Prediction of co-occurrence of Antimicrobial Resistant (AMR) genes in Salmonella and Enterococcus using Bayesian Networks and Association Rule Mining

by

Vipul Shinde

(Under the Direction of John Miller)

Abstract

Antimicrobial Resistance (AMR) in bacteria is a global threat with increased prevalence found in isolates from food animals including those in the United States. This is due to the emergence of new mechanisms giving rise to multi-drug-resistant bacterial strains. In this research project, retrospective Antimicrobial Susceptibility Testing (AST) surveillance datasets for Salmonella and Enterococcus bacteria collected by the Food and Drug Administration (FDA) were utilized to determine the co-occurrence of AMR to different antibiotics. For this purpose, a Bayesian Network was implemented and trained and interesting rules were generated using association rule mining. Whole genomic sequence (WGS) data was also used to detect AMR genes and check for co-occurrence.

Index words: Antimicrobial Resistance (AMR), Bayesian Networks, Association Rule Mining, Multidrug Resistance, *Salmonella*, *Enterococcus*

Prediction of co-occurrence of Antimicrobial Resistant (AMR) genes in Salmonella and Enterococcus using Bayesian Networks and Association Rule Mining

by

Vipul Shinde

B.E., Mumbai University, June 2018

A Thesis Submitted to the Graduate Faculty of the
University of Georgia in Partial Fulfillment of the Requirements for the Degree.

Master of Science

Athens, Georgia

2022

Prediction of co-occurrence of Antimicrobial Resistant (AMR) genes in Salmonella and Enterococcus using Bayesian Networks and Association Rule Mining

by

Vipul Shinde

Major Professor: John Miller

Committee: Tianming Liu

Charlene Jackson

Electronic Version Approved:

Ron Walcott

Vice Provost for Graduate Education and Dean of the Graduate School

The University of Georgia

December 2022

DEDICATION

I dedicate this to my loving parents who have always believed in me and supported me unconditionally throughout my life.

ACKNOWLEDGMENTS

I would first like to thank my thesis advisor, Dr. John Miller for his invaluable patience and constant support. This thesis would not have been possible without his input at crucial moments, and his ability to steer me to the right path. I would like to also thank Dr. Tianming Liu and Dr. Charlene Jackson for agreeing to be on my committee. Additionally, I would like to thank Dr. Adam Rivers, Dr. Charlene Jackson, and Dr. Jonathan Frye who motivated me to keep going when I approached them with doubts related to the feasibility of my thesis. Additionally, this endeavor would not have been possible without the generous support from the US Department of Agriculture - ARS, who funded my research. Finally, I must thank all my friends and family for providing me with constant support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Antimicrobial resistance (AMR) has emerged as one of the most serious and growing global public health threats. Antimicrobial products and compounds have been used to kill or slow the spread of microorganisms such as bacteria, viruses, fungi, etc. These products successfully treat various diseases and are used in human and veterinary medicine. But, resistance to these compounds was demonstrated in target pathogens only a few years after their therapeutic use in humans began (Alanis, 2005). According to the 2019 Centers for Disease Control and Prevention (CDC) report, more than 2.8 million antimicrobial-resistant infections occur each year resulting in nearly 35,000 deaths in the United States (US) (Centers for Disease Control, 2019). This is due to the rise and spread of multi-drug resistance (MDR) bacteria, also called "superbugs" (Davies & Davies, 2010). Superbugs are strains of bacteria that have become resistant to most antibiotics and other medications used to treat the infections caused by them.

In this research, the whole-genome sequencing (WGS) dataset for *Salmonella* and *Enterococcus* collected from the National Center for Biotechnology Information (NCBI) website was used initially to train a Bayesian network to check for the co-occurrence of AMR genes and find patterns. Secondly, associations between resistance to different antibiotics in both *Salmonella* and *Enterococcus* were also identified. This paper focuses on finding interesting association rules with high support and confidence for AMR in different strains of bacteria. The main objective of the paper is to build and generate patterns in the AMR analysis using Bayesian Networks and Association Rule Mining.

This thesis is organized into four sections along with an appendix. Chapter 2 contains all the required information on background, techniques, and algorithms to understand the outputs that will be generated in Chapters 3 and 4. It contains information related to AMR in bacteria, Bayesian networks, and association rule mining. This thesis utilizes the past collected genome samples along with the Antimicrobial

Susceptibility Testing (AST) data and uses that knowledge to train the Bayesian networks and find co-occurrence among AMR genes and relationships between resistance to different antibiotics. In Chapter 3, the collection and feature extraction of the WGS dataset is covered. Before training the Bayesian network for Salmonella, hierarchical clustering was performed. Chapter 4 covers the details about the dataset and setup of the experiments and the selection of metrics for finding the associations among antibiotic resistance. The AST dataset for *Salmonella* and *Enterococcus* was collected by the U.S. Food and Drug Administration (FDA) from different locations across the United States during the period 2014-2019. For training the Bayesian Network, the Hill-climbing approach was used along with Bayesian Information Criterion (BIC) as the scoring metric. For checking the quality of the association rules, five interest measures (IMs) such as support, lift, confidence, leverage, and conviction were evaluated. In Chapter 5, conclusions on the findings and potential future work are presented.

CHAPTER 2

LITERATURE REVIEW AND BACKGROUND

§ 2.1 Antimicrobial Resistance (AMR)

Antimicrobial agents have been widely used in livestock and poultry since the 1950s (Mathew AG, 2007). The use of antimicrobial agents in the production of food animals has offered proven benefits such as improved animal health, higher production, and, in some cases, reduction in foodborne pathogens (Mathew AG, 2007). But resistance to these agents has emerged posing a serious global threat of increasing concern for human, animal, and environmental health. There has been an increase in the prevalence of Antimicrobial Resistance (AMR), including multi-drug resistance (MDR) in bacteria isolated from U.S food animals (Frye & Jackson, 2013). The phenotypic expression of antimicrobial resistance to a particular antimicrobial agent may be encoded by a number of different resistance genes.

In the case of MDR, multiple unrelated resistance genes exist within the same bacterium, resulting in simultaneous resistance to multiple antimicrobial agents of different classes (Alanis, 2005). These MDR mechanisms imply that antimicrobial use can select not only for resistance to one drug but also for resistance to other antimicrobials. Therefore, finding the patterns of associations between resistance to multiple antimicrobials is important in order to reduce the subsequent development of AMR. *Salmonella* infections are the second most common cause of bacterial foodborne illness in the United States. It is estimated that approximately 1.4 million *Salmonella* infections occur each year, resulting in 17,000 hospitalizations and 585 deaths in the United States (Mead et al., 1999; Voetsch et al., 2004). *Enterococcus faecalis* and *Enterococcus faecium* have emerged as MDR pathogens in critically ill patients (Sood et al., 2008). Different mechanisms for antibiotic resistance in *Salmonella* and *Enterococcus* have been described in Frye and Jackson (2013).

§ 2.2 Probability

A random variable is a quantity with an associated probability distribution. It can be either discrete (i.e., have a countable range) or continuous (have an uncountable range). The probability of a random variable X taking on a value from set A is defined as the probability measure of X, denoted by $P(X \in A)$. In particular, the probability distribution associated with a discrete random variable is a probability mass function (pmf), whereas one associated with a continuous random variable is a probability density function (pdf). The cumulative distribution function (CDF) is the probability that the random variable is less than or equal to a certain value x and is defined as $F_X(x) = P(X \leq x)$ in both cases. For a discrete random variable, the pmf is given by the difference.

$$p_X(x_k) = F_X(x_k) - F_X(x_{k-1}) \tag{2.1}$$

and the pdf for a continuous random variable is given by the derivative.

$$f_X(x) = \frac{d}{dx} F_X(x) \tag{2.2}$$

There are three types of probabilities:

- Joint Probability

  A joint probability is the probability of two different events occurring at the same time. For the discrete random variables X and Y, the joint pmf is given as:

  $$p_{X,Y}(x, y) = P(X = x, Y = y) \tag{2.3}$$

  where (x, y) is a possible pair of values for the random variables X and Y.

- Marginal Probability

  From a joint pmf, a marginal pmf can be derived. The marginal PMFs for the random variables X and Y are given as:

$$p_X(x) = P(X = x) = \sum_j p_{X,Y}(x, y_j) \tag{2.4}$$

$$p_Y(y) = P(Y = y) = \sum_i p_{X,Y}(x_i, y) \tag{2.5}$$

where i and j correspond to all possible values of random variables X and Y respectively.

- Conditional Probability

  Conditional probability is the probability of an event X occurring, given that another event Y is known to have occurred. It is denoted mathematically as $P(X \mid Y)$. A conditional pmf is calculated as follows:

$$p_{X|Y}(x \mid y) = P(X = x \mid Y = y) = \frac{p_{X,Y}(x, y)}{p_Y(y)} \tag{2.6}$$

§ 2.3 Bayes' Theorem on Conditional Probability

The conditional probability is often found in the following form based on Bayes' rule.

$$P(A \mid B) = \frac{\text{P}(B \mid A)\ \text{P(A)}}{\text{P(B)}} \tag{2.7}$$

Under Bayes' rule, P(A) is known as the prior probability, P($B \mid A$) as the likelihood, P(B) as the evidence, and P($A \mid B$) as the posterior probability. If this conditional probability is presented simply as P(A), that is, P($A \mid B$) = P(A) then A and B are independent events as knowing about event B tells nothing about the probability of event A. Similarly, it is possible for A and B to be conditionally independent given the occurrence of another event C: P($A \cap B \mid C$) = P($A \mid C$) P($B \mid C$). This statement says that, given that C has occurred, and knowing that B has also occurred tells nothing about the probability of A having occurred.

§ 2.4 Bayesian Network

Bayesian networks (BNs) are a part of the probabilistic graphical models family. Their graphic structures can be used to represent causal knowledge about an uncertain domain and can be referenced to reflect the interpretation of particular input data (Pearl, 1985). A Bayesian network is a directed acyclic graph (DAG) where the nodes represent the random variables and the edges between the nodes express probabilistic relationships between these variables. Thus, if the graph G = (V, E), where V is a finite set of vertices or nodes and E is a finite set of edges, contains a set of random variables $\mathbf{X} = \{X_1, X_2, \ldots X_n\}$, then an edge between node X_i and X_j represents a statistical dependence between the two corresponding variables. Also, a directed arrow indicates that the value of variable X_j is dependent on the value of X_i. Here, the node X_i is referred to as a parent node and X_j as a child node. The factorization of the joint probability distribution P(X) can be written as the product of individual density functions depending on the parent variables:

$$P(X) = \prod_{j=1}^{n} P(X_j \mid \Pi_{X_j}) \tag{2.8}$$

where Π_{X_j} is the set of parents for node X_j and n is the total number of nodes in G. With Bayesian structure learning, we want to find the optimal DAG that best captures the statistical dependencies among variables in a given dataset. There are two approaches that can be used to search throughout the DAG space and find the best-fitting one for the given dataset. The first is score-based algorithms, which search for all possible DAGs and a score is assigned to each DAG based on how well it fits the data. The DAG with the best score is then kept as the final structure (D. Heckerman, 1999). This approach seems computationally intractable as the number of possible DAGs is massive and the time-complexity increases exponentially with the number of nodes (Robinson, 1977). There are many scores that have been proposed for finding the optimal DAG that maximizes the posterior probability (Daly et al., 2011). Scores such as BDeu, BDs, and BDla which belong to the Bayesian Dirichlet (BD) family, Bayesian Information Criterion (BIC), Akaike Information Criterion (AIC), and Minimum Description Length (MDL) have been proposed (Daly et al., 2011). The BIC was used as the scoring metric for training the Bayesian network. BIC score

gives an estimate of the model's performance where a model with a lower BIC score is selected. The BIC score is calculated as follows:

$$BIC = -2 * Loglikelihood + k * log(N) \tag{2.9}$$

where N is the sample size of the training dataset and k is the number of parameters. A Likelihood function is a measure of how well a particular model fits the data and is given by:

$$L(\theta) = L(\theta \mid x_1, x_2, \ldots x_n) = f(x_1, x_2, \ldots x_n \mid \theta) \tag{2.10}$$

where f is the probability mass function. The logarithmic transformation to the base e of the likelihood function is called the Log-likelihood. The score-based approaches are basically searching problems consisting of two parts: the search algorithm to find the optimal DAG from the search space of all possible DAGs and the scoring metric to generalize how well the BN fits the given dataset. The different score-based search algorithms are as follows:

1. Exhaustive Search: The Exhaustive search approach scores each possible DAG and returns the DAG with the highest score. But, this search algorithm is only effective for small networks, preventing local optimization algorithms from finding optimal structures consistently. Therefore, finding the ideal DAG is not always possible.

2. HillClimb Search: The HillClimb search approach is a heuristic approach that works well with larger networks. It uses a greedy local search approach that starts with an empty DAG and keeps on adding-removing edges to maximize the score. But, this search algorithm stops when local maxima are achieved.

3. Chow-Liu: The Chow-Liu algorithm is a type of search approach which finds the maximum-likelihood tree-structure graph where each node has exactly one parent. This approach can determine the DAG quickly on large datasets, as the complexity can be limited by restricting it to a tree structure. But, for this algorithm to work a root node needs to be set. The algorithm has

three steps, viz. compute the mutual information (MI) for all pairs of variables, find the maximum weight spanning tree that connects all vertices in a graph and pick any node as root node assigning an outward going arrow.

§ 2.5 Inference via Bayesian Network

Bayesian networks can be used to compute the posterior probability distribution of one or more variables in the network by conditioning the observed values for the other variables in the network. This can be done by using variable elimination which performs marginalization over Joint Probability Distribution (JPD) (Cooper, 1999). A trained Bayesian network encodes a joint probability distribution, so it contains all the information needed to compute the marginal or conditional probabilities of the nodes in the network. In our case, we will be able to ask the network questions as follows:

- P(X_1 = present | X_5 = absent and X_6 = present)
- P(X_3 = absent | X_1 = present)
- P(X_7 = present | X_1 = present and X_1 = present)

where X_1, X_2,....,X_n represent different antibiotics and present, absence denotes the AMR resistance.

§ 2.6 Association Rule Mining

Agrawal first stated the formal explanation of the association rule mining problem in Agrawal et al. (1993). Given a set of items, $I = \{I_1, I_2, \ldots I_n\}$, and a set of transactions T such that $T \subseteq I$, D be the database with different transaction records T's. Then, the association rule is an implication of form $X \rightarrow Y$, where $X, Y \subset I$ are a disjoint set of items called itemsets. Here, X is called the antecedent, and Y is called the consequent, the association rule stating X gives rise to Y. Association rule mining has two important basic measures: support and confidence which can be used to prune the rules that are not interesting as we are only interested in frequently seen patterns.

Table 2.1: Example database with 6 items and different transactions at a supermarket store

Transaction_id	Milk	Eggs	Bread	Butter	Diaper	Coke	Beer
1000	0	1	0	0	1	0	1
1001	1	0	1	1	0	1	0
1002	1	1	1	1	1	1	0
1003	0	1	0	0	0	1	0
1004	1	0	1	0	1	0	1
1005	1	0	1	1	1	1	0

§ 2.7 Interest Measures (IM)

1. **Support**

   Support is defined as the frequency or number of times an itemset occurs in the given transactional database (Agrawal et al., 1993). So, by the above definition, support is the statistical significance of the association rule. Suppose the support of an itemset or item is 0.05%, then it means that the itemset or item occurs in only 0.05 percent of the total transactions in the database. In many cases, such low-support items can appear randomly, so it may not be interesting to watch. We can prune or filter the itemsets with a specified minimum-support threshold to find the "frequent itemsets". Support is calculated by the following formula:

$$Support(X \rightarrow Y) = P(X \cap Y) = \frac{\text{number of transactions containing X \& Y}}{\text{Total number of transactions in D}} \tag{2.11}$$

   Using Table 2.1 as an example, the itemset {Milk, Bread} has a support of 4/6 = 0.66 meaning it occurs in 66% of all transactions.

2. **Confidence**

   While support corresponds to the rule's statistical significance, confidence is a measure of the rule's strength (Agrawal et al., 1993). Confidence is defined as the conditional probability P(X | Y), which

is the probability of X and Y occurring together given that the transaction already contains X. Suppose the confidence of a rule is 90%, then it means that 90 percent of the transactions that contain X also contain Y giving us a great measure to validate the rule. The confidence is 1 or 100% if the antecedent and consequent always occur together. Confidence is given as follows:

$$Confidence(X \rightarrow Y) = P(X \mid Y) = \frac{\text{Support } (X \rightarrow Y)}{\text{Support (X)}} \tag{2.12}$$

Again, looking at Table 2.1, the rule {Milk, Bread} $\rightarrow$ {Butter} is (3/6)/(4,6) = 0.75, meaning that every time a customer buys milk and bread, 75% of the time they also buy butter along with it.

3. **Lift**

Lift is a measure of the rule's importance (Brin et al., 1997). Lift is the ratio of the confidence of the rule and the expected confidence of the rule. It is used to measure how often a rule's antecedents and consequences occur together more than would be expected if they were statistically independent. Lift for a rule is defined as follows:

$$Lift(X \rightarrow Y) = \frac{\text{Confidence } (X \rightarrow Y)}{\text{Support (Y)}} = \frac{\text{Support } (X \rightarrow Y)}{\text{Support (X) Support(Y)}} \tag{2.13}$$

- A lift value > 1 indicates that the antecedent and consequent occur more often together than expected, meaning that the occurrence of one item has a positive effect on the occurrence of other items.
- A lift value < 1 indicates that the antecedent and consequent occur less often together than expected, meaning that the occurrence of one item has a negative effect on the occurrence of other items.
- A lift value = 1 indicates that the antecedent and consequent occur almost as often together as expected, meaning that the occurrence of both items is independent of each other.

For example, the rule {Diaper} $\rightarrow$ {Beer} has a lift of (2/6) / $((4/6) \times (2/6))$ = 1.5 which means that there is a direct implication between purchasing diapers and beer.

4. **Leverage**

   Leverage calculates the difference between the observed frequency of items (in an itemset) occurring together and the frequency that would be expected if the items were independent (Piatetsky-Shapiro, 1991). This definition is almost identical to lift, except that lift calculates the ratio while leverage finds the difference because of which leverage is able to favor itemsets with higher support. A leverage value of 0 means that both X and Y are independent, a value greater than 0 indicates a positive correlation between X and Y while a value less than 0 shows that X and Y are negatively correlated. The leverage of a rule is calculated as follows:

$$Leverage(X \rightarrow Y) = Support(X \rightarrow Y) - Support(X) \times Support(Y) \tag{2.14}$$

5. **Conviction**

   A conviction can be interpreted as the ratio of the probability of X occurring without Y, given they are dependent on the actual frequency of occurrence of X without Y (Brin et al., 1997). Unlike lift, conviction is sensitive to rule direction as it also uses information about the absence of consequent. A conviction value of 1 indicates that X and Y are independent.

$$Conviction(X \rightarrow Y) = \frac{\text{1 - Support (Y)}}{\text{1 - Confidence } (X \rightarrow Y)} \tag{2.15}$$

   For example, the rule {Diaper} $\rightarrow$ {Beer} has a conviction of $\frac{1-(2/6)}{1-(1/2)}$ = 1.33 stating that the two given itemsets are positively related.

§ 2.8 The Apriori Algorithm

Association rule mining can be broken down into two sub-problems: The first problem is to find the itemsets whose number of occurrences exceeds a certain defined threshold. In the database, these itemsets are called frequent or large itemsets. The second problem is generating association rules from these large itemsets using minimum confidence constraints (Agrawal et al., 1993). Once we have determined the

large itemsets, the solution to the second sub-problem is straightforward. Hence, many of the approaches focus on solving the first sub-problem. One such approach that is efficient and used widely for finding large itemsets is called the Apriori Algorithm. It was introduced by R. Agrawal and R. Srikant in 1994 (Agrawal, Srikant, et al., 1994). The first problem can be further decomposed into two sub-problems: firstly generating candidate large itemsets and second generation of frequent itemsets (Zhao & Bhowmick, 2003). The itemsets whose support exceeds the minimum support threshold are called large or frequent itemsets, while those itemsets that are expected to be large or frequent are called candidate itemsets.

There are two processes involved in finding all the large or frequent itemsets in a given database D in the Apriori algorithm. Initially, the candidate itemsets are generated, and then the database is scanned to check the support count for all corresponding itemsets. During the first scan of the database, the support count for each item is calculated and the large 1-itemsets are generated by removing those itemsets whose supports are below the minimum support threshold. Only candidate itemsets containing the same specified number of items are generated and checked in each pass. The candidate k-itemsets are generated after the $(k-1)^{th}$ passes over the database by combining the frequently used k-1-itemsets. According to the Apriori property, all the candidate k-itemsets are pruned by checking if their sub (k-1)-itemsets are present in the frequent itemsets. If not, they should be removed as the Apriori property states that every sub (k-1)-itemsets of the frequent k-itemsets should be frequent.

CHAPTER 3

CO-OCCURRENCE OF AMR GENES IN BACTERIA

§ 3.1 Dataset Overview

The genome assembly dataset downloaded from the NCBI website for *Salmonella enterica*, *Enterococcus faecalis*, and *Enterococcus faecium* (Wheeler et al., 2007) was used in the analysis. This can be done using the command-line tool called datasets which is used to query and download genome sequence data for the required organism from the NCBI database. For example, for downloading the genome data for *E. faecalis*, the following command was used:

datasets download genome taxon "Enterococcus faecalis" –filename filename.zip

In total, around 325,000 genomes for *Salmonella*, 15,124 genomes for *E. faecium*, and 7,631 genomes for *E. faecalis* were downloaded. This dataset mostly contains bacteria genomes collected from humans and some from retail food animals. Since we were interested in finding the co-occurrence of AMR genes in these bacteria, all the AMR genes present in these genome files were identified. The AMRFinderPlus tool was used for finding all the AMR, metal, heat, and virulence resistance genes from the genome assembly.

§ 3.2 AMRFinderPlus

AMRFinderPlus is a bioinformatics tool that is used to identify AMR genes, point mutations, and other classes of genes including stress, acid, biocide, metal, heat, and virulence resistance genes in both protein and nucleotide sequences (Feldgarden et al., 2021). The tool uses the Reference Gene Catalog database along with a tool like Basic Local Alignment Search Tool (BLAST) internally to match the given sequence with the genes from the database. This tool also provides other details such as gene class, subclass, full-sequence name, element type, etc. There is also an option to only retrieve the AMR genes by removing the –plus option.

AMRFinderPlus with the –plus option was used for all the genome sequences for both *Salmonella* and *Enterococcus*. The total number of AMR genes along with other resistant genes is shown in Table 3.1 below.

Table 3.1: Genome Sequence Dataset Along with the AMR Genes for *Salmonella* and *Enterococcus*

Bacterial Species	*Salmonella*	*Enterococcus faecalis*	*Enterococcus faecium*
Number of Genomes	310743	6706	14617
AMR genes	369	129	142

§ 3.3 Hierarchical Clustering

The Bayesian network was first trained before with random 10K genomes from the WGS dataset for *Salmonella* which contains 310K genomes collected from the NCBI website. Due to computational constraints, the entire network with the 310K genomes along with the 369 AMR gene nodes could not be trained as the computational complexity is more than exponential in N nodes for the greedy search-based hill-climbing algorithm (Scutari et al., 2019). Since the first 10k genomes were randomly selected, there was a potential loss of information as genomes of interest could be located anywhere in the dataset. To avoid that, genomes were clustered hierarchically and at least one genome was selected from each of the 15K cluster nodes. In this way, the representation and diversity of all the genomes in the dataset were preserved.

Clustering is an unsupervised machine learning technique that groups similar data points such that the data points in one group are similar to each other than points in the other group, where each group forms a cluster. Hierarchical clustering was first introduced by Stephen Johnson in 1967 (Johnson, 1967). In agglomerative hierarchical clustering, in the beginning, each data point is considered to be an individual cluster. At each iteration, similar clusters are merged with other clusters until one or a number of specified clusters (k) are formed. With the help of hierarchical clustering, the cut-off (for the dendrogram) can be defined later and the specified number of clusters will be formed. The 325K genomes were first hashed using the dashing software (Baker & Langmead, 2019) and then the Mash distance (Ondov et al., 2016) was used to generate the distance linkage matrix (325K vs 325K) which was used as the input for training

the hierarchical clustering done with the use of python package SciPy (Virtanen et al., 2020). Different clustering algorithms like single linkage, average linkage, and weighted linkage were evaluated to compute the clusters using agglomerative clustering.

But, as the number of genomes is reduced, a few genes may be lost as there were genomes that did not contain any AMR genes. The dendrogram tree was cut at a distance t, where around 15K clusters of genomes were present.

§ 3.4 Structure Learning Using Bayesian Network

The Python package *bnlearn* was used to perform structure learning using a Bayesian network. The hill-climbing algorithm was used along with the BIC as a score metric to train the model. Due to the high number of AMR genes in the *Salmonella* dataset, a very complex static plot is formed from which it is hard to interpret associations. The BN for the clustered *Salmonella* genome dataset with 279 nodes (genes) is shown below.

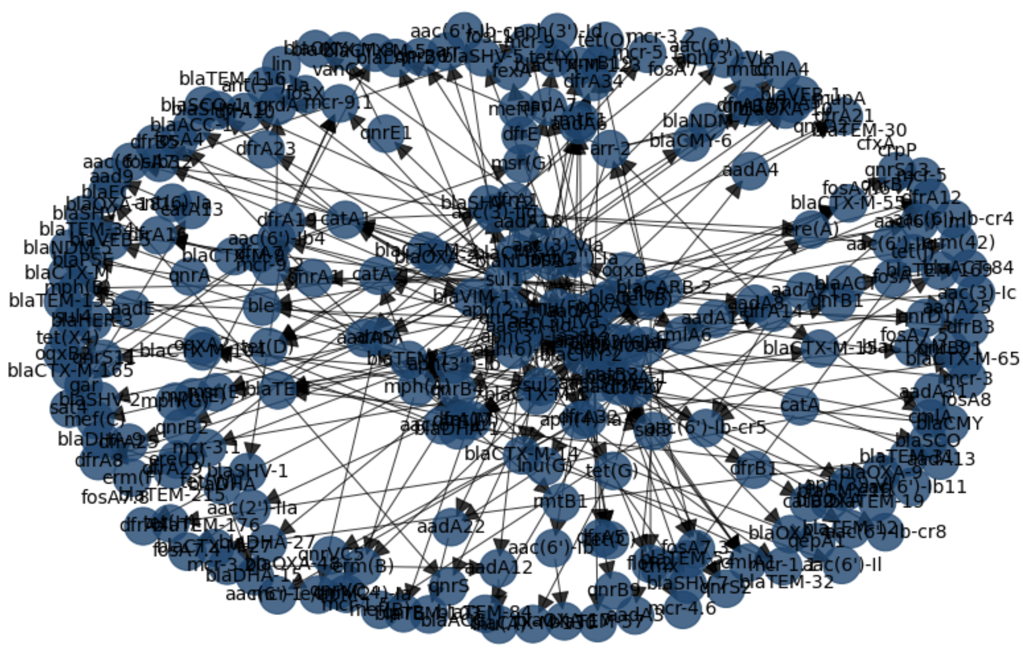


Figure 3.1: DAG for clustered *Salmonella* data learned with structure Bayesian learning

Also, the BN for *E. faecium* with 14617 genomes, 142 AMR genes, and *E. faecalis* with 6706 genomes, 129 AMR genes were trained directly using *bnlearn* Python package without any clustering as the training time was within the computational power due to fewer AMR genes and genome samples. The static DAG for both bacterial species is shown below.

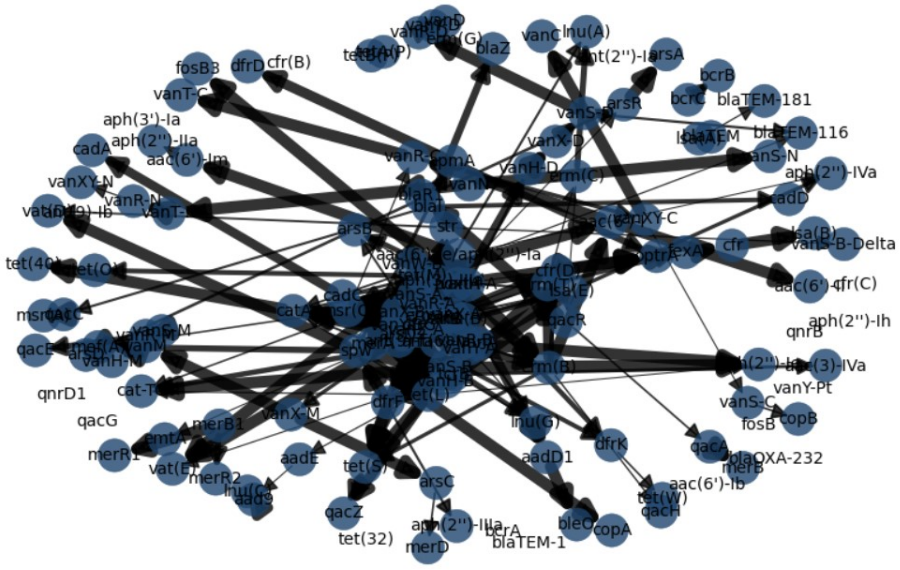


Figure 3.2: DAG for clustered *E. faecium* data learned with structure Bayesian learning

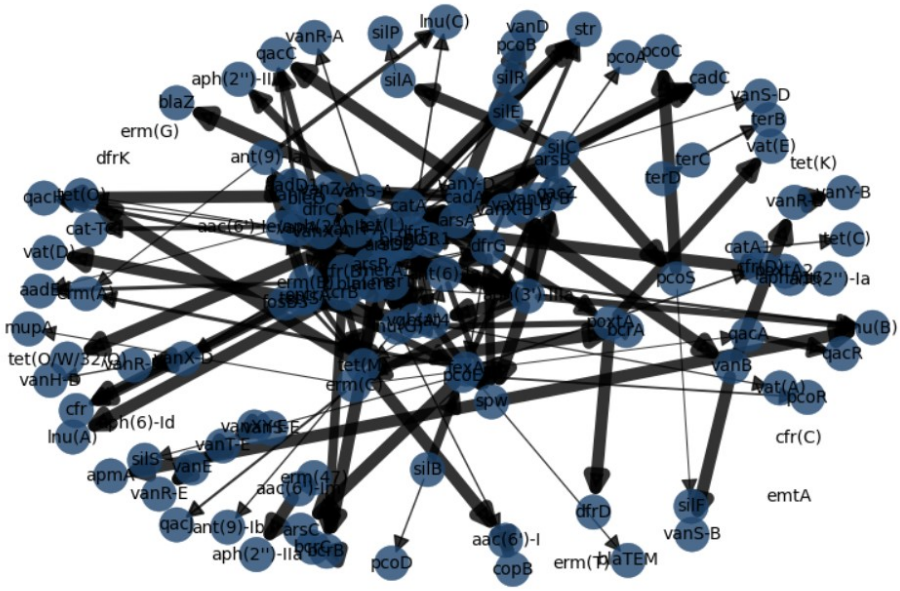


Figure 3.3: DAG for clustered *E. faecalis* data learned with structure Bayesian learning

Since we were not able to interpret the graph due to the high number of nodes, the graph was plotted in interactive mode where the nodes can be moved around; however, that type of plot is not able to be displayed here.

§ 3.5 Inference

With the help of trained BN and its encoded joint probability distribution for each AMR gene node, the probability of the presence/absence of a particular AMR gene can be inferred given that one already knows the presence/absence of other AMR genes. For example, the probability of the presence of the *catA13* (type A-13 chloramphenicol O-acetyltransferase) gene given that *aad9* (ANT-9 family aminoglycoside nucleotidyltransferase) is absent/present can be inferred from the CPD table as shown below.

Table 3.2: Example Conditional Probability Table of *catA13* AMR gene in *Salmonella*

aad9 / catA13	*catA13 (True)*	*catA13 (False)*
aad9 (True)	0.2605	0.7395
aad9 (False)	0.0283	0.9717

In the next section, we will be working with the AST data to check for associations between resistance to different antibiotics.

CHAPTER 4

ASSOCIATIONS AMONG RESISTANCE TO DIFFERENT ANTIBIOTICS

§ 4.1 Dataset Overview

The antimicrobial susceptibility testing data for *Salmonella* and *Enterococcus* isolated from retail food animals were obtained from the FDA database for the period 2014-2019. In total there were 4471 and 5992 sample isolates available for *Salmonella* and *Enterococcus* respectively. Each of the tested isolates was classified as either resistant or susceptible based on the minimum inhibitory concentration (MIC) breakpoints (CLSI, 2020).

§ 4.1.1 *Salmonella* data

Out of the 15 antimicrobials that the *Salmonella* isolates were tested against, the antimicrobials that were used for the analysis were Ampicillin (AMP), Amoxicillin–clavulanic acid (AMC), Azithromycin (AZI), Chloramphenicol (CHL), Ceftriaxone (AXO), Cefoxitin (FOX), Ciprofloxacin (CIP), Colistin (COT), Gentamicin (GEN), Nalidixic acid (NAL), Streptomycin (STR) and Tetracycline (TET). Sulfisoxazole was not used for this study as there was not a single isolate resistant to it in the dataset.

From the total of 4471 isolates that were tested, 2 (0.04%) were resistant to 10 of the 12 antimicrobials, 310 (6.93%) to at least 8, 622 (13.91%) to at least 5, 1078 (24.11%) to at least 3 and 2977 (66.58%) to at least 1 of the antimicrobials. For the association rule mining, the 2977 isolates resistant to at least one antibiotic were used for generating rules and training the Bayesian network. The breakpoints used to determine resistance and the number of resistant isolates is shown below in Table 4.1.

Table 4.1: Resistance breakpoints and prevalence of antimicrobial resistance for the 4471 *Salmonella* isolates obtained from retail meat between 2014-2019.

Antimicrobial	**Abbreviations**	**Resistance breakpoints (ug/ml)**	**Number (%) of resistant isolates**
Ampicillin	AMP	≥ 32	877 (19.61)
Amoxicillin–clavulanic acid	AMC	$\geq 32/16$	494 (11.04)
Azithromycin	AZI	≥ 32	2 (0.04)
Chloramphenicol	CHL	≥ 32	387 (8.65)
Ceftriaxone	AXO	≥ 4	481 (10.75)
Cefoxitin	FOX	≥ 32	207 (4.62)
Ciprofloxacin	CIP	≥ 1	654 (14.62)
Colistin	COT	≥ 4	216 (4.83)
Gentamicin	GEN	≥ 16	589 (13.17)
Nalidixic acid	NAL	≥ 32	641 (14.33)
Streptomycin	STR	≥ 32	2050 (45.85)
Tetracycline	TET	≥ 16	2214 (49.51)

§ 4.1.2 *Enterococcus* data

Out of the 15 antimicrobials that the *Enterococcus* isolates were tested against, the antimicrobials that were used for the analysis were Chloramphenicol (CHL), Ciprofloxacin (CIP), Daptomycin (DAP), Kanamycin (KAN), Lincomycin (LIN), Linezolid (LZD), Penicillin (PEN), Streptomycin (STR), Nitrofurantoin (NIT), Tetracycline (TET), Tigecycline (TGC) and Quinupristin/Dalfopristin (SYN). The *Enterococcus* data was divided into two based on two predominant species: *Enterococcus faecalis* and *Enterococcus faecium*. The breakpoints used to determine resistance and the number of resistant isolates is shown below in Table 4.2.

- *E. faecium*

  From the total of 1129 isolates that were tested, 7 (0.62%) were resistant to 9 of the 12 antimicrobials, 217 (19.22%) to at least 6, 702 (62.17%) to at least 4, and 1127 (99.82%) to at least 1 of the antimicrobials. For association rule mining, the 1127 isolates for generating rules were used.

- *E. faecalis*

  From the total of 4863 isolates that were tested, 4 (0.08%) were resistant to 8 of the 12 antimicrobials, 251 (5.16%) to at least 6, 2051 (42.58%) to at least 4, and 4829 (99.30%) to at least 1 of the antimicrobials. For the association rule mining, the 4829 isolates for generating rules were used.

Table 4.2: Resistance breakpoints and prevalence of antimicrobial resistance for the 1129 *E. faecium* and 4863 *E. faecalis* isolates obtained from retail meat between 2014-2019.

Antimicrobial	**Abbreviations**	**Resistance breakpoints (ug/ml)**	**Number (%) of resistant isolates *E. faecium***	**Number (%) of resistant isolates *E. faecalis***
Chloramphenicol	CHL	≥ 32	15 (1.32)	185 (3.80)
Ciprofloxacin	CIP	≥ 4	364 (32.24)	2367 (48.67)
Daptomycin	DAP	≥ 8	146 (12.93)	5 (0.10)
Kanamycin	KAN	≥ 1024	100 (8.85)	839 (17.25)
Lincomycin	LIN	≥ 8	237 (20.99)	78 (1.60)
Linezolid	LZD	≥ 8	3 (0.26)	4 (0.08)
Penicillin	PEN	≥ 16	164 (14.52)	4 (0.08)
Streptomycin	STR	>1000	147 (13.02)	608 (12.50)
Nitrofurantoin	NIT	≥ 128	1067 (94.50)	49 (1.00)
Tetracycline	TET	≥ 16	505 (44.72)	1641 (33.74)
Tigecycline	TGC	≥ 0.5	4 (0.35)	25 (0.51)
Quinupristin/Dalfopristin	SYN	≥ 4	308 (27.28)	111 (2.28)

§ 4.2 Structure Learning Using Bayesian Network

As discussed in the literature review, Bayesian network learning can be divided into **structure learning** and **parameter learning**. The package *bnlearn* which is available in Python has the ability to easily implement both these methods (Taskesen, 2020).

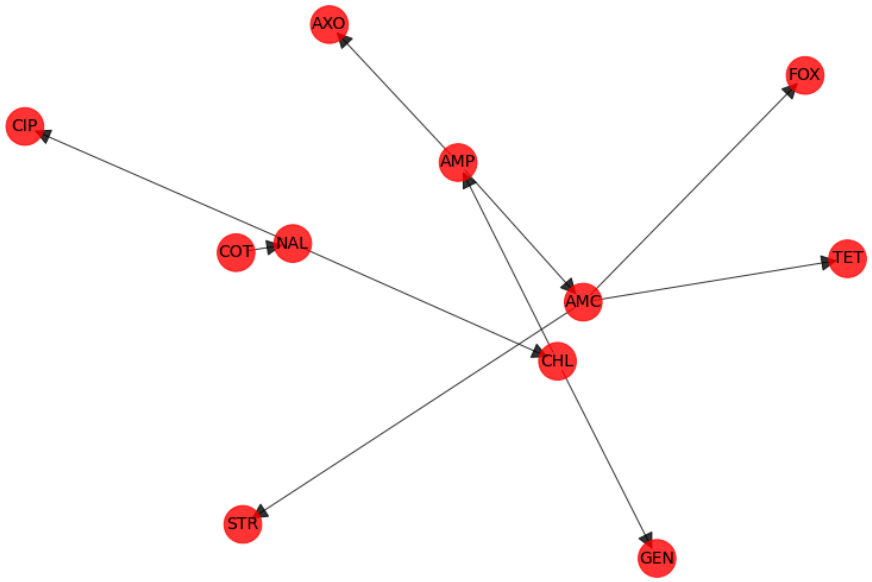
1. *Structure Learning*

   Given a set of random variables X, estimate a DAG that best fits the statistical dependencies among the variables in a given dataset.

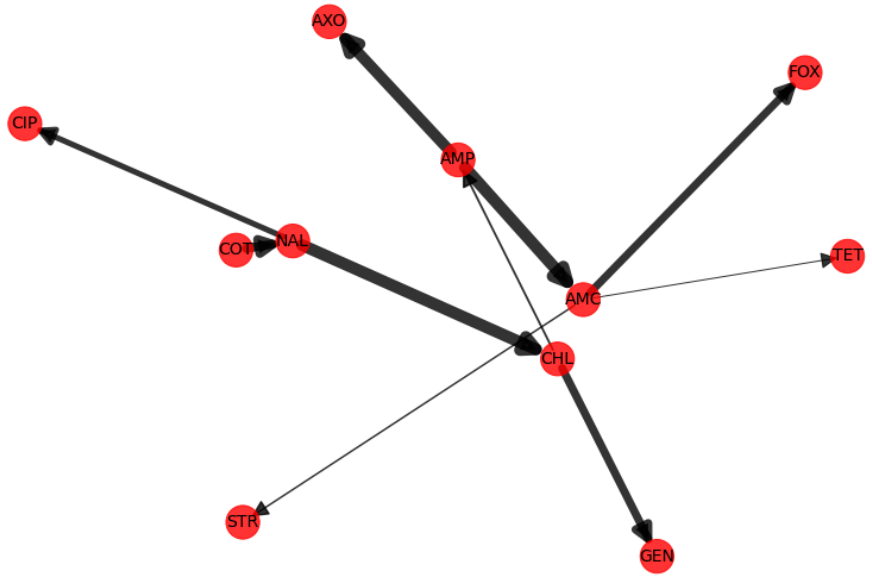
2. *Parameter Learning*

   Given a set of random variables X and a DAG that captures the statistical dependencies among them, calculate the conditional probability distributions for each variable in a given dataset.

bnlearn has multiple parameters, allowing the ability to choose the structure learning method, the score type to compare DAGs, and search among models with a set maximum in-degree (incoming) for nodes, etc. The package allows the use of multiple structure learning algorithms like Exhaustive Search, Hill-climb Search, Chow-Liu Algorithm, Tree-augmented Naive Bayes, and score types such as BIC (Bayesian Information Criterion), K2 and BDeu (Bayesian Dirichlet). For this thesis, the Hill Climb approach was used along with BIC as the score type for learning the structure of the Bayesian network with different in-degree nodes.



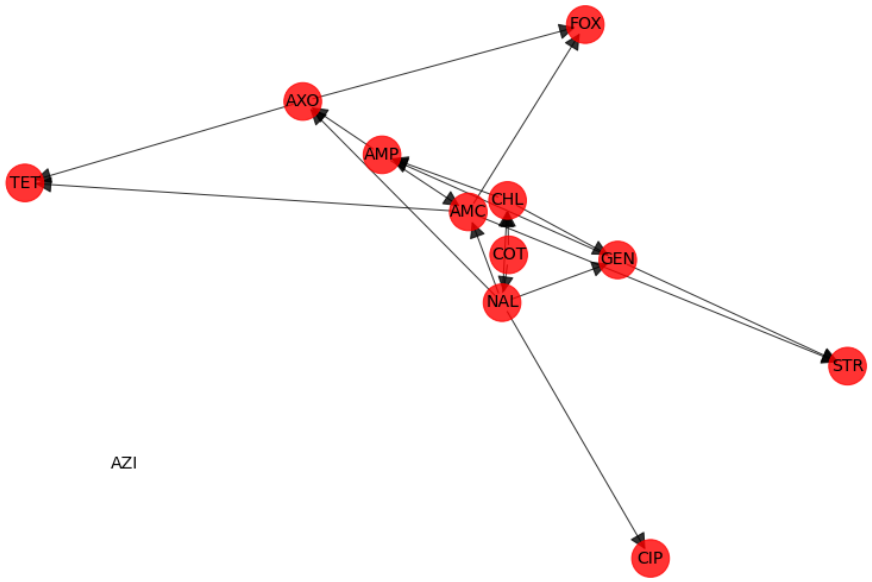
(a) estimated DAG



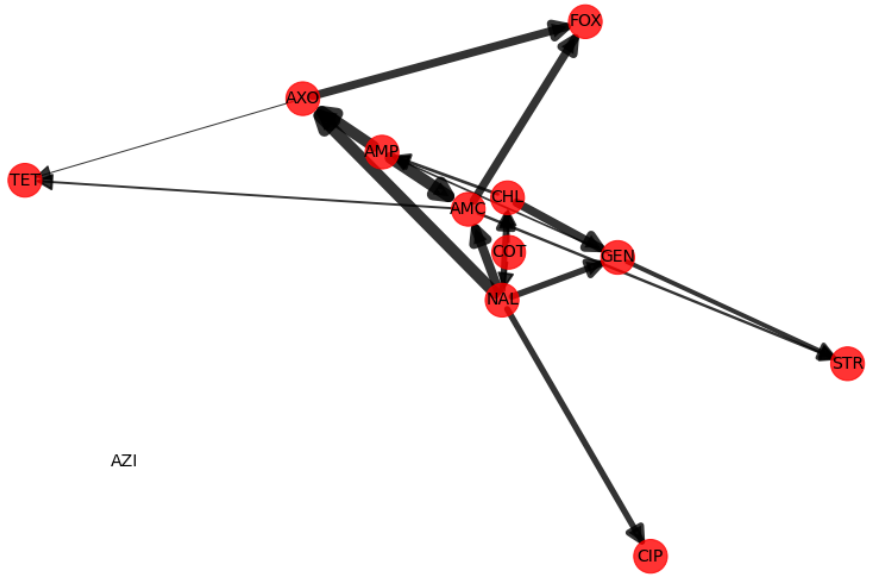
(b) DAG with edge significance

Figure 4.1: Structure Bayesian Learning for *Salmonella* data with max-indegree = 1

Panel a above shows the estimated Bayesian network DAG when the number of parent nodes is limited to one. In panel b, the edge significance depicts the chi-square statistic calculated for the test of independence. Darker edges represent higher statistic values indicating a strong association between the two nodes.



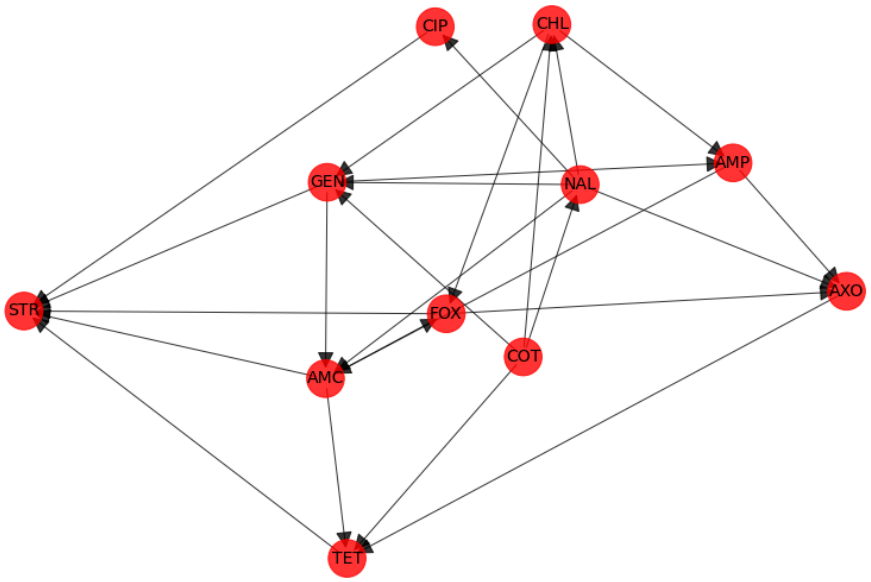
(a) estimated DAG



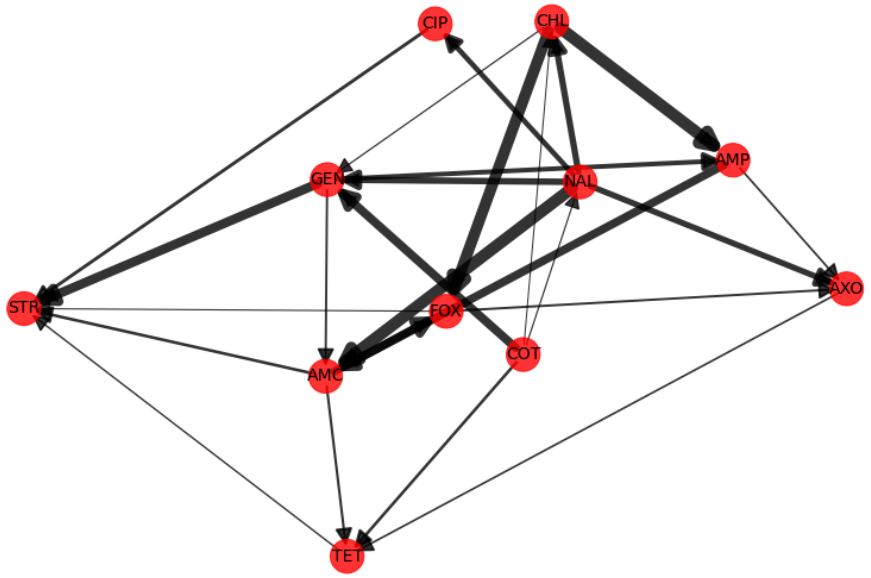
(b) DAG with edge significance

Figure 4.2: Structure Bayesian Learning for *Salmonella* data with max-indegree = 2

Changing the max parents for a given node to two changes some relationships between the antibiotics resistance. Now, there is a new edge between [NAL] –> [AXO] with high edge significance. [AZI] was cut out of the graph because there weren't enough resistant samples for the BN network to learn from.



(a) estimated DAG



(b) DAG with edge significance

Figure 4.3: Structure Bayesian Learning for *Salmonella* data with max-indegree = *None*

Here, the parent node restriction is removed and the Bayesian network assigns the number of parents based on the training data. STR has the maximum number of parents = 4.

§ 4.3 Chi-square Test of Independence

The chi-square independence test is used to determine if there is a statistical dependency among two categorical independent variables. Karl Pearson introduced the idea of the chi-square test and hypothesis testing in 1900 (Pearson, 1900). Pearson's chi-square test is given as follows:

$$\chi^2 = \sum_{k=1}^{n} \frac{(O_k - E_k)^2}{E_k} \tag{4.1}$$

where

- χ^2 is the chi-square test statistic
- O is the observed frequency
- E is the expected frequency

Similar to all the hypothesis tests, the chi-square test of independence evaluates a null and alternative hypothesis. If we want to know the relationship between variable 1 and variable 2, we can use the chi-square independence test. *bnlearn* Python package provides us with the option for performing chi-square tests for all the detected edges and also pruning the non-significant edges to reduce the False Discovery Rate (FDR).

- Null hypothesis, (H_0): The null hypothesis states that there is no relationship between variable 1 and variable 2.
- Alternative hypothesis, (H_1): The alternative hypothesis is that there is a relationship between variable 1 and variable 2

After we have calculated the chi-square statistic, we need to compare it with the critical chi-square value found in the critical chi-square table which follows a chi-square distribution first published by Elderton (Elderton, 1902). Based on the degrees of freedom (dof) which is the number of independent pieces of information used to calculate a statistic and the significance level or alpha (α) which is a value set

in advance as the threshold for statistical significance. If the p-value is greater than the significance level, the null hypothesis is not rejected and the result is not statistically significant. If the p-value is smaller than the significance level, the result is interpreted as refusing the null hypothesis and reported as statistically significant.

In our case, the significance level (α) was 0.05, and from Table 4.3, the degree of freedom is 1. So according to the chi-square table, the critical value is 3.841. Here, we can see that the edge [CHL <-> FOX] is excluded because it was not significant i.e p < 0.05, and its chi-square value is 1.43485 which is less than our critical value of 3.841.

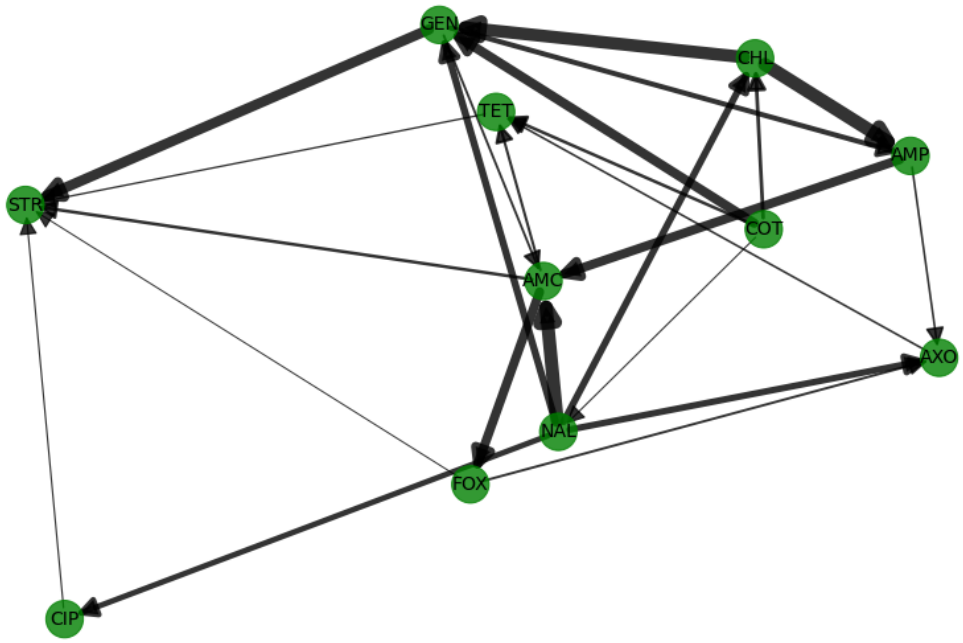


Figure 4.4: Pruned DAG for *Salmonella* learned with structure Bayesian learning

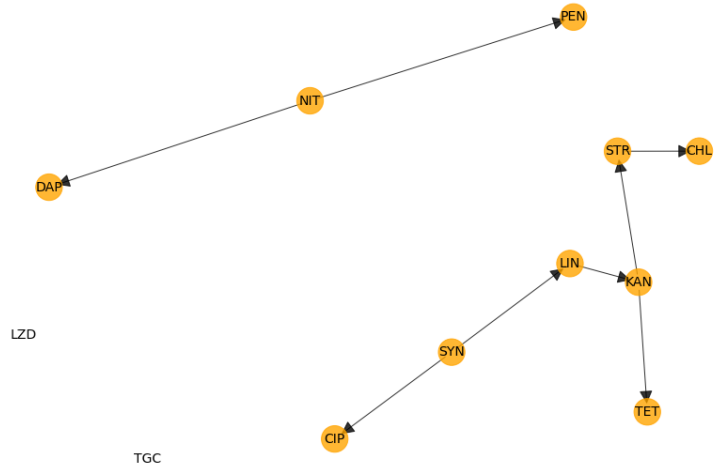
Final trained Bayesian network DAG for the *Salmonella* dataset after the pruning is done and removing the non-significant edge [CHL <-> FOX]

Table 4.3: Independence test for pruning the Bayesian network - *Salmonella*

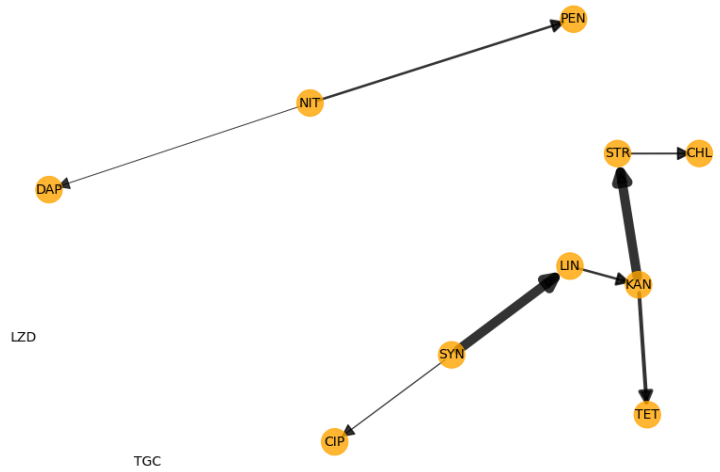
Source	Target	stat_test	p_value	chi_square	dof
AMC	FOX	True	4.72104e-206	938.263	1
AMC	STR	True	1.85515e-53	236.91	1
AMC	TET	True	2.58346e-39	172.092	1
AMP	AMC	True	1.75103e-309	1414.17	1
AMP	AXO	True	4.28622e-291	1329.55	1
AXO	TET	True	3.39571e-09	34.9426	1
CHL	GEN	True	1.91405e-213	972.27	1
CHL	AMP	True	2.6071e-63	282.108	1
CHL	FOX	**False**	0.230974	1.43485	1
CIP	STR	True	0.000265615	13.2985	1
COT	NAL	True	2.19916e-154	700.614	1
COT	CHL	True	3.11265e-148	672.329	1
COT	GEN	True	1.99166e-115	521.505	1
COT	TET	True	1.23817e-14	59.4756	1
FOX	AXO	True	5.49055e-205	933.361	1
FOX	STR	True	8.97837e-25	105.61	1
GEN	STR	True	9.15571e-34	146.694	1
GEN	AMC	True	6.49211e-07	24.7603	1
GEN	AMP	True	9.26667e-52	229.122	1
NAL	CIP	True	0	2872.65	1
NAL	CHL	True	1.14921e-233	1065.3	1
NAL	AMC	True	3.6733e-31	134.788	1
NAL	AXO	True	4.67793e-103	464.65	1
NAL	GEN	True	2.5296e-138	626.762	1
TET	STR	True	3.47679e-23	98.3666	1

§ 4.3.1 *E. faecalis*

In this section, structure learning for *E. faecalis* was performed to generate a DAG that captures dependencies among resistance to different antibiotics.



(a) estimated DAG

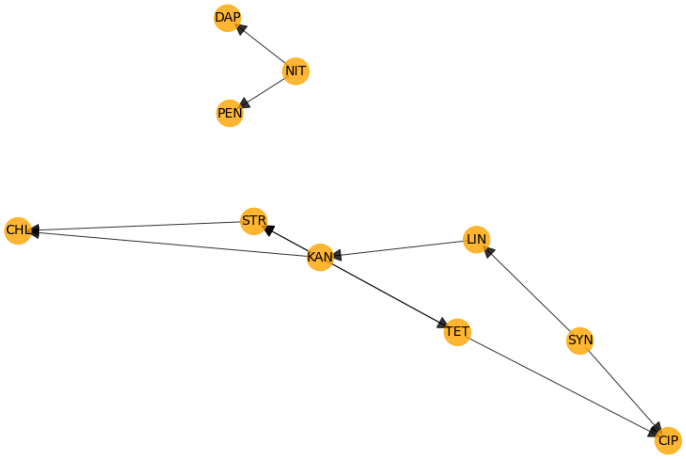


(b) DAG with edge significance

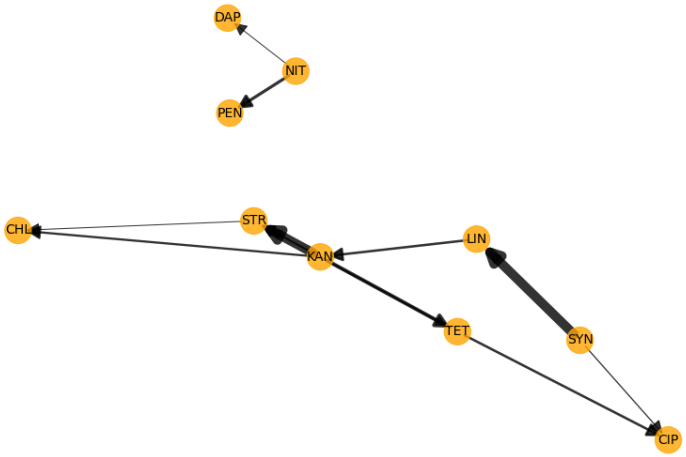
Figure 4.5: Structure Bayesian Learning for *E. faecalis* data with max-indegree = 1

Here, LZD and TGC are present without any associations as there are not enough positive isolates for these two antibiotics for the BN to learn from. Also, it can be seen that [DAP, NIT, PEN] form a separate tree graph.

bnlearn gives the option to restrict the maximum number of in-degree edges coming to a node using the *max_indegree* parameter.



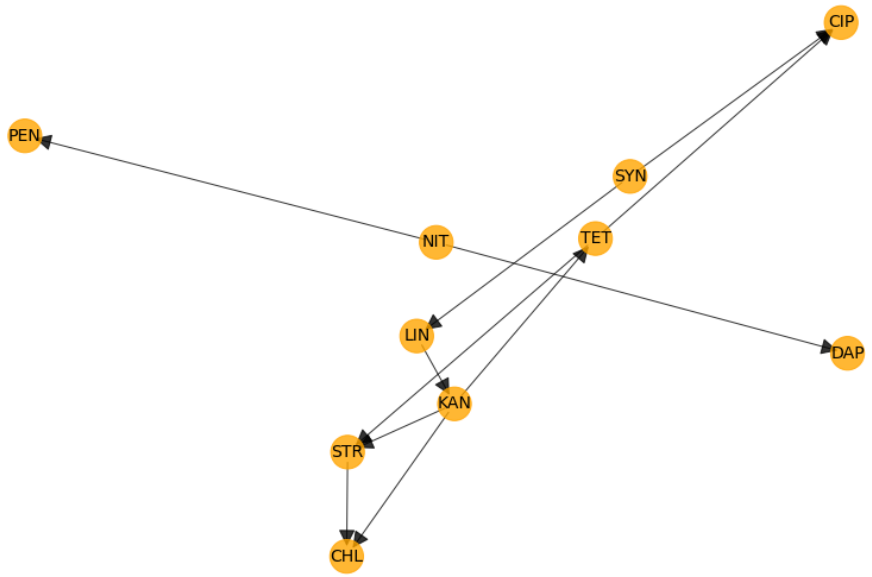
(a) estimated DAG



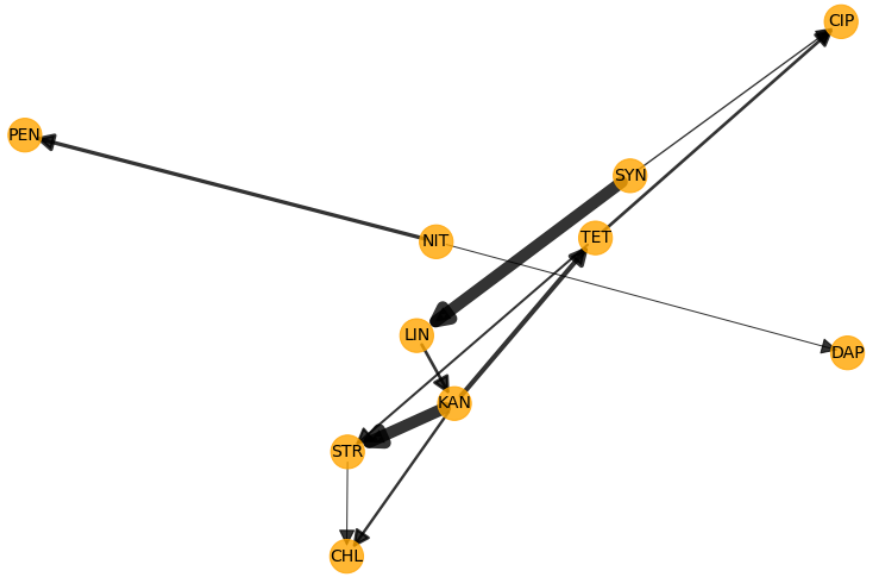
(b) DAG with edge significance

Figure 4.6: Structure Bayesian Learning for *E. faecalis* data with max-indegree = 2

After increasing the maximum number of parents a node can have to two, [CHL] and [CIP] now have two incoming nodes while the rest of the associations remain the same.



(a) estimated DAG



(b) DAG with edge significance

Figure 4.7: Structure Bayesian Learning for *E. faecalis* data with max-indegree = *None*

This is the trained Bayesian network DAG without any pre-set value of maximum parents per node. But the trained DAG still has a maximum number of parents two.

Next, the chi-square test of independence was performed to check for any non-significant edges. The significance level or alpha (α) is set at 0.05 again and the output is given below in Table 4.4.

Table 4.4: Independence test for pruning the Bayesian network - *E. faecalis*

source	**target**	**stat_test**	**p_value**	**chi_square**	**dof**
KAN	STR	True	8.78134e-282	1286.7	1
KAN	TET	True	6.718e-98	440.952	1
KAN	CHL	True	8.95525e-46	201.685	1
LIN	KAN	True	9.06238e-05	15.3226	1
NIT	DAP	True	7.00526e-54	238.85	1
NIT	PEN	True	7.13803e-35	151.763	1
STR	CHL	True	1.23285e-52	233.138	1
TET	STR	True	2.03216e-73	328.507	1
TET	CIP	True	0.000287483	13.1502	1
SYN	LIN	True	0	3349.3	1
SYN	CIP	True	3.11568e-14	57.66	1

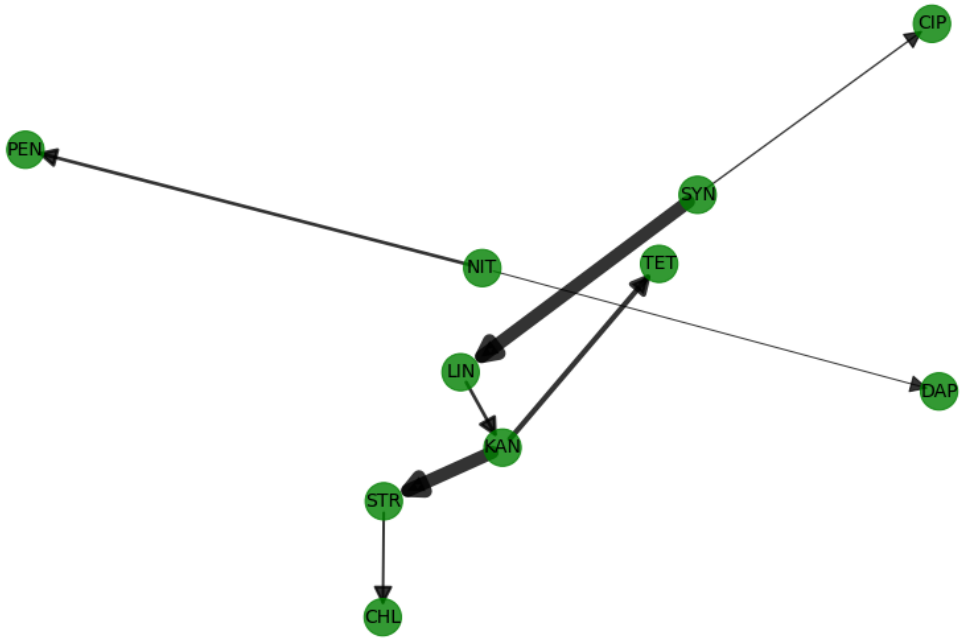
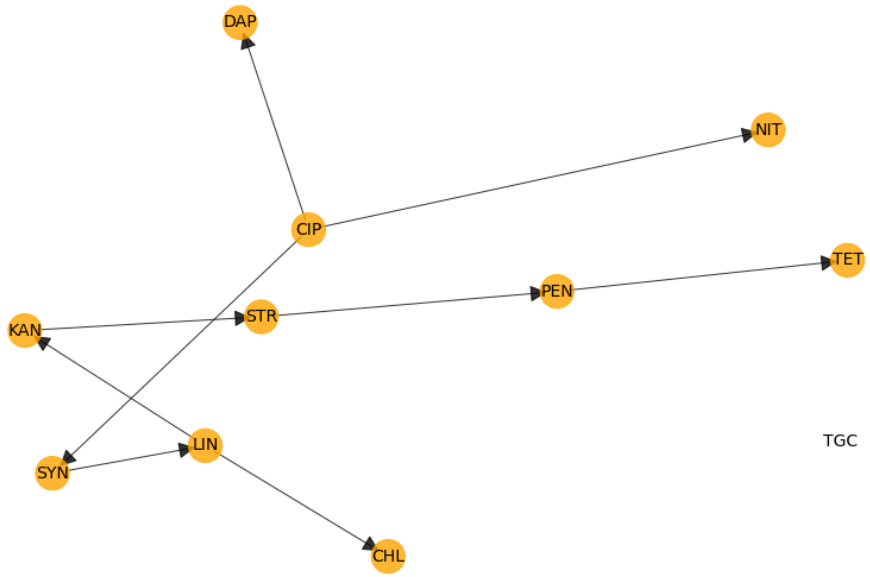


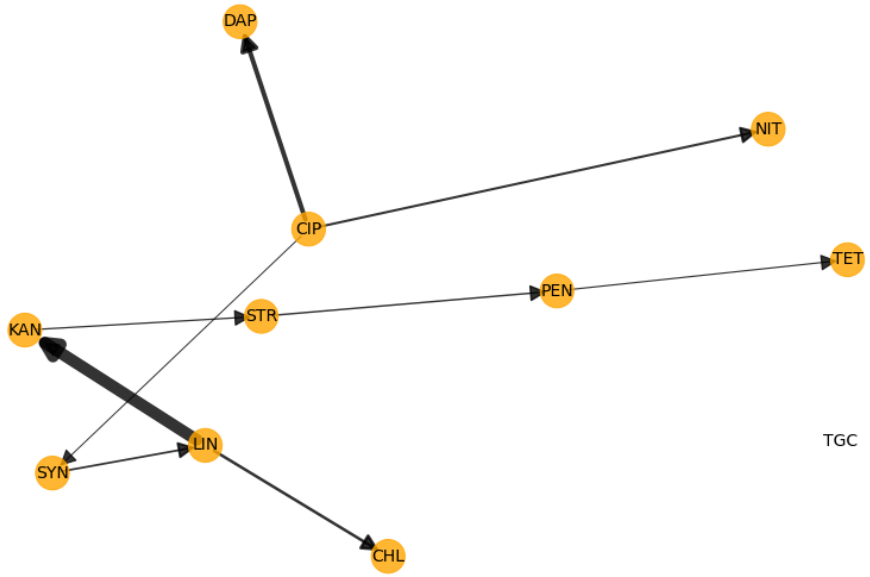
Figure 4.8: Pruned DAG for *E. faecalis* learned with structure Bayesian learning

§ 4.3.2 *E. faecium*

Similar to *E. faecalis*, the DAG was built that best describe the associations between different antibiotics.



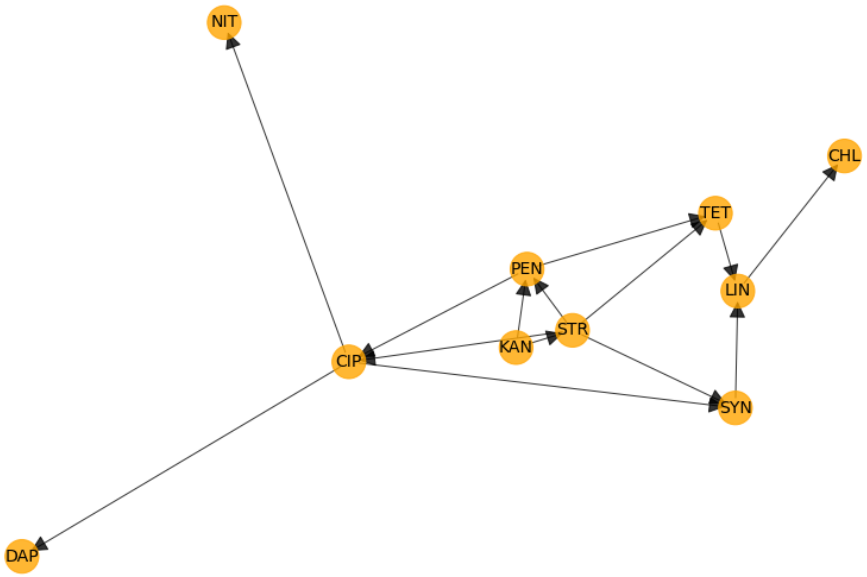
(a) estimated DAG



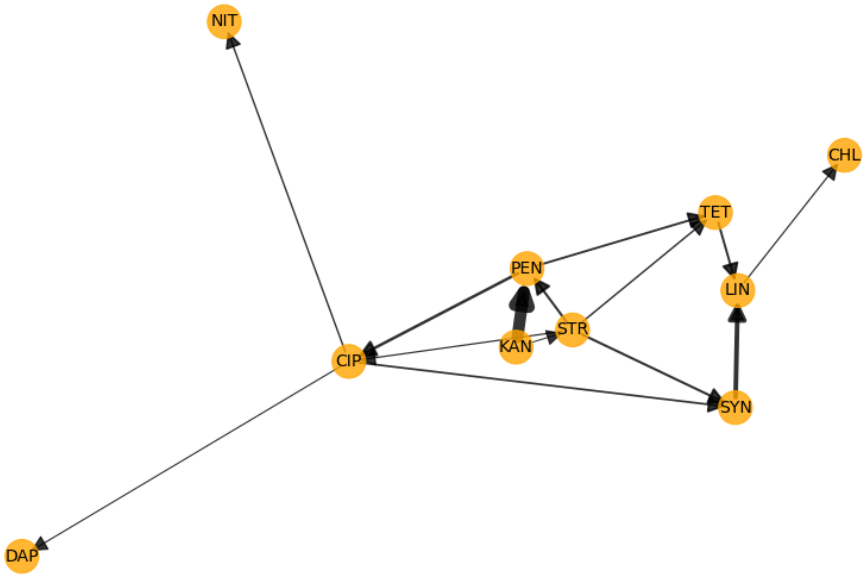
(b) DAG with edge significance

Figure 4.9: Structure Bayesian Learning for *E. faecium* data with max-indegree = 1

TGC is not associated with other nodes of the DAG since there were only 4 isolates that had tested positive for resistance and hence, it was pruned out. Panel b above suggests a strong association between [LIN] –> [KAN].



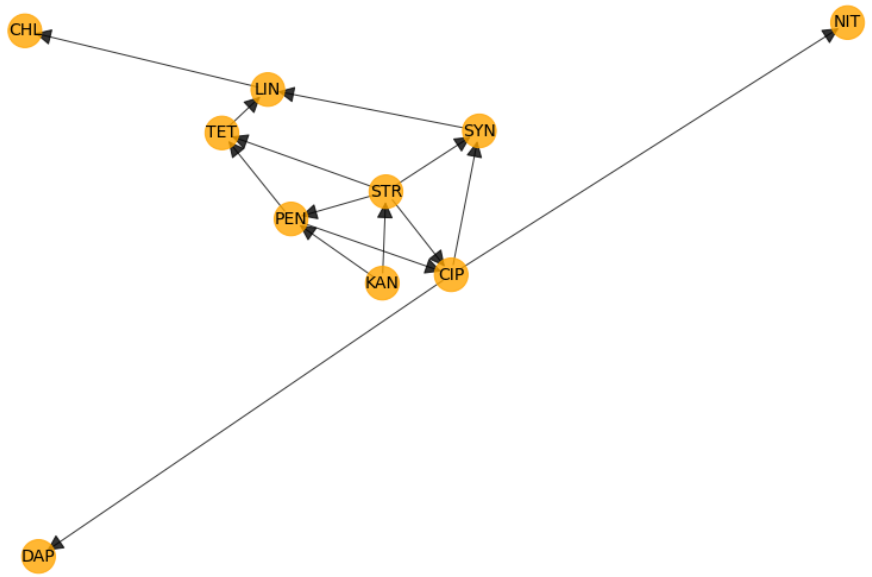
(a) estimated DAG



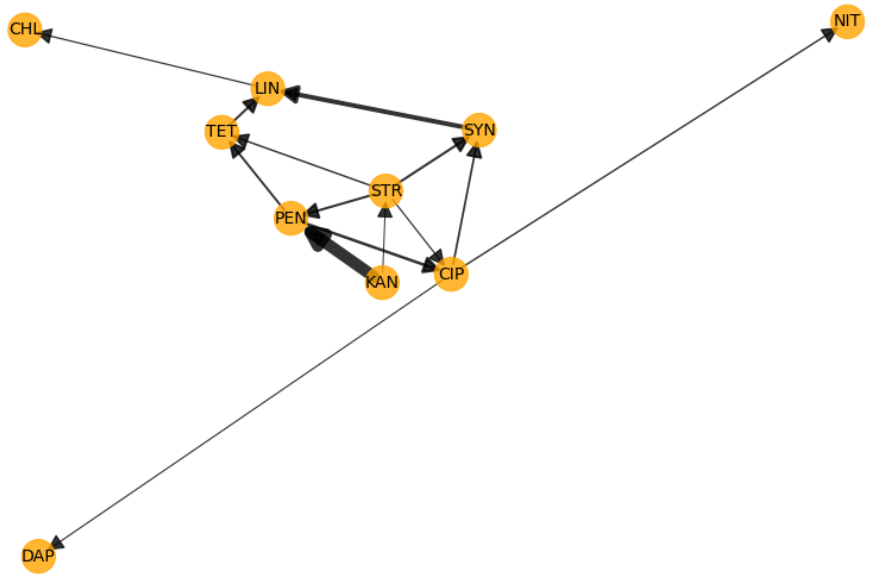
(b) DAG with edge significance

Figure 4.10: Structure Bayesian Learning for *E. faecium* data with max-indegree = 2

By changing the maximum number of parents per node, there is a change in associations. It can be seen that [LIN] [KAN] are no longer associated while there is a highly significant association seen between resistance to [KAN] and [PEN].



(a) estimated DAG



(b) DAG with edge significance

Figure 4.11: Structure Bayesian Learning for *E. faecium* data with max-indegree = *None*

Here, the maximum number of parents limit was removed and the Bayesian network was trained. But, it can be seen that there is no difference between this graph and when the maximum number of parents was limited to two.

Table 4.5: Independence test for pruning the Bayesian network - *E. faecium*

source	target	stat_test	p_value	chi_square	dof
CIP	SYN	True	4.70848e-15	61.3788	1
CIP	NIT	True	1.83005e-09	36.1467	1
CIP	DAP	True	0.00020405	13.7934	1
KAN	STR	True	1.24918e-55	246.871	1
KAN	PEN	True	5.09121e-19	79.3928	1
LIN	CHL	**False**	0.0909094	2.8582	1
PEN	TET	True	1.83679e-28	122.453	1
PEN	CIP	True	4.53012e-06	21.0263	1
STR	PEN	True	1.13001e-20	86.92	1
STR	SYN	True	3.49938e-10	39.3739	1
STR	TET	True	6.48412e-21	*88.0186*	1
STR	CIP	True	6.15414e-06	20.4397	1
TET	LIN	True	8.00459e-18	73.9518	1
SYN	LIN	True	6.45169e-169	767.448	1

The table above shows that the edge [LIN <-> CHL] fails the chi-square independence test and is termed a non-significant edge. The rest of all the edges pass the chi-square test so they will be retained in the final pruned Bayesian network for *E. faecium*

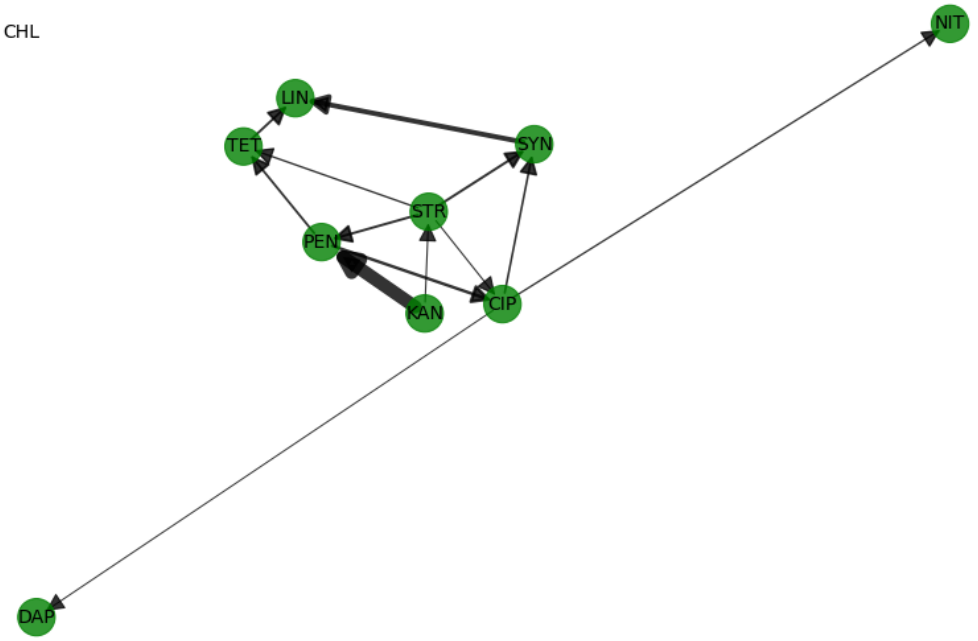


Figure 4.12: Pruned DAG for *E. faecium* learned with structure Bayesian learning

§ 4.4 Generating Association Rules Using Apriori

In this section, the association rules were generated using the *mlxtend* package available in python (Raschka, 2018). *mlxtend* provides multiple options for selecting the algorithm for frequent itemset generation like *apriori*, *fpgrowth*, and *fpmax* along with a function to generate the association rules. Different Interest Measures (IMs) such as Support, Confidence, Lift, Leverage, and Conviction can be measured which have been discussed in the literature review.

One aspect to consider in association rule mining is that some of the discovered rules may occur by chance rather than true associations. To keep the False Discovery rate (Type I error) to under 5%, the itemsets can be pruned according to one of the interest measures, by setting an appropriate threshold. In our case, a minimum support threshold was used to prune the insignificant or non-relevant itemsets (Tan et al., 2004).

Table 4.6: Association rules sorted by Lift for *Salmonella* AST dataset

antecedents	consequents	support	confidence	lift	leverage	conviction
['NAL']	['CIP']	0.214646	0.99688	4.537785	0.167344	250.0912
['CIP']	['NAL']	0.214646	0.977064	4.537785	0.167344	34.21216
['AMC']	['AMP']	0.165939	1	3.394527	0.117055	inf
['AXO']	['AMP']	0.159893	0.989605	3.359241	0.112295	67.86026
['COT']	['TET']	0.070877	0.976852	1.288475	0.015868	10.4481
['AMC', 'AXO']	['FOX']	0.063823	0.954774	13.73122	0.059175	20.57366
['FOX', 'AMC']	['AXO']	0.063823	0.984456	6.092984	0.053348	53.93886
['FOX', 'AMP']	['AXO']	0.063823	0.984456	6.092984	0.053348	53.93886
['FOX', 'AMP']	['AMC']	0.06483	1	6.026316	0.054072	inf
['FOX', 'AXO']	['AMC']	0.063823	1	6.026316	0.053232	inf
['AMP', 'NAL']	['AXO']	0.093383	0.968641	5.995103	0.077806	26.73654
['AMP', 'CIP']	['AXO']	0.093383	0.961938	5.953615	0.077698	22.02779
['CHL', 'CIP']	['NAL']	0.11085	1	4.644306	0.086982	inf
['CIP', 'COT']	['NAL']	0.067518	1	4.644306	0.05298	inf
['GEN', 'CIP']	['NAL']	0.117904	0.997159	4.631112	0.092445	276.2083
['CIP', 'AXO']	['NAL']	0.09439	0.996454	4.627837	0.073994	221.2805
['AMP', 'CIP']	['NAL']	0.096406	0.99308	4.612165	0.075503	113.3866
['CIP', 'STR']	['NAL']	0.162244	0.98773	4.58732	0.126876	63.95163
['TET', 'CIP']	['NAL']	0.181727	0.983636	4.568308	0.141947	47.95282
['AMP', 'NAL']	['CIP']	0.096406	1	4.551988	0.075227	inf
['CHL', 'NAL']	['CIP']	0.11085	1	4.551988	0.086498	inf
['NAL', 'COT']	['CIP']	0.067518	1	4.551988	0.052685	inf
['GEN', 'NAL']	['CIP']	0.117904	1	4.551988	0.092002	inf
['NAL', 'STR']	['CIP']	0.162244	0.997934	4.542583	0.126528	377.6728
['NAL', 'AXO']	['CIP']	0.09439	0.996454	4.535846	0.07358	220.049

Table 4.7: Association rules sorted by Lift for *E. faecium* AST dataset

antecedents	consequents	support	confidence	lift	leverage	conviction
['SYN']	['LIN']	0.724535	0.996346	1.26107	0.149995	57.44818
['STR']	['LIN']	0.128432	0.986395	1.248475	0.025561	15.42914
['KAN']	['LIN']	0.086802	0.98	1.240381	0.016822	10.49601
['STR']	['NIT']	0.127547	0.979592	1.036513	0.004493	2.690877
['DAP']	['NIT']	0.126661	0.979452	1.036365	0.004444	2.672572
['PEN']	['NIT']	0.141718	0.97561	1.032299	0.004434	2.25155
['CIP']	['NIT']	0.659876	0.973856	1.030444	0.019496	2.100531
['LIN', 'DAP']	['SYN']	0.104517	0.95935	1.319252	0.025293	6.711072
['LIN', 'STR']	['SYN']	0.123118	0.958621	1.318249	0.029723	6.592855
['NIT', 'STR']	['SYN']	0.121346	0.951389	1.308305	0.028596	5.612046
['SYN', 'STR']	['LIN']	0.123118	1	1.265695	0.025845	inf
['NIT', 'SYN']	['LIN']	0.677591	0.996094	1.260751	0.140141	53.73959
['CIP', 'SYN']	['LIN']	0.441984	0.996008	1.260642	0.091382	52.58503
['NIT', 'STR']	['LIN']	0.126661	0.993056	1.256906	0.025889	30.22852
['PEN', 'SYN']	['LIN']	0.126661	0.993056	1.256906	0.025889	30.22852
['TET', 'SYN']	['LIN']	0.354296	0.992556	1.256273	0.072274	28.19929
['SYN', 'DAP']	['LIN']	0.104517	0.991597	1.255059	0.02124	24.98051
['TET', 'STR']	['LIN']	0.104517	0.991597	1.255059	0.02124	24.98051
['KAN', 'SYN']	['LIN']	0.081488	0.989247	1.252085	0.016406	19.52259
['CIP', 'STR']	['LIN']	0.10806	0.983871	1.245281	0.021284	13.01506
['KAN', 'NIT']	['LIN']	0.081488	0.978723	1.238765	0.015706	9.866253
['KAN', 'CIP']	['LIN']	0.070859	0.97561	1.234824	0.013475	8.606732
['KAN', 'TET']	['LIN']	0.065545	0.973684	1.232387	0.01236	7.976971
['PEN', 'TET']	['LIN']	0.117803	0.956835	1.211061	0.02053	4.863153

Table 4.8: Association rules sorted by Lift for *E. faecalis* AST dataset

antecedents	consequents	support	confidence	lift	leverage	conviction
['STR']	['TET']	0.123586	0.988487	1.491934	0.04075	29.3096
['KAN']	['TET']	0.168209	0.97497	1.471533	0.0539	13.48177
['STR']	['SYN']	0.124614	0.996711	1.019992	0.002442	6.938927
['KAN']	['SYN']	0.17191	0.996424	1.019699	0.003321	6.383508
['CHL']	['SYN']	0.037837	0.994595	1.017827	0.000663	4.222702
['CIP']	['SYN']	0.483858	0.994085	1.017306	0.008231	3.859126
['CHL']	['LIN']	0.038042	1	1.016301	0.00061	inf
['KAN']	['LIN']	0.172527	1	1.016301	0.002767	inf
['LIN']	['SYN']	0.977175	0.993103	1.016301	0.015673	3.309685
['SYN']	['LIN']	0.977175	1	1.016301	0.015673	inf
['STR']	['LIN']	0.12482	0.998355	1.014629	0.0018	9.752005
['CIP']	['LIN']	0.485092	0.99662	1.012866	0.006162	4.745682
['TET']	['SYN']	0.652067	0.984171	1.00716	0.004636	1.442029
['TET']	['LIN']	0.655151	0.988827	1.004946	0.003224	1.435534
['KAN', 'STR']	['TET']	0.08575	0.997608	1.5057	0.0288	141.0524
['LIN', 'STR']	['TET']	0.123381	0.988468	1.491905	0.040681	29.26139
['SYN', 'STR']	['TET']	0.123175	0.988449	1.491877	0.040611	29.21318
['CIP', 'STR']	['TET']	0.060457	0.986577	1.489052	0.019856	25.13973
['KAN', 'CIP']	['TET']	0.080403	0.9775	1.475351	0.025905	14.9976
['KAN', 'SYN']	['TET']	0.167798	0.976077	1.473203	0.053898	14.10524
['KAN', 'LIN']	['TET']	0.168209	0.97497	1.471533	0.0539	13.48177
['TET', 'CHL']	['SYN']	0.034752	1	1.023359	0.000793	inf
['KAN', 'CIP']	['SYN']	0.082254	1	1.023359	0.001877	inf
['TET', 'CIP']	['SYN']	0.309685	0.998674	1.022001	0.006667	17.21036
['LIN', 'STR']	['SYN']	0.124614	0.998353	1.021673	0.002643	13.85503

§ 4.5 Results

To validate the Bayesian network, the dataset was randomly split into 80/20% training and testing tests. The Conditional Probability Distributions (CPDs) for the single-node interaction pairs predicted by the Bayesian Network trained on the training set were compared to the CPDs calculated from the testing set. Since CPDs were compared, we will be using the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) were used as evaluation metrics. The performance of the Bayesian networks on different datasets was as follows:

Table 4.9: Performance of Bayesian network models on Test data

Bacteria species	**Train set samples**	**Test set samples**	**RMSE**	**MAE**
Salmonella	2344	633	0.0009	0.0237
E. faecium	891	236	0.0017	0.0320
E. faecalis	3875	954	0.0063	0.0312

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this thesis, the associations and co-occurrence of resistance patterns among antibiotics and AMR genes respectively were identified using Bayesian networks and association rule mining. The trained Bayesian models were able to achieve a low MAE of 0.0237, 0.0320, and 0.0312 on the *Salmonella*, *E. faecium* and *E. faecalis* testing datasets respectively. Training the models with different numbers of maximum in-degree nodes was evaluated to check for different pair-level implications. We have shown how we can generate association rules for identifying resistance patterns to different antibiotics.

In the literature review, research projects which used anything other than Bayesian networks and association rule mining for solving a similar problem to ours were non-existent. In the future, different association rules can be mined and compared for different periods of time to check for any changes in the resistance association patterns over the years. Also, cross-validation can be performed to train the Bayesian network with different portions of the training data and check how well the model generalizes on the testing data. A similar approach can be used for finding the associations among resistance to different antibiotics can in other bacterial strains.

AMR resistance remains difficult to interpret, but machine learning and statistics can be used to push the boundaries of understanding patterns that follow seemingly random paths.

BIBLIOGRAPHY

Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *SIGMOD Rec.*, *22*, 207–216. https://doi.org/10.1145/170036.170072

Agrawal, R., Srikant, R., et al. (1994). Fast algorithms for mining association rules. *1215*, 487–489.

Alanis, A. J. (2005). Resistance to antibiotics: Are we in the post-antibiotic era? *Arch Med Res*, *36*, 697–705. https://doi.org/10.1016/j.arcmed.2005.06.009

Baker, D. N., & Langmead, B. (2019). Dashing: fast and accurate genomic distances with HyperLogLog. *Genome Biol.*, *20*(1), 265.

Brin, S., Motwani, R., Ullman, J. D., & Tsur, S. (1997). Dynamic itemset counting and implication rules for market basket data.

Centers for Disease Control, P. (2019). Antibiotic resistance threats in the United States, 2019. Atlanta, GA: US Department of Health and Human Services, CDC; 2019.

CLSI. (2020). Performance standards for antimicrobial susceptibility testing. *30th ed. CLSI supplement M100*, *Vol M100. Wayne, PA*.

Cooper, G. (1999). An overview of the representation and discovery of causal relationships using bayesian networks. *Computation, causation, and discovery*, 4–62.

D. Heckerman, G. C., C. Meek. (1999). *A bayesian approach to casual discovery*. MIT Press.

Daly, R., Shen, Q., & Aitken, S. (2011). Learning bayesian networks: Approaches and issues. *The Knowledge Engineering Review*, *26*(2), 99–157. https://doi.org/10.1017/S0269888910000251

Davies, J., & Davies, D. (2010). Origins and evolution of antibiotic resistance. *Microbiology and molecular biology reviews*, *74*(3), 417–433.

Elderton, W. P. (1902). Tables for testing the goodness of fit of theory to observation. *Biometrika*, *1*(2), 155–163.

Feldgarden, M., Brover, V., Gonzalez-Escalona, N., Frye, J. G., Haendiges, J., Haft, D. H., Hoffmann, M., Pettengill, J. B., Prasad, A. B., Tillman, G. E., et al. (2021). Amrfinderplus and the reference gene catalog facilitate examination of the genomic links among antimicrobial resistance, stress response, and virulence. *Scientific reports*, *11*(1), 1–9.

Frye, J. G., & Jackson, C. R. (2013). Genetic mechanisms of antimicrobial resistance identified in Salmonella enterica, Escherichia coli, and Enteroccocus spp. isolated from us food animals. *Frontiers in microbiology*, *4*, 135.

Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika*, *32*(3), 241–254.

Mathew AG, L. S., Cissell R. (2007). Antibiotic resistance in bacteria associated with food animals: A united states perspective of livestock production. *Foodborne Pathog Dis*, *4*, 115–133. https://doi.org/10.1089/fpd.2006.0066

Mead, P. S., Slutsker, L., Dietz, V., McCaig, L. F., Bresee, J. S., Shapiro, C., Griffin, P. M., & Tauxe, R. V. (1999). Food-related illness and death in the united states. *Emerging infectious diseases*, *5*(5), 607.

Ondov, B. D., Treangen, T. J., Melsted, P., Mallonee, A. B., Bergman, N. H., Koren, S., & Phillippy, A. M. (2016). Mash: Fast genome and metagenome distance estimation using minhash. *Genome biology*, *17*(1), 1–14.

Pearl, J. (1985). Bayesian networks: A model cf self-activated memory for evidential reasoning. *Proceedings of the 7th conference of the Cognitive Science Society, University of California, Irvine, CA, USA*, 15–17. Retrieved October 8, 2022, from https://ftp.cs.ucla.edu/pub/stat_ser/r43-1985.pdf

Pearson, K. (1900). X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *50*(302), 157–175.

Piatetsky-Shapiro, G. (1991). Discovery, analysis, and presentation of strong rules.

Raschka, S. (2018). Mlxtend: Providing machine learning and data science utilities and extensions to python's scientific computing stack. *The Journal of Open Source Software*, *3*(24). https://doi.org/10.21105/joss.00638

Robinson, R. (1977). Counting unlabeled acyclic digraphs. *Combinatorial Mathematics*, *622*, 28–43. https://doi.org/10.1007/BFb0069178

Scutari, M., Vitolo, C., & Tucker, A. (2019). Learning bayesian networks from big data with greedy search: Computational complexity and efficient implementation. *Statistics and Computing*, *29*(5), 1095–1108.

Sood, S., Malhotra, M., Das, B., Kapil, A., et al. (2008). Enterococcal infections & antimicrobial resistance. *Indian journal of medical research*, *128*(2), 111.

Tan, P.-N., Kumar, V., & Srivastava, J. (2004). Selecting the right objective measure for association analysis. *Inf. Syst.*, *29*(4), 293–313. https://doi.org/10.1016/S0306-4379(03)00072-3

Taskesen, E. (2020). *Bnlearn - library for bayesian network learning and inference* (Version 0.3.22). https://erdogant.github.io/bnlearn

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2

Voetsch, A. C., Van Gilder, T. J., Angulo, F. J., Farley, M. M., Shallow, S., Marcus, R., Cieslak, P. R., Deneen, V. C., Tauxe, R. V., & Group, E. I. P. F. W. (2004). Foodnet estimate of the burden of illness caused by nontyphoidal Salmonella infections in the united states. *Clinical infectious diseases*, *38*(Supplement_3), S127–S134.

Zhao, Q., & Bhowmick, S. S. (2003). Association rule mining: A survey. *Nanyang Technological University, Singapore*, *135*.

APPENDIX

APPENDIX A

§ .0.1 Streamlit Web Application for Predicting the Probability of co-occurrence of AMR genes

We deployed the trained Bayesian network models for all three *Salmonella*, *Enterococcus faecalis* and *Enterococcus faecium* using Steamlit - a python package for deploying trained Machine Learning models on a web application.

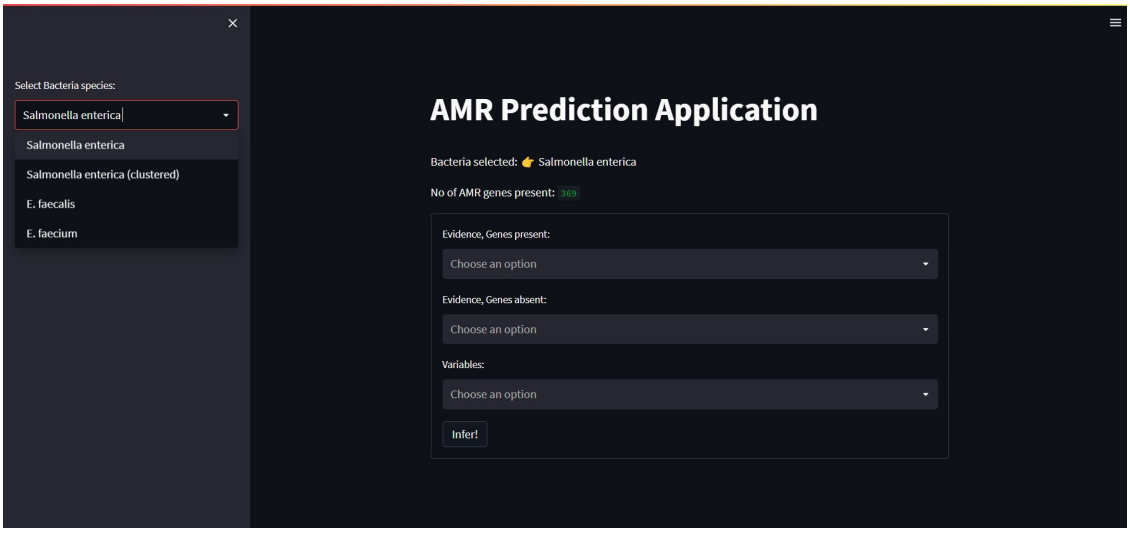


Figure 1: Streamlit Web application UI

We have 3 options to select from, viz, evidence-absent & present genes and the target variables or genes for which we need to predict the probability. We could ask the web app certain questions such as given that we already know the presence or absence of these sets of genes, what is the likelihood or probability that we are going to see the presence of these AMR genes? A sample example is shown in Figure 2 below.

	arsC	p
0	0	0.9346
1	1	0.0654

Figure 2: Streamlit Web application Inference