wiese for agreeing to be my examiner and to Igor Shinkar for being the chair of my thesis defense.

I would also like to express my gratitude to Elijah Willie, who had been working with me for my entire MSc education here at SFU; thanks for the fruitful discussions, both research and non-research related. I would also like to thank Matthew Nguyen and Brian Lee for their help and hard work in obtaining interesting results for the project. I am also grateful to the members of the Computational Epidemiology lab for the interesting discussions and enjoyable lunch breaks.

Last but not least, my sincere thanks to my family, my girlfriend and my friends. I would not be able to thrive through my MSc education and achieve what I have today without their unremitting support and love.
# Table of Contents

Approval ii
Abstract iii
Dedication iv
Acknowledgements v
Table of Contents vi
List of Tables viii
List of Figures ix

1 Introduction 1
  1.1 Strain Diversity Problem .............................................. 1
  1.2 The Antimicrobial Resistance Problem ............................... 3
  1.3 Connecting Both Problems .............................................. 5

2 Strain Diversity Problem 6
  2.1 Related Works ............................................................ 6
    2.1.1 Kallisto ............................................................. 7
    2.1.2 StrainEst ........................................................... 8
    2.1.3 DESMAN .............................................................. 9
    2.1.4 Comparison and Comments ....................................... 9
  2.2 Methods ................................................................. 10
    2.2.1 Overview ............................................................ 10
    2.2.2 The Allele Diversity Problem ..................................... 11
    2.2.3 The Strain Diversity Problem .................................... 16
    2.2.4 Implementation ..................................................... 20
    2.2.5 Data simulation .................................................... 21
  2.3 Results ........................................................................... 23
    2.3.1 Simulated Data ....................................................... 23
## Antimicrobial Resistance Problem

3

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Related Works</td>
<td>31</td>
</tr>
<tr>
<td>3.1.1 Machine Learning for Classifying TB Drug-Resistance from Sequencing Data</td>
<td>32</td>
</tr>
<tr>
<td>3.1.2 Deep Learning Predicts TB Drug Resistance from Sequencing Data</td>
<td>33</td>
</tr>
<tr>
<td>3.2 Dataset</td>
<td>33</td>
</tr>
<tr>
<td>3.3 Methods</td>
<td>35</td>
</tr>
<tr>
<td>3.3.1 Calling Single Nucleotide Polymorphisms (SNPs)</td>
<td>35</td>
</tr>
<tr>
<td>3.3.2 Cross Validation Protocol</td>
<td>37</td>
</tr>
<tr>
<td>3.3.3 Cross Dataset Evaluation</td>
<td>38</td>
</tr>
<tr>
<td>3.3.4 Machine Learning Classifiers</td>
<td>38</td>
</tr>
<tr>
<td>3.3.5 Resistance Determinant Gene</td>
<td>38</td>
</tr>
<tr>
<td>3.4 Results and Analysis</td>
<td>39</td>
</tr>
<tr>
<td>3.4.1 Exploratory Data Analysis</td>
<td>39</td>
</tr>
<tr>
<td>3.4.2 Within and Cross Dataset Evaluation</td>
<td>39</td>
</tr>
<tr>
<td>3.4.3 Resistance Determinant Genes</td>
<td>41</td>
</tr>
<tr>
<td>3.4.4 Resistance Determinant SNPs</td>
<td>43</td>
</tr>
<tr>
<td>3.5 Discussion</td>
<td>45</td>
</tr>
</tbody>
</table>

## Conclusion

4

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Summary of Contributions</td>
<td>51</td>
</tr>
<tr>
<td>4.2 Future Work</td>
<td>52</td>
</tr>
</tbody>
</table>

Bibliography

Appendix A Strain Diversity Problem

Appendix B Antimicrobial Resistance Problem
Table 2.1 Average (AVG) and standard deviation (SD) of precision for each gene of the Boreliia MLST scheme across all parameters combination. . . . 23
Table 2.2 Average (AVG) and standard deviation (SD) of recall for each gene of the Boreliia MLST scheme across all parameters combination. . . . 23
Table 2.3 Average (AVG) and standard deviation (SD) of TVD for each gene of the Boreliia MLST scheme across all parameters combination. . . . 24
Table 2.4 Average and standard deviation of different statistics for each evolutionary mechanisms for SDP simulation. This simulation assumes perfect input data from ADP. EM represents EvoMod. . . . . . . . 24
Table 2.5 Average and standard deviation of different statistics for each evolutionary mechanisms for the ADP stage of full pipeline simulation. EM represents EvoMod. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25
Table 2.6 Average and standard deviation of different statistics for each evolutionary mechanisms for the SDP stage of full pipeline simulation. EM represents EvoMod. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25
Table 2.7 Full pipeline benchmark simulation: Comparison between our algorithm and strainEST on the ADP problem. k is the number of existing strains to simulate. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25
Table 2.8 Full pipeline benchmark simulation: Comparison between our algorithm and strainEST on the SDP problem. k is the number of existing strains to simulate. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25
Table A.1 Average and standard deviation of different statistics for full pipeline simulation of 4 evolutionary mechanisms. (Top) ADP comparison (Middle) strainEST on SDP (Bottom) Our method on SDP . . . . . . . . 61
Table B.1 Details of lineage . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 72
List of Figures

Figure 2.1  (Upper) T-DBG and 2 transcripts. (Bottom) $k_1, k_2, k_3, k_4$ highlighted in green are $k$-mers of a read. $k_1$ and $k_2$ are associated to the compatibility class which consists of transcript A and transcript B. $k_3$ and $k_4$ are associated to the compatibility class consists of transcript A. The read is associated to an equivalence class consisting of only transcript A. Non-informative $k$-mers such as $k_1, k_2$ are skipped which speeds up the process. .................................................. 7

Figure 2.2  A dataset with two samples and an MLST scheme of three loci (genes clpA, clpX, nifS). The strain type distributions require 5 different strains as the strain (clpA_1,clpX_1, nifS_7) appears in both distributions. .................................................. 10

Figure 2.3  Distribution of the number of existing and novel strains per tick sample. .................................................. 26

Figure 2.4  (Left) Cumulated proportion of the 10 existing strains in all 24 samples (In each bar, different colors represent different samples). (Right) Similar graph for the 60 novel strains. .................................................. 27

Figure 2.5  Nodes with prefix ‘n’ represents novel strains, ‘e’ represents existing. The edges connecting two strains are labeled with the edit distance between them. The x-axis and y-axis are normalized distance used for visualizing the tree. .................................................. 28

Figure 2.6  Minimum spanning tree of strains found in 24 samples. The edges connecting two strains are labeled with the edit distance between them. The x-axis and y-axis are normalized distance used for visualizing the tree. .................................................. 29

Figure 3.1  WDNN model where each output corresponds to resistance towards a drug. .................................................. 34

Figure 3.2  Number of SNPs used to predict resistance of common drugs for ReSeqTB and BCCDC. .................................................. 35

Figure 3.3  Number of SNPs used to predict resistance of different drugs the remaining drugs for ReSeqTB .................................................. 35
Figure 3.4  Number of susceptible and resistant samples for common drugs for ReSeqTB and BCCDC. ........................................ 36
Figure 3.5  Number of susceptible and resistant samples for the remaining drugs for ReSeqTB ................................................ 36
Figure 3.6  Number of samples which are resistant to different number of drugs. (Left) ReSeqTB (Right) BCCDC ................................. 37
Figure 3.7  Description of each fold for a 10-fold cross validation process. (Left) This is the workflow for choosing the best parameters for each model. Models were trained on the train set and an optimal probability threshold was chosen for classification. The validation set was used for choosing the best parameters and test set was left untouched. (Right) Once the best parameters were found, we trained on both train and validation sets, predicted on test sets and report the performance on the test set for each fold. ........................................ 37
Figure 3.8  Description of machine learning pipeline: From the reads, we called SNPs and these SNPs are treated as features for any machine learning model. We also performed a kbest feature selection before training any model. At the very end, we predict the response for each strain. ................................................................. 38
Figure 3.9  ReSeqTB: The axes are the first 2 components of PCA and the drug in study is isoniazid. .................................................. 40
Figure 3.10  ReSeqTB: The axes are the first 2 components of PCA and the drug in study is pyrazinamide. ........................................... 41
Figure 3.11  BCCDC: The axes are the first 2 components of PCA and the drug in study is isoniazid ................................................. 42
Figure 3.12  BCCDC: The axes are the first 2 components of PCA and the drug in study is pyrazinamide. ................................. 43
Figure 3.13  Common Drugs: Test AUC for within and cross dataset evaluation. "X train; Z test" means models are trained on X dataset and tested on Z dataset. (Top) Isoniazid (Middle) Pyrazinamide (Bottom) MDR 44
Figure 3.14  ReSeqTB: Test AUC for other drugs ............................. 45
Figure 3.15  ReSeqTB: Difference in test AUC before and after permuting the rows of a submatrix corresponding to a gene. (Left) Isoniazid (Right) Streptomycin ......................................................... 45
Figure 3.16  BCCDC: Difference in test AUC before and after permuting the rows of a submatrix corresponding to a gene. (Left) Isoniazid (Right) Streptomycin ......................................................... 46
Figure 3.17  ReSeqTB: Non-zero coefficients of the LASSO model with PLS-transformed feature matrix (Left) Isoniazid (Right) Streptomycin

Figure 3.18  BCCDC: Non-zero coefficients of the LASSO model with PLS-transformed feature matrix (Left) Isoniazid (Right) Streptomycin

Figure 3.19  ReSeqTB: Non-zero coefficients of the LASSO model (Left) Isoniazid (Right) Rifampicin

Figure 3.20  ReSeqTB: Feature importance of Random Forest (Left) Isoniazid (Right) Rifampicin

Figure 3.21  BCCDC: Non-zero coefficients of the LASSO model (Left) Isoniazid (Right) Rifampicin

Figure 3.22  BCCDC: Feature importance of Random Forest (Left) Isoniazid (Right) Rifampicin

Figure A.1  Box plots of precision and recall of our method for different combinations of coverages and edit distance parameters over 40 simulations for 8 genes of the Borrelia MLST scheme

Figure A.2  Box plots of TVD of our method for different combinations of coverages and edit distance parameters over 40 simulations for 8 genes of the Borrelia MLST scheme

Figure A.3  Box plots of precision and recall of Kallisto for different combinations of coverages and edit distance parameters over 40 simulations for 8 genes of the Borrelia MLST scheme

Figure A.4  Box plots of TVD of Kallisto for different combinations of coverages and edit distance parameters over 40 simulations for 8 genes of the Borrelia MLST scheme

Figure A.5  Our method: Box plots of (1) ADP statistics (2) Soft-Precision and Soft-Recall on SDP (3) EMD on SDP where k=1

Figure A.6  Our method: Box plots of (1) ADP statistics (2) Soft-Precision and Soft-Recall on SDP (3) EMD on SDP where k=2

Figure A.7  strainEST: Box plots of (1) ADP statistics (2) Soft-Precision and Soft-Recall on SDP (3) EMD on SDP where k=1

Figure A.8  strainEST: Box plots of (1) ADP statistics (2) Soft-Precision and Soft-Recall on SDP (3) EMD on SDP where k=2

Figure B.1  Common Drugs: Test AUC for within and cross dataset evaluation. "X train; Z test" means models are trained on X dataset and tested on Z dataset. (Top) Ethambutol (Middle) Rifampicin (Bottom) Streptomycin
Figure B.2  ReSeqTB: Difference in test AUC before and after permuting the rows of a submatrix corresponding to a gene.  

Figure B.3  BCCDC: Difference in test AUC before and after permuting the rows of a submatrix corresponding to a gene.  

Figure B.4  ReSeqTB: Non-zero coefficients of the LASSO model with PLS-transformed feature matrix for different drugs. The y-axis is the magnitude of the coefficient of the Lasso model.  

Figure B.5  BCCDC: Non-zero coefficients of the LASSO model with PLS-transformed feature matrix for different drugs. The y-axis is the magnitude of the coefficient of the Lasso model.
Chapter 1

Introduction

Bacterial populations often exhibit an impressive amount of genomic diversity and such diversity decreases the efficacy of treatment for humans. Furthermore, the development of antimicrobial resistance further aggravates the effectiveness of these drug treatments. In this thesis, we investigate the strain diversity problem and the antimicrobial resistance problem within bacterial pathogens, both of which have strong connections and exhibit a great impact on the study of host genetics, eco-epidemiology and antibiotic drug design. We believe that this thesis demonstrates an interesting and comprehensive view on the integration of different computational and mathematical techniques in solving greatly related biological problems in the area of bioinformatics. We solved the strain diversity problem using knowledge from the fields of algorithms and optimization, whereas for the antimicrobial resistance problem, we tackled the problem from a machine learning perspective. An analysis of our results shows that the methods we presented are promising and will hopefully serve as a future reference for both problems.

The outline of the thesis is as follows: we will first introduce some background for both problems in this chapter and demonstrate crucial connections between them. Next, we proceed to present both problems individually as chapters of their own. Finally, we draw new inferences from both studies and discuss future directions. The work presented here on the strain diversity problem was presented in RECOMB-Seq 2018 and submitted to the Bioinformatics journal for its RECOMB-Seq special track. Most of the work presented on the antimicrobial resistance problem will be published on bioRxiv.

1.1 Strain Diversity Problem

Several studies demonstrated that hosts such as humans and arthropods are often infected by heterogeneous populations of bacterial pathogens [58]. Due to the emergence of improved genome sequencing technologies, useful information can be derived from sequencing data which can be utilized for unraveling within-host pathogen strain diversity. The study of within-host diversity is beneficial in helping researchers study the impact of diversity on
human health, epidemiology and pathogen evolution [5, 46]. For instance, understanding the within-host diversity of *Borrelia Burgdorferi* not only illuminates the transmission or mutation patterns of the bacteria, it also helps us in designing better treatment protocols, thus increasing the efficacy of Lyme disease treatments [18, 5].

*Borrelia* is a bacterium carried by deer ticks (*Ixodes scapularis*) and is the most common vector-borne disease in the United States [58]. Early symptoms of Lyme disease include rash-shaped like a bull’s eye, fever, headache, and swollen lymph nodes; if left untreated, serious symptoms such as facial paralysis, heart and neurological disorders may take place. Subsequent laboratory testing including an enzyme immunoassay (EIA) screening test and a confirmatory immunoblot (IB) test are usually carried out to detect the presence of *Borrelia* within patients. Treatment for Lyme disease consists of 2 to 4 weeks of antibiotics and a longer course of treatment may be required in certain situations, which might be related to antibiotic resistance issues or incorrect antibiotic prescription [1].

The study of bacterial pathogens has revealed an impressive genetic diversity that had not been fully suspected prior to the advent of genome sequencing technologies. This diversity may have been an adaptive response to challenges such as the variability in host genetics, environmental conditions, and, in the case of pathogens affecting humans, the introduction of antibacterial drugs [22, 11, 56, 3]. One of the emerging fields studying the effect of this diversity is eco-epidemiology, which tries to capture and interpret the combined influence of biological and environmental factors on human disease. It has been found that up to six genetically different strains can affect a single host [52, 9]. Furthermore, this diversity may result from both clonal evolution within the host as well as multiple infection events [58]. Unfortunately, techniques such as bacterial culture are difficult to apply to reveal the whole range of diversity in bacteria like *Borrelia*, a situation common to many bacterial pathogens. Next-generation sequencing (NGS) techniques such as whole-genome sequencing (WGS) with short reads have revolutionized our ability to investigate the genomic diversity of bacteria and other organisms [39]. Recently, an adaptation of WGS technology to *Borrelia*, called whole-genome capture, has been proposed which is able to reliably filter out irrelevant DNA (such as host DNA) [13]. This novel approach for generating sequence data for *Borrelia* nicely complements a highly reproducible strain-typing scheme known as multi-locus sequence typing (MLST), which has been developed and found to be useful for different pathogens in a number of contexts [40].

MLST is a summary of the bacterial genotype using the alleles of several (typically 6 to 9) housekeeping genes, which may be further grouped into closely related strain types (see [42] for a recent evaluation of MLST tools). In the case of *B. burgdorferi*, several hundred strain types have been characterized using the MLST scheme developed in [41], while only 56 fully sequenced *B. burgdorferi* genomes¹ are currently available in the NCBI

databases. MLST strain types thus provide a finer-grained picture of the strain diversity of this pathogen, which motivates the need for developing novel diversity estimation methods that combine NGS data and the wealth of strain types already characterized by MLST.

In principle, this problem is a special instance of estimating the diversity and abundance of microbial strains from metagenomics data, a problem for which several accurate methods have recently been developed [44, 2]. In our work, we study the strain diversity of *Borrelia*. We hypothesize that the degree of heterogeneity in the *Borrelia* samples in the study could provide interesting insights about their transmission pattern and mutation events. The goal of this study is to investigate co-infection patterns, which is to discover the number of different types of *Borrelia* strains and their abundances in each sample. We explored the strain diversity of *Borrelia* in a multi-locus sequence typing (MLST) context, which is one of the many ways to define a strain type, and proposed an optimization pipeline to understand complex infections within each sample. Although the pipeline is introduced specifically for *Borrelia* strain diversity, the idea and methods are generalizable for other pathogens.

### 1.2 The Antimicrobial Resistance Problem

Antimicrobial resistance is an event whereby a population of microbes such as viruses or a bacteria gains the capability, via genetic mutation, to resist against drugs which are used to treat the disease it causes. Antibiotic resistance in bacteria poses a major problem for the efficacy of treatment regimens; such bacteria include *Mycobacterium tuberculosis* and *Staphylococcus aureus* [17]. Due to numerous reasons such as incorrect drug prescriptions and prematurely halting treatments, bacteria gain the ability to circumvent the effects of drugs or their mechanisms of action and attain resistance.

Tuberculosis (TB) is a serious infectious disease caused by *Mycobacterium Tuberculosis (MTB)* and it is the 9th leading cause of death in the world, with roughly 1.4 million people dying due to the disease every year [60]. Based on the latest WHO TB report, TB incidence is decreasing at approximately 2% each year, and about 16% of the cases resulted in deaths due to the disease. In 30 of the high TB burden countries, the TB incidence accounts for as high as 86% of the global incidence rate [60]. More efforts are needed to reach the milestones of 4-5% incidence per year and a 10% death rate, as indicated in the End TB Strategy. In order to achieve these milestones, researchers and scientists have been working hard on addressing one of the most pressing issues in TB, drug resistance. MTB is vulnerable to acquiring drug resistance, especially multi-drug resistance (MDR) \(^2\) which poses challenges for an effective treatment for the disease. 600,000 new cases of resistance towards rifampicin, one of the strongest first-line drugs, were reported in 2016, and 490,000 of the cases are MDR[60]. MDR arises and continues to spread largely due to 2 reasons:

\(^2\)MDR is a resistance towards rifampicin and isoniazid, the 2 most effective first line-drugs.
mishandling of TB drug treatments and transmission from person-to-person, where poor quality drugs, stopping treatment prematurely, and inappropriate drug prescriptions cause the TB within patients to develop drug resistance, and these drug-resistant MTB strains are spread to new patients[60].

The golden rule in detecting drug resistance is phenotypic drug susceptibility testing (DST) via bacterial culture, where MTB is cultured in a Lowenstein-Jensen medium and is exposed to a particular drug, and its response to the drug is then observed. This process is slow and expensive, and can take up to 2 months to obtain the results. The next tools which are faster and commonly used for drug resistance detection are commercial genotypic line probe assays (LPA) such as Xpert MTB/RIF, GenoType MTBDRplus. However, such assays only predict resistance for isoniazid and/or rifampicin; they also consider very common mutations related to resistance and are usually low in sensitivity. Due to recent advancements in whole-genome sequencing (WGS) technology, researchers are able to sequence the whole genome of bacteria within a few hours at a tremendously lower cost and extract more information about genotypic resistance markers. Mutation libraries and databases based on single nucleotide polymorphisms (SNPs) were constructed to diagnose resistance, for instance TBProfiler [19] and TBDreaMDB [47]. However, such libraries are not able to capture the association between different resistance markers in predicting resistance; moreover, the libraries constructed are highly dependent on the diversity of the samples sequenced.

Machine learning and deep learning has been successfully applied to different areas of research such as natural language processing, computer vision, and more recently bioinformatics, including genome-wide association studies (GWAS) and cancer prediction [36, 53]. Leveraging the abundance of WGS data, researchers have also applied machine learning tools to predicting antibiotic resistance in MTB; for instance, in [61] they applied machine learning models such as logistic regression, support vector machines, and naive Bayes classifier in predicting resistance of 8 drugs from WGS data; in [15], the authors presented a deep learning model, multi-task wide and deep neural network (MD-WDNN) for multi-drug resistance prediction, where they combined logistic regression and neural networks with various deep learning techniques such as dropout and batch normalization.

In our work, we hypothesize that machine learning models are robust in predicting resistance across diversified dataset and interesting findings can be discovered such as gene and mutation markers related to resistance by analyzing these models. Therefore, the goal of this study is to construct powerful machine learning models in predicting multiple drugs resistance for a given Tuberculosis(TB) sample, which are robust to highly noisy data. Moreover, inference analyses are conducted to investigate the genes and mutations which are highly related to resistance.
1.3 Connecting Both Problems

Both problems are challenging and significant in the study of bacterial pathogens in epidemiology and human health. Especially in antibiotic treatment regimens, it is crucial to prescribe the correct antibiotic drugs in order to avoid strains infecting the patients from achieving multidrug resistance and any unnecessary side effects of the drugs [17]. In order to achieve this, we need to identify the different types of bacterial strains infecting each patient and prescribe the right drugs to eliminate these strains effectively, in which the former task is precisely the strain diversity problem and the latter one would rely on accurate machine learning models for resistance prediction. The methods in both problems in general contribute to an advanced platform where clinicians may sequenced the DNA of the patients and diagnose the patients efficiently using available powerful big data and cloud technologies. Due to the increase in availability of powerful computers such as clusters and data warehouse, we believe that this platform will be reasonably accessible for the public. Although the bacteria in study for both problems are different in this thesis, the methods and techniques are generalizable and may serve as references for similar problems in different pathogens.
Chapter 2

Strain Diversity Problem

Several investigations on bacterial pathogens such as *Borrelia* have demonstrated within-host mixed infections which were not fully understood before the availability of advanced genome sequencing technologies. In this chapter, we introduced the first paradigm for extracting MLST information in the presence of within-host heterogeneity, which is also able to simultaneously take multiple samples into account. Our method is based on mixed integer linear programming (MILP), and consists of two main stages which solve 2 different problems each, the allele diversity problem and the strain diversity problem.

The allele diversity problem starts by filtering the short reads in each sample, selecting those that closely match known (database) alleles in at least one of the housekeeping genes in the MLST scheme, and then assigns fractional abundances to each allele of each gene, ensuring that as few such alleles as possible are used that can explain the data well. In the strain diversity problem, it assigns combinations of these alleles, with corresponding proportions, to each sample, while maximizing the usage of known strains and minimizing the number of novel strains, a parsimony-based approach that has been shown to perform well in related contexts [16].

We evaluate our approach on simulated samples and find that it is accurate in identifying both the fractional allele composition at each housekeeping gene, as well as the complete strain types present in each sample. We then apply it to a dataset of 24 real tick samples containing *B. burgdorferi* extracted via whole-genome capture, and find a substantial amount of diversity, as well as a number of new strains. Our work demonstrates that accurate strain typing via MLST is achievable from WGS data even in the presence of substantial within-host heterogeneity, and provides a robust and reproducible pipeline for performing this strain typing.

2.1 Related Works

We hereby present some methods which we compared with for solving both the allele diversity problem and strain diversity problem.
2.1.1 Kallisto

We compared our method for solving the first stage of the problem, the allele diversity problem to Kallisto. Kallisto is an RNA-seq quantification program which pseudoaligns reads to a reference, constructing a list of transcripts which are compatible with each read and quantifies the abundance of each transcript [7]. Although we are not dealing with RNA-seq data, Kallisto can be implemented on our data by considering the reference as the database of alleles of a gene and the list of transcripts as the alleles identified which the reads are most likely originated from.

Kallisto performs pseudoalignments which do not involve actual alignment i.e. matching each nucleotide base of a read to the reference and it only generates a list of transcripts which the reads could have originated from. A pseudoalignment can be obtained through fast hashing of the $k$-mers of the reference transcriptome de Bruijn graph (T-DBG). Therefore, this gives Kallisto an edge over mapping-based approaches such as TopHat, Bowtie.

![Figure 2.1: (Upper) T-DBG and 2 transcripts. (Bottom) $k_1, k_2, k_3, k_4$ highlighted in green are $k$-mers of a read. $k_1$ and $k_2$ are associated to the compatibility class which consists of transcript A and transcript B. $k_3$ and $k_4$ are associated to the compatibility class consists of transcript A. The read is associated to an equivalence class consisting of only transcript A. Non-informative $k$-mers such as $k_1, k_2$ are skipped which speeds up the process.](image)

Kallisto is a $k$-mer based approach (Figure 2.1). First of all, it generates a T-DBG where each node $v$ in T-DBG is a $k$-mer. Each transcript (allele in our case) is a colored path on the T-DBG and a path covering of the T-DBG is a set of colored paths in which the union of it covers all edges of T-DBG. In their work, they define $k$-compatibility class as a set of transcripts associated to a particular $k$-mer or a node $v$ in T-DBG. Breaking down an error-free read to $k$-mers, a $k$-mer of the read is associated to a $k$-compatibility class and the read is associated to an equivalence class, which is the intersection of the $k$-compatible classes of all $k$-mers of the read. The equivalence class of a read shows a list of transcripts.
that a read could potentially originated from and it is also used for abundance estimation of the transcripts.

The abundance estimation part of Kallisto involves maximizing a likelihood function using the Expectation-Maximization (EM) algorithm. The likelihood function used is:

\[ L(\alpha) \propto \prod_{e \in E} \left( \sum_{t \in e} \frac{\alpha_t}{l_t} \right)^{c_e} \]

where \( E \) is the set of equivalence classes identified through pseudoalignment, \( t \) is a transcript, \( l_t \) is the (effective) length of transcript \( t \) and \( c_e \) is the number of times equivalence class \( e \) is observed. The parameters estimated are \( \alpha_t \), which is the probability of observing the set of reads from transcript \( t \). The optimal parameters, \( \alpha_t^* \) is then the abundance of transcript \( t \).

### 2.1.2 StrainEst

We compared our method for solving the second stage of the problem, strain diversity problem, to StrainEst. StrainEst is a reference-based method that utilizes single nucleotide variants (SNV) profiles of existing genomes of a particular species to identify coexisting strains within samples and also quantify the abundances of each strain in a mixed metagenomic sample. First of all, a genome in this context is defined as a strain \( S \) and it is the concatenated sequences of alleles of the strain \( S \) described in a MLST framework. StrainEst uses a species representative (SR) to build 2 groups of representative genomes for different purposes. Among hundreds of strains characterized in the MLST scheme, we chose one genome randomly as SR and 99 random genomes are chosen for the remaining genomes.

The first group of representative genomes is used to generate SNV profiles which are representative of the species. Among all genomes \( G_1, G_2, \ldots \), genomes with Mash distance > 0.1 from SR are filtered and the remaining genomes are clustered using a complete linkage hierarchical clustering method, where for each genome cluster, the genome with the smallest average distance from other members of its cluster is chosen as a representative genome \( R_i \). Hence, a list of representative genomes \( R_1, R_2, \ldots \) are chosen and can be regarded as genomes which represent the variability of the species the best. Next, each \( R_i \) is mapped to SR to generate a SNV profile \( SNV_i \), a vector of changes in nucleotide bases. Doing this for all \( R_i \) generates a SNV matrix. This matrix is used later for the modeling process of abundance estimation.

The second group of representative genomes is used for alignment, where metagenomic reads are mapped onto. The reason of using a set of representative genomes for alignment instead of SR is to avoid bias towards strains which are closely related to SR. The procedure of selecting this group of genomes are similar as the SNV profiling group, but the number of

\[ ^1 \text{The maximum number of genomes that StrainEst takes as input is 100.} \]
clusters formed can be set by the user where higher number of clusters increases sensitivity but requires more computational power. Next, the representative genomes $A_1, A_2, \ldots$ are aligned against SR using Bowtie2 to obtain consensus sequences (MA). Given a metagenomic sample (MG), its raw reads are aligned with MA using Bowtie2 and the frequency of occurrences of the variants at SNV positions identified in the SNV profiling group are extracted.

The information about variants extracted from the alignment of metagenomic reads with MA and the SNV profiles were used to predict the relative abundances of each representative genomes $R_i$ using a Lasso regression model.

2.1.3 DESMAN

We will briefly explain another strain profiling tool, DESMAN which was implemented for resolving strain variants in metagenomics. DESMAN (De Novo Extraction of Strains from Metagenomes) is a non-reference based approach in resolving strain haplotypes and frequencies directly from metagenomic reads. When references of a bacteria species are not well documented, a de-Novo based method such as DESMAN is preferred in resolving strain variants and abundance estimation.

First of all, DESMAN co-assembles metagenomic reads into contigs and the contigs are binned into metagenome-assembled genomes (MAGs) using CONCOCT binning algorithm. Each MAG bin can be thought of genomes which contains highly similar strains and it is constructed leveraging varying coverage of multiple co-assembled contigs across a large set of strains present in different metagenome samples. Reads are also mapped back to the contigs. In order to identify strain variants, DESMAN first identifies core genes within each MAG. There are two types of core genes: single-copy core genes (SCG) and single-copy core species genes (SCSG). SCG are genes which are known to be core in all bacteria and archaea in [4], in the absence of reference genomes. If reference genomes of some species are available, the genes within those genomes are denoted as SCSG. Given the reads which were previously mapped to the contigs, DESMAN computes the base frequencies at each position of all SCG and SCSG. These base frequencies are used to determine the number of strains present in each sample, their abundances and their exact sequences in the core genes. Once the number of strains, their abundances and core gene sequences are identified, the second component of DESMAN is to determine the remaining accessory genes of each strain by inferring through variant base frequencies and gene coverage across samples.

2.1.4 Comparison and Comments

Kallisto [7] does not pin-point the exact coordinates where the reads are originated from, which possibly lead to high number of false alleles, which we will observe in the Results section.
De-novo methods, such as DESMAN [44], cannot take advantage of known reference strains, alleles and are likely to be confounded by the high similarity observed between strain types. StrainEst [2] is a reference-based approach which can be compared to our method, however it only considers a limited set of reference genomes and their diversity models are not well adapted to handle the very high similarity between strain types, which is the case for *Borrelia* strains. Moreover, none of the reference-based methods consider the detection of novel strain types. A possible route to bypassing these issues is to extend the MLST approach to use NGS data, a research avenue that has been actively explored lately [62, 25].

However, so far the vast majority of MLST methods have been designed without accounting for within-host diversity, and do not provide accurate results in this context.

## 2.2 Methods

### 2.2.1 Overview

**Terminology.** An *MLST scheme* is composed of a set of loci together with a database of known alleles for each locus [42]. An *allele distribution* for a given locus is a set of alleles for this locus together with a proportion assigned to each allele; the proportions must be non-negative and add up to 1. A *strain type* is an assignment of a specific allele to each gene of the MLST scheme. A *strain type distribution* is a set of strain types together with a proportion assigned to each strain type; the proportions must once again be non-negative and add up to 1. A *sample* is a WGS dataset obtained from a single host that contains the sequence data from one or several pathogen strains present in the host (see Fig. 2.2).

![Figure 2.2: A dataset with two samples and an MLST scheme of three loci (genes clpA, clpX, nifS). The strain type distributions require 5 different strains as the strain (clpA_1,clpX_1, nifS_7) appears in both distributions.](image)

---

**10**
Data. In the present work we use the traditional \textit{B. burgdorferi} MLST scheme \cite{41} composed of 8 housekeeping genes having a combined total of 1,726 known alleles\footnote{https://pubmlst.org/borrelia}. For each locus, the various known alleles differ from one another primarily by single nucleotide polymorphisms (SNPs), with small indels also appearing in 4 out of the 8 genes. The number of known strain types is 753.

Problems and contribution overview. The problems we address in this work take as input (1) an MLST scheme together with databases of known alleles and strain types and (2) WGS data for a set of samples, that are mapped, using a short-read mapper of choice, onto the database of known alleles for the provided MLST scheme. It then proceeds in two stages, each addressing a specific problem, which we solve using MILPs:

- The Allele Diversity Problem. For a given sample and a given locus of the MLST scheme, given the mappings of DNA reads onto the known alleles for this locus, detect the alleles present in the sample and the corresponding allele distribution.

- The Strain Diversity Problem. Given a set of samples and an allele distribution for each locus at each sample, compute a strain type distribution per sample that requires the smallest number of novel strain types \textit{among all considered samples}, which are closely distant from known strains.

2.2.2 The Allele Diversity Problem

We split the Allele Diversity Problem into two subproblems: 1) allele detection in a sample and 2) computing the proportion of each identified allele. 

The Allele Detection Problem. We formulate the problem of allele detection as a variant of the Set Cover problem as follows. The input of the Allele Detection Problem (ADP) is composed of a set of \( m \) reads \( \mathcal{R} = \{r_1, \ldots, r_m\} \), a set of \( n \) alleles \( \mathcal{A} = \{a_1, \ldots, a_n\} \) for the chosen locus, and a set of mappings of the reads onto the alleles, encoded by a matrix \( M \), where \( m_{ij} \) is the sum of the normalized Phred scores\footnote{We remind the reader that a Phred score of \( Q \) implies that the corresponding base has a probability \( 10^{-Q/10} \) of being incorrect, so a low Phred score reduces the confidence that the observed mismatch is actually a true mismatch. Phred quality scores vary from 33 to 126.} of the mismatched bases in the mapping of read \( r_i \) onto allele \( a_j \) (we set it to \( \infty \) if \( r_i \) does not map onto \( a_j \)). For instance, if read \( r_i \) maps to allele \( a_j \) with 2 mismatches with base quality scores of 60 and 80, respectively, then \( m_{ij} = \frac{60-33}{126-33} + \frac{80-33}{126-33} = 0.796 \). Each allele \( a_j \) implicitly defines a subset of \( \mathcal{R} \) (the reads aligning with the allele), with each read \( r_i \) being weighted by \( m_{ij} \). Informally, we then aim at selecting a subset of alleles covering the set of reads, while minimizing the
sum of the number of required alleles and the sum of the corresponding weights. The ADP is thus very similar to the Uncapacitated Facility Location Problem, and we discuss this observation in the Appendix.

Formally, we define an edge-weighted bipartite graph whose vertex set is $\mathcal{R} \cup \mathcal{A}$ and whose incidence matrix is $M$. A read cover is a subset of edges of this graph such that each read belongs to exactly one edge; the cost of a read cover is the number of allele vertices it is incident to plus the sum of the weights of the edges in the cover. The ADP aims at finding a read cover of minimum weight, the allele vertices incident on the edges of the cover representing the selected alleles.

**Theorem 1.** *The Allele Detection Problem is NP-hard.*

First of all, we transform the Allele Detection Problem into a decision problem as follows: Given inputs (1) a set of reads $\mathcal{R} = \{r_1, r_2, \ldots \}$, (2) a set of $n$ alleles $\mathcal{A} = \{a_1, \ldots, a_n\}$ for the chosen locus and (3) a target cost $C$, is there a read cover whose cost is at most $C$?

To show its NP-completeness we use a reduction from the 3-dimensional matching problem, formulated as follows: Given inputs (1) three disjoint sets, $X$, $Y$ and $Z$, each of size $n$, (2) a collection of $m \geq n$ triples $\{(x_i, y_i, z_i)\}_{1 \leq i \leq m}$, with $x_i \in X, y_i \in Y, z_i \in Z \forall i$, is there a three-dimensional matching, i.e. a subset $M \subseteq \{1, 2, \ldots, m\}$ with $|M| = n$ such that $\bigcup_{i \in M} \{x_i, y_i, z_i\} = X \cup Y \cup Z$ (i.e. the corresponding triples contain each element exactly once)?

**Proof:**

- It is clear that this problem is in NP, since given a particular subset $\mathcal{B}$ of alleles, it is easy to check, in polynomial time, that they form a read cover and that the cost of this read cover is at most $C$.

- From a given input to the 3-dimensional matching problem we construct an instance of the allele diversity problem (with reads over a binary alphabet) as follows:

Select $3n$ binary strings, each of length $L - 2$, such that the Hamming distance between any pair of them is at least 1 (for instance by using a Hamming code), pad them with a single 0 at the beginning and at the end to a total length $L$, and associate exactly one of them to each of the elements of $X \cup Y \cup Z$. For an element $e$ denote by $s(e)$ the associated binary string.

Let the set of reads be $\mathcal{R} := \{s(e)\}_{e \in X \cup Y \cup Z}$. There are exactly $3n$ reads. We assume that the base quality scores at all the positions of all the reads equal the maximum possible base quality score, so that after scaling, each mismatch contributes exactly 1 to the objective function.

---

$^4$The proof was initially written by L. Chindelevitch, edited and elaborated by me.
For each triplet \((x_i, y_i, z_i)\) we construct an allele of length \(5L\), \(s(x_i)1^Ls(y_i)1^Ls(z_i)\), where \(1^L\) is the string of \(L\) 1’s and the product is via concatenation. In other words, \(\mathcal{A} := \{s(x_i)1^Ls(y_i)1^Ls(z_i)\}_{1 \leq i \leq m}\). Lastly, we set \(C := n\). Since \(L\) can be chosen to be linear in \(n\), and the codewords can be constructed in linear time with respect to \(n\) as well, this reduction can be performed in polynomial time.

- **Claim:** There is a read cover with cost at most \(n\) if and only if there is a three-dimensional matching.

  - **Proof\(\Leftarrow\):** By construction, for any \(e \in X \cup Y \cup Z\), \(s(e)\) maps to any allele \(\in \mathcal{A}\) with no error if a triplet \(\in X \times Y \times Z\) contains \(e\), and to any allele with cost at least 1 if a triplet doesn’t contain \(e\) (since the Hamming distance between any pair of encodings of the elements differ by at least 1, and a mapping overlapping with the “sentinel” string \(1^L\) also incurs at least one mismatch due to the presence of a 0 at either end). It follows that if we have a three-dimensional matching \(M^* = \{(x_i^*, y_i^*, z_i^*)\}_{1 \leq i \leq n}\) of size \(n\), we have a read cover of cost at most \(n\) by picking alleles from \(A\) based on \(M^*\), as there are \(n\) alleles and no errors incurred on any \(s(x_i^*), s(y_i^*), s(z_i^*)\).

  - **Proof\(\Rightarrow\):** On the other hand, suppose we have a read cover of cost at most \(n\). Since each allele can explain at most 3 reads with no errors, the case of no errors requires at least \(n\) alleles for the \(3n\) reads, and the only way that this can be achieved with exactly \(n\) alleles is if each allele explains three different reads with no errors, meaning that the triplets corresponding to the \(n\) chosen alleles form a three-dimensional matching. If there are \(k > 0\) errors, then at most \(n - k\) alleles can be used. It follows that at most \(3(n - k)\) of the \(3n\) reads incur no errors, and the remaining \(3k\) reads must each incur at least one error, so the number of errors incurred must be at least \(3k\), contradicting the hypothesis that there were only \(k\) errors.

**Intuition** Before describing our ILP, we comment on the relevance of our formulation for selecting a set of alleles from short reads. Our objective function aims to minimize the sum of the number of alleles and the weight of each read based on the Phred scores; the latter part aims at explaining the data (reads) using as few errors/mismatches as possible, accounting for the base quality score of the mismatches, while the former part ensures that an allele is not introduced unnecessarily to reduce the contribution of the mismatches and their quality for a small number of reads. Our experiments on simulated data show that this objective function leads to extremely accurate results.

**An Integer Linear Program for the Allele Detection Problem.** First we introduce the following notation: \(R_j = \{r_i : m_{ij} \neq \infty\}\) represents the set of reads mapping onto allele \(a_j\) (i.e. covered by allele \(a_j\)), and \(M_i = \{m_{ij} | 1 \leq j \leq n\} - \{\infty\} = \{q_{i1}, ..., q_{i|M_i|}\}\) represents the distinct summed Phred scores for read \(r_i\). The decision variables of the ILP are:
• $x_j = 1$ if allele $a_j$ is chosen, and $0$ otherwise.

• $y_{ik} = 1$ if a mapping of read $r_i$ with score $q_{ik}$ is chosen, and $0$ otherwise.

The objective function is

\[
\min \left( \sum_{i=1}^{\mid R \mid} \sum_{k=1}^{\mid M_i \mid} q_{ik} \cdot y_{ik} + \sum_{j=1}^{n} x_j \right).
\]

Finally, the constraints of the ILP are the following ones:

• If $y_{ik} = 1$, then there must be some chosen allele $a_j$ onto which $r_i$ maps with score $q_{ik}$.

• There is a unique score with which read $r_i$ can be mapped onto the selected alleles.

These constraints can be translated as follows:

\[
\sum_{\{j \mid r_i \in R_j, m_{ij} = q_{ik}\}} x_j \geq y_{ik} \forall i, k \quad \sum_{k=1}^{\mid M_i \mid} y_{ik} = 1 \forall i.
\]

Post-processing. If the above 0-1 ILP has multiple optimal solutions, we resort to a likelihood based method to select one, precisely GAML [6], a probabilistic model for genome assembly. Given a set of solutions where each solution represents a set of alleles, we measure the likelihood of observing the set of reads given a solution and pick the solution which maximizes the likelihood criterion. If there are multiple solutions maximizing the likelihood criterion, we pick one arbitrarily.

Computing allele proportions. Finally, once the alleles have been identified for a given locus, we compute the proportion of each allele. The principle is to assign a weight to each allele based on the read mappings (edges) selected by the ILP, and to normalize these weights to obtain proportions. First, we filter out all reads that map equally well (i.e. with the same score $k$) onto all selected alleles. Then every chosen allele gets an initial weight of $0$. Next, for every non-discarded read, say $r_i$, we consider all the alleles it maps onto with optimal score (say $q_{ik}$ if $y_{ik} = 1$); assuming there are $h$ such alleles, we increase the weight of each by $1/h$. We then normalize the weights of the alleles to define their respective proportions. As an example, consider the matrix $M = (m_{ij})$ below (restricted to the chosen alleles):

\[
\begin{pmatrix}
2.1 & 2.1 & 2.1 \\
1.0 & 1.0 & 2.0 \\
2.1 & 1.1 & 1.1
\end{pmatrix}
\]

The entry $m_{11}$ can be thought as the mapping of read $r_1$ to allele $a_1$ with normalized quality score 2.1. For the matrix shown, $r_1$ is not used for proportion computation as it
maps equally well to all identified alleles. Allele $a_1$ gets $\frac{1}{2}$ from $r_2$; allele $a_2$ gets $\frac{1}{2}$ from $r_2$ and $\frac{1}{2}$ from $r_3$, which is a total of 1; and allele $a_3$ gets $\frac{1}{2}$ from $r_3$. Summing over all alleles and normalizing, we obtain proportions $\frac{1}{4}$ for $a_1$, $\frac{1}{2}$ for $a_2$, and $\frac{1}{4}$ for $a_3$.

**Alternative Formulation of the Allele Detection Problem**

The seemingly intuitive way to formulate the ADP is to introduce a decision variable $b_{ij}$ representing the assignment of read $r_i$ to allele $a_j$. Here we present the alternative formulation of ADP based on this idea as an integer linear program. Given there are $n$ alleles and $m$ reads, we denote the set of reads $R = \{r_1, ..., r_m\}$ and the set of alleles $A = \{a_1, ..., a_n\}$. Also, $m_{ij}$ is the score for mapping read $r_i$ to allele $a_j$, the entries of the matrix input for ADP as described previously. Next, the decision variables and objective function are as follows:

- $z_j = 1$ if allele $a_j$ is chosen, and 0 otherwise.
- $b_{ij} = 1$ if the mapping of read $r_i$ to allele $a_j$ is chosen and 0 otherwise.
- $\min \left( \sum_{i=1}^{m} \sum_{j=1}^{n} m_{ij} \cdot b_{ij} + \sum_{k=1}^{n} z_k \right)$

The constraints are:

- All reads are covered: $\sum_{j=1}^{n} b_{ij} = 1 \ \forall i$
- If read $r_i$ is assigned to allele $a_j$, then allele $a_j$ must be chosen: $z_j \geq b_{ij} \ \forall i, j$

**Proof of the Equivalence of Both Formulations**

Denote our formulation as (1) and the alternative formulation as (2). We prove that both formulations are equivalent by showing that a feasible solution of (1) can be translated to a feasible solution of (2), with both having the same objective value and vice versa.

("⇒") Prove that a feasible solution of (1) can be translated into a feasible solution of (2). Given a feasible solution $s_1 = (x; Y)$ where $x = (x_j)$ and $Y = (y_{ik})$. We denote a solution for (2), $s_2 = (z; B)$ where $z = (z_j)$ and $B = (b_{ij})$. Also, recall that the matrix input for the ADP is $M = (m_{ij})_{m \times n}$.

- For all $j$, assuming the indices for both refer to the same allele, let $z_j = x_j$.
- For all $i$, due to the 2nd constraint in (1), $\exists! k^*$ such that $y_{ik^*} = 1$ and $y_{ik} = 0 \ \forall k \neq k^*$.
  Denote $H_{ik^*} = \{ j \mid m_{ij} = q_{ik^*} \land x_j = 1, j \in \{1, ..., n\}\}$. Choose an index $j^* \in H_{ik^*}$, let $b_{ij^*} = 1$ and $b_{ij} = 0$ for all $j \in H_{ik^*} - \{j^*\}$. For $j \not\in H_{ik^*}$, let $b_{ij} = 0$.

**Claim:** $s_2$ is a feasible solution for (2) with same objective value as (1).

**Proof:**

- Constraint 1: $\sum_{j=1}^{n} b_{ij} = 1 \ \forall i$. This is true by construction of $b_{ij}$. 


• Constraint 2: \( z_j \geq b_{ij} \forall i, j \). If \( b_{ij} = 1 \), then \( j = j^* \Rightarrow j \in H_{ik^*} \Rightarrow x_j = 1 = z_j \Rightarrow z_j \geq b_{ij} \). The case for \( b_{ij} = 0 \) is trivial as \( z_j \) is always \( \geq 0 \).

• Objective value: The summation of the allele variables is the same in both formulations. For the other summation term:

\[
\sum_{i=1}^{\left| R \right|} \sum_{k=1}^{\left| M_i \right|} q_{ik} y_{ik} = \sum_{i=1}^{\left| R \right|} \left( q_{ik^*} y_{ik^*} + \sum_{k \neq k^*} q_{ik} y_{ik} \right)
\]

\[
= \sum_{i=1}^{\left| R \right|} \left( m_{ij^*} b_{ij^*} + \sum_{j \in \{1, \ldots, n\} - \{j^*\}} m_{ij} b_{ij} \right) \quad (b_{ij} = 0 \, , \, j \neq j^*)
\]

\[
= \sum_{i=1}^{\left| R \right|} \sum_{j=1}^{n} m_{ij} b_{ij}
\]

("⇐") Prove that a feasible solution of (2) can be translated to a feasible solution of (1). Given a feasible solution \( s_2 = (z; B) \).

• Let \( x_j = z_j \)

• For all \( i \), due to first constraint of (2), \( \exists j \in \{1, \ldots, n\} \), say \( j^* \), where \( b_{ij^*} = 1 \). Also, say the corresponding score \( q_{ik^*} = m_{ij^*} \). Let \( y_{ik^*} = 1 \), \( y_{ik} = 0 \) \( \forall q_{ik} \in M_i - \{q_{ik^*}\} \).

**Claim:** \( s_2 \) is a feasible solution for (2) with same objective value as (1).

**Proof:**

• Constraint 1: If \( y_{ik} = 1 \), \( k = k^* \). Due to how we construct the solution, \( \exists j^* \) such that \( b_{ij^*} = 1 \wedge m_{ij^*} = q_{ik^*} \). Due to the 2nd constraint of (2), \( x_j^* = z_j^* \geq b_{ij^*} = 1 = y_{ik^*} \).

The case of \( y_{ik} = 0 \) is trivial.

• Constraint 2: \( \sum_{k=1}^{\left| M_i \right|} y_{ik} = \left( y_{ik^*} + \sum_{k \neq k^*} y_{ik} \right) = (1 + 0) = 1 \)

• Objective value: The summation of the allele variables is the same. For the other summation term:

\[
\sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} b_{ij} = \sum_{i=1}^{m} \left( b_{ij^*} w_{ij^*} + \sum_{j \neq j^*} w_{ij} b_{ij} \right)
\]

\[
= \sum_{i=1}^{m} \left( q_{ik^*} y_{ik^*} + \sum_{k \neq k^*} q_{ik} y_{ik} \right) \quad (\text{as } y_{ik} = 0 \text{ for } k \neq k^*)
\]

\[
= \sum_{i=1}^{\left| R \right|} \sum_{k=1}^{\left| M_i \right|} q_{ik} y_{ik}
\]

**2.2.3 The Strain Diversity Problem**

Once alleles present in each sample and their proportions have been identified, this information is passed to the second stage of the pipeline. Its goal is to compute strain types and proportions in all samples jointly, minimizing the number of novel strains required to
explain the given allele distributions plus an error term measuring the total discrepancy between each given allele proportion and the proportions of strains having this allele. The rationale behind minimizing the number of new strains is driven by parsimony considerations; we would like to explain the data present in all samples using known strains as much as possible. The error terms allow some flexibility to modify the allele proportions by bounding each error to be \( \leq \epsilon \) (in our analysis we set the bound to \( \epsilon = 0.1 \), or 10%).

**Definition and Tractability of The Strain Diversity Problem.** The Strain Diversity Problem (SDP) can be defined as follows. It takes as input four elements: (1) the set \( G_{ij} = \{g_{ij1}, g_{ij2}, \ldots \} \) of all alleles selected for locus \( j \) in sample \( i \) (2) the set \( P_{ij} = \{p_{ij1}, p_{ij2}, \ldots \} \) of proportions of these alleles, (3) a database \( \Omega \) of known strain types, (4) an error bound \( \epsilon \in [0, 1] \). From now, we assume that there are \( \ell \) loci and \( m \) samples.

From this input, we generate, for each sample \( i \) the set of all possible strain types defined as the Cartesian product \( G_i = G_{i1} \times G_{i2} \times \cdots \times G_{i\ell} \) which we denote by \( V_i = \{V_{i1}, V_{i2}, \ldots, V_{iH_i}\} \) with \( H_i = \prod_{j=1}^{\ell} |G_{ij}| \). We also denote by \( K \) the number of strain types that appear in at least one \( V_i \) and we define the set \( S = \{S_1, \ldots, S_K\} \) of all such strain types. We assign a weight \( w_j \) to each \( S_j \in S \), where \( w_j = N \cdot \min_{s \in \Omega} d(s, S_j) \), where \( d \) is the edit distance metric and \( N = \frac{1}{\max_j w_j} \), a normalizing term. These weights measure the distance to the closest known strain, so strains belonging to \( \Omega \) are assigned a weight 0. The intuition behind such weighting scheme is again in the line of a parsimonious approach, where a novel strain introduced should not be too different from the reference strains which are well studied.

A solution to the SDP is fully described by assigning to every strain type \( V_{ih} \) from \( V_i \) a proportion \( \pi_{ih} \) for this strain type in sample \( i \) (where \( \pi_{ih} = 0 \) if the strain type is deemed to be absent from sample \( i \)). A strain type from \( S \setminus \Omega \) is said to be present in a solution if it is given a non-zero proportion in at least one sample; we denote by \( S_n \) the set of such novel strain types. The cost of a solution is then defined as

\[
\sum_{\{h | S_h \in S_n\}} w_h + \sum_{i,j} e_{ij} \tag{2.1}
\]

where the latter term of the cost represents the deviation from the input alleles proportions for sample \( i \) at locus \( j \). This cost function penalizes the introduction of highly different novel strains from known strains and the error introduced in the proportions of the selected alleles.

The SDP aims at finding a solution of minimum cost, *i.e.* one that explains the provided allele distributions as much as possible with known strains and novel strains which are closer to known strains, and also sticks to the desired proportions as closely as possible. As expected, this problem is intractable; its decision version is proven to be NP-complete by a reduction from the 3-partition problem.
Theorem 2. The Strain Diversity Problem is NP-hard. 5

We turn the Strain Diversity Problem into a decision problem by putting the objective function value into the input. This decision problem version is then formulated as follows: Given inputs (1) The set $G_{ij} = \{g_{ij1}, g_{ij2}, \ldots\}$ of all alleles selected for locus $j$ in sample $i$, together with the set $P_{ij} = \{p_{ij1}, p_{ij2}, \ldots\}$ of proportions of these alleles, (2) A database $\Omega$ of known strain types, (3) An error bound $\epsilon \in [0, 1]$, (4) A target cost $C$. Let the set of all possible strain types for sample $i$, the Cartesian product $G_i = G_{i1} \times G_{i2} \times \cdots \times G_{i\ell}$, be denoted by $V_i = \{V_{i1}, V_{i2}, \ldots, V_{ih}\}$ with $H_i = \prod_{j=1}^\ell |G_{ij}|$. For simplicity, assuming the weights for each novel strain are set to 1. The question is: Is there a set of proportions $\pi_{ih}$ for each strain type $V_{ih} \in V_i$ in each sample $i$, satisfying the validity constraint

$$
\nu_{ijk} := \sum_{(i,k)\,|\,g_{ijk} \in V_{ik}} \pi_{ik} \in [p_{ijk} - \epsilon, p_{ijk} + \epsilon] \forall g_{ijk} \in G_{ij},
$$

such that the total cost is at most equal to the target $C$, i.e.

$$
|\{V_{ih}|\pi_{ih} > 0\} - \Omega| + \sum_{i,j,k} |p_{ijk} - \nu_{ijk}| \leq C?
$$

To show its NP-completeness we use a reduction from the 3-partition problem, formulated as follows:

Input: A multiset of $n = 3m$ positive integers $x_1, x_2, \ldots, x_n$ with sum $S = mB$ for some integer $B$, and satisfying $\frac{B}{4} < x_i < \frac{B}{2}$ for each $1 \leq i \leq n$.

Question: Does there exist a partition of this multiset into $m$ disjoint triples $T_1, T_2, \ldots, T_m$, such that the sum of each triple is exactly $B$?

Proof:

- The SDP problem is clearly in NP because given a set of proportions $\pi_{ih}$, it is easy to check, in linear time in the size of the input, that the validity constraints are satisfied, and that the cost is at most $C$.

- Now, we construct an instance of SDP from an instance of 3-partition problem. Let us choose $\epsilon = \frac{1}{4m}$. Let us look at a single sample with two loci, 1 and 2, where the first locus has the $n = 3m$ possible alleles $g_1, g_2, \ldots, g_n$ and the second one has the $m$ possible alleles $h_1, h_2, \ldots, h_m$. Let the target allele distribution for locus 1 be $g_1$ with proportion $\frac{2}{B}$, $g_2$ with proportion $\frac{2}{B}$, ..., $g_n$ with proportion $\frac{2}{B}$, and let the target allele distribution for locus 2 be $h_i$ with proportion $\frac{1}{m}$ for each $1 \leq i \leq m$.

Finally, let the library strains be $\Omega := \emptyset$, and let the target cost be $C := n$. It is clear that the transformation is polynomial in the input size.

5This proof was initially written by L. Chindelevitch, edited and elaborated by me.
• We now prove the following:
  **Claim:** The original 3-partition problem has a “yes” answer if and only if the transformed version (SDP) has cost at most $C$.

• **Proof($\Rightarrow$):** Suppose that there is a partition of the multiset into disjoint triples $T_1, T_2, \ldots, T_m$. Then, for each $j$, we can take the 3 integers, say $x_{j_1}, x_{j_2}, x_{j_3}$, in a triple $T_j$, and use the 3 new strains $(g_{j_1}, h_j)$ with proportion $\frac{x_{j_1}}{S}$, $(g_{j_2}, h_j)$ with proportion $\frac{x_{j_2}}{S}$, and $(g_{j_3}, h_j)$ with proportion $\frac{x_{j_3}}{S}$, to cover the allele $h_j$ completely, with no error (since $\frac{x_{j_1}}{S} + \frac{x_{j_2}}{S} + \frac{x_{j_3}}{S} = \frac{B}{S} = \frac{1}{m}$ by construction). Thus, we need a total of $n = 3m$ strains to cover all the alleles in both loci with no error, so the cost of our solution is indeed $n$.

• **Proof($\Leftarrow$):** Conversely, suppose that there is a solution with cost at most $n$. Note first that because $\epsilon = \frac{1}{4m} = \frac{B}{4Bm} = \frac{B}{4S} < \frac{x_i}{S}$ for any $1 \leq i \leq n$, it follows that no allele in locus 1 can be eliminated from the solution while satisfying the validity constraint. It follows that at least $n = 3m$ strains are needed in any valid solution, so the cost cannot be smaller than $n$.

Furthermore, in order to get a cost of exactly $n$, there needs to be no error on the proportions and exactly $n = 3m$ strains, so each allele in locus 1 must be in exactly one strain. Thus, the allele $h_j$ for locus 2 must be associated with a subset $T_j$ of alleles of locus 1, and the $T_j$ must be disjoint and therefore form a partition of the alleles of locus 1. Since $\frac{B}{4} < x_i < \frac{B}{2}$ for each $i$, we must have exactly 3 alleles in each subset $T_j$, since adding up 2 of the $\frac{x_i}{S}$ results in a sum below the target proportion $\frac{B}{S} = \frac{1}{m}$ for $h_j$, and adding up 4 of them results in a sum above it, thus incurring a non-zero error on the proportions.

By considering the sum of the proportions in each triple $T_j$ containing three alleles, say $g_{j_1}, g_{j_2}, g_{j_3}$, we see that we must have

$$\frac{x_{j_1}}{S} + \frac{x_{j_2}}{S} + \frac{x_{j_3}}{S} = \frac{1}{m} = \frac{B}{S} \Rightarrow x_{j_1} + x_{j_2} + x_{j_3} = B.$$ 

As the $T_j$ are disjoint, this proves that the answer to the original problem is “yes”, completing the proof.

**An MILP for the Strain Diversity Problem.** We now describe an MILP that solves the SDP. The decision variables of the MILP are the following:

- Binary usage variables $a_k$, $1 \leq k \leq K$, where $a_k = 1$ if strain type $S_k$ is chosen to explain the observed allele distribution in at least one of the samples, and 0 otherwise.

- Proportion variables $\pi_{ih}$ encoding the proportion of strain type $V_{ih}$ in sample $i$; their values are constrained to be in $[0, 1]$.  


• Error variables $e_{ijk}$ encoding the absolute error of the observed proportion $p_{ijk}$ of allele $g_{ijk}$ for locus $j$ in sample $i$ from the assigned proportions, in sample $i$, of the strain types containing this allele; their values are constrained to be in $[0, \epsilon]$.

The objective function of the MILP is

$$\min_{a, \pi, e} \left( \sum_{\{k \mid S_k \notin \Omega\}} w_k a_k + \sum_{i,j,k} e_{ijk} \right)$$

Finally the constraints of the MILP are the following:

• For any allele $g_{ijk} \in G_{ij}$, the sum of the proportions of the strain types from $V_i$ that contain this allele, denoted $\nu_{ijk}$, belongs to $[p_{ijk} - \epsilon, p_{ijk} + \epsilon]$.

• For each sample $i$, the strain type proportions must form a distribution: $\sum_{h=1}^{H_i} \pi_{ih} = 1$.

• If the assigned proportion for some strain type $V_{ih} = S_k$ in a sample $i$ is non-zero, then $S_k$ must be chosen: $a_k \geq \pi_{ih}$.

• Conversely, if a strain is chosen, then it must be assigned a non-zero proportion:

$$0 \leq a_k - \frac{1}{|\{\pi_{ih} \mid V_{ih} = S_k\}|} \cdot \sum_{\{(i,h) \mid V_{ih} = S_k\}} \pi_{ih} \leq 1 - \delta$$

where $\delta$ is a tolerance chosen to match the smallest allowed proportion; we use $\delta = 0.001$. This constraint is needed because the binary decision variables for the usage of existing strains have coefficient 0 in the objective function, so setting these variables to 1 will not incur any cost in the objective function. If we do not impose such a constraint, we could end up with an incorrect solution where some existing strains have zero proportions, while the strain usage variables are set to 1, which would then need to be post-processed. Including the constraint eliminates the possibility of such a spurious solution being produced.

• The absolute error between the input proportion and the assigned proportion for allele $g_{ijk}$ for locus $j$ in sample $i$: $e_{ijk} = |p_{ijk} - \nu_{ijk}|$. This is encoded by the following 2 constraints: $e_{ijk} \geq T_{ijk} - p_{ijk}$ and $e_{ijk} \geq p_{ijk} - T_{ijk}$ where $T_{ijk} = \sum_{\{k \mid g_{ijk} \in V_i\}} \pi_{ik}$. Note that since $e_{ijk}$ is part of the objective function, it will be equal to the error in any optimal solution.

2.2.4 Implementation

All scripts are written in Python 2.7 and there approximately 5543 line of codes. Both ILPs are formulated and solved using the Python API of IBM’s CPLEX 12.6.3.0. For the ADP, each sample and each locus may require a different number of variables in the ILP. To
evaluate the practical resources requirements of our ILP, we choose the sample SRR2034336, which has the largest number of reads among our samples. The average number of variables across each gene for this sample is 20,112, the maximum RAM usage is \( \sim 1.5 \text{GB} \), and the time taken for all 8 genes is \( \sim 33 \) minutes on a 4 CPUs Intel® Xeon® machine. For the MILP solving the SDP on all 24 samples, there are a total of 21,885 variables, with 10,682 strain type variables, 10,795 proportion variables and 408 error variables. Due to the computational complexity of the MILP, we output a solution as long as the relative gap tolerance \(^6\) is within 10% and after a time limit of 24 hours.

### 2.2.5 Data simulation

Given the absence of benchmarks available for estimating diversity at the level of precision considered in this work, we conducted several simulations to justify our methods solving the ADP and SDP. All reads are simulated as using ART \(^{27}\), following the characteristics of the reads from the real data set described in Section 2.3.2.

**ADP simulation.** For each locus of the *Borrelia* MLST scheme, we drew a random number \( k \in [2, 7] \), selected a random allele from the database and selected \( k - 1 \) other alleles, each at edit distance from the first chosen one at most \( d \) (a given parameter). Next, we randomly assigned proportions to each selected allele, which sums up to 1, then generate reads with coverage \( c \). To align the simulated reads to the alleles of the database, we used Bowtie v0.12.7 \(^{37}\). We used parameters \( c \in \{30, 100, 300\} \) and \( d \in \{5, 10, 15, 20, 25\} \) and for each combination of parameters we ran 40 simulations. For this experiment, we compared our results with the results obtained with Kallisto \(^{7}\) a recent method for isoform abundance estimation that has also been applied to metagenomics.

**SDP simulation** For this simulation we selected random strain types distributions and tested the ability of our SDP method to recover the true diversity given perfect alleles calls. We considered 5 different mechanisms to generate strain types distributions. EvoMod1: We select a random existing strain \( S \), which is then mutated \( m = 2 \) times to obtain a new strain \( S' \), where each mutation results in an allele which has edit distance at most \( d = 15 \) from the original allele in \( S \). The total number of strains simulated is 2 (1 existing and 1 novel). EvoMod2: We repeat EvoMod1 in parallel from two starting existing strains. The total number of strains simulated is 4 (2 existing and 2 novel). EvoMod2e/EvoMod2n: We apply EvoMod2 then remove a random existing/novel strain. EvoMod3: we apply EvoMod2, then apply a recombination event on two randomly chosen strains out of the 4 available strains. For all experiments, once the strains are chosen, we assign random proportions to each.

**Full pipeline simulation.** We generated strain type distributions as in the SDP simulations above, then generated reads as in the ADP simulations. The generated reads were

\(^6\)The relative gap tolerance is defined as the gap between the best integer objective value and the best node objective value (which is related to the LP relaxation of the problem)
then fed to the ADP solver, and the ADP results were provided as input to the SDP solver. We compared our pipeline with strainEST [2], a recent method to estimate the strain composition and abundance in metagenomics datasets. However, strainEST does not predict novel strain types. Hence, to complement EvoMod1, 2, 2e and 2n, we added an additional simulation where we randomly pick \( k = \{1, 2\} \) existing strains and assign them random proportions.

**Statistics.** For each experiment, we recorded the following statistics: Precision, Recall and Total Variation Distance. Precision and recall are defined as usual, respectively \( \frac{TP}{TP + FP} \) and \( \frac{TP}{TP + FN} \), where \( TP, FP, FN \) are the number of true positive calls, false positive calls, and false negative calls. The Total Variation Distance (TVD) [38, p. 50] is defined as
\[
TVD = \frac{1}{2} \sum_{a \in S} |Pred(a) - True(a)|,
\]
where \( Pred \) and \( True \) are the predicted distribution and the true distribution, respectively, and \( S \) is the set of all possible outcomes. TVD basically describes the average amount of distribution to ‘move’ from \( Pred \) to \( True \) and conversely.

The statistics described above rely on a stringent measure of accuracy in calling alleles, strain types or proportions. For example, a novel strain type called which differs from the true simulated strain type by a single SNP would be considered as a False Positive. To account for this, we considered 3 additional statistics: Earth-Mover’s distance (EMD), soft-precision and soft-recall. Soft precision and soft recall are similar to precision and recall, however, a strain is considered a TP if it differs from the true strain type by at most 5 SNPs. EMD [43] is similar in principle to TVD but is finer as it considers the edit distances between strains and is commonly used in genomics to evaluate haplotype reconstruction methods [35]. Formally, EMD is defined as follows: Let \( T = \{T_i, t_i\}_{i=1}^{T} \) be the set of true strains \( T_i \) with its respective proportion \( t_i \) and \( P = \{P_j, p_j\}_{i=1}^{P} \) be the set of predicted strains \( P_j \) with predicted proportion \( p_j \). Also, denote \( d_{ij} \) as the edit distance between \( T_i \) and \( P_j \). EMD measures the minimum amount of work to transform the predicted distribution defined by \( P \) into true distribution defined by \( T \), where the amount of work is also weighted by distances \( d_{ij} \). It involves solving a transportation problem which can be modeled as a network flow problem, where the problem is to obtain a flow \( F = \{f_{ij}\} \) which minimizes
\[
\sum_{ij} f_{ij} d_{ij}
\]
and the \( EMD = \frac{\sum_{ij} f_{ij} d_{ij}}{\sum_{ij} f_{ij}} \). It can be thought as a network flow problem in the following sense: we assign fractions of each \( P_i \) to any \( T_j \) (flow \( f_{ij} \)), constrained by the maximum amount of fractions that \( P_j \) and \( T_j \) can give and take, which is \( p_i \) and \( t_j \) (capacities), where the assignments \( f_{ij} \) are weighted by \( d_{ij} \).

In our case where the distribution of proportions and edit distances are bounded by a maximum value (hence a finite metric space), there is a maximum value to the EMD. Imagine each strain in \( P \) differs from each strain in \( T \) at the largest possible edit distance, say \( D \), then \( EMD = \frac{x_1 D + x_2 D + \ldots + x_n D}{x_1 + \ldots + x_n} \) where \( n \) is the number of strains in \( P \) and \( x_i \) is the proportion of predicted strain \( S_i \) in \( P \). Note that \( \sum_{i=1}^{n} x_i = 100 \), the total proportion. Hence, the maximum possible \( EMD = D \). \( D \) is the maximum edit distance between strains,
theoretically it is around $8 \times 600 = 2400$ as a strain is defined by a set of 8 genes here and 600 is the average length of a gene. Clearly, the least value for EMD is 0 when $P = T$.

2.3 Results

2.3.1 Simulated Data

**ADP simulation** Table 2.1, table 2.2 and table 2.3 show the comparison of our method and Kallisto. Overall, our method obtained very high precision and recall statistics. As Kallisto introduces a large number of alleles with extremely small abundances, we only retained alleles with larger than 1% proportion. Our method performs better in terms of precision and comparable in TVD, while Kallisto recall is better. Precision shows the confidence of the method in calling alleles which are indeed true alleles; recall shows the ability of the method in recovering true alleles. The reason Kallisto has a higher recall could be the case where Kallisto infers much more alleles, hence has a higher chance to cover all of the true alleles. Precisely the same reason leads to a lower precision for Kallisto as it is calling more false positives. Gene-by-gene boxplots for our method and Kallisto are available in the Appendix.

<table>
<thead>
<tr>
<th>Precision</th>
<th>clpA</th>
<th>clpX</th>
<th>nifS</th>
<th>pepX</th>
<th>pyrG</th>
<th>recG</th>
<th>rplB</th>
<th>uvrA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Our method</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVG</td>
<td>0.99</td>
<td>0.98</td>
<td>0.96</td>
<td>0.96</td>
<td>0.97</td>
<td>0.98</td>
<td>0.99</td>
<td>0.98</td>
</tr>
<tr>
<td>SD</td>
<td>0.009</td>
<td>0.012</td>
<td>0.024</td>
<td>0.016</td>
<td>0.024</td>
<td>0.013</td>
<td>0.007</td>
<td>0.011</td>
</tr>
<tr>
<td><strong>Kallisto</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVG</td>
<td>0.97</td>
<td>0.94</td>
<td>0.89</td>
<td>0.93</td>
<td>0.93</td>
<td>0.89</td>
<td>0.95</td>
<td>0.93</td>
</tr>
<tr>
<td>SD</td>
<td>0.014</td>
<td>0.014</td>
<td>0.027</td>
<td>0.03</td>
<td>0.02</td>
<td>0.021</td>
<td>0.012</td>
<td>0.023</td>
</tr>
</tbody>
</table>

Table 2.1: Average (AVG) and standard deviation (SD) of precision for each gene of the *Borellia* MLST scheme across all parameters combination.

<table>
<thead>
<tr>
<th>Recall</th>
<th>clpA</th>
<th>clpX</th>
<th>nifS</th>
<th>pepX</th>
<th>pyrG</th>
<th>recG</th>
<th>rplB</th>
<th>uvrA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Our method</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVG</td>
<td>0.95</td>
<td>0.94</td>
<td>0.90</td>
<td>0.94</td>
<td>0.92</td>
<td>0.95</td>
<td>0.94</td>
<td>0.96</td>
</tr>
<tr>
<td>SD</td>
<td>0.022</td>
<td>0.027</td>
<td>0.05</td>
<td>0.034</td>
<td>0.032</td>
<td>0.028</td>
<td>0.043</td>
<td>0.026</td>
</tr>
<tr>
<td><strong>Kallisto</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVG</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.98</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>SD</td>
<td>0.004</td>
<td>0.005</td>
<td>0.003</td>
<td>0.006</td>
<td>0.006</td>
<td>0.011</td>
<td>0.006</td>
<td>0.005</td>
</tr>
</tbody>
</table>

Table 2.2: Average (AVG) and standard deviation (SD) of recall for each gene of the *Borellia* MLST scheme across all parameters combination.

**SDP and full pipeline simulation** The results are presented in Table 2.4, 2.5 and 2.6. Based on table 2.4, given perfect input data, our SDP algorithm performed extremely...
Table 2.3: Average (AVG) and standard deviation (SD) of TVD for each gene of the *Borellia* MLST scheme across all parameters combination.

<table>
<thead>
<tr>
<th>Gene</th>
<th>clpA</th>
<th>clpX</th>
<th>nifS</th>
<th>pepX</th>
<th>pyrG</th>
<th>recG</th>
<th>rplB</th>
<th>uvrA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Our method</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVG</td>
<td>0.077</td>
<td>0.080</td>
<td>0.119</td>
<td>0.087</td>
<td>0.110</td>
<td>0.082</td>
<td>0.089</td>
<td>0.069</td>
</tr>
<tr>
<td>SD</td>
<td>0.015</td>
<td>0.010</td>
<td>0.039</td>
<td>0.024</td>
<td>0.019</td>
<td>0.028</td>
<td>0.030</td>
<td>0.020</td>
</tr>
<tr>
<td><strong>Kallisto</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVG</td>
<td>0.029</td>
<td>0.041</td>
<td>0.085</td>
<td>0.046</td>
<td>0.0047</td>
<td>0.068</td>
<td>0.032</td>
<td>0.05</td>
</tr>
<tr>
<td>SD</td>
<td>0.011</td>
<td>0.015</td>
<td>0.028</td>
<td>0.022</td>
<td>0.018</td>
<td>0.018</td>
<td>0.011</td>
<td>0.022</td>
</tr>
</tbody>
</table>

Table 2.4: Average and standard deviation of different statistics for each evolutionary mechanisms for SDP simulation. This simulation assumes perfect input data from ADP. EM represents EvoMod.

<table>
<thead>
<tr>
<th>Evolutionary Mechanism</th>
<th>Soft-Precision</th>
<th>Soft-Recall</th>
<th>EMD</th>
<th>Precision</th>
<th>Recall</th>
<th>TVD</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM1</td>
<td>0.98 ± 0.11</td>
<td>0.96 ± 0.13</td>
<td>0.64 ± 1.7</td>
<td>0.85 ± 0.28</td>
<td>0.86 ± 0.23</td>
<td>0.15 ± 0.29</td>
</tr>
<tr>
<td>EM2</td>
<td>0.96 ± 0.12</td>
<td>0.98 ± 0.076</td>
<td>0.71 ± 1.18</td>
<td>0.81 ± 0.21</td>
<td>0.88 ± 0.14</td>
<td>0.17 ± 0.22</td>
</tr>
<tr>
<td>EM2e</td>
<td>0.98 ± 0.11</td>
<td>0.97 ± 0.1</td>
<td>0.34 ± 0.81</td>
<td>0.91 ± 0.20</td>
<td>0.92 ± 0.17</td>
<td>0.1 ± 0.23</td>
</tr>
<tr>
<td>EM2n</td>
<td>0.96 ± 0.13</td>
<td>0.95 ± 0.12</td>
<td>0.6 ± 1.35</td>
<td>0.86 ± 0.23</td>
<td>0.88 ± 0.16</td>
<td>0.14 ± 0.25</td>
</tr>
<tr>
<td>EM3</td>
<td>0.90 ± 0.17</td>
<td>0.88 ± 0.13</td>
<td>4.6 ± 7.58</td>
<td>0.76 ± 0.21</td>
<td>0.76 ± 0.17</td>
<td>0.22 ± 0.24</td>
</tr>
</tbody>
</table>

Comparison with strainEST

We compared our methods to strainEST in full pipeline simulation with 2 set of different experiments: (1) benchmark simulation where only existing strains are simulated (2) 4 different evolutionary mechanisms, where novel strains are involved. Overall, our method outperforms strainEST significantly in most of the situations and the performances are summarized in the tables 2.7, 2.8 and A.1. For the SDP stage, our method’s EMD for $k = 2$ however performed worse than strainEst and this might due to the case where strainEst predicts the dominant strain out of the 2 much better than our method.
<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>TVD</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM1</td>
<td>0.96 ± 0.07</td>
<td>0.91 ± 0.09</td>
<td>0.07 ± 0.058</td>
</tr>
<tr>
<td>EM2</td>
<td>0.93 ± 0.07</td>
<td>0.91 ± 0.07</td>
<td>0.26 ± 0.16</td>
</tr>
<tr>
<td>EM2e</td>
<td>0.93 ± 0.08</td>
<td>0.91 ± 0.08</td>
<td>0.34 ± 0.25</td>
</tr>
<tr>
<td>EM2n</td>
<td>0.92 ± 0.09</td>
<td>0.9 ± 0.09</td>
<td>0.34 ± 0.25</td>
</tr>
<tr>
<td>EM3</td>
<td>0.94 ± 0.07</td>
<td>0.92 ± 0.08</td>
<td>0.29 ± 0.15</td>
</tr>
</tbody>
</table>

Table 2.5: Average and standard deviation of different statistics for each evolutionary mechanisms for the ADP stage of full pipeline simulation. EM represents EvoMod.

<table>
<thead>
<tr>
<th></th>
<th>Soft-Precision</th>
<th>Soft-Recall</th>
<th>EMD</th>
<th>Precision</th>
<th>Recall</th>
<th>TVD</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM1</td>
<td>0.96 ± 0.14</td>
<td>0.99 ± 0.079</td>
<td>4.1 ± 7.0</td>
<td>0.44 ± 0.34</td>
<td>0.58 ± 0.40</td>
<td>0.62 ± 0.37</td>
</tr>
<tr>
<td>EM2</td>
<td>0.79 ± 0.21</td>
<td>0.91 ± 0.16</td>
<td>68.8 ± 74.6</td>
<td>0.32 ± 0.19</td>
<td>0.44 ± 0.27</td>
<td>0.78 ± 0.2</td>
</tr>
<tr>
<td>EM2e</td>
<td>0.72 ± 0.24</td>
<td>0.88 ± 0.22</td>
<td>98.9 ± 89.4</td>
<td>0.36 ± 0.26</td>
<td>0.5 ± 0.30</td>
<td>0.72 ± 0.26</td>
</tr>
<tr>
<td>EM2n</td>
<td>0.76 ± 0.23</td>
<td>0.9 ± 0.19</td>
<td>98.6 ± 90</td>
<td>0.36 ± 0.25</td>
<td>0.52 ± 0.30</td>
<td>0.71 ± 0.24</td>
</tr>
<tr>
<td>EM3</td>
<td>0.68 ± 0.20</td>
<td>0.79 ± 0.2</td>
<td>83.7 ± 64</td>
<td>0.29 ± 0.2</td>
<td>0.35 ± 0.22</td>
<td>0.83 ± 0.16</td>
</tr>
</tbody>
</table>

Table 2.6: Average and standard deviation of different statistics for each evolutionary mechanisms for the SDP stage of full pipeline simulation. EM represents EvoMod.

<table>
<thead>
<tr>
<th></th>
<th>Our Algorithm</th>
<th>StrainEST</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>k=1</td>
<td>0.96 ± 0.057</td>
<td>0.88 ± 0.08</td>
</tr>
<tr>
<td>k=2</td>
<td>0.94 ± 0.061</td>
<td>0.9 ± 0.06</td>
</tr>
</tbody>
</table>

Table 2.7: Full pipeline benchmark simulation: Comparison between our algorithm and strainEST on the ADP problem. k is the number of existing strains to simulate.

<table>
<thead>
<tr>
<th></th>
<th>Soft-Precision</th>
<th>Soft-Recall</th>
<th>EMD</th>
</tr>
</thead>
<tbody>
<tr>
<td>k=1, our method</td>
<td>0.99 ± 0.08</td>
<td>1 ± 0</td>
<td>0.623 ± 0.625</td>
</tr>
<tr>
<td>k=1, strainEST</td>
<td>0.35 ± 0.43</td>
<td>0.48 ± 0.5</td>
<td>8.07 ± 7.39</td>
</tr>
<tr>
<td>k=2, our method</td>
<td>0.90 ± 0.18</td>
<td>0.99 ± 0.08</td>
<td>79.2 ± 90.3</td>
</tr>
<tr>
<td>k=2, strainEST</td>
<td>0.37 ± 0.32</td>
<td>0.49 ± 0.34</td>
<td>12.5 ± 8.7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>TVD</th>
</tr>
</thead>
<tbody>
<tr>
<td>k=1, our method</td>
<td>0.48 ± 0.39</td>
<td>0.68 ± 0.47</td>
<td>0.406 ± 0.42</td>
</tr>
<tr>
<td>k=1, strainEST</td>
<td>0.24 ± 0.4</td>
<td>0.3 ± 0.46</td>
<td>0.4 ± 0.33</td>
</tr>
<tr>
<td>k=2, our method</td>
<td>0.4 ± 0.25</td>
<td>0.6 ± 0.31</td>
<td>0.67 ± 0.24</td>
</tr>
<tr>
<td>k=2, strainEST</td>
<td>0.2 ± 0.24</td>
<td>0.28 ± 0.32</td>
<td>0.714 ± 0.352</td>
</tr>
</tbody>
</table>

Table 2.8: Full pipeline benchmark simulation: Comparison between our algorithm and strainEST on the SDP problem. k is the number of existing strains to simulate.
2.3.2 Application to Real Data

The sequencing data we analyzed are from 24 tick samples infected with *B. burgdorferi*, collected using the standard tick dragging method [23] in 2007 from 8 different sites in Vermont, New York, Massachusetts and Connecticut. For each tick sample, the *B. burgdorferi* genome was captured as described in [13]. The sequencing data is composed of $2 \times 76$bp paired-end reads and the number of read pairs ranges from $2.7 \cdot 10^4$ to $2.7 \cdot 10^6$ over all tick samples (coverages ranging from 5X to 500X).

Based on the output of the pipeline, 60 novel and 10 existing strains were inferred to be potential candidates to explain the strain diversity in this large sample of ticks. The total error component of the MILP for solving the SDP amounts to 1.258, or an average of 0.05 per sample. The total proportion of new strains is 14.67 in these 24 samples. On average, each new strain constitutes roughly 40% of the diversity of its sample. For each sample having novel strains, 76% of its genotype is composed of novel strains. Fig. 2.3 further illustrates the diversity, showing a wide range of strain composition in each of the 24 samples, with an average of 3 strains and a maximum of 9 strains infecting each sample, consistent with previous reports [52]. This suggest that the diversity of the *B. burgdorferi* strain types might be much larger than what was known so far.

Figure 2.3: Distribution of the number of existing and novel strains per tick sample.

To further refine our analysis, Fig. 2.4 illustrates the distribution of strains types in the 24 tick samples and the respective contribution to the total diversity of each strain type. Although we observe that 2 of the 10 detected existing strains are present in more than one sample, only 5 out of the 60 novel strains appear in more than one sample.

It is striking to observe that most strain types appear in exactly one tick sample each. Also we can observe that for 11 samples, we do not detect any existing strains. This suggests the possibility that some of these strain types have been improperly called, and that the correct call should have been another strain type, extremely close to this one in terms of sequence similarity; a reasonable cause for such errors could be a mistake while solving the
ADP, in which case a wrongly called allele could be extremely close (in terms of sequence similarity) to the correct allele. Although there are only 7 strains in total being shared by multiple samples, we observed that 2 strains (both novel strains) are shared by samples from the same geographical location. Strain type 56 and 34 (both novel) are shared by SRR2034335, SRR2034360 where both samples were sampled at Franklin D. Roosevelt State Park. This observation suggests that the algorithm is identifying some strain types that are indeed likely to be shared due to their geographical proximity.

Due to the possibility of wrong allele calls leading to introducing novel strains, we also computed a minimum spanning tree (MST) (Figure 2.5, Figure 2.6) of the 70 strains found in these 24 samples, with edge connecting 2 strains having weight defined by the edit distance between the sequences of the alleles over the 8 genes of the MLST scheme. We can observe clusters of strains which are very close to each other, for example, the circled tree branch in the upper left, consists of nodes n25, n26, e6, e3, n13, n20, n14, n27, n28, n29. These nodes can be grouped as a cluster with strains of edit distance at most 5 bases. It is worth taking note that this distance is taken across 8 genes where each is approximately 580 bases long. Furthermore, mapping back to samples which contain these strains, n13 and n14 are in sample SRR2034335; n25, n26, n27, n28, n29 are in sample SRR2034342. Similar observations were seen on nodes which are in the star structure circled in Figure 2.6 where any 2 nodes in the structure differ by at most 11 bases. These observations indicate that either strains in these samples mutate into extremely similar strains, or it indicates that wrong allele call leads to relatively higher number of novel strains inferred by our method. In other words, if the ground truth is the case where a strain evolves into an extremely similar strain, then our algorithm is able to detect this evolutionary event. Otherwise, this observation could mean that the number of strains introduced is inflated by a wrong allele call in the ADP, where for instance, if the sample is infected by a single strain and our ADP
algorithm infers 3 alleles in a particular locus, then at least 3 strains would be needed to explain the sample in the SDP.

Figure 2.5: Nodes with prefix 'n' represents novel strains, 'c' represents existing. The edges connecting two strains are labeled with the edit distance between them. The x-axis and y-axis are normalized distance used for visualizing the tree.

2.4 Discussion

In this chapter, we presented an optimization-based pipeline for estimating the within-host strain diversity of a pathogen from WGS data analyzed in the MLST framework. This work belongs to a growing body of work on understanding genomic diversity within complex biological systems. It is a specific instance of estimating the diversity of a bacterial pathogen from metagenomics data, focusing on within-host diversity and taking advantage of the availability of a large database of known MLST strain types. However, to the best of our knowledge, this is the first work that aims to address strain diversity at this resolution, taking advantage of the large body of known strains available through MLST experiments.

Our approach is composed of two main steps, each of a different nature; the first step detects the alleles present in a sample from the sequence data, while the second step esti-
mates the strain diversity based on the output of the first one. In both steps we follow a parsimonious approach that aims at explaining the input using as few alleles or novel strains as possible. The main contribution of our work is the formulation and the solution of the Strain Diversity Problem for a group of samples. The main challenge of this problem is the need to consider a potentially large set of samples at once. While this leads to a relatively complex MILP, with a large number of variables (whose number is guided by the number of potentially present novel strain types), we believe that the ability to consider a large set of samples at once is an important part of the model, for example for analyzing sequencing data from pathogen hosts originating from a single geographical area. Our work shows that this problem, despite its complexity, can actually be solved using reasonable computational resources and good accuracy.

Our experiments on real data suggest avenues for future research; in particular, the multiplicity of optimal solutions, a common problem in applying optimization algorithms in bioinformatics; this is obviously problematic as calling a wrong allele in a single sample during the first step might force the MILP computing the strain types to introduce a
new strain type. We can observe in our results on real data groups of very closely related strain types, sometimes differing by a single SNP, which likely results from this issue. At the moment, our approach to this problem is to post-process the result of our pipeline to identify clusters of closely related strains, but other more principled approaches could be explored. In particular, it might be interesting to address the ADP using all samples at once, in order to harness the signal from other samples in handling multiple optimal solutions in a given sample. This approach would follow a general parsimony principle that is already at the core of the SDP. Finally, it might be worth investigating a single optimization problems that would consider all variable positions observed in the MLST genes and infer jointly alleles, strain types and proportions. This is however likely to lead to a combinatorial explosion of the size of a MILP encoding this unified approach. Moreover, we may also consider a different penalty for novel strains. For instance, if we have extra information about the relations between the occurrences of alleles at different genes, we are able to adjust the penalty term $w_k$ which takes this into account, possibly based on a likelihood approach i.e. among all combinations of alleles that the MILP considers to explain the sample, some combinations are more likely to occur.

Notwithstanding the aforementioned issues, our experiments with real data suggest a strikingly high diversity in our dataset of 24 tick samples, with the need to introduce dozens of novel strains in order to explain the sequencing data generated from these ticks. This is not altogether surprising since the library of known strains might be limited, and within-host (or, more precisely, within-vector) evolution might result in the presence of a number of strains that only differ by a small number of SNPs in one or two loci of the MLST scheme. Furthermore, our results on the average number of strains present within a tick are consistent with previous work on within-host diversity [52], which suggest that hosts can be infected by an average of 2 to 3 and up to 6 strains. It is remarkable that our MILP formulation produces a similar result without being given any prior information on the expected diversity, or constrained by any principle other than parsimony.

Our work is, to our knowledge, the first comprehensive approach to the problem of reference-based detection of pathogen diversity in a collection of related samples that considers novel strain types. Our two-step pipeline, based on the principle of parsimony implemented through (mixed) integer linear programming, appears to perform extremely well on simulated data and produces reasonable results on a real dataset. We expect that both our approach and our publicly available pipeline will participate to the development of accurate and efficient tools for quantifying the within-host diversity of bacterial pathogens.
Chapter 3

Antimicrobial Resistance Problem

MTB is vulnerable in acquiring drug resistance, especially MDR which poses challenges in effective treatments for the disease. By leveraging the abundance of NGS data, we aimed to construct robust machine learning models predicting across different datasets and draw interesting inferences on resistance related gene and mutation markers by studying trained models. In this chapter, we investigated SNPs within 23 candidate genes of two highly varying diversity datasets, ReSeqTB which comprises heterogeneous samples from different countries and BCCDC which is a homogeneous dataset. We applied some machine learning models which were presented in [61] and introduced ElasticNet in solving our problem. Apart from machine learning algorithms, we also applied a $\chi^2$ feature selection procedure in our machine learning pipeline to remove highly noisy features and prevent overfitting, which is a very common challenge in machine learning. For each model, we performed a grid search and a 10-fold cross validation to choose the optimal parameters. We conducted further analysis on resistance determinant genes using 2 methods, namely Random Permutation of Submatrix and Gene-wise Partial Least Square Regression with LASSO Classifier. We also conducted inference analysis on highly significant SNPs which are responsible for resistance determination by analyzing machine learning models such as Logistic Regression and Random Forest.

3.1 Related Works

In this section, we will review some methods which were previously used to predict drug resistance in MTB from SNPs profiles. The work by Y. Yang et al. [61] to our knowledge is the first to apply machine learning approaches in solving this problem and the work by [15] is the first to apply a deep learning approach in predicting resistance of MTB. Both works have demonstrated promising results over curated libraries in terms of specificities and sensitivities. However, both works focused solely on the prediction aspect of a machine learning problem and have not fully explored tremendous insights and inferences which can be obtained through studying the machine learning models. Inference analysis is important.
especially in biological problems, where in our case biologists or clinicians are interested in studying the factors \textit{i.e.} genes or SNPs which are highly significant in determining resistance. Moreover, previous works have not demonstrated the robustness of their models in predicting major accessible datasets of different diversity.

### 3.1.1 Machine Learning for Classifying TB Drug-Resistance from Sequencing Data

The focus of this work was resistance prediction on 8 drugs \textit{i.e.} isoniazid, rifampicin, ethambutol, pyrazinamide, ciprofloxacin, moxifloxacin, ofloxacin, streptomycin and multidrug-resistance (MDR), across 1839 UK MTB bacterial isolates using different machine learning models. SNPs were called for each sample and 23 candidate genes which were previously reported to be resistance-related were targeted for this study, such genes are \textit{ahpC, eis, embA, embB, embC, embR, fabG1, gidB, gyrA, gyrB, inhA, iniA, iniC, katG, manB, ndh, pncA, rmlD, rpoB, rpsA, rrs, tlyA}. Furthermore, different feature sets were studied to investigate association of different SNPs in resistance prediction. Feature set F1 contains all SNPs, F2 contains 108 SNPs which were reported in [57] and F3 is a subset of F1 for a particular drug where only genes related to resistance of the drug are considered. F1 describes the association of SNPs for resistance prediction in a larger scale, F2 describes the association of SNPs within highly determinant SNPs and F3 describes the direct determinants.

For exploratory data analysis, they performed Principal Component Analysis (PCA) and Sparse Logistic Principal Component Analysis (SL-PCA) on the feature matrix of isoniazid. The first 2 components were used to visualize lineages and resistance in order to study any underlying genetic variation structures of their isolates. They demonstrated that SL-PCA clustered the isolates better in terms of resistance and lineages.

Before the dataset is split into training and test sets, it was balanced by ensuring the number of susceptible and resistant samples were equal. It is worth to take note that their dataset is highly imbalanced \textit{i.e.} number of susceptible samples are much higher than the number of resistant samples, about 7 - 40 times higher depending on the drug under study. Hence, many samples were essentially omitted from further analysis and prediction. The dataset was split into a training set (80%) and a test set (20%), in which a 5-Fold cross validation (CV) was performed on the training set to choose the optimal parameters for the machine learning models.

Machine learning models including logistic regression with $L_1$ and $L_2$ penalties (LR\(_L1\), LR\(_L2\)), support vector machine (SVM) with linear kernel and $L_2$ penalty (SVM\(_L2\)), SVM with radial basis function kernel (SVM\(_RBF\), random forest (RF), Naive Bayes Classifier (NB) and Bernoulli mixture models (BMM) were applied to solve drug resistance problem in this work. In general, they found that machine learning models performed better than any direct association methods on all drugs and all feature sets, with an average of at least 32
95% AUC \(^1\) for first line drugs and at least 80% for second line drugs. SVM_RBF had consistently scored high AUC for all drugs and feature sets.

### 3.1.2 Deep Learning Predicts TB Drug Resistance from Sequencing Data

The focus of this work is to introduce a deep learning model in predicting resistance of 11 drugs simultaneously for 3601 MTB isolates. Moreover, they considered dataset of different diversity for training and validation (1379 isolates were sequenced and analyzed by their group and 2222 isolates were from ReSeqTB). This work also focuses on SNPs within a set of 28 candidate genes and promoter regions. They considered two different ways of reflecting SNPs as features for their machine learning models: 1) the usual binary labels where 0 represents a SNP is present for a sample and 1 otherwise, 2) derived features where rarer mutations are grouped by gene locus (coding, intergenic and promoter regions). Each coding region is split into 3 different types of single nucleotide variants \(i.e.\) substitution, frameshift insertion/deletion and non-frameshift insertion/deletion. Each non-coding region is grouped into either insertion/deletion or substitution. For the resistance label, if a sample has phenotypic data for a particular drug, then it’s labeled 0 or 1 as susceptible or resistant; if the sample has missing phenotypic data, it is assigned a status of 0.5.

Their main contribution is a multidrug wide and deep neural network model (MD-WDNN), which merges a logistic regression model and a deep neural network (NN) in predicting resistance to all drugs simultaneously. Figure 3.1 refers to the structure of their model. The logistic regression portion is the part which connects the concatenation layer and the outputs via a sigmoid activation function and the NN portion is the non-linear transformation using ReLu function on the input data. Hence, we can treat the concatenation layer as a combination of input data and a non-linear abstraction of the input data. Different techniques such as dropout [50], batch normalization [28] and \(L_2\) regularization were applied to the MLP. The network was trained using Adam optimizer [34] and the loss function is a variant of binary cross entropy where the model is not penalized due to missing resistance labels and the function upweights the sparser class.

Based on their results, MD-WNN scores higher AUC than any single drug prediction models including single drug WDNN, logistic regression, random forest. On the independent validation set, they achieved 94% and 89% for first line and second line drugs respectively.

### 3.2 Dataset

In this work, we focused on two geographically distinct datasets: Relational Sequencing TB Data Platform (ReSeqTB) [51] and isolates from British Columbia Centre for Disease Con-

---

\(^1\)AUC is the area under the receiver operating characteristics curve, which is a plot of true positive rate vs. false positive rate.
ReSeqTB is a consortium dedicated to the development of next-generation TB diagnostics, which comprises global isolates from 32 different countries. The BCCDC dataset only consists of isolates sampled and sequenced in British Columbia, Canada. The ReSeqTB dataset has phenotypes for numerous drugs but we focused on first and second line drugs including isoniazid (INH), rifampicin (RIF), ethambutol (EMB), pyrazinamide (PZA), streptomycin (SM), ciprofloxacin (CIP), moxifloxacin (MXF), ofloxacin (OFX), kanamycin (KM), amikacin (AMK), capreomycin (CAP). For the BCCDC dataset, it only contains phenotypes for 5 drugs, INH, RIF, EMB, PZA and SM.

In this work, we predicted single drug resistance and focused on SNPs within 23 candidate genes which were previously reported to confer resistance [57]. There are a total of 3867 SNPs within the 23 genes for ReSeqTB dataset and 1938 SNPs for BCCDC dataset. Figure 3.2 and 3.3 refer to the number of SNPs used for resistance prediction for different drugs. There are a total of 4598 samples having phenotypes for at least 1 of the 11 drugs for the ReSeqTB data; 1136 samples having phenotypes for at least 1 of the 5 drugs for the BCCDC data. In general, both dataset are imbalanced i.e. the number of susceptible samples is higher than the number of resistant samples. Figure 3.4 and 3.5 show the composition of susceptible and resistant samples for each drug. Also, figure 3.6 shows that among the 11 drugs we considered for ReSeqTB, most samples which are resistant to at least 1 drug are resistant to 4 drugs; among the 5 drugs we considered for BCCDC, most samples which are resistant to at least 1 drug are resistant to 2 drugs. We predicted resistance for MDR and there are a total of such 867 resistant samples for ReSeqTB dataset; For BCCDC dataset, there are only 20 such cases.
3.3 Methods

3.3.1 Calling Single Nucleotide Polymorphisms (SNPs)

For ReSeqTB dataset, we were given SNPs called for each sample in csv files. These calls were performed by the ReSeqTB consortium using the Unified Variant Pipeline (UVP)[21].
An in-house pipeline was used to call SNPs for the BCCDC dataset. The *Mycobacterium tuberculosis H37Rv* reference genome was used in both pipelines. For each drug, we translated the SNPs called and phenotypic data into a feature matrix $X$ and a label vector $y$ respectively, which are required for a machine learning problem. $X$ is a binary matrix $X = (x_{ij})$ where $x_{ij}=0$ indicates sample $i$ does not have SNP $j$, $x_{ij}=1$ indicates it has the SNP. $y$ is basically a 0/1-vector where 0 indicates susceptible and 1 indicates resistant. For instance, $y[i] = 1$ means that sample $i$ is resistant.
3.3.2 Cross Validation Protocol

We performed a 10-fold cross validation (CV) and the details are shown in Figure 3.7. In order to choose the best set of parameters for each model, we split the dataset into 80% training, 10% validation and 10% test in each fold. Only the training and validation sets were used in the process of parameter tuning. We trained the model on the training set and searched for an optimal probability threshold which yielded the highest accuracy on the training set. Next, we predicted on the validation set, where we then picked the best parameters for the model based on the average of AUC (Area under the Receiver Operating Characteristics (ROC) curve$^2$) across 10-fold on the validation set. Once the set of best parameters were found, we trained on train and validation sets, then predicted on the test set and reported the AUC of the test set for each fold.

Figure 3.7: Description of each fold for a 10-fold cross validation process. (Left) This is the work flow for choosing the best parameters for each model. Models were trained on the train set and an optimal probability threshold was chosen for classification. The validation set was used for choosing the best parameters and test set was left untouched. (Right) Once the best parameters were found, we trained on both train and validation untouched sets, predicted on test sets and report the performance on the test set for each fold.

$^2$The ROC curve is a plot of true positive rate vs false positive rate.
3.3.3 Cross Dataset Evaluation

The reason for training and testing on datasets of different diversity is to test the robustness of models and to test if the models performed decently well on any dataset. We would expect that models which are trained on a diversified dataset should perform equally well or better on a homogeneous dataset. For cross testing, we trained the models on ReSeqTB dataset, with optimal parameters chosen from CV from the same dataset (as depicted in Figure 3.7), then predicted on the BCCDC dataset and vice versa. Due to possible bias in ReSeqTB data when testing on BCCDC data, we excluded 160 Canadian samples from ReSeqTB.

3.3.4 Machine Learning Classifiers

We considered some of the models presented in [61], Logistic Regression with $L_1$ and $L_2$ penalty (LR_L1 and LR_L2), support vector machine with linear kernel and $L_2$ penalty (SVM_L2), support vector machine with radial basis kernel (SVM_RBF) and random forest (RF). In our work, we investigated ElasticNet too. Also, we performed feature selection using a $\chi^2$ univariate test to choose the best $k$ features for prediction, where $k$ was chosen using CV. Figure 3.8 illustrates the work flow of our machine learning pipeline. We coded our scripts in Python 2.7 and the machine learning models used are implemented in scikit-learn package.

![Figure 3.8: Description of machine learning pipeline: From the reads, we called SNPs and these SNPs are treated as features for any machine learning model. We also performed a kbest feature selection before training any model. At the very end, we predict the response for each strain.](image)

3.3.5 Resistance Determinant Gene

We investigated resistance determinant genes for each drug in a machine learning context by the following two methods:

- **Random Permutation of Submatrix**: For each gene $g$ in the list of 23 candidate genes, we permuted the rows of a submatrix of the feature matrix $X$, where the columns of the submatrix are SNPs within $g$. Denoting this permuted matrix as $X'$, we then input $X'$ to the machine learning models for prediction. Denote $AUC_g(X)$ and $AUC_g(X')$ as the test AUC using $X$ as feature matrix and test AUC using $X'$ as feature matrix respectively, we computed $\Delta_g = AUC_g(X) - AUC_g(X')$. If $\Delta_g$ is large with a positive
value, we conclude that gene $g$ is highly associated to resistance determination for the drug;

- **Gene-wise Partial Least Square (PLS) Regression with LASSO Classifier:** PLS is a regression model which decomposes $X$ into a set of components explaining the covariance between $X$ and $y$ as much as possible [26]. We performed PLS on $X$ gene-by-gene and keeping only 1 component for each gene. Hence, the PLS-transformed matrix $PLS(X)$ is an $n \times 23$ matrix where $n$ is the number of samples. We then trained and predicted resistance using LASSO ($LR_{L1}$) model with $PLS(X)$ as the input matrix. LASSO selects a subset of features for prediction [55] and by investigating the coefficients of the selected features in the model, we are able to learn which gene is important for conferring resistance of a particular drug.

3.4 Results and Analysis

3.4.1 Exploratory Data Analysis

We performed Principal Component Analysis (PCA) on the feature matrices of isoniazid and pyrazinamide for both dataset to investigate the relations between lineages \(^3\) and resistance. Figure 3.9, 3.10, 3.11 and 3.12 show the plots of the first 2 components of PCA with samples grouped by lineages. The exact lineage represented by each index is presented in the appendix. For ReSeqTB, samples can be clearly grouped into clusters of lineages but within each cluster, the resistance and susceptible samples are non-separable, which suggests lineage is not a good predictor for resistance. For BCCDC, similar conclusion applies for pyrazinamide but for isoniazid, samples are not clustered well with lineages.

3.4.2 Within and Cross Dataset Evaluation

For cross testing, we predicted resistance for common drugs between ReSeqTB and BCCDC, which are INH, PZA, EMB, RIF, SM and multidrug-resistant (MDR), which is a resistance to both isoniazid and rifampicin. Figure 3.13 illustrates these results.

For isoniazid, consistent observations across different models are noticed, where (ReSeqTB train, ReSeqTB test) has the highest AUC and (ReSeqTB train, BCCDC test) has the lowest AUC. Surprisingly, training on a diversified dataset ReSeqTB, does not help the models in predicting better on a homogeneous dataset, BCCDC. One reason which explains this is a diversified dataset has more SNPs which are associated to lineages or geographical locations of the samples which they originated from, hence introducing more noise to the models. This could also be justified by comparing (BCCDC train, BCCDC test) and (BCCDC train, ReSeqTB test) where the latter has a higher AUC. For pyrazinamide, (BCCDC

\(^3\)Strains from the same lineage originated from the same geographical location and share some common mutations which are unique to them.
Figure 3.9: ReSeqTB: The axes are the first 2 components of PCA and the drug in study is isoniazid.

The advantage of diversified dataset seems to be especially obvious in MDR, where (ReSeqTB train, BCCDC test) has the highest AUC. The reason in which the advantage is apparent in MDR might be due to inclusion of more supportive features, where SNPs related to resistances of isoniazid and rifampicin both play significant roles in prediction. Similar plots for ethambutol, rifampicin and streptomycin are in the appendix.

We also predicted resistance for other remaining drugs (second-line drugs) for ReSeqTB, which is illustrated in figure 3.14. The models performances on these second line drugs fluctuates more and are not as consistent as first-line drugs where test AUC are at least 0.85. This observation was also reported in [61] and the reason could be highly imbalanced dataset and scarce samples.
3.4.3 Resistance Determinant Genes

Random Permutation of Submatrix

We considered machine learning models of different mechanisms for the investigation of gene importance using random permutation of the rows of feature matrix, as this would provide an unbiased view in judging the importance. We chose the models LR_L1, LR_L2, SVM_L2, SVM_RBF and RF for this study and we concluded a gene is important for resistance prediction if and only if all models gave fairly consistent results.

ReSeqTB: Based on figure 3.15, for isoniazid, genes \(embB\), \(gidB\), \(katG\), \(rpoB\) have positive difference, especially \(katG\) which has 4-5% difference in test AUC and \(rpoB\) which has 1-2% difference. \(katG\) is a crucial resistant determinant gene for isoniazid [10] and the reason \(rpoB\), which is an important gene for rifampicin resistance [54], has a significant effect is due to MDR, where samples are usually resistant to both drugs as both drugs are the strongest first-line drugs for TB treatment. For streptomycin, \(rpoB\), \(rpsL\) and \(rrs\) have positive difference, especially \(rrs\) and \(rpsL\) which have 1-2% difference and both were reported to be gene determinants for the drug [49]. Similar plots for the other drugs are presented in the Appendix.
Similar observations as ReSeqTB can also be concluded here. However, the magnitude of difference for BCCDC is much higher (approximately 5-10 times), where permutation of \textit{katG} for isoniazid has a difference of 15 - 20\% in test AUC and permutation of \textit{rpsL} for streptomycin has a difference of 20 - 30\% in test AUC.

\textbf{Gene-wise PLS with LASSO Classifier}

We reduced the feature matrix gene-by-gene using PLS and ended up with a transformed matrix with 23 columns, each representing a gene, which was then used as an input for a LASSO classifier. The non-zero coefficients of the LASSO and their value can be analyzed to study resistance determinant genes.

\textbf{ReSeqTB:} Based on figure 3.17, \textit{katG} is clearly the most significant gene for isoniazid identified by the LASSO classifier. For streptomycin, \textit{rpsL} is the most significant gene and \textit{katG}, \textit{rrs}, \textit{rpsA} are also identified as significant. \textit{rrs} was reported as one of the resistant determinant gene for streptomycin [49] and \textit{rpsA} has a negative coefficient which might indicate compensatory mutation or having mutations in this gene is more likely to contribute to susceptibility. Similar plots are presented in the appendix.

Figure 3.11: BCCDC: The axes are the first 2 components of PCA and the drug in study is isoniazid
BCCDC: Based on figure 3.18, \textit{gyrA} was identified as the most significant gene for isoniazid instead of \textit{katG} or \textit{rpoB}. This observation might indicate that this method is less suitable for a homogeneous dataset. Similar observations can be seen for streptomycin, where significant genes such as \textit{rpsL} and \textit{rrs} are weighted less in the LASSO model.

\subsection*{3.4.4 Resistance Determinant SNPs}

We investigated two models, LASSO and Random Forest, and analyzed if the important features (SNPs) selected by these models are highly significant in the literature.

\textit{ReSeqTB}: Figure 3.19 refers to the non-zero coefficients of a subset of SNPs of the LASSO model. For isoniazid, mutation 944G>A in gene \textit{katG} has the highest coefficient value and this mutation has been reported to occur frequently in resistant isolates [14, 12, 29, 30]; mutation 1304A>T, 1349C>T in \textit{rpoB} were also reported in the literature [14]. Some of the \textit{rpoB} SNPs were captured by the model most likely due to MDR. For rifampicin, mutations 1333C>T, 1333C>G, 1304A>T were reported to occur frequently in resistant isolates [33, 32]. Figure 3.20 shows the feature importances for some features of the RF model. We observed that some of the highly ranked SNPs identified for both drugs are the same as the ones identified by LASSO such as \textit{katG}_{944G} > C and \textit{rpoB}_{1349C} > T.
Figure 3.13: Common Drugs: Test AUC for within and cross dataset evaluation. "X train; Z test" means models are trained on X dataset and tested on Z dataset. (Top) Isoniazid (Middle) Pyrazinamide (Bottom) MDR

**BCCDC:** Figure 3.21 refers to the non-zero coefficients of a subset of SNPs of the LASSO model. For isoniazid, many highly ranked mutations are in *katG* and *rpoB*. Furthermore, mutations in genes such as *iniA*, *inhA* and *embB* which are associated to isoniazid resistance [24] were identified by the LASSO model. Again, as observed in ReSeqTB dataset, mutation in gene *katG* at position 944 and *rpoB* at position 1349 were also found in BCCDC dataset. The significance of mutation *katG* 944G>C is again observed in the feature importance of random forest model in Figure 3.22. For rifampicin, the aforementioned resistance related mutations for ReSeqTB were also captured by the LASSO model. Similar mutations were also highly ranked in the random forest model such as *rpoB*_1349C > T and *rpoB*_1304A > T.
3.5 Discussion

In summary, we predicted TB drug resistance using different machine learning models on 2 major accessible datasets, ReSeqTB and BCCDC, and performed the study on resistant determinant genes and mutations in a machine learning context.

The first main contribution of our work is demonstrating robustness of models when training and testing on datasets of different diversity. Within dataset testing shows that test AUC is consistently high (at least 85%) for first line drugs and cross dataset testing
Figure 3.16: BCCDC: Difference in test AUC before and after permuting the rows of a submatrix corresponding to a gene. (Left) Isoniazid (Right) Streptomycin

Figure 3.17: ReSeqTB: Non-zero coefficients of the LASSO model with PLS-transformed feature matrix (Left) Isoniazid (Right) Streptomycin

Figure 3.18: BCCDC: Non-zero coefficients of the LASSO model with PLS-transformed feature matrix (Left) Isoniazid (Right) Streptomycin

shows that training on a more diversified dataset does not necessarily contribute to better predictions due to multiple factors such as imbalanced dataset and small sample size. A diversified dataset might only be useful if the set of features considered are mostly supportive
Figure 3.19: ReSeqTB: Non-zero coefficients of the LASSO model (Left) Isoniazid (Right) Rifampicin

Figure 3.20: ReSeqTB: Feature importance of Random Forest (Left) Isoniazid (Right) Rifampicin

Figure 3.21: BCCDC: Non-zero coefficients of the LASSO model (Left) Isoniazid (Right) Rifampicin
SNPs which are related to the resistance, where we observed in the case of MDR. MDR is a resistance towards 2 different drugs, hence the number of available SNPs which are related to MDR is higher, hence making prediction of MDR easier. The main challenge or a possible future work for considering two major datasets is the consistency of SNP calling algorithm and accessible raw reads data. Currently, we only have raw reads for BCCDC but not for ReSeqTB. Although multiple SNP calling pipeline such as UVP and Snippy [48] should yield approximately same results, we should however consider a unified algorithm which is applicable to all datasets to reduce the bias that might be propagated to the machine learning process. Also, we may consider more than 23 candidate genes for prediction. However, doing so will again increase the number of noisy SNPs in our data and a better feature selection process is definitely needed if more genes are studied. We may also filter out lineage-related SNPs, as observed from our lineage analysis where lineage might not be a good predictor for resistance. In any case, our work shows that machine learning models do perform consistently well across these two datasets.

Apart from achieving decent results on predictions, it is equally important to study the trained models and understand resistance-related markers, which was not investigated by other works. In our work, the second contribution we showed is the study of resistant determinant genes by permuting the rows of the submatrix of the corresponding gene and performing gene-wise PLS with LASSO. Both methods give consistent results on the ReSeqTB dataset but not for the BCCDC dataset, where the random permutation method matches well with the literature but not for the gene-wise PLS method in which diversity of dataset might be the culprit. Moreover, we studied resistance determinant mutations by studying the LASSO and random forest models. Our analysis showed that some of the highly ranked SNPs were reported as resistance-associated and this justified the efficacy of our methods, however some of the other SNPs identified were previously not studied, masking the fact whether these SNPs are true signals or just noise. Therefore, the main
and next challenge is to construct reliable feature selection process, either from a machine learning perspective preferably, or expertise-based feature engineering, in eliminating noisy SNPs. By effectively eliminating noise, we are also able to include more supportive SNPs, which will potentially amplify the advantage of training models on a diverse dataset.

As the essence of this work lies in the aforementioned contributions, dealing with a growing number of MTB samples available, a possible challenge is to leverage the abundance of data by incorporating a neural network model, which has recently gained its fame due to its performance in numerous challenges. Moreover, a tougher challenge would be to construct an architecture which predicts well and is insightful for resistance related markers. A feasible architecture which achieves this and could be done in the future is a network which merges multiple different neural networks, where each corresponds to a particular gene. The weights of the edges in each gene-based network can be analyzed to reflect resistance related SNPs and the weights of the edges of the network which merge all gene-based networks can be analyzed to reflect resistance related genes, hence achieving analysis presented in this work in a single model.

Our work presented here can be thought as a preliminary exploratory study and to the best of our knowledge, is the first which investigates resistance markers at different resolution by analyzing machine learning models trained on 2 major accessible datasets. We expect that this explorative study will serve as a reference for studying drug resistance in multiple pathogens and for the inference perspective of machine learning on the drug resistance problem.
Chapter 4

Conclusion

The work presented here solves 2 different problems which are closely related, the strain diversity problem and the antimicrobial resistance problem. In order to develop effective treatments for either Lyme disease or TB, it is crucial to correctly predict resistance of a patient or sample towards a particular drug and to achieve this, the composition of strains within the sample should be clearly identified, which is precisely the two problems presented. Apart from the efficacy of treatments, understanding within-host strain diversity for pathogen such as Borrelia illuminates the transmission or mutation patterns of the bacteria which then enhances our understanding on multiple infection events and evolutionary mechanisms of the pathogen. For the drug resistance problem, we believe our work demonstrated a different perspective to biologists and clinicians for understanding resistance related markers by analyzing machine learning models.

In a broader perspective, methods introduced for both problems hopefully contribute to the development of a powerful, convenient and easily accessible online platform for strain identification and drug resistance diagnosis. This online platform would most likely consists of front end web access, cloud storage and back end analytics. Therefore, user would only need to upload their data to the cloud or query from a database, proceed to run the a particular algorithm and detailed results would be output onto the web application.

In a broader perspective, methods introduced for both problems hopefully contribute to the development of a powerful, convenient and easily accessible online platform for strain identification and drug resistance diagnosis. This online platform would most likely consists of front end web access, cloud storage and back end analytics. Therefore, clinicians would only need to upload the sequenced reads data to the cloud or query from a database, proceed to run the a particular algorithm and detailed results would be output onto the web application. The emergence of advanced NGS technology generates an ample amount of sequence data which requires huge hardware storage, therefore prompting the need to store these data on cloud storage and clusters. Analysis and computations can be done with advanced parallel and distributed computing tools, possibly tools such as Hadoop, Microsoft Azure, Spark, which are widely used in big data analysis in different industries. The platform
should also ensure iterative and incremental development on the optimization pipeline and machine learning models whenever new techniques or data are available for training. The development of such platform requires tremendous effort from multiple parties, which might require an extensive amount of time. However, it will be one of the first which brings medical health diagnosis and facilities to a more accessible level.

4.1 Summary of Contributions

The work presented in this thesis contributes to the understanding of pathogen in their transmission and mutation pattern, as well as the development of drugs resistance classification. These investigations provide new insights to the design of better treatment protocol from the perspective of strain diversity and the development of resistance. Here, two different problems, strain diversity and antimicrobial resistance, were solved.

In the strain diversity problem, we proposed and implemented a two-stage optimization pipeline to deconvolute strain diversity of *Borrelia* samples in a MLST framework, where the first stage is to solve the ADP problem and the results are used as input for the second stage, the SDP problem. In general, both stages are inspired by a parsimonious approach, which is to minimize the number of alleles or strains to describe the data with minimum errors involved. Our pipeline also takes all samples into consideration for the SDP problem. Our results on simulation show that our pipeline worked extremely well on the ADP problem and the SDP problem given perfect input data from ADP, worked well on the full pipeline simulation in the sense that the pipeline was able to detect strains which are closely distant from the true strains. Moreover, our method was shown to be comparable or better than available tools which solve similar problems. For real data, we observed a high number of novel strains were introduced and the numbers could be inflated due to wrong allele calls in the ADP stage. To further analyze this, we computed a minimum spanning tree of all strains and observed that strains identified in each sample were extremely similar, often differing only by few nucleotide bases. It is also worth to take note that the optimization pipeline introduced here can serve as references to solve similar problem for other pathogens or microbes.

In the antimicrobial resistance problem, we explored different machine learning models in predicting first and second line drug resistance of MTB isolates from diversified datasets, ReSeqTB and BCCDC. Results show that training on diversified dataset does not necessarily help prediction, in which case it depends on other factors and number of informative SNPs present. Moreover, we focused more on inferential study where we introduced 2 methods to study resistance determinant genes and investigated LASSO, random forest models to study resistance determinant mutations. It is surprising to observe that multiple SNPs and gene markers which are highly ranked in our analysis were reported as resistance determinants.
in different studies. Therefore, we believe that our preliminary work presented here paves way to discover possible new resistance determinants mutation or gene markers.

4.2 Future Work

Strain Diversity Problem

In order to improve the overall pipeline of the strain diversity problem, more efforts should be performed in perfecting the ADP stage as minor errors in the ADP will be magnified in the SDP. Here are some suggestions worth exploring in the future:

- One advantage of Kallisto over our method is the high number of alleles they inferred, hence achieving higher recall. We may introduce a coefficient $\alpha$ to the allele usage term in the objective function. $i.e. \alpha \cdot \sum_{j=1}^{n} x_j$ in order to achieve similar effect. $\alpha$ can be chosen based on simulated data.

- The abundance estimation method of alleles is rather simple. A more sophisticated method such as a regression model can be investigated to model the abundance distribution of alleles using the distribution of mismatches or scores. For instance, we may utilize simulated data to generate an abundance of score profiles and with the true proportions known to us, we may fit a linear regression model.

- Our allele detection pipeline runs on each gene independently but in reality, the presence of a particular allele at a gene might be associated to the presence of an allele at another gene. Such associations are not captured in our method. Therefore, the most straightforward way is to utilize the comprehensive MLST library of strains and calculate the likelihood of observing each allele at a particular gene given a subset of alleles in other genes. This might also reduce the complexity in the SDP’s optimization problem as it encourages usage of existing strains and possibly decrease the search space for solutions.

- In order to accommodate a larger number of samples while retaining a reasonable amount of speed for the SDP pipeline, the number of variables should be decreased. There are numerous strain candidates in the MILP (the $S_k$) which are not usable for the samples we are considering due to how we construct these candidates $i.e.$ taking the cross product of alleles of all genes. A filtering process, potentially based on a probabilistic ranking on each candidate strain can be performed before running the MILP.

Antimicrobial Resistance Problem

Here are some suggestions worth exploring in the future:
• Complex models such as gradient boosting machines (GBM) and neural networks should be explored. Preferably, a model which shares the benefits of both accurate predictions and high interpretability should be constructed. A neural network example is depicted in the Discussion section.

• Raw reads of ReSeqTB should be requested and a unified SNP calling algorithm should be used on each dataset to ensure no bias is introduced. A 3rd publicly available dataset, PATRIC [59] database can be considered to increase the number of samples, hence allowing the usage of big data-driven models such as neural networks and providing more concrete justifications on the methods we introduced.

• Better feature selection process or regularization technique can be employed as we have seen that high number of noisy features is detrimental to any predictors and our analysis. For example, a univariate selection test (such as $\chi^2$) proceeding with a forward stepwise feature selection can be useful. Dropout meets Multiple Additive Regression Trees (DART) [45] is a tree-based technique (similar to RF) which marries dropout technique in deep learning could be useful as a substitute for RF or GBM due to its smarter regularization and feature selection technique.
Bibliography


[38] David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. Markov chains and mixing times. 2009.


Appendix A

Strain Diversity Problem

The ADP and the Uncapacitated Facility Location Problem

Our formulation of the ADP is very similar to that of the Uncapacitated Facility Location Problem (UFLP), in which a set of customers needs to be served by facilities placed in a subset of a finite set of possible locations, such that the total cost of opening the facilities plus the distances between each customer and the closest facility is minimized. Indeed, if we consider reads to be customers, alleles to be facilities, and distances to be the sum of normalized base quality scores of the corresponding mappings, the ADP becomes a special case of the UFLP. However, our problem is somewhat simpler than the general UFLP because the set of possible distances is limited. We exploit this fact in our ILP formulation which, unlike the formulation for the UFLP, has fewer variables and constraints.

Related to the Uncapacitated Facility Location Problem, already known to be NP-hard, our NP-hardness proof is of independent interest because it actually produces a set of reads and a set of alleles that have the desired distances, whereas not every instance of the UFLP may be obtainable from actual reads and alleles.

WGS data preprocessing.

Before running the pipeline, we map the reads of each sample to a database of alleles for each gene. We use the memory efficient short read mapper Bowtie (v0.12.7) for read mapping. Next, we use SAMTOOLS (v1.3.1) to process the SAM file and extract information such as reads that are mapped as a pair, the alleles that are involved in the mappings, the number of mismatches, the base qualities, and the position of the mismatches.
Comparison of 4 EvoMods

The results for different evolutionary mechanisms presented for our algorithm here are different from those presented in the Results section, due to the database size constraint by strainEST.

<table>
<thead>
<tr>
<th></th>
<th>StrainEST</th>
<th>Our Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>EM1</td>
<td>0.55 ± 0.26</td>
<td>0.51 ± 0.34</td>
</tr>
<tr>
<td>EM2</td>
<td>0.64 ± 0.16</td>
<td>0.57 ± 0.24</td>
</tr>
<tr>
<td>EM2e</td>
<td>0.52 ± 0.16</td>
<td>0.46 ± 0.24</td>
</tr>
<tr>
<td>EM2n</td>
<td>0.61 ± 0.18</td>
<td>0.56 ± 0.23</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Soft-Precision</th>
<th>Soft-Recall</th>
<th>EMD</th>
<th>Precision</th>
<th>Recall</th>
<th>TVD</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM1</td>
<td>0.42 ± 0.46</td>
<td>0.28 ± 0.27</td>
<td>13.6 ± 10.9</td>
<td>0.27 ± 0.41</td>
<td>0.18 ± 0.24</td>
<td>0.83 ± 0.26</td>
</tr>
<tr>
<td>EM2</td>
<td>0.45 ± 0.31</td>
<td>0.33 ± 0.18</td>
<td>13.6 ± 7.84</td>
<td>0.3 ± 0.31</td>
<td>0.21 ± 0.18</td>
<td>0.79 ± 0.26</td>
</tr>
<tr>
<td>EM2e</td>
<td>0.22 ± 0.26</td>
<td>0.18 ± 0.17</td>
<td>17.86 ± 9.04</td>
<td>0.13 ± 0.23</td>
<td>0.1 ± 0.15</td>
<td>0.912 ± 0.178</td>
</tr>
<tr>
<td>EM2n</td>
<td>0.43 ± 0.32</td>
<td>0.38 ± 0.23</td>
<td>12.7 ± 7.68</td>
<td>0.26 ± 0.3</td>
<td>0.22 ± 0.22</td>
<td>0.8 ± 0.251</td>
</tr>
</tbody>
</table>

Table A.1: Average and standard deviation of different statistics for full pipeline simulation of 4 evolutionary mechanisms. (Top) ADP comparison (Middle) strainEST on SDP (Bottom) Our method on SDP
Box Plots of Our Method on ADP simulation
Figure A.1: Box plots of precision and recall of our method for different combinations of coverages and edit distance parameters over 40 simulations for 8 genes of the Borrelia MLST scheme
Figure A.2: Box plots of TVD of our method for different combinations of coverages and edit distance parameters over 40 simulations for 8 genes of the Borrelia MLST scheme.
Box plots of Kallisto on ADP simulation
Figure A.3: Box plots of precision and recall of Kallisto for different combinations of coverages and edit distance parameters over 40 simulations for 8 genes of the Borrelia MLST scheme
Figure A.4: Box plots of TVD of Kallisto for different combinations of coverages and edit distance parameters over 40 simulations for 8 genes of the Borrelia MLST scheme
Box Plots of Our Method and strainEST on Benchmark Simulation

Recall that the benchmark simulation is one of the simulations used to compare to strainEST.

Figure A.5: Our method: Box plots of (1) ADP statistics (2) Soft-Precision and Soft-Recall on SDP (3) EMD on SDP where k=1
Figure A.6: Our method: Box plots of (1) ADP statistics (2) Soft-Precision and Soft-Recall on SDP (3) EMD on SDP where k=2

Figure A.7: strainEST: Box plots of (1) ADP statistics (2) Soft-Precision and Soft-Recall on SDP (3) EMD on SDP where k=1
Figure A.8: strainEST: Box plots of (1) ADP statistics (2) Soft-Precision and Soft-Recall on SDP (3) EMD on SDP where k=2
Appendix B

Antimicrobial Resistance Problem

Machine Learning Models

- Logistic Regression
  Models the logits of success \( i.e. f(x) = \log \frac{p(x)}{1-p(x)} = X\beta \) where \( p(x) \) is the probability of being resistant given \( x \). A penalty term \( h(\beta) \) can be added, \( f(x) = X\beta + h(\beta) \) for regularization. If \( h(\beta) = ||\beta||_1 \), then it is a LASSO regression model (L1 norm); if \( h(\beta) = ||\beta||_2 \), then it is a ridge regression model (L2 norm); if \( h(\beta) = \alpha ||\beta||_1 + (1 - \alpha)||\beta||_2 \), it is ElasticNet. [55, 63]

- Support Vector Machines (SVM)
  The idea is to construct a decision boundary in some subspace which maximizes the margin between resistant and susceptible classes, up to certain degree of error \( \epsilon \). This is also noted as soft-margin SVM. The decision boundary can be constructed without explicitly mapping the data to the subspace and this is achieved by using kernel functions, for example linear kernel and radial basis function kernel. [20]

- Random Forest
  It is an ensemble learning algorithm which uses many decision trees for classification, where for each tree, a random subset of features is chosen for constructing the tree. [8]

- Partial Least Square Regression (PLS)
  A supervised linear model which performs dimensionality reduction. It first identifies a set of new features \( Z_1, Z_2, ..., Z_m \) which are linear combinations of the original features \( x_1, ..., x_n \) where \( m < n \), then fit a linear model using least square with the new set of features. It can be thought as a dimensionality reduction method which aims to explain both the variance between \( X \) and \( y \), and within \( X \). [31]
<table>
<thead>
<tr>
<th>Lineage Number</th>
<th>Lineage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Indo Oceanic</td>
</tr>
<tr>
<td>2</td>
<td>East Asian</td>
</tr>
<tr>
<td>3</td>
<td>East African Indian</td>
</tr>
<tr>
<td>4</td>
<td>Euro American</td>
</tr>
<tr>
<td>5</td>
<td>West Africa 2</td>
</tr>
<tr>
<td>6</td>
<td>Bovine</td>
</tr>
<tr>
<td>7</td>
<td>West Africa 1</td>
</tr>
<tr>
<td>888</td>
<td>Unknown</td>
</tr>
</tbody>
</table>

Table B.1: Details of lineage

Figure B.1: Common Drugs: Test AUC for within and cross dataset evaluation. 'X train; Z test' means models are trained on X dataset and tested on Z dataset. (Top) Ethambutol (Middle) Rifampicin (Bottom) Streptomycin
Figure B.2: ReSeqTB: Difference in test AUC before and after permuting the rows of a submatrix corresponding to a gene.
Figure B.3: BCCDC: Difference in test AUC before and after permuting the rows of a submatrix corresponding to a gene.
Figure B.4: ReSeqTB: Non-zero coefficients of the LASSO model with PLS-transformed feature matrix for different drugs. The y-axis is the magnitude of the coefficient of the Lasso model.

Figure B.5: BCCDC: Non-zero coefficients of the LASSO model with PLS-transformed feature matrix for different drugs. The y-axis is the magnitude of the coefficient of the Lasso model.
For ReSeqTB, all drugs except kanamycin, the PLS-Lasso method ranks the most important gene in conferring resistance as the highest. For example, \textit{rpoB} for rifampicin, \textit{pncA} for capreomycin, \textit{gyrA} for ofloxacin. However, some of the other genes which were highly ranked are not reported in the literature, which might indicate too many noisy SNPs are involved in those genes or they are indeed correlated to resistance which is not extensively discussed in the literature. Further investigations about the functions of these genes are needed to confirm this result. For BCCDC, as mentioned in earlier chapter, due to the homogeneous nature of the dataset, this method might not be useful. However, for rifampicin, it is still able to detect \textit{rpoB} as the important gene.