Analysis of VRE Transmission in a Major Hospital Setting Using Hierarchical Clustering and Bayesian Phylodynamic Methods

by

Arvon Anthony Clemons II

Biochemistry and Biotechnology BS, University of Missouri – St. Louis, 2017

Biochemistry and Biotechnology MS, University of Missouri – St. Louis, 2017

Submitted to the Graduate Faculty of the

Graduate School of Public Health in partial fulfillment

of the requirements for the degree of

Master of Science

University of Pittsburgh

2022

UNIVERSITY OF PITTSBURGH

GRADUATE SCHOOL OF PUBLIC HEALTH

This thesis was presented

by

Arvon Anthony Clemons II

It was defended on

April 22, 2022

and approved by

Jenna C. Carlson, PhD, Assistant Professor, Department of Biostatistics, School of Public Health, University of Pittsburgh

Ada O. Youk, PhD, Associate Professor and Vice Chair of Education, Department of Biostatistics, School of Public Health, University of Pittsburgh

Daria Van Tyne, PhD, Assistant Professor, Department of Medicine, Division of Infectious Diseases, University of Pittsburgh

Thesis Advisor: Jenna C. Carlson, PhD, Assistant Professor, Department of Biostatistics, School of Public Health, University of Pittsburgh

Copyright © by Arvon Anthony Clemons II

2022

Analysis of VRE Transmission in a Major Hospital Setting Using Hierarchical Clustering and Bayesian Phylodynamic Methods

Arvon Anthony Clemons II, MS

University of Pittsburgh, 2022

Healthcare-associated infections (HAIs) can prolong and add substantial costs to hospital stays. One study estimated that 1 out of 25 hospitalized patients were expected to be infected by a HAI on a daily basis. Minimizing HAIs would increase the quality of healthcare within hospitals; thus infection prevention methods must utilize various strategies to identify the cause of HAIs and develop interventions to reduce cases.

Currently many healthcare institutions utilize whole-genome sequencing (WGS) in identifying outbreaks and combine them with epidemiological methods in developing protocols to minimize the size of an outbreak. A recent example would be researchers in the University of Pittsburgh – Medical Center Presbyterian Hospital (UPMC), who have developed a machine-learning based method to incorporate WGS data with electronic health records (EHRs) to determine the most likely routes of transmission during an outbreak of vancomycin-resistant enterococci (VRE).

Using a ground truth (GT) dataset based on the VRE outbreak, we performed an assessment to compare two methods for categorizing bacterial isolates into transmission routes. We compared hierarchical clustering methods with a Bayesian phylodynamic model to determine which classification had the most similarity to the GT dataset.

Our analysis proved inconclusive in identifying a method with superior performance due to computational limitations for the Bayesian phylodynamic model, however the urgency and time

constraints of an active outbreak have shown to be better suited for the hierarchical clustering method and we recommend the Bayesian phylodynamic model as part of a retrospective analysis of an outbreak. This analysis which identifies routes of infectious disease transmission within a hospital setting could be utilized in optimizing infection prevention strategies within the hospital setting and lower the rate of HAIs – making a positive public health impact through reducing cost of care and increasing quality of care.

Table of Contents

List of Tables

List of Figures

Preface

I dedicate this preface to those who have helped me reached this point of my academic journey.

Thank you Dr. Jenna Carlson for your continuous support, patience and willingness to lend me an ear whenever I needed it during this arduous process. Spending three-fourths of my public health graduate experience from home during a global pandemic is not what I or any other student would ever have expected but your extra effort helped me see through it. I extend a warm thank you to Dr. Ada Youk as well for her advice, wisdom and guidance. Your enthusiasm and openness really helped me feel welcome in the program and helped Pittsburgh feel like a new home.

I thank Dr. Daria Van Tyne for welcoming me into her lab and providing me the opportunity to work on a project that would give me so much enthusiasm as well as her advice.

I would also like to thank the faculty of the Human Genetics department and Infectious Diseases & Microbiology department. For me this thesis is a combination of all that I have learned in the fields of Biostatistics, Infectious Diseases, and Population Genetics during my time at the University of Pittsburgh. The skills I've learned from your diverse coursework helped make this thesis possible in so many ways and I hope that is clearly reflected in the following text.

I also thank my cohort for their support and fun memories during my time in the program and I wish all the best of luck in their future careers as I know they would do the same for me.

Last but certainly not least I thank my friends and family from St. Louis, it was hard spending so much time away from you in Pittsburgh but I know you all had me in your thoughts just as I had you in mine. Thank you for your continuing support.

1.0 Introduction

Healthcare-associated infections (HAIs) are a major concern in the US and are known to increase health care costs substantially while also contributing to morbidity and mortality. In 2011 the CDC conducted a prevalence survey on HAIs through the National Healthcare Safety Network (NHSN) in 10 geographically diverse states and estimated that 1 out of 25 hospitalized patients were expected to be infected by a HAI on a daily basis (Magill, 2014). It was estimated that there were 648,000 patients with 721,800 HAIs within acute care US hospitals. Through data-driven coordination regional healthcare facilities could substantially reduce HAIs (Slayton, 2015). Medical cost savings from preventing HAIs could range from $25 - $31.5 billion (Douglass S., 2009). As such, the U.S. Department of Health and Human Services (HHS) has recognized the reduction of HAIs as an Agency Priority Goal (HHS.gov, 2021). One of the goals is to use data to facilitate core methods of infection prevention (IP) such as: (1) implementation of electronic health records (EHRs) for antibiotic treatment, (2) clinical staff duty records for logging environmental cleaning and disinfection activities, and (3) automate identification of HAIs in contrast to manual chart reviews (Atreja, 2008). In particular, whole-genome sequencing (WGS) is prominently used in outbreak investigations within hospitals and is used in tandem with epidemiological methodology to identify transmission routes with some limitations (Sundermann, 2019). To overcome some of the limitations of using traditional IP with WGS, researchers at the University of Pittsburgh developed a tool called Enhanced Detection System for Healthcare Association Transmission (EDS-HAT) which incorporates WGS surveillance with machine learning EHR data (Sundermann, 2021). This tool has shown promise to enhancing prior IP methods, in one case

EDS-HAT was successfully used to identify a previously undiscovered transmission route through poor interventional radiology technician aseptic technique (Sundermann, 2020).

WGS data in outbreak surveillance is often done through hierarchical clustering methods, in order to identify clusters of cases which may indicate a common transmission route. However, the use of hierarchical clustering comes with limitations for identifying transmission routes. Dissimilarity metrics are used for hierarchical clustering, in particular single-nucleotide polymorphisms (SNP), but lack geotemporal inference. As such, hierarchical clustering can only be used to infer genetically related cases. However, with Bayesian phylogenetics, geographical and temporal information can be incorporated to illustrate the genetic ancestry of cases. This additional information can indicate migration history and thus transmission routes. In this thesis, I will compare hierarchical clustering with Bayesian phylogenetics and contrast the performance of clusters generated from both methods as input into EDS-HAT in correctly identifying transmission routes.

2.0 Methods

The aim of this project was to combine whole-genome sequences (WGS) of bacterial isolates from hospital-associated disease surveillance of antibiotic resistant bacterial pathogens with electronic health records to identify outbreaks and transmission routes. Specifically, we performed and compared two different machine learning methods (hierarchical clustering and Bayesian phylodynamic modeling) to identify clusters of isolates and likely transmission routes for vancomycin-resistant enterococci (VRE).

2.1 Data

2.1.1 Data Source

The dataset for these analyses is composed of de novo genome assemblies and genetic dissimilarity matrices of 267 VRE isolates collected from unique patients as part of the EDS-HAT project conducted at University of Pittsburgh – Medical Center Presbyterian Hospital (UPMC). Whole-genome sequencing (WGS) isolates were collected from December 2016 to September 2018. The VRE isolates were selected for WGS based upon positive identification with the following criteria: >3 days of hospital stay after admission and/or any procedure or prior inpatient stay within 30 days of isolate collection date. The 267 VRE isolates were cultured from 7 different on-patient sites (e.g., rectal swab), 47 geographic locations (e.g., PUH-10N), and were restricted to sequencing type (ST) ST-736 and ST-1471 – the two largest ST groups collected.

2.1.2 Data Cleaning and Pre-processing

Single Nucleotide Polymorphisms (SNPs) are a common source of variation within bacterial populations and can be used to chart the evolutionary history of variants (Dong et al. Gut Pathog, 2017). As such, examining dissimilarity of SNPs is useful for tracking the history of a bacterial outbreak and identifying the spatial epidemiology of independent strains. The core genome of a bacterial species is a group of highly conserved shared genes (Segerman Front. Cell. Infect. Microbiol, 2012). The goal of this study was to perform comparative analysis of the SNPs within the core genome of the WGS isolates to generate a phylogenetic history which would then be used to infer the infection transmission route between patients among the isolates. To perform these analyses, we first identified the structural functions within the assemblies (Stein et al., 2001) using annotation software Prokka v. 1.14.5 (Seeman T. Bioinformatics, 2014) to annotate and identify the core genomes for each isolate. After annotation, we next performed an alignment of each of the core genome sequences to identify the SNPs. However, during alignment there can be gaps (indels) within the sequences which are not informative for comparative analysis and could lead to false positives during SNP identification (Olson et al., 2015). As such we 'masked' the indels in order to improve the quality of SNP calls (Yun and Yun., 2014). Using Roary v. 3.13 (Page et al., 2015) we performed a fast core gene multiple sequence alignment across all annotated assemblies per ST with the MAFFT algorithm option (Katoh, 2002). Then using SNP-sites v. 2.5.1 (Page et al., 2016) we extracted the monomorphic sites, excluding any indels, followed by an extraction of the core genome SNP sequences into a multi-FASTA format. Subsequent data wrangling was handled using base R v.4.0.5 'Shake and Throw' within RStudio v.1.4.1717 'Juliet Rose' and packages as described in Table 1.

Table 1 R Data Wrangling Software

Package	Version
Readxl	1.3.1
Stringr	1.4.0
Readr	1.4.0
Dplyr	1.0.6
Data.table	1.14.0

2.2 Hierarchical Clustering

Broadly, the objective of hierarchical clustering is to find sub-groups among objects and categorize those which are most similar, yet different enough from the remaining objects, into the same cluster. Agglomerative ("bottom-up") clustering is the most common form of hierarchical clustering (Lance and Williams, 1967). The algorithm requires a dissimilarity metric in which we measure and define the distance between the *n* elements within the data (Murtagh, 2011; ISLR 2013). In turn the relationship between these elements can be best visualized in a tree-like figure called a dendrogram, which can illustrate the evolutionary relationship between organisms. As such, hierarchical clustering can be used to study the ancestry/origin of antibiotic resistant VRE cultures in a hospital environment, using maximum SNP count difference (i.e., the number of differences between two isolates) as the dissimilarity metric.

We begin with *n* elements, each within its own cluster ("singleton"), and in a series of successive steps the two closest pairwise elements are clustered together so that at the end of each

step there are $n - 1$ elements until there remains only one cluster. The theory of agglomerative hierarchical clustering is as follows:

$$d(C_i \cup C_j, C_k) = \alpha_i d(C_i, C_k) + \alpha_j d(C_j, C_k) + \beta d(C_i, C_j) + \gamma |d(C_i, C_k) - d(C_j, C_k)| +$$

$$\sigma_i h(C_i) + \sigma_j h(C_j) + \epsilon h(C_k) \quad (2.1)$$

Where the dissimilarity, $d(\cdot)$, between a newly-amalgamated class $C_i \cup C_j$ and another class C_k is defined by the above equation. For the above, $h(C_i)$ is the height in the dendrogram of class C_i and $\Theta \equiv (\alpha_i, \alpha_j, \beta, \gamma, \delta_i, \delta_j, \varepsilon)$ is a set of parameters who values specify the linkage methods (described below). This dissimilarity function is calculated iteratively for pairs of classes, beginning with singleton clusters until only one cluster remains.

The R packages Dendextend v.1.15.1 (Tal Galili, 2015), Cluster v. 2.1.2 (Maechler, M., 2021) and Corrplot v.0.88 (Taiyun Wei, 2021) were used for conducting hierarchical clustering and visualization.

2.2.1 Linkages

As elements are grouped together into clusters it is necessary to define a function to determine the dissimilarity between two clusters. This function is known as a "linkage" and they can drastically affect the interpretation of the relationship between elements of the dendrogram (ISLR, 2013).

Using equation 2.1 without the terms $\{\delta_i, \delta_j, \varepsilon\}$ we can use a general agglomerative algorithm with varying parameter values which define a unique linkage strategy. For each of the below linkages, w_i is the weight associated with class C_i (usually the number of items contained within the class) and $w_+ \equiv w_i + w_j + w_k$ is the sum of the weights across classes (Lance and

Williams, 1967; Gordon, 1987). Then with each iteration of the algorithm the dissimilarity measures will increase monotonically provided that:

$$\left(\boldsymbol{\alpha_i} + \boldsymbol{\alpha_j} + \boldsymbol{\beta} \geq \mathbf{1}\right); \gamma = 0 \quad (2.2)$$

The choice of $\{\alpha_i, \alpha_j, \beta\}$ defines the linkage method. There are several commonly used weighting schemes. The ones considered for this thesis – Single, Unweighted Paired Group Method Arithmetic mean (UPGMA), and Ward's Method – are given below.

2.2.1.1 Single Linkage

The single linkage (also known as nearest neighbor) method is relatively straightforward; the distance between groups is defined as the distance between the two closest elements in each respective group (Lance and Williams, 1966).

From equation 2.1 let $\alpha_i = \alpha_j = 0.5$; $\beta = 0$ and $\gamma = -0.5$.

In subsequent iterations, as groups grow, they will continue to move closer to some elements and further from others – as such single linkage is an effective "space-contracting" method.

Conceptually, this method could be viewed as grouping elements as part of a chain, as the two most dissimilar members are categorized together because they are even more dissimilar to two other members. Hence only linking "nearest neighbors".

2.2.1.2 UPGMA Linkage

The unweighted paired group method arithmetic mean (UPGMA) linkage, will be referred to as "average" linkage for the remainder of this thesis, is commonly used in phylogenetics between microbial isolates within a species as it was designed specifically for taxonomy of organisms (Sokal and Michener, 1958).

From equation 2.1 let $\alpha_i = w_i / (w_i + w_j)$, $\alpha_j = w_j / (w_i + w_j)$ and $\beta = \gamma = 0$.

Hence, per equation 2.2, the resulting dissimilarity measures will always be monotonically increasing. Conceptually this method groups elements together into *class* or other collective labels in which groups of various shapes or border outlines can be drawn.

2.2.1.3 Ward's Method Linkage

Ward's method was developed for creating mutually exclusive subsets while minimizing variation from the mean of the subset. To achieve this, in successive steps, the error sum of squares is minimized when categorizing objects into clusters (Ward, 1963). In each iteration, the pair of objects having the smallest error sum of squares are clustered together; this process continues until only one cluster remains. We apply this linkage method to the SNP dissimilarity matrix with the intention to combine isolates with similar SNP dissimilarity into the same cluster.

From equation 2.1 let $\alpha_i = (w_i + w_k) / w_+$; $\alpha_j = (w_j + w_k) / w_+$; $\beta = -w_k / w_+$ *and* $\gamma = 0$.

When applying these weights for each class $C_{(.)}$ the parameters for equation 2.1 are calculated accordingly.

Conceptually, this method can be viewed as grouping elements into a dense *type* which differs from the *class* of average linkage through having a sort of "cloud" group with a heavy center where other points can be scattered freely but are few outside of this center.

2.2.2 Cophenetic Distance

The cophenetic distance is the dissimilarity at which two objects may be combined into a single group. On a dendrogram, this could be considered the height on the tree in which to perform a 'cut' when lumping two branches together (Sokal and Rohlf., 1962). Prior literature which used

hierarchical clustering methods for bacterial disease surveillance used a range of SNP cut-offs to define isolates within the same culture as such we selected a cophenetic distance of 15 as our cut-off (Sundermann, 2019). Any singleton isolate clusters were subsequently removed from further analysis.

2.3 Bayesian Phylogenetics

Phylogenetics originated as means to elucidate the evolutionary history and relationship of organisms (Hall, 2006). A natural implementation of evolutionary theory with the assumption of a common ancestor, molecular epidemiologists can utilize phylogenetic methods to form phylogenetic trees (phylogenies) of microorganisms and infer ancestry across isolates within closely-related or the same species. Bayesian phylogenetics is a model-based method for constructing a phylogeny through inference from Bayesian statistics.

Bayesian inference features probability distributions to describe the uncertainty of unknowns. Central to Bayesian inference is Bayes' theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2.3}$$

where A and B are different events and $P(B) \neq 0$.

Additionally in Bayes' theorem there are additional labels for the terms of the equation (Table 2).

Table 2 Labels From Bayes's Theorem

Term	Definition
P(A|B) – Conditonal Probability	The probability of event A occurring given that B is true. Alternatively called the **posterior probability** of A given B.
P(B|A) – Conditional Probability	The probability of event B occurring given that A is true. Alternatively called the **likelihood** of A given a fixed B.
P(A) & P(B) – Marginal Probability	The probabilities of observing A and B respectively without any given conditions. Alternatively known as the **prior probability.**

The use of Bayesian inference in phylogenetics can be explained as follows. Let *D* be the observed data and Θ the unknown parameter. After assigning a distribution to the prior $f(\Theta)$, using pre-existing knowledge about Θ, we can then use Eq 2.4 to calculate the distribution of the posterior of Θ:

$$f(\Theta|D) = \frac{1}{z} f(\Theta) f(D|\Theta) \tag{2.4}$$

Where the probability of the data given $f(D|\Theta)$ is the likelihood. The normalizing constant $z = \int f(\Theta) f(D|\Theta)$ means that $f(\Theta|D)$ must integrate to 1 as a proper statistical distribution. Thus Eq 2.4 indicates the posterior is proportional to the prior multiplied by the likelihood. Hence from the prior and likelihood one can calculate the probability of Θ.

Bayesian phylogenetic methods have grown in popularity since their original debut in the late 20th century. Currently there is a wide diversity of applications for phylogenetic analysis such as in comparative linguistics (Bouckaert, 2012), analyzing species diversification (Nascimento, 2017), and tracing the geographic spread and evolutionary history of influenza A (H1N1) virus during the 2009 pandemic (Smith, 2009). This popularity can be credited to the development of high-performance computing resources in tandem with innovative models which enable researchers to conduct analysis on complex data (Nascimento, 2017). Bayesian phylogenetics require the use of genetic data such as amino acid or DNA sequence alignments as input, a nucleotide substitution model, and selection of informative parameters (priors) within the model.

After selection of the data, substitution model, and priors, the key next step is running a Markov chain Monte Carlo (MCMC) algorithm. The MCMC is of great importance as it allows faster computation of the posterior. To explain the value of MCMC we must first acknowledge the high dimensionality for calculating z in eq. 2.4. Monte Carlo methods must sample from a high dimensional probability distribution, which are very difficult to analyze and requires large amounts of computation. However, estimating the expected values (sampling) from these distributions can be done using a simulation technique where, if it is difficult to sample directly from the probability density function $p(\chi)$, or if $p(\chi)$ is unknown, we instead sample from the distribution $q(\gamma)$ and obtain a sample of χ values as some function of the corresponding γ values. From here we begin to use Markov chain methods to simulate a sequence of samples underneath the conditional distribution $\chi = g(\gamma)$ given $h(\gamma) = h_0$. Computations using ratios of the form $p(\chi') / p(\chi)$, where χ' and χ are sample points, allow us to ignore the normalizing constant z in eq. 2.4 and thus simplifies computation.

Rather than solving such complex integrals directly, MCMC allows sampling from a distribution while ignoring the normalizing constant *z*. Instead, a sequence of posterior samples forms a Markov chain simulation in which either the posterior mean, standard deviation or entire distribution may be estimated from a correlating sample underneath a conditional distribution (Hastings, 1970). After a set number of iterations, the chain becomes long enough to cease sampling and the posterior estimate is accepted provided an acceptable effective sample size (ESS) coincides with the parameter of interest. For constructing the MCMC chain we use BEAUti v.2.6.5 to construct the BEAST-subject XML file from each ST core genome alignment. The MCMC is run through the BEAST 2 v. 2.6.4 (Bouckaert, 2019) software. BEAST 2 is an open-source platform for Bayesian phylogenetic analysis with a package management system allowing independent researcher developed models to be used for inference (Bouckaert, 2014). After completion of each model run the log file is analyzed using the Tracer v.1.7.2 application (Rambaut A., 2018). This application allows us to diagnose and summarize each MCMC chain by providing estimated values of the posterior such as the sample mean, standard deviation, highest posterior density interval and the ESS.

The ESS can be considered the number of independent draws from the posterior distribution and it is commonly recommended to have a minimum ESS of 200 for any posterior estimate to be accepted.

2.3.1 bModel Test

bModelTest is a BEAST 2 package for co-estimating the nucleotide site substitution model of a phylogeny (i.e. the mutation rate of each nucleotide and whether all nucleotide changes occur at the same rate or at differing rates in a set of sequences). *bModelTest* indicates the most likely

time-reversible nucleotide substitution model to use, taking into account sequence regions which are constant (invariable sites) and the remaining regions which have heterogenous rates of change best described underneath a discrete gamma probability distribution (Bouckaert, 2017) Specifically, it describes the transition and transversion rates which are denoted as r_{ac}, r_{ag}, r_{at}, r_{cg}, r_{ct} and r_{gt} which are indicated by a six-digit model number *M* in which each digit refers to one of up to six rates in the same order as each rate. For example, model number 123456 indicates all nucleotide substitution rates are independent and unique, model number 111111 indicates all nucleotide substitution rates are equal, and model number 112345 indicates r_{ac} and r_{ag} are equal with all other rates being unique. The graphical output is a series of nested models, where arrows point towards a model that is a subset of another. The area of each circle is proportional to model with the highest posterior support while the color indicates whether the model is 95% credible (blue) or not (red). The 95% credibility in Bayesian Statistics is known as highest posterior density (HPD) is analogous to the 95% confidence interval (CI) of frequentist statistics. We used default prior selection with the exception of parameters depicted in Table 3. This protocol is an adaptation of the Substitution Model Averaging tutorial hosted on the *Taming the BEAST* community teaching resource (Joëlle Barido-Sottani, 2018)

Table 3 BModelTest BEAUti Settings

BEAUti Tab Name	Parameter Name	Setting
MCMC	Chain Length	10,000,000; 10% burn-in
Site Model	Mutation Rate	Estimate
Site Model	Nucleotide Substitution Model Set	transistionTransversionSplit
Partitions	Split	1+2+3; link site models; link clock models; link trees
Clock Model	Strict Clock	Clock.rate = 1.0
Priors	Tree.t	Coalescent Constant Population

The results from the *bModelTest* showed strongest support for the 121321 nucleotide substitution model for both ST groups, by both metrics of having the highest posterior support and passing the 95% credible interval (Figure 1 and Figure 2). Furthermore, for both ST groups, no invariable sites were identified, the estimated gamma rate heterogeneity was 0, and nucleotide base frequencies were estimated to be even. Nucleotide conversion rates did differ for each ST group (Table 4).

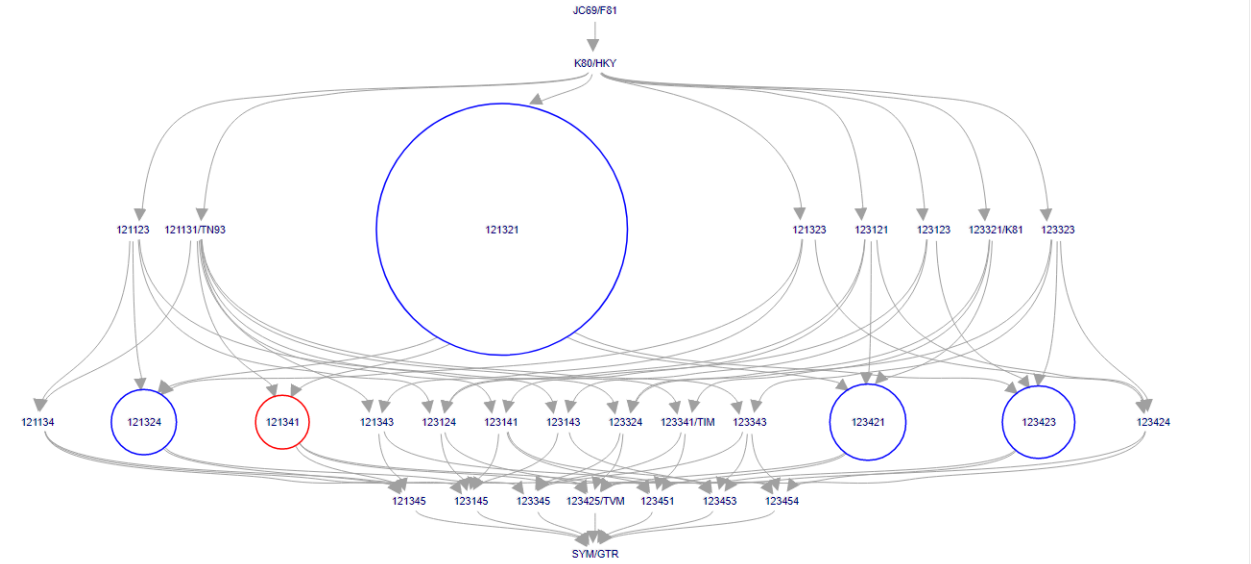


Figure 1 ST1471 BModelTest Indicator

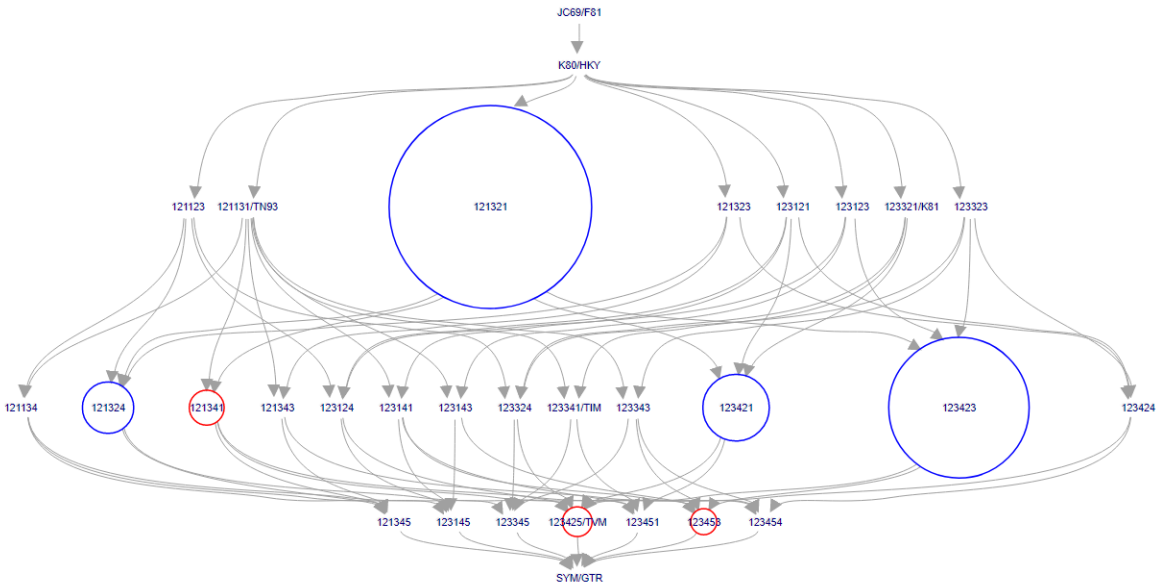


Figure 2 ST736 BModelTest Indicator

Therefore, we linked the respective nucleotides in our prior parameters in the subsequent MASCOT models to the estimated nucleotide conversion rates (Table 4).

Table 4 BModelTest Estimated Nucleotide Conversion Rates

Nucleotide Conversion	ST1471	ST736
A > C	0.502	0.472
A > G	2.142	2.236
A > T	0.505	0.452
C > G	0.208	0.151
C > T	2.142	2.234
G > T	0.502	0.454

2.3.2 MASCOT

MASCOT is a BEAST2 package which uses structured coalescence approximation in order to infer the ancestral migration history of phylogenies with at least 3 or 4 different states (Muller, 2017). The structured coalescence method itself is derived from coalescent theory as applied to population genetics. Coalescent theory is a model of common ancestry through successive inheritance of alleles, assuming each allele variant is equally likely to be passed down (Kingman, 1982). By following ancestral history backwards in time, alleles across the population should merge into a single individual through a sequence of coalescence events. Using structured coalescence, the population structure can be inferred from multiple phylogenies under the assumption that a MCMC is treated as a continuous temporal element and migration of the population is dependent on the phylogenies. Since our MCMC chain uses sample collection dates as part of calculating height between phylogeny nodes, rooted at the earliest timepoint, then we have a temporal element within the model. Both ancestral lineage and state may be inferred from estimates of migration rates in the posterior. MASCOT incorporates information from the entire

phylogeny in order to calculate the probability of being in a particular state using a forwards/backwards approach to estimate time between the phylogeny tree root and the coalescent event. Our protocol closely follows the MASCOT v.2.1.2 tutorial hosted on *Taming the BEAST* community teaching resource (Joëlle Barido-Sottani, 2018). Using each geographic location as a state we can infer the ancestry and migration rates for each isolate. For each ST group we ran MCMC chains at lengths of 500,000, 10% burn-in, in replicates of 3 using default prior selection with the exception of parameters settings depicted in Table 5. Isolate collection dates were used for the tip dates underneath the "Tip Dates" tab and hospital unit were selected as the location (Fig 3and 4).

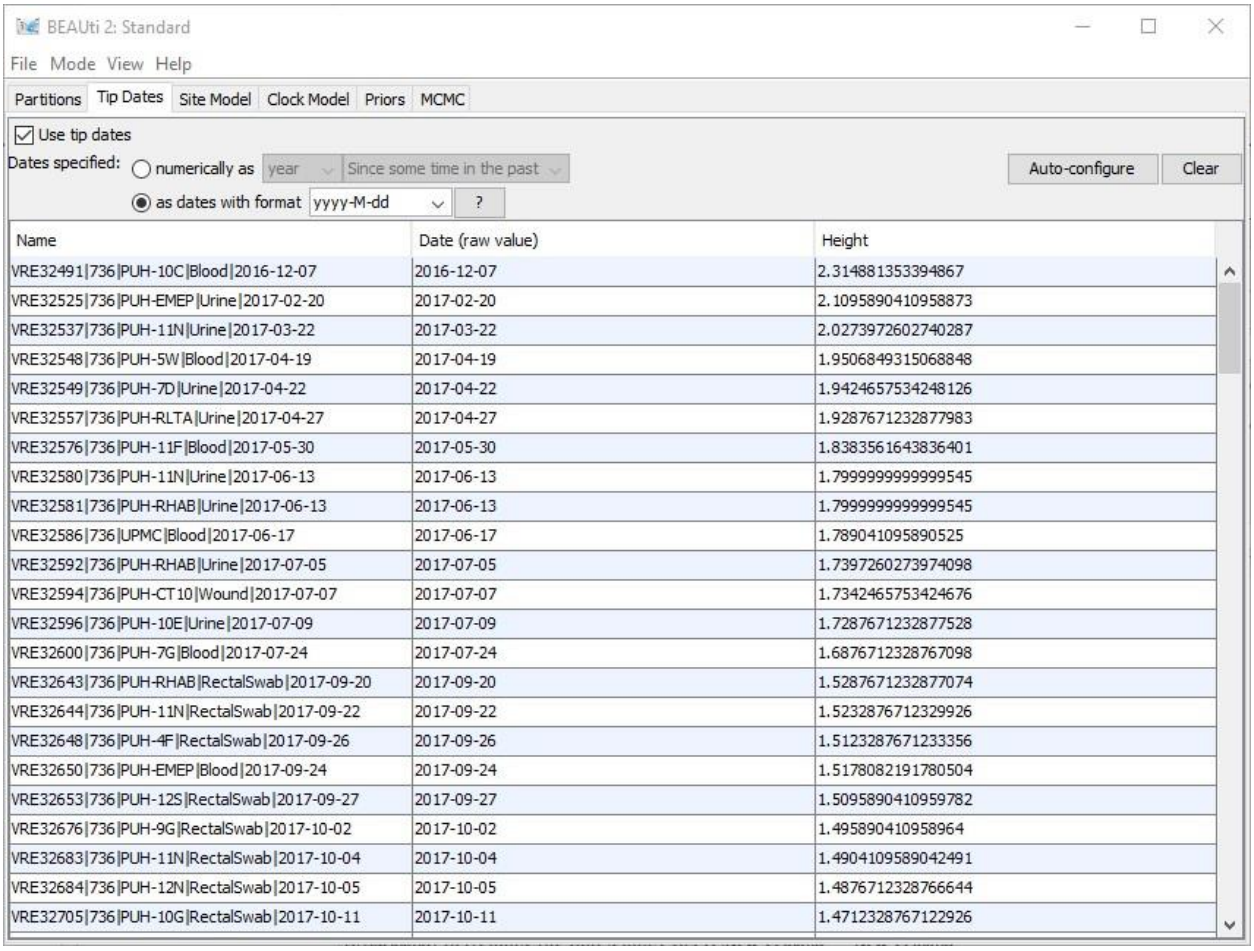


Figure 3 BEAUti Tip Dates

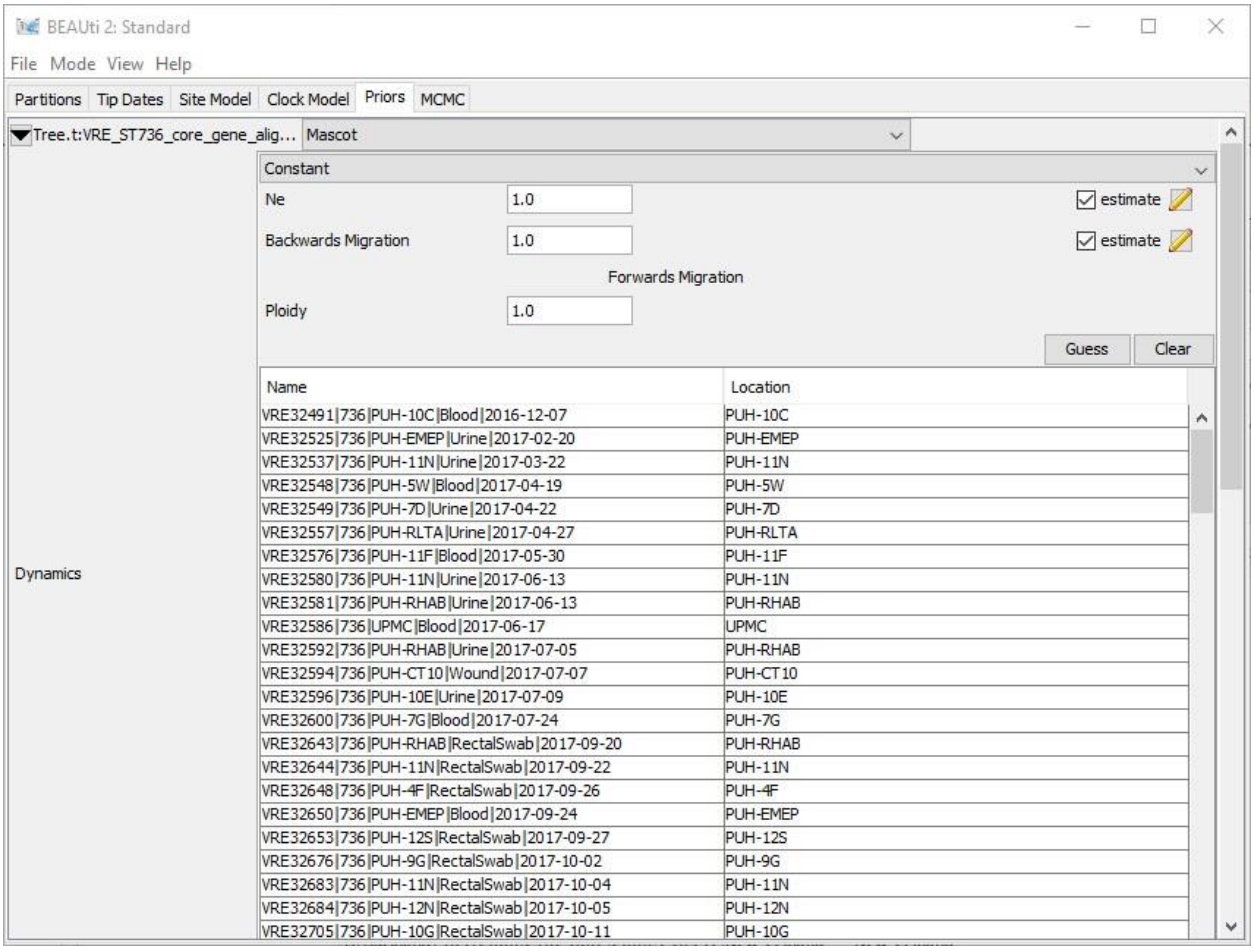


Figure 4 BEAUti Location

Table 5 Selected MASCOT BEAUti Settings

BEAUti Tab Name	Parameter Name	Settings
Site Model	----	Gamma Site Model;
Site Model	Substitution Model	GTR; link rates and input values as indicated from BModelTest; Set Frequencies to "All Equal"
Priors	clockRate.c	Log Normal; M = $4.9E^{-5}$; S = 1.25; Mean In Real Space
Priors	migrationConstant.t	Exponential; M = 1
Priors	NeConstant.t	Log Normal; M = 0; S = 1

2.3.3 Maximum Clade Credibility Tree

Upon completion of all MASCOT runs, ST group log output files were aggregated together using LogCombiner v.2.6.5 and then imported into TreeAnnotator v.2.6.4 with 10% burn-in and mean node heights to generate two Maximum Clade Credibility (MCC) Trees. The MCC Tree is a summary of all of the phylogenies generated from every iteration (step) in a Bayesian phylogenetic inference. As posterior clade probabilities are additive, this tree serves as a point estimate of the total probability across all phylogenies.

Each MCC Tree is then exported in the Newick format then imported into R. Using the Phylogram package (Wilkinson SP, Davy SK, 2018) the trees are converted into dendrograms which are then processed in the same manner as the hierarchical cluster generated dendrograms with a cophenetic height cutoff at 15.

2.4 Assessment of Methods

To assess the comparative performance of hierarchical clustering and Bayesian phylogenetic methods in identifying epidemiologically plausible transmission routes, we performed each type of method and compared the results to a Ground Truth (GT) dataset, obtained by an inference model of direct and indirect transmission patterns which incorporates electronic health data and machine learning to score possible transmission routes for each isolate.

2.4.1 Transmission Route Identification

The transmission route for each isolate was identified using a machine-learning model based on SNP dissimilarity and electronic health data. This unique algorithm was developed by researchers in Carnegie Mellon University and created a scoring system to rank the log likelihood of a particular geographical location being the source of transmission (Miller JK, Chen J., 2019). This algorithm was used to form the GT dataset, which consists of the most likely transmission source for each isolate. The geographic location with the highest rank for each isolate was used to label the determined transmission source. If there were no geographic location ranked then the given label is “NA”.

2.4.1.1 DITO Model

The direct/indirect transmission outbreak model (DITO) is a statistical inference model designed to detect and characterize the root-cause of bacterial outbreaks in the hospital setting (Miller JK, Chen J., 2019). Its intended purpose is to be a highly flexible model which combines WGS-based SNP distance metrics with EHR to both explain outbreak root-causes and recommend

bacterial isolates for WGS to improve performance using machine-learning. For the purpose of evaluating the clusters we generated through hierarchical and Bayesian methods, we utilized DITOs' capabilities for detecting and characterizing outbreaks. When conducting its inference, DITO utilizes the maximum likelihood approach in order to generate a set of samples and rank transmission routes. Using the analysis output we labeled clusters into hospital units based on rank and settled tied ranks with differing locations by selecting the greater log-likelihood score.

2.4.2 Rand Index

The Rand Index is a statistical assessment criterion developed specifically for measuring similarity between data clusters. It involves calculating the number of ways a total number of N objects can be paired. This is a useful metric for comparing clustering methods to assess how similar the classification of all data points are across two different methods. However, there is always a possibility that an object has been classified the same by random chance which would falsely inflate the Rand Index. A modified version of the original, the adjusted rand index (ARI) adjusts for the chance grouping of elements (Rand, 1971) using a method that is analogous to the Expected Count for contingency tables to normalize the calculations and reduce the effect of chance (Hubert, 1985). Using the R package Mclust v. 5.4.7 (Scrucca L., 2016), we assessed the accuracy of the identified clusters from each method per ST by comparing the resulting clusters to those from the GT dataset, with a score range from 0.0 to 1.0 in which a 1.0 indicates 100% similarity.

3.0 Results

3.1 Hierarchical Clustering

3.1.1 Comparison of Linkages

In comparison of single vs. average linkages, we observed very similar clustering of isolates for both ST1471 and ST736 (Fig. 5 and 6) across both dendrograms and as measured through cophenetic correlation. However, when comparing either of the aforementioned linkage methods to Ward's, we saw a much greater difference between results, with more nodes pairing isolates into unique clusters with Ward's method. (Fig. 7 - 10). These relationships between isolate pairings and linkage are readily visualized using Tanglegrams (each side depicts a dendrogram created from the indicated clustering method - unique nodes are indicated by dash lines and shared sub-trees are connected through colored bars). In Figures 5 and 6 the clusters between single and average linkage are largely shared with extremely few differences. Figures 7 - 10 however shows that the clustering using Ward's Method linkage has much more of a difference compared to either Single or Average methods, as there are many unique nodes. Figure 11 and 12 confirms what we observed in Figures 5 – 10 in that clusters formed from ward's method are drastically different from either Single or Average through displaying the cophenetic correlation scores, with ST1471 showing a larger margin of difference than ST736.

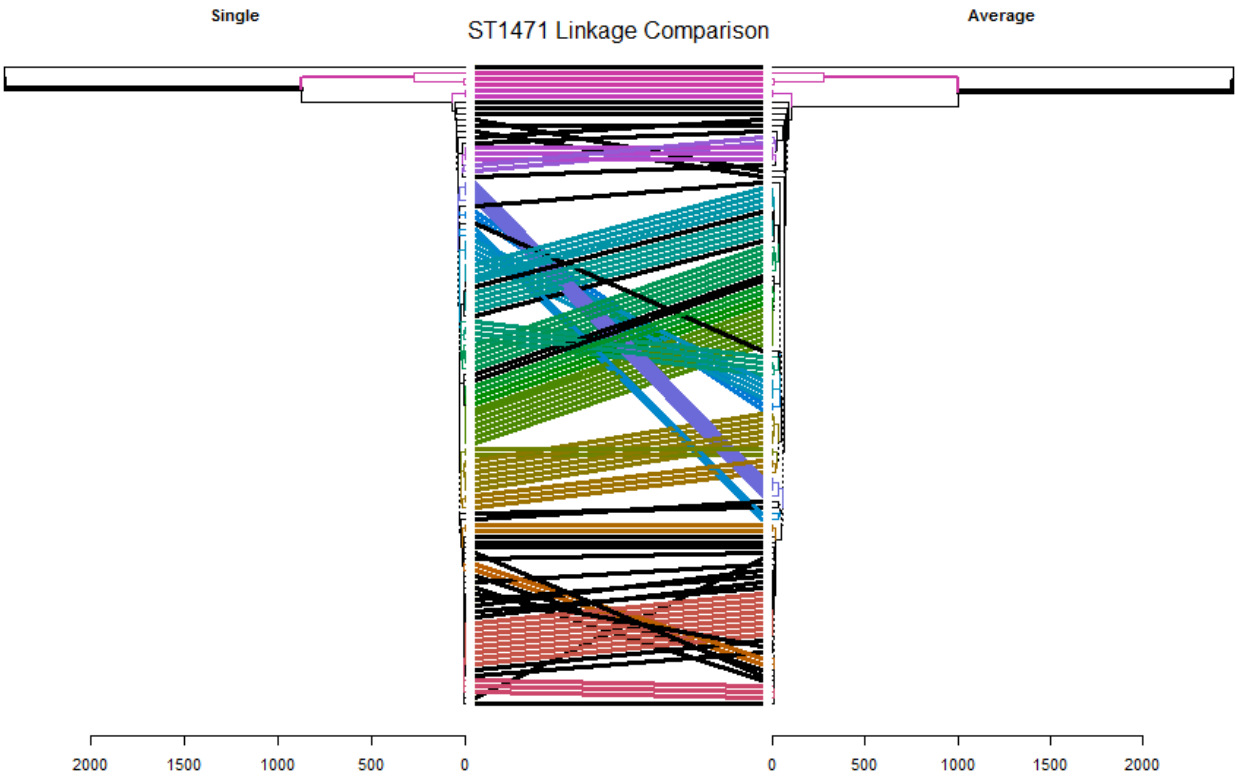


Figure 5 ST1471 Single vs Average Tanglegram

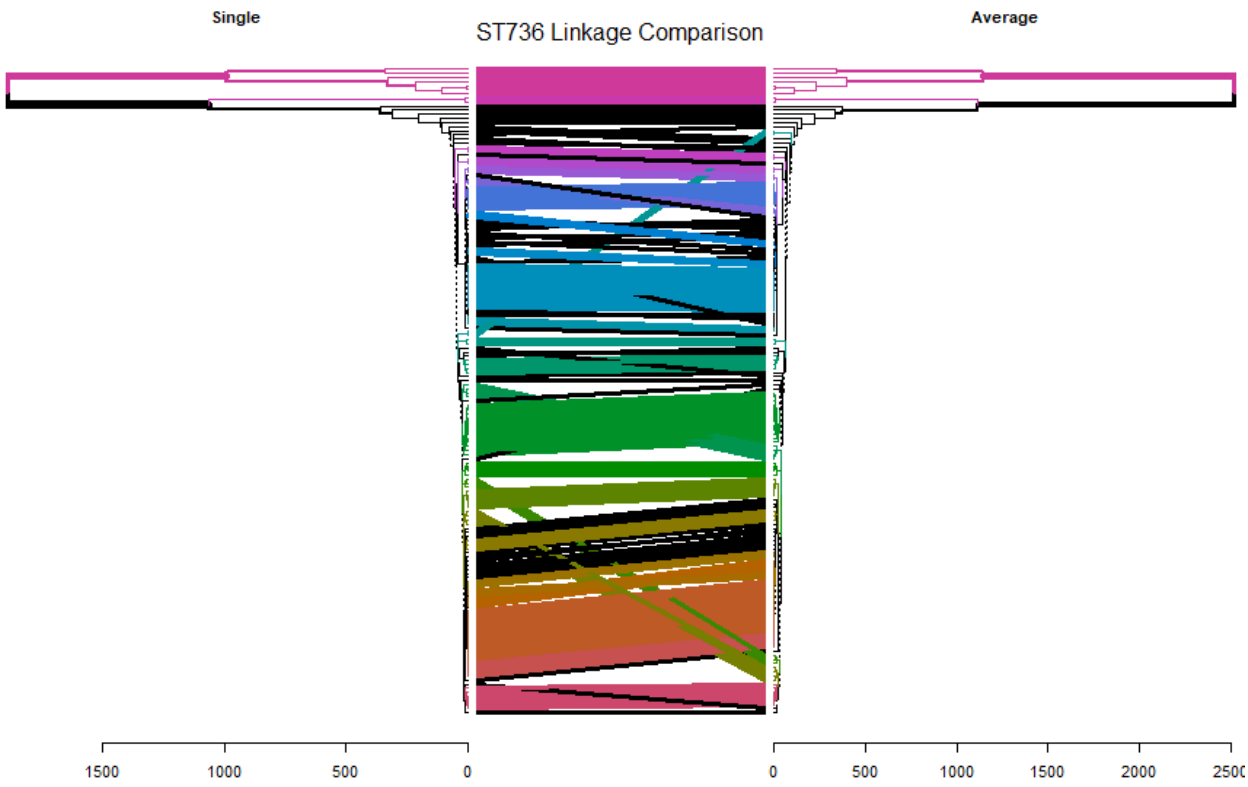


Figure 6 ST736 Single vs Average Tanglegram

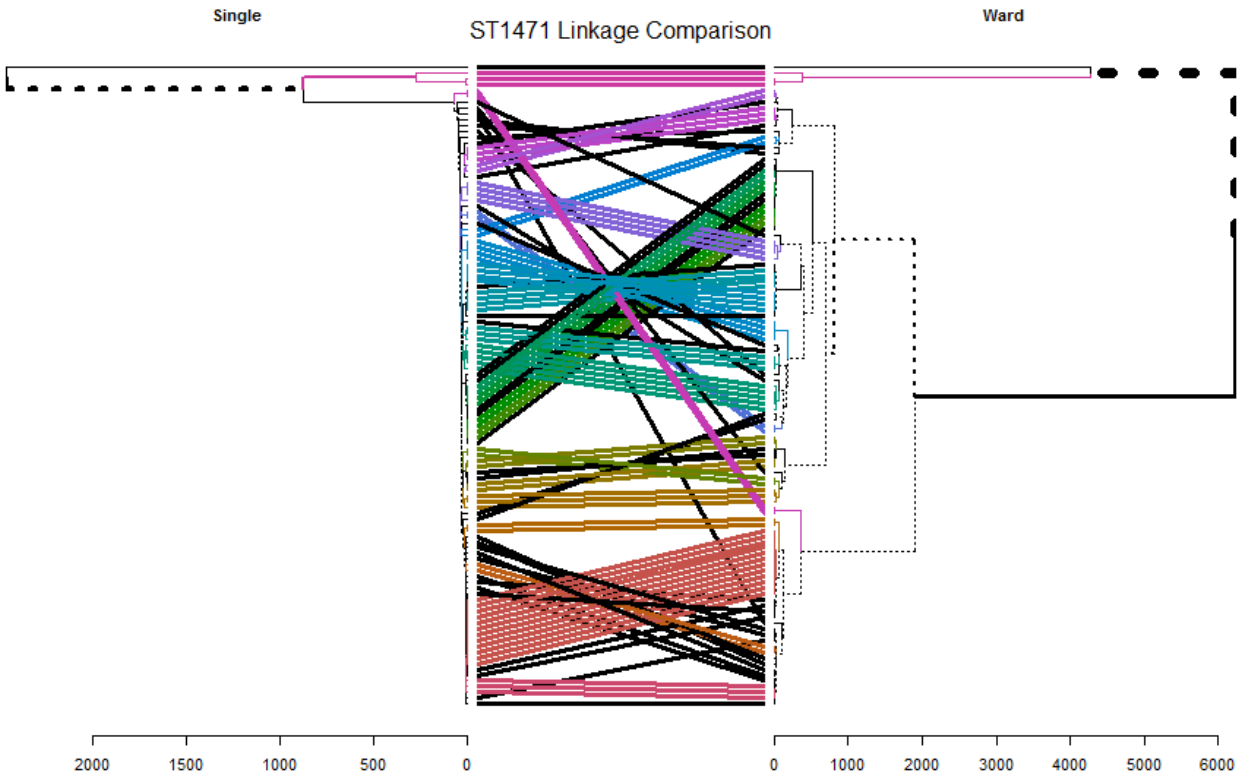


Figure 7 ST1471 Single vs Ward Tanglegram

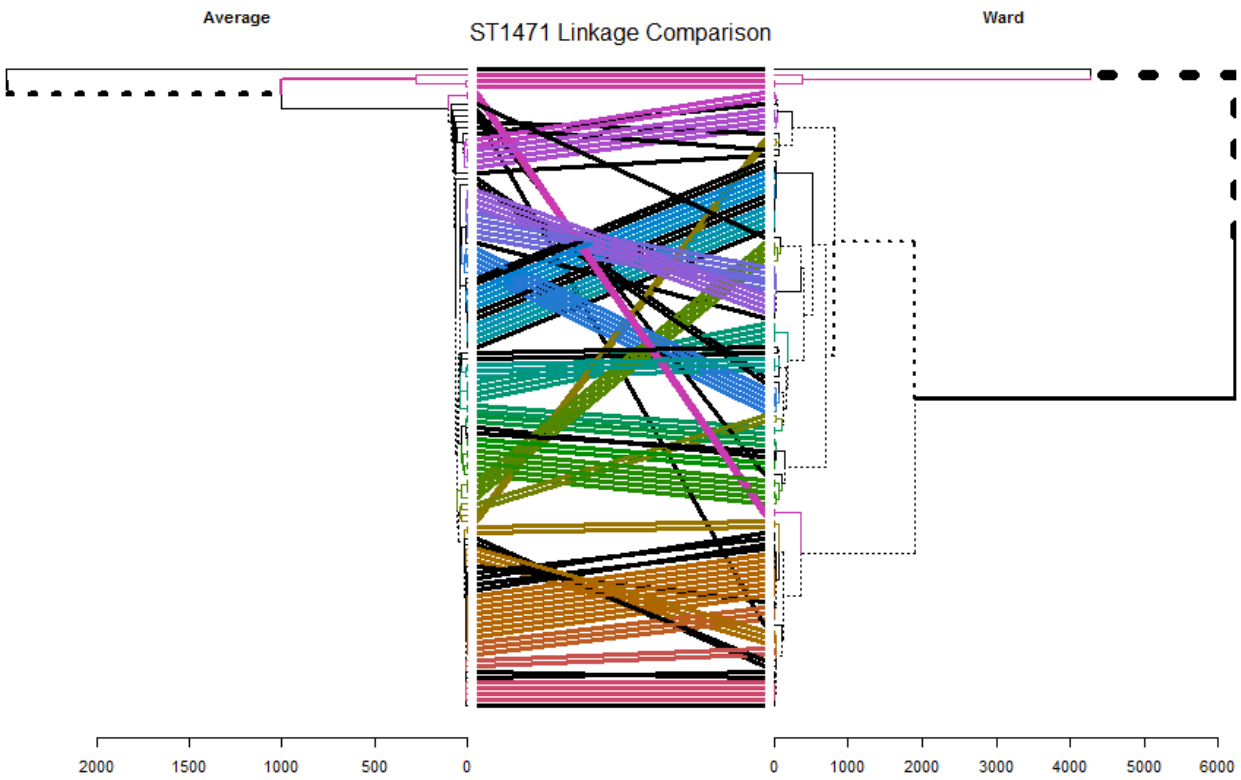


Figure 8 ST1471 Average vs Ward Tanglegram

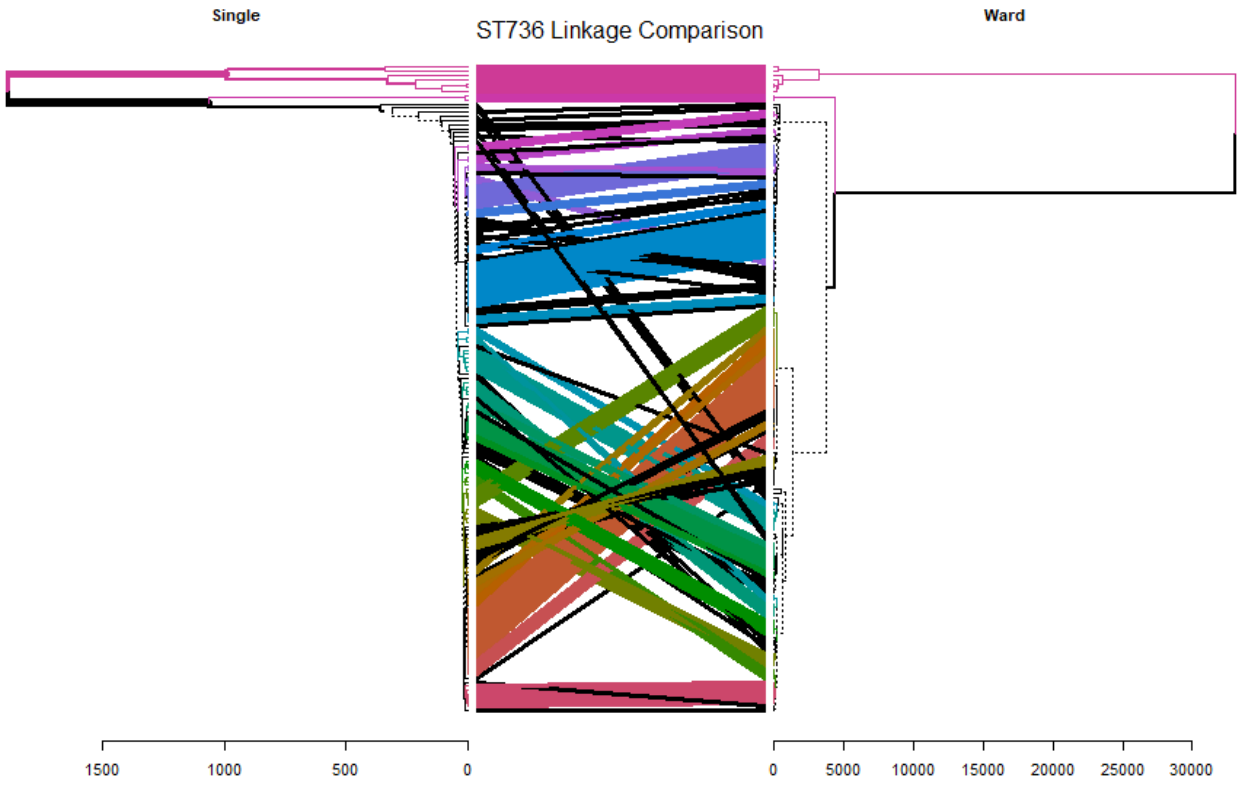


Figure 9 ST736 Single vs Ward Tanglegram

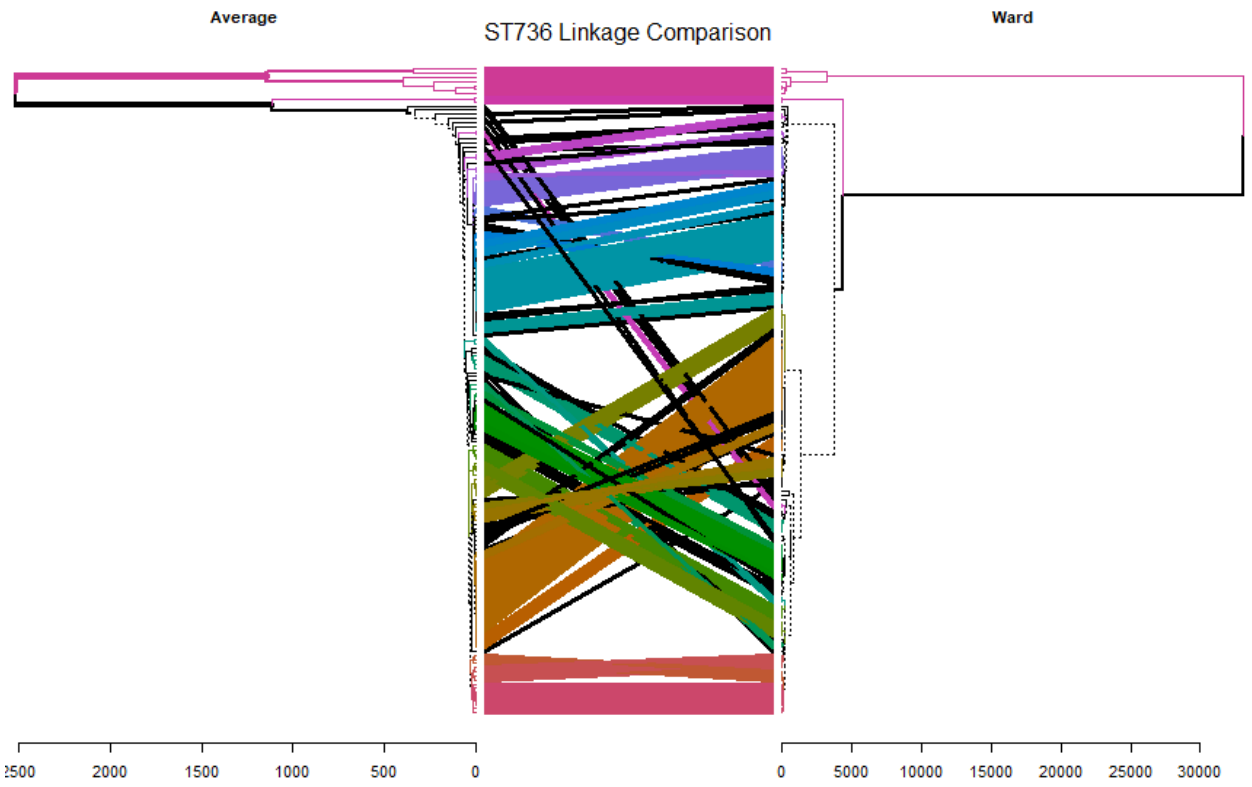


Figure 10 ST736 Average vs Ward Tanglegram

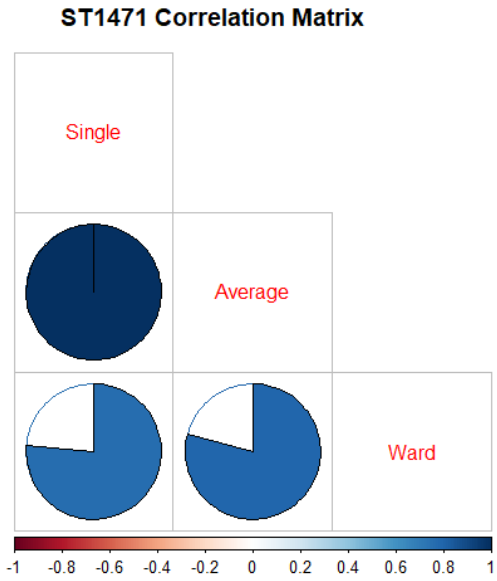


Figure 11 ST1471 Correlation Plot

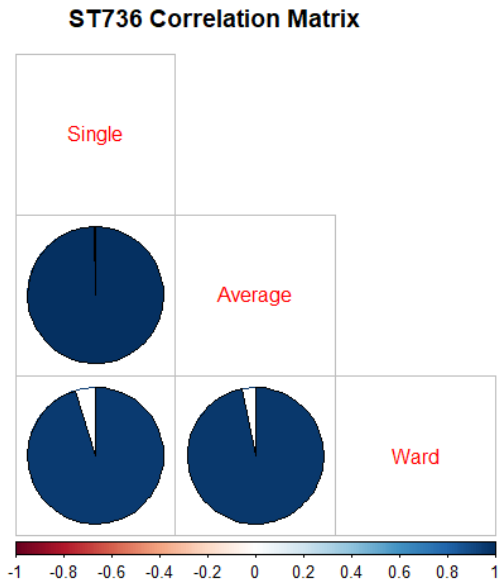


Figure 12 ST736 Correlation Plot

3.2 Bayesian Phylodynamic Models

The posterior estimated values from the combined log files for each ST type had ESS values far below the acceptable threshold, with most being < 100. This brought a lot of uncertainty as to the reliability of the tree generated from each ST type (Fig 13 – 14). Diamonds indicate the isolate sample collection dates and branch age are indicated by as an integer. Several distinct clades (which each may indicate a hospital unit) are visible for each ST type. Using a height cut-off of 15, in similarity to the dendrograms generated from hierarchical clustering, 13 and 16 unique clusters were identified for ST1471 and ST736 respectively.

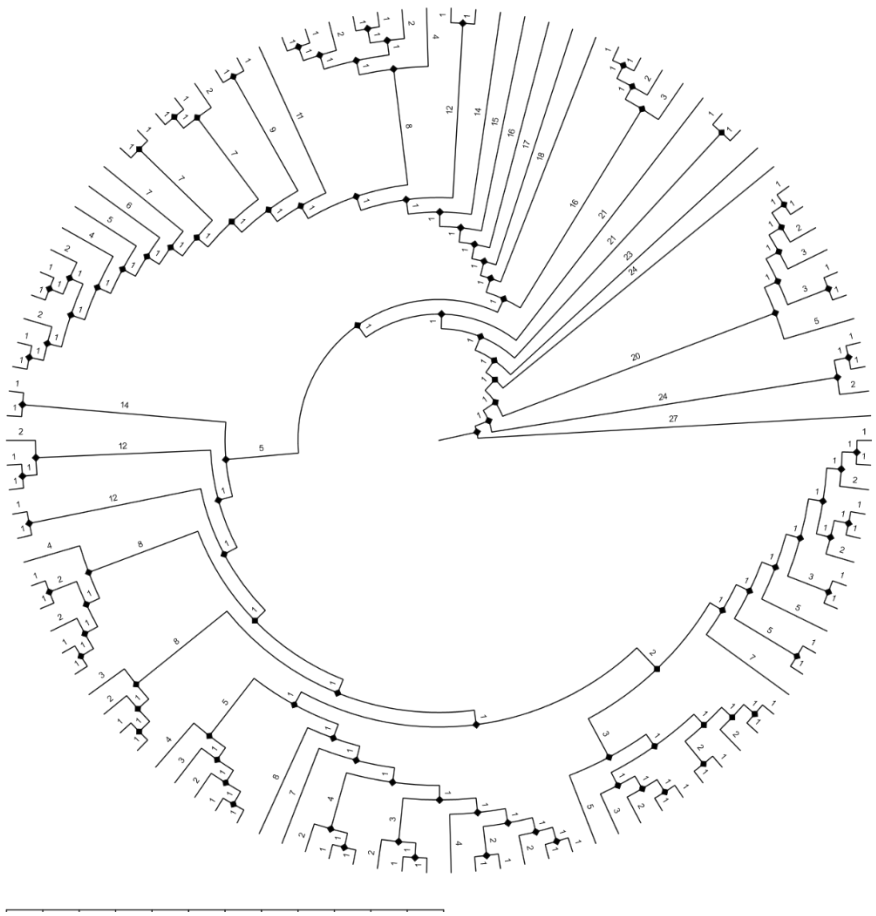


Figure 13 ST1471 Cladogram

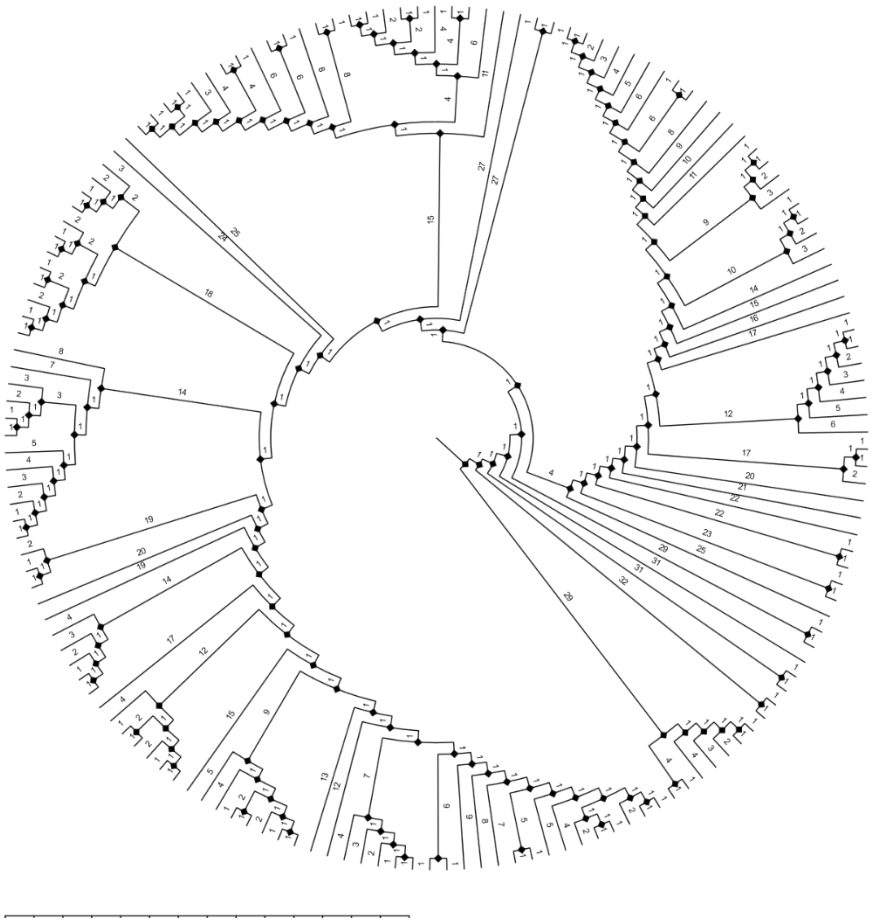


Figure 14 ST736 Cladogram

3.3 Transmission Route Classification

The clusters between Single and Average linkage were similar and prior research recommends Average linkage, so we decided to identify the most likely route of transmission only using the Average and Ward's clusters. In doing so, the Adjusted Rand Index identified the Average method as superior to both Ward's and Bayesian-derived clusters in correctly identifying the same route as the Ground Truth dataset (Table 6). Upon further investigation, we identified that the clusters formed using Ward's method were mostly subclusters of those obtained from the Average method.

Table 6 Adjusted Rand Index for All Methods

ST	Average	Ward's	Bayesian
736	0.9638	0.4011	0.5694
1471	1	0.4808	0.3018

4.0 Discussion

Hierarchical Clustering and Bayesian Phylogenetics methods have great potential for identifying infectious disease transmission routes within a population. Each method has appealing traits for many researchers but come with their own unique limitations. In this thesis, our aim was to evaluate and compare these two methods for the purpose of inferring the transmission route for VRE isolates in the hospital setting. We found that hierarchical clustering with an Average linkage had better concordance with a GT dataset than a Bayesian phylogenetic model using geotemporal data to identify disease transmission history in a hospital setting. However computational limitations affected the performance of the Bayesian phylogenetic model and a repeat assessment without these limitations would be necessary for a fair comparison.

Some of the aforementioned limitations became apparent such as a lack of computation time when running each MCMC, which lead to models with inadequate ESS values for each run. All estimated posterior values from a MCMC run should be at least 200 for them to be considered acceptable. When ESS values are too low the estimate of the posterior is considered poor. To increase the ESS for a parameter there are multiple methods such as increasing the sampling frequency or increasing the chain length in which both would increase computational time dramatically. For our analysis we ran MCMCs sampled at a frequency of every 5000 steps on a chain length of 500,000. We may have achieved the desired ESS values had we run chain lengths of at least 10 million at similar frequency. However, running such models for both ST 1471 and ST 736 in triplicate would have taken much more computational resources than available.

Another consideration is that the MASCOT model was inadequate for our analysis and that an alternative model could have performed better. One such model could be the Multi-Type Birth-

Death model (MTBD) (Kühnert, 2016). This model was designed with the intent to reconstruct the phylodynamic history of structured populations and is well-suited for charting the transmission history of a pathogen within structured populations. In this model discrete typed subpopulations (such as hospital wards) are quantified and within each subpopulation a 'type-change' process (i.e. exposed to infected) is quantified chronologically. Various parameters such as birth, death, and sampling are allowed to change overtime and these parameters are inferred as part of which differs between the subpopulations and their influence on one another. This model has been used successfully in other publications regarding other pathogens such as SARS-CoV-2 (Nabil, 2020) and multi-drug resistant tuberculosis (Pečerska, 2021). However, this model is far more computationally intensive than MASCOT due to the added complexity of type-changes.

For hierarchical clustering methods there also were alternative methods worth considering which may have improved performance. Such methods could have been the core genome distance methods we have chosen to identify genetically distinct isolates. One popular alternative used in bacterial genomics would be cgMLST (Mellmann, 2011; Neumann, 2019). This has been used with great success for real-time analysis of various outbreaks such as with *E. coli* and *E. faecalis*, through combining whole genome sequencing with high-throughput next-generation sequencing for a highly discriminatory typing schema. However a comparative assessment of genomic typing methods found that there were very high concordance between SNP and cgMLST (Henri, 2017). Hence it is possible that cgMLST would produce very similar cluster classifications and have little impact on our results.

Perhaps the largest limitation in this analysis would the metric we used for comparison across classification methods - the Adjusted Rand Index. The 100% ARI from the average method for ST1471 raised suspicion and upon investigation of the labels, we confirmed that roughly 81%

(raw proportion) of the labels truly matched. However, for the sample space of labels that were shared between the GT clusters and Average clusters there were indeed a 100% match. Suggesting the ARI calculation is blind to labels which are exclusive to a particular subset. For example, if calculating the ARI between two subsets {X = 1, 2, 4} and {Y = 1, 2, 3} the ARI will take into account only the classes 1 and 2 which then biases the ARI score. Recalculation using either an alternative R package or custom function that will not ignore labels exclusive to one method or another would be required. As such the results assessing the transmission route classification to the GT dataset remains inconclusive.

5.0 Conclusion

HAIs make it necessary to understand the transmission routes of an outbreak as part of an effective infection control strategy. Through understanding the relationship of cases during an infectious disease outbreak hospital staff can create effective interventions that would minimize cases and provide effective care for their already vulnerable patients. WGS is a key tool in understanding the evolutionary history of a pathogen and, in tandem with EHR, allows reconstruction of the phylogeography of pathogen strains. This has been applied to the hospital setting with great success and has directly led to a change of policy to prevent further cases (Sunderman, 2021).

In this thesis, we compared two methods for identifying the transmission route of an outbreak of VRE in the hospital setting, one method being a simple hierarchical clustering and the other an advanced Bayesian phylodynamic model. These methods utilized WGS of isolates from multiple patients to determine the relationship between cases. The hierarchical clustering method utilized WGS SNP data to determine how closely related cases were and categorize them. The Bayesian phylodynamic model utilized WGS alignment data in combination with spatial-temporal data to categorize cases. Both methods were assessed by comparing to the ground truth dataset using the Adjusted Rand Index for scoring similarity. The results from our comparative analyses where largely inconclusive due to a limitation on computational resources for the Bayesian phylodynamic model and a discovered error in the Adjusted Rand Index software. However, the displayed time constraint of using Bayesian phylodynamic methods when reconstructing the history of an infectious disease outbreak in the hospital setting supports our recommendation to

utilize the simpler and quicker hierarchical clustering during an active outbreak and reserving the Bayesian phylodynamic model for retrospective analysis.

Appendix A Descriptive Statistics

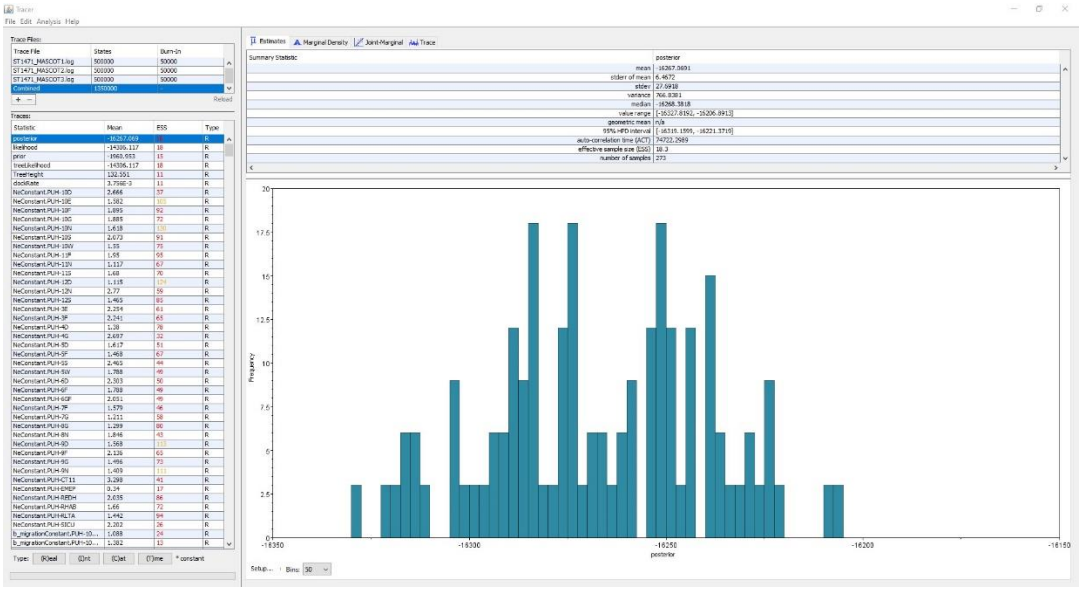
Count of VRE Positive Cultures Unit and Room			
	Geographic Location	ST1471(n)	ST736(n)
	PUH - 10N	1	2
	PUH - 11N	8	11
	PUH - EMEP	3	4
	PUH - 6GF	1	2
	PUH - 5W	1	3
	PUH - RLTA	5	4
	PUH - 12N	3	11
	PUH - 3F	3	3
	PUH - 6D	2	1
	PUH - 11F	2	2
	PUH - RHAB	3	7
	PUH - 10E	4	3
	PUH - 7G	3	5
	PUH - CT11	4	5
	PUH - 12D	2	2
	PUH - 10S	8	6
	PUH - 12S	6	16
	PUH - 5D	3	1
	PUH - 9G	1	6
	PUH - 10G	5	3
	PUH - 4D	3	0
	PUH - 8G	3	1
	PUH - 8N	1	2
	PUH - SICU	3	0
	PUH - 10F	3	4
	PUH - 10D	2	4
	PUH - 5S	1	3
	PUH - 7F	4	1
	PUH - 6F	2	3
	PUH - 3E	3	6
	PUH - 5F	1	1
	PUH - 9D	2	4
	PUH - 10W	2	3
	PUH - 4G	4	0
	PUH - 11S	5	4
	PUH - 9N	1	1
	PUH - 9F	1	3
	PUH - REDH	1	2
	PUH - 10C	0	2
	PUH - 7D	0	6
	UPMC	0	1
	PUH - CT10	0	5
	PUH - 4F	0	1
	PUH - 8W	0	2
	PUH - 8D	0	1
Sum	—	110.00	157.00
Mean	—	2.44	3.49
SD	—	1.97	3.12

Appendix Figure 1 Hospital Unit Isolate Count

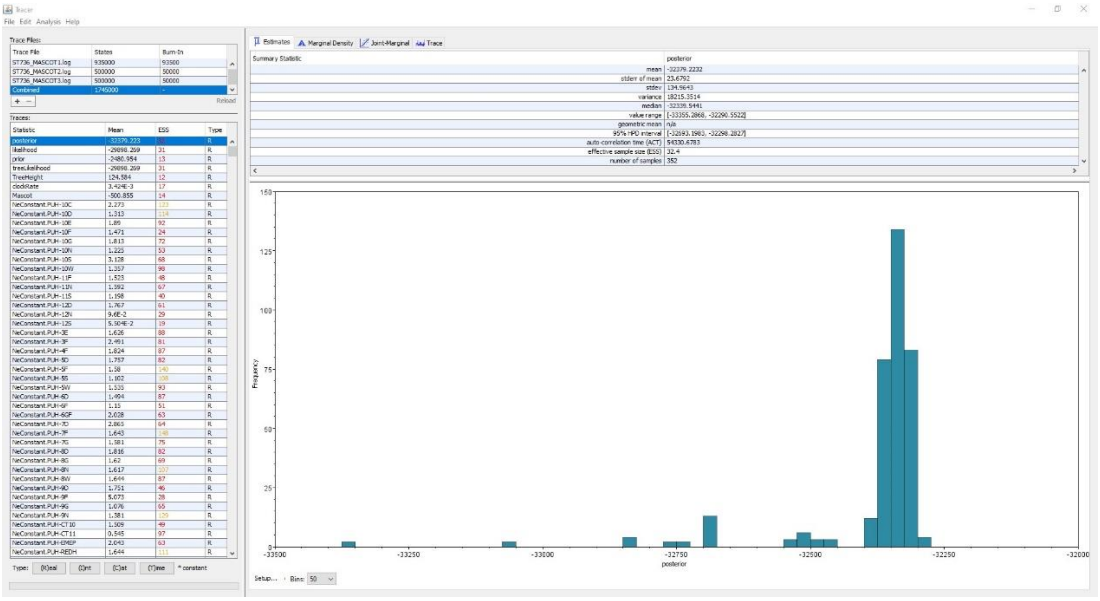
VRE Positive Isolates		
Sequencing Type		
	ST	n
	736	157
	1471	110
Sum	—	267

Appendix Figure 2 ST Type Isolate Count

Appendix B Tracer Output



Appendix Figure 3 ST1471 Combination



Appendix Figure 4 ST736 Combination

Appendix C Linux CLI

Appendix C.1 Prokka Annotation

for file in *.fasta; do tag=${file%.fasta}; prokka --prefix "$tag" --locustag "$tag" --cpus 12 --usegenus --mincontiglen 200 --genus Enterococcus --species faecium --strain "$tag" --force --addgenes --kingdom Bacteria --gcode 11 --outdir ~/VRE_ST1471/annotations/"$tag"_prokka "$file"; done

for file in *.fasta; do tag=${file%.fasta}; prokka --prefix "$tag" --locustag "$tag" --cpus 12 --usegenus --mincontiglen 200 --genus Enterococcus --species faecium --strain "$tag" --force --addgenes --kingdom Bacteria --gcode 11 --outdir ~/VRE_ST736/annotations/"$tag"_prokka "$file"; done

Example

prokka --prefix VRE32493 --locustag VRE32493 --cpus 12 --usegenus --mincontiglen 200 --genus Enterococcus --species faecium --strain VRE32493 --force --addgenes --kingdom Bacteria --gcode 11 --outdir ~/VRE_ST1471/annotations VRE32493.fasta

Appendix C.2 Roary Alignment

cp ~/VRE_ST1471/annotations/*/*.gff ~/VRE_ST1471/alignment; roary -e -n -v -p 32 *.gff

cp ~/VRE_ST736/annotations/*/*.gff ~/VRE_ST736/alignment; roary -e -n -v -p 32 *.gff

Appendix C.3 SNP-sites Masking and Core Genome SNP extraction

Mask alignments from gaps/indels
snp-sites -mcb -o ~/VRE_ST1471/alignment/VRE_ST1471_core_gene_alignment_clean.fna ~/VRE_ST1471/alignment/VRE_ST1471_core_gene_alignment.fna
snp-sites -mcb -o ~/VRE_ST736/alignment/VRE_ST736_core_gene_alignment_clean.fna ~/VRE_ST736/alignment/VRE_ST736_core_gene_alignment.fna

```
# Extract core SNPs
snp-sites -mc -o ~/VRE_ST1471/alignment/VRE_ST1471_core_gene_SNP.fna ~/VRE_ST1471/alignment/VRE_ST1471_core_gene_alignment_clean.fna
snp-sites -mc -o ~/VRE_ST736/alignment/VRE_ST736_core_gene_SNP.fna ~/VRE_ST736/alignment/VRE_ST736_core_gene_alignment_clean.fna
```

Appendix D R Code

Appendix D.1 Core Genome Alignment Sequence ID Renaming

```
# Import Sample Collection Info
VRE_db <- readxl::read_xlsx("~/School/Pitt/VRE Thesis/data_for_arvon/VREs_Sample_Collection_Info.xlsx")
VRE_db <- subset(VRE_db, select = c("SpecimenID", "ST", "Geographic Location", "Source", "CultDate"))

# Remove white space in variable entries
VRE_db$`Geographic Location` <- stringr::str_replace_all(VRE_db$`Geographic Location`, " ", "")
VRE_db$Source <- stringr::str_replace_all(VRE_db$Source, " ", "")
head(VRE_db)
## A tibble: 6 x 5
# SpecimenID ST    `Geographic Location` Source         CultDate
# <chr>      <chr> <chr>                <chr>          <dttm>
#   1 VRE32491   736   PUH-10C              Blood          2016-12-07 00:00:00
# 2 VRE32493   1471  PUH-10N              Tissue/Surgical 2016-12-08 00:00:00
# 3 VRE32503   1471  PUH-11N              Wound          2017-01-05 00:00:00
# 4 VRE32504   1471  PUH-EMEP             Blood          2017-01-10 00:00:00
# 5 VRE32514   1471  PUH-6F               Wound          2017-01-20 00:00:00
# 6 VRE32525   736   PUH-EMEP             Urine          2017-02-20 00:00:00

# Export .txt file listing new Sequence IDs
write.table(VRE_db, file = "~/School/Pitt/VRE Thesis/Annotation and Alignment/fasta_rename.txt", sep = "|", row.names = F, quote = F)

# Import new Sequence IDs as character vector
fasta_rename <- readr::read_file("~/School/Pitt/VRE Thesis/Annotation and Alignment/fasta_rename.txt")
fasta_rename <- unlist(strsplit(fasta_rename, "\\r"))
fasta_rename <- stringr::str_remove(fasta_rename[-1], "\\n")
fasta_rename <- fasta_rename[-length(fasta_rename)]

# Name vector of new Sequence IDs
patterns <- stringr::str_remove(fasta_rename, "\\|(.+)$")
names(fasta_rename) <- patterns
head(fasta_rename)
# VRE32491                    VRE32493                    VRE32503
```

```
# "VRE32491|736|PUH-10C|Blood|2016-12-07" "VRE32493|1471|PUH-10N|Tissue/Surgical|2016-12-08" "VRE32503|1471|PUH-11N|Wound|2017-01-05"
# VRE32504 VRE32514 VRE32525
# "VRE32504|1471|PUH-EMEP|Blood|2017-01-10" "VRE32514|1471|PUH-6F|Wound|2017-01-20" "VRE32525|736|PUH-EMEP|Urine|2017-02-20"

# Import core genome SNP alignment as character vector
ST1471_fasta <- readr::read_file("~/School/Pitt/VRE Thesis/Annotation and Alignment/VRE_ST1471_core_gene_alignment.fna")
ST1471_fasta <- unlist(strsplit(ST1471_fasta, "\\n"))
# ST1471_fasta <- stringr::str_remove(ST1471_fasta, "\\n")
# ST1471_fasta <- ST1471_fasta[-length(ST1471_fasta)]

head(ST1471_fasta) #print to verify

ST736_fasta <- readr::read_file("VRE_ST736_core_gene_alignment.fna")
ST736_fasta <- unlist(strsplit(ST736_fasta, "\\n"))
# ST736_fasta <- stringr::str_remove(ST736_fasta, "\\n")
# ST736_fasta <- ST736_fasta[-length(ST736_fasta)]

head(ST736_fasta) #print to verify

# Rename Sequence IDs and export fasta file
cat(stringr::str_replace_all(ST1471_fasta, fasta_rename),
    file = "VRE_ST1471_core_gene_alignment.fna",
    sep = "\n")

cat(stringr::str_replace_all(ST736_fasta, fasta_rename),
    file = "VRE_ST736_core_gene_alignment.fna",
    sep = "\n")

##########Example##########
cat(
  ">VRE32493", "AGCT",
  ">VRE32503", "CAGT",
  ">VRE32504", "TCAA",
  file = "example.fasta", sep = "\n"
)

cat(
  "SpecimenID|ST|Geographic Location|Source|CultDate",
  "VRE32491|736|PUH - 10C|Blood|2016-12-07",
  "VRE32493|1471|PUH - 10N|Tissue/Surgical|2016-12-08",
  "VRE32503|1471|PUH - 11N|Wound|2017-01-05",
  "VRE32504|1471|PUH - EMEP|Blood|2017-01-10",
  "VRE32514|1471|PUH - 6F|Wound|2017-01-20",
```

```
  file = "example.txt", sep = "\n"
)

# Read in from example.txt
fasta_rename <- readr::read_file("example.txt")
fasta_rename <- unlist(strsplit(fasta_rename, "\\r"))
fasta_rename <- stringr::str_remove(fasta_rename[-1], "\\n")
fasta_rename <- fasta_rename[-length(fasta_rename)]
# Remove everything after first | to get the pattern to match off of
patterns <- stringr::str_remove(fasta_rename, "\\|(.+)$")
# Make replacement a named character vector in the form of pattern = replacement
names(fasta_rename) <- patterns

fasta_rename # print to verify

example_fasta <- readr::read_file("example.fasta")
example_fasta <- unlist(strsplit(example_fasta, "\\r"))
example_fasta <- stringr::str_remove(example_fasta, "\\n")
example_fasta <- example_fasta[-length(example_fasta)]

example_fasta #print to verify

cat(stringr::str_replace_all(example_fasta, fasta_rename),
    file = "example.fasta",
    sep = "\n")


# Name vector of new Sequence IDs
patterns <- stringr::str_remove(fasta_rename, "\\|(.+)$")
names(fasta_rename) <- patterns
head(fasta_rename)
# VRE32491                                VRE32493                                VRE32503
#                "VRE32491|736|PUH-10C|Blood|2016-12-07"                "VRE32493|1471|PUH-
10N|Tissue/Surgical|2016-12-08"         "VRE32503|1471|PUH-11N|Wound|2017-01-05"
# VRE32504                                VRE32514                                VRE32525
#   "VRE32504|1471|PUH-EMEP|Blood|2017-01-10"                              "VRE32514|1471|PUH-
6F|Wound|2017-01-20"          "VRE32525|736|PUH-EMEP|Urine|2017-02-20"

# Import core genome SNP alignment as character vector
ST1471_fasta <- readr::read_file("~/School/Pitt/VRE Thesis/Annotation and
Alignment/VRE_ST1471_core_gene_alignment.fna")
ST1471_fasta <- unlist(strsplit(ST1471_fasta, "\\n"))
# ST1471_fasta <- stringr::str_remove(ST1471_fasta, "\\n")
# ST1471_fasta <- ST1471_fasta[-length(ST1471_fasta)]

head(ST1471_fasta) #print to verify
```

```
ST736_fasta <- readr::read_file("VRE_ST736_core_gene_alignment.fna")
ST736_fasta <- unlist(strsplit(ST736_fasta, "\\n"))
# ST736_fasta <- stringr::str_remove(ST736_fasta, "\\n")
# ST736_fasta <- ST736_fasta[-length(ST736_fasta)]

head(ST736_fasta) #print to verify


# Rename Sequence IDs and export fasta file
cat(stringr::str_replace_all(ST1471_fasta, fasta_rename),
    file = "VRE_ST1471_core_gene_alignment.fna",
    sep = "\n")

cat(stringr::str_replace_all(ST736_fasta, fasta_rename),
    file = "VRE_ST736_core_gene_alignment.fna",
    sep = "\n")

##########Example##########
cat(
  ">VRE32493", "AGCT",
  ">VRE32503", "CAGT",
  ">VRE32504", "TCAA",
  file = "example.fasta", sep = "\n"
)

cat(
  "SpecimenID|ST|Geographic Location|Source|CultDate",
  "VRE32491|736|PUH - 10C|Blood|2016-12-07",
  "VRE32493|1471|PUH - 10N|Tissue/Surgical|2016-12-08",
  "VRE32503|1471|PUH - 11N|Wound|2017-01-05",
  "VRE32504|1471|PUH - EMEP|Blood|2017-01-10",
  "VRE32514|1471|PUH - 6F|Wound|2017-01-20",
  file = "example.txt", sep = "\n"
)
# Read in from example.txt
fasta_rename <- readr::read_file("example.txt")
fasta_rename <- unlist(strsplit(fasta_rename, "\\r"))
fasta_rename <- stringr::str_remove(fasta_rename[-1], "\\n")
fasta_rename <- fasta_rename[-length(fasta_rename)]
# Remove everything after first | to get the pattern to match off of
patterns <- stringr::str_remove(fasta_rename, "\\|(.+)$")
# Make replacement a named character vector in the form of pattern = replacement
names(fasta_rename) <- patterns

fasta_rename # print to verify
```

```
example_fasta <- readr::read_file("example.fasta")
example_fasta <- unlist(strsplit(example_fasta, "\\r"))
example_fasta <- stringr::str_remove(example_fasta, "\\n")
example_fasta <- example_fasta[-length(example_fasta)]

example_fasta #print to verify

cat(stringr::str_replace_all(example_fasta, fasta_rename),
    file = "example.fasta",
    sep = "\n")
```

Appendix D.2 Hierarchical Clustering

```
library(cluster)
library(dendextend)
library(dplyr)

## Import Data
setwd("~/School/Pitt/VRE Thesis/data_for_arvon")
ST1471 <- unlist(stringr::str_extract_all(list.files(), "VRE_ST1471.+matrix.+"))
ST736 <- unlist(stringr::str_extract_all(list.files(), "VRE_ST736.+matrix.+"))

ST1471_Matrices <- lapply(ST1471, data.table::fread, header = TRUE)
ST736_Matrices <- lapply(ST736, data.table::fread, header = TRUE)

names(ST1471_Matrices) <- ST1471
names(ST736_Matrices) <- ST736
DistMat <- function(x) {
  # function to convert data.frame into dissimilarity matrix w/ first column as rownames
  x <- as.dist(x[,-1])
  x <- as.matrix(x)
  return(x)
}

rmFct_Gap <- function(x){
  # function to remove gaps in ordinal variables
  x <- factor(x) # column as factors
  x <- ordered(x, sort(as.numeric(levels(x)))) # reorder levels
  levels(x) <- 1:length(levels(x)) # relevel
  return(x)
}
```

```
LinkClust <- function(dist, linkage, ...){
  # function to create a list of agnes clusters using multiple linkages
  clustH <- lapply(linkage, function(x) agnes(dist,
                                  method = x))
  names(clustH) <- linkage
  return(clustH)
}

`%>%` <- magrittr::`%>%`

Index <- function(x){
  # function to change rownames to 1:nrows
  rownames(x) <- 1:nrow(x)
  return(x)
}

# Create list of matrices
ST1471_Matrices <- lapply(ST1471_Matrices, DistMat)
ST736_Matrices <- lapply(ST736_Matrices, DistMat)
# Convert matrices to dist class
ST1471_Dist <- lapply(ST1471_Matrices, as.dist)
ST736_Dist <- lapply(ST736_Matrices, as.dist)
C1471 <- LinkClust(ST1471_Dist$VRE_ST1471_SNP_matrix_core.csv,
                linkage = c("single", "average", "ward"))
C736 <- LinkClust(ST736_Dist$VRE_ST736_SNP_matrix_core.csv,
                linkage = c("single", "average", "ward"))



# Height Threshold
H <- 15

# Get Isolate Clusters
## We will cut at height 15 as was done in `find_clusters.py`
S1471                                                                                    <-
as.data.frame(cbind(colnames(ST1471_Matrices$VRE_ST1471_SNP_matrix_core.csv),
                          cutree(C1471$single %>%
                                   cophenetic() %>%
                                   agnes(method = "single"),
                                 h = H)))
S736 <- as.data.frame(cbind(colnames(ST736_Matrices$VRE_ST736_SNP_matrix_core.csv),
                        cutree(C736$single %>%
                                 cophenetic() %>%
```

```
                        agnes(method = "single"),
                      h = H)))

A1471 <-
as.data.frame(cbind(colnames(ST1471_Matrices$VRE_ST1471_SNP_matrix_core.csv),
                    cutree(C1471$average %>%
                             cophenetic() %>%
                             agnes(method = "average"),
                           h = H)))
A736 <- as.data.frame(cbind(colnames(ST736_Matrices$VRE_ST736_SNP_matrix_core.csv),
                    cutree(C736$average %>%
                             cophenetic() %>%
                             agnes(method = "average"),
                           h = H)))



W1471 <-
as.data.frame(cbind(colnames(ST1471_Matrices$VRE_ST1471_SNP_matrix_core.csv),
                    cutree(C1471$ward %>%
                             cophenetic() %>%
                             agnes(method = "ward"),
                           h = H)))
W736 <- as.data.frame(cbind(colnames(ST736_Matrices$VRE_ST736_SNP_matrix_core.csv),
                    cutree(C736$ward %>%
                             cophenetic() %>%
                             agnes(method = "ward"),
                           h = H)))
LETTERS702 <- c(LETTERS, sapply(LETTERS, function(x) paste0(x, LETTERS)))

# Remove Clusters w/ single isolate
## Generate vector of clusters w/ single isolate
S1471 %>%
  group_by(V2) %>%
  tally() %>%
  filter(n > 1) %>% select(V2) %>%
  unlist() -> L1471

S736 %>%
  group_by(V2) %>%
  tally() %>%
  filter(n > 1) %>% select(V2) %>%
  unlist() -> L736
```

```
## Removal
S1471 %>%
  Index() %>%
  filter(V2 %in% L1471) -> S1471

S736 %>%
  Index() %>%
  filter(V2 %in% L736) -> S736

# Repeat for other linkage methods

A1471 %>%
  group_by(V2) %>%
  tally() %>%
  filter(n > 1) %>% select(V2) %>%
  unlist() -> L1471

A736 %>%
  group_by(V2) %>%
  tally() %>%
  filter(n > 1) %>% select(V2) %>%
  unlist() -> L736

A1471 %>%
  Index() %>%
  filter(V2 %in% L1471) -> A1471



A736 %>%
  Index() %>%
  filter(V2 %in% L736) -> A736

W1471 %>%
  group_by(V2) %>%
  tally() %>%
  filter(n > 1) %>% select(V2) %>%
  unlist() -> L1471

W736 %>%
  group_by(V2) %>%
  tally() %>%
  filter(n > 1) %>% select(V2) %>%
  unlist() -> L736
```

```
W1471 %>%
  Index() %>%
  filter(V2 %in% L1471) -> W1471

W736 %>%
  Index() %>%
  filter(V2 %in% L736) -> W736

# Label Clusters
S1471$V2 <- rmFct_Gap(S1471$V2)
S1471$V2 <- LETTERS702[as.numeric(S1471$V2)]


S736$V2 <- rmFct_Gap(S736$V2)
S736$V2 <- LETTERS702[as.numeric(S736$V2)]

A1471$V2 <- rmFct_Gap(A1471$V2)
A1471$V2 <- LETTERS702[as.numeric(A1471$V2)]

A736$V2 <- rmFct_Gap(A736$V2)
A736$V2 <- LETTERS702[as.numeric(A736$V2)]

W1471$V2 <- rmFct_Gap(W1471$V2)
W1471$V2 <- LETTERS702[as.numeric(W1471$V2)]

W736$V2 <- rmFct_Gap(W736$V2)
W736$V2 <- LETTERS702[as.numeric(W736$V2)]

## Write Isolate Labels to .csv
## Format VRE_[ST]_[Linkage]_[distance]

# write.table(S1471, file = "./VRE_1471_Single_SNP.csv",
#             quote = FALSE, row.names = FALSE, col.names = FALSE, sep = ",")
# write.table(S736, file = "./VRE_736_Single_SNP.csv",
#             quote = FALSE, row.names = FALSE, col.names = FALSE, sep = ",")
#
# write.table(A1471, file = "./VRE_1471_Average_SNP.csv",
#             quote = FALSE, row.names = FALSE, col.names = FALSE, sep = ",")
# write.table(A736, file = "./VRE_736_Average_SNP.csv",
#             quote = FALSE, row.names = FALSE, col.names = FALSE, sep = ",")
#
#
# write.table(W1471, file = "./VRE_1471_Ward_SNP.csv",
#             quote = FALSE, row.names = FALSE, col.names = FALSE, sep = ",")
# write.table(W736, file = "./VRE_736_Ward_SNP.csv",
#             quote = FALSE, row.names = FALSE, col.names = FALSE, sep = ",")
```

```
C1471$single %>%
  cophenetic() %>%
  agnes(method = "single") %>%
  as.dendrogram() -> DS1471
C736$single %>%
  cophenetic() %>%
  agnes(method = "single") %>%
  as.dendrogram() -> DS736
C1471$average %>%
  cophenetic() %>%
  agnes(method = "average") %>%
  as.dendrogram() -> DA1471
C736$average %>%
  cophenetic() %>%
  agnes(method = "average") %>%
  as.dendrogram() -> DA736
C1471$ward %>%
  cophenetic() %>%
  agnes(method = "ward") %>%
  as.dendrogram() -> DW1471
C736$ward %>%
  cophenetic() %>%
  agnes(method = "ward") %>%
  as.dendrogram() -> DW736

tanglegram(dendlist(DS1471, DA1471), margin_inner = 0.5, common_subtrees_color_branches =
TRUE,
        main_left = "Single", main_right = "Average", main = "ST1471 Linkage Comparison",
cex_main = 1)

tanglegram(dendlist(DS1471, DW1471), margin_inner = 0.5, common_subtrees_color_branches
= TRUE,
        main_left = "Single", main_right = "Ward", main = "ST1471 Linkage Comparison",
cex_main = 1)

tanglegram(dendlist(DA1471, DW1471), margin_inner = 0.5, common_subtrees_color_branches
= TRUE,
        main_left = "Average", main_right = "Ward", main = "ST1471 Linkage Comparison",
cex_main = 1)
tanglegram(dendlist(DS736, DA736), margin_inner = 0.5, common_subtrees_color_branches =
TRUE,
        main_left = "Single", main_right = "Average", main = "ST736 Linkage Comparison",
cex_main = 1)
```

```
tanglegram(dendlist(DS736, DW736), margin_inner = 0.5, common_subtrees_color_branches = TRUE,
        main_left = "Single", main_right = "Ward", main = "ST736 Linkage Comparison", cex_main = 1)

tanglegram(dendlist(DA736, DW736), margin_inner = 0.5, common_subtrees_color_branches = TRUE,
        main_left = "Average", main_right = "Ward", main = "ST736 Linkage Comparison", cex_main = 1)

library(corrplot)

corrplot(cor.dendlist(dendlist("Single" = DS1471, "Average" = DA1471, "Ward" = DW1471)),
       "pie", "lower", title = "ST1471 Correlation Matrix", mar = c(0, 0, 2,0), tl.pos = "d")

corrplot(cor.dendlist(dendlist("Single" = DS736, "Average" = DA736, "Ward" = DW736)),
       "pie", "lower", title = "ST736 Correlation Matrix", mar = c(0, 0, 2,0), tl.pos = "d")

SA1471 <- cor_cophenetic(DS1471, DA1471)
SW1471 <- cor_cophenetic(DS1471, DW1471)
AW1471 <- cor_cophenetic(DA1471, DW1471)

SA736 <- cor_cophenetic(DS736, DA736)
SW736 <- cor_cophenetic(DS736, DW736)
AW736 <- cor_cophenetic(DA736, DW736)
```

Appendix D.3 Dendrogram Visualization

```
C1471$single %>%
  cophenetic() %>%
  agnes(method = "single") %>%
  as.dendrogram() -> DS1471
C736$single %>%
  cophenetic() %>%
  agnes(method = "single") %>%
  as.dendrogram() -> DS736
C1471$average %>%
  cophenetic() %>%
  agnes(method = "average") %>%
  as.dendrogram() -> DA1471
C736$average %>%
  cophenetic() %>%
  agnes(method = "average") %>%
  as.dendrogram() -> DA736
C1471$ward %>%
  cophenetic() %>%
  agnes(method = "ward") %>%
  as.dendrogram() -> DW1471
C736$ward %>%
  cophenetic() %>%
  agnes(method = "ward") %>%
  as.dendrogram() -> DW736

tanglegram(dendlist(DS1471, DA1471), margin_inner = 0.5, common_subtrees_color_branches =
TRUE,
        main_left = "Single", main_right = "Average", main = "ST1471 Linkage Comparison",
cex_main = 1)

tanglegram(dendlist(DS1471, DW1471), margin_inner = 0.5, common_subtrees_color_branches
= TRUE,
        main_left = "Single", main_right = "Ward", main = "ST1471 Linkage Comparison",
cex_main = 1)

tanglegram(dendlist(DA1471, DW1471), margin_inner = 0.5, common_subtrees_color_branches
= TRUE,
        main_left = "Average", main_right = "Ward", main = "ST1471 Linkage Comparison",
cex_main = 1)
tanglegram(dendlist(DS736, DA736), margin_inner = 0.5, common_subtrees_color_branches =
TRUE,
        main_left = "Single", main_right = "Average", main = "ST736 Linkage Comparison",
cex_main = 1)
```

```
tanglegram(dendlist(DS736, DW736), margin_inner = 0.5, common_subtrees_color_branches = TRUE,
        main_left = "Single", main_right = "Ward", main = "ST736 Linkage Comparison", cex_main = 1)

tanglegram(dendlist(DA736, DW736), margin_inner = 0.5, common_subtrees_color_branches = TRUE,
        main_left = "Average", main_right = "Ward", main = "ST736 Linkage Comparison", cex_main = 1)
library(corrplot)

corrplot(cor.dendlist(dendlist("Single" = DS1471, "Average" = DA1471, "Ward" = DW1471)),
       "pie", "lower", title = "ST1471 Correlation Matrix")

corrplot(cor.dendlist(dendlist("Single" = DS736, "Average" = DA736, "Ward" = DW736)),
       "pie", "lower", title = "ST736 Correlation Matrix")

SA1471 <- cor_cophenetic(DS1471, DA1471)
SW1471 <- cor_cophenetic(DS1471, DW1471)
AW1471 <- cor_cophenetic(DA1471, DW1471)

SA736 <- cor_cophenetic(DS736, DA736)
SW736 <- cor_cophenetic(DS736, DW736)
AW736 <- cor_cophenetic(DA736, DW736)
```

Appendix D.4 Bayesian Clustering

```
setwd("~/School/Pitt/VRE Thesis/data_for_arvon")

# Custom Functions
Index <- function(x){
  # function to change rownames to 1:nrows
  rownames(x) <- 1:nrow(x)
  return(x)
}

rmFct_Gap <- function(x){
  # function to remove gaps in ordinal variables
  x <- factor(x) # column as factors
```

```
  x <- ordered(x, sort(as.numeric(levels(x)))) # reorder levels
  levels(x) <- 1:length(levels(x)) # relevel
  return(x)
}

# Import Newick tree files as dendrogram class object
ST1471_Tree <- phylogram::read.dendrogram("~/School/Pitt/VRE Thesis/XML/ST1471_Export.tree")
ST736_Tree <- phylogram::read.dendrogram("~/School/Pitt/VRE Thesis/XML/ST736_Export.tree")

# cut dendrograms at height 15
H <- 15

# Get isolate clusters
C1471 <- as.data.frame(dendextend::cutree(ST1471_Tree, H))
C736 <- as.data.frame(dendextend::cutree(ST736_Tree, H))

# Remove row and column names
C1471 <- cbind(row.names(C1471), C1471)
C736 <- cbind(row.names(C736), C736)

rownames(C1471) <- c()
rownames(C736) <- c()

colnames(C1471) <- c("V1", "V2")
colnames(C736) <- c("V1", "V2")

# Truncate Sample ID names
C1471[, 1] <- stringr::str_remove(C1471[, 1], "\\|.*")
C736[, 1] <- stringr::str_remove(C736[, 1], "\\|.*")

# Create alphabetic cluster labels
LETTERS702 <- c(LETTERS, sapply(LETTERS, function(x) paste0(x, LETTERS)))

# Remove Clusters w/ single isolate
## Generate vector of clusters w/ single isolate
library(dplyr)
C1471 %>%
group_by(V2) %>%
  tally() %>%
  filter(n > 1) %>% select(V2) %>%
  unlist() -> L1471

C736 %>%
  group_by(V2) %>%
```

```
  tally() %>%
  filter(n > 1) %>% select(V2) %>%
  unlist() -> L736

## Removal
C1471 %>%
  Index() %>%
  filter(V2 %in% L1471) -> C1471

C736 %>%
  Index() %>%
  filter(V2 %in% L736) -> C736

# Label clusters
C1471$V2 <- rmFct_Gap(C1471$V2)
C1471$V2 <- LETTERS702[as.numeric(C1471$V2)]

C736$V2 <- rmFct_Gap(C736$V2)
C736$V2 <- LETTERS702[as.numeric(C736$V2)]

## Write Isolate Labels to .csv
## Format VRE_[ST]_MASCOT

# write.table(C1471, file = "./VRE_1471_MASCOT.csv",
#             quote = FALSE, row.names = FALSE, col.names = FALSE, sep = ",")
# write.table(C736, file = "./VRE_736_MASCOT.csv",
#             quote = FALSE, row.names = FALSE, col.names = FALSE, sep = ",")
```

Appendix D.5 Adjusted Rand Index Calculation

```
library(dplyr)
library(gt)

# Import metadata
setwd("~/School/Pitt/VRE Thesis/data_for_arvon/736 and 1471")
metadf <- read.csv("~/School/Pitt/VRE Thesis/data_for_arvon/VRE_Metadata_Filtered.csv")
%>%
  select(SpecimenID, ST)
metadf$Truth <- rep(NA_character_, nrow(metadf))
# metadf$Average <- rep(NA_character_, nrow(metadf))
```

```
# metadf$Ward <- rep(NA_character_, nrow(metadf))

# Import Ground Truth clusters
Clusters <- lapply(list.files()[-1], xlsx::read.xlsx, sheetIndex = 1)
names(Clusters) <- list.files()[-1]

# Label Ground Truth isolates
metadf$Truth[metadf$SpecimenID %in% Clusters$VRE_ST1471D.xlsx$SpecimenID] <- "D"
metadf$Truth[metadf$SpecimenID %in% Clusters$VRE_ST1471G.xlsx$SpecimenID] <- "G"
metadf$Truth[metadf$SpecimenID %in% Clusters$VRE_ST1471I.xlsx$SpecimenID] <- "I"
metadf$Truth[metadf$SpecimenID %in% Clusters$VRE_ST1471K.xlsx$SpecimenID] <- "K"
metadf$Truth[metadf$SpecimenID %in% Clusters$VRE_ST1471L.xlsx$SpecimenID] <- "L"
metadf$Truth[metadf$SpecimenID %in% Clusters$VRE_ST1471M.xlsx$SpecimenID] <- "M"
metadf$Truth[metadf$SpecimenID %in% Clusters$VRE_ST1471N.xlsx$SpecimenID] <- "N"
metadf$Truth[metadf$SpecimenID %in% Clusters$VRE_ST1471O.xlsx$SpecimenID] <- "O"
metadf$Truth[metadf$SpecimenID %in% Clusters$VRE_ST1471P.xlsx$SpecimenID] <- "P"
metadf$Truth[metadf$SpecimenID %in% Clusters$VRE_ST1471Q.xlsx$SpecimenID] <- "Q"
metadf$Truth[metadf$SpecimenID %in% Clusters$VRE_ST736A.xlsx$SpecimenID] <- "A"
metadf$Truth[metadf$SpecimenID %in% Clusters$VRE_ST736B.xlsx$SpecimenID] <- "B"
metadf$Truth[metadf$SpecimenID %in% Clusters$VRE_ST736C.xlsx$SpecimenID] <- "C"
metadf$Truth[metadf$SpecimenID %in% Clusters$VRE_ST736D.xlsx$SpecimenID] <- "D"
metadf$Truth[metadf$SpecimenID %in% Clusters$VRE_ST736E.xlsx$SpecimenID] <- "E"
metadf$Truth[metadf$SpecimenID %in% Clusters$VRE_ST736F.xlsx$SpecimenID] <- "F"
metadf$Truth[metadf$SpecimenID %in% Clusters$VRE_ST736I.xlsx$SpecimenID] <- "I"
metadf$Truth[metadf$SpecimenID %in% Clusters$VRE_ST736J.xlsx$SpecimenID] <- "J"
metadf$Truth[metadf$SpecimenID %in% Clusters$VRE_ST736K.xlsx$SpecimenID] <- "K"
metadf$Truth[metadf$SpecimenID %in% Clusters$VRE_ST736L.xlsx$SpecimenID] <- "L"

setwd("E:/School/Pitt/VRE Thesis/data_for_arvon")
# list of all SNP linkage clusters
Linkages <- lapply(unlist(stringr::str_extract_all(list.files(), ".*SNP.csv$")),
                   read.csv, header = F)

names(Linkages) <- unlist(stringr::str_extract_all(list.files(), ".*SNP.csv$"))

# list of all MASCOT
Bayesians <- lapply(unlist(stringr::str_extract_all(list.files(), ".*MASCOT.csv$")),
                    read.csv, header = F)
names(Bayesians) <- unlist(stringr::str_extract_all(list.files(), ".*MASCOT.csv$"))

# join cluster labels for each linkage to metadata
metadf <- left_join(metadf, Linkages$VRE_1471_Average_SNP.csv, by = c("SpecimenID" =
"V1"))
metadf <- left_join(metadf, Linkages$VRE_736_Average_SNP.csv, by = c("SpecimenID" =
"V1"))
metadf <- left_join(metadf, Linkages$VRE_1471_Ward_SNP.csv, by = c("SpecimenID" = "V1"))
```

```
metadf <- left_join(metadf, Linkages$VRE_736_Ward_SNP.csv, by = c("SpecimenID" = "V1"))
metadf <- left_join(metadf, Bayesians$VRE_1471_MASCOT.csv, by = c("SpecimenID" = "V1"))
metadf <- left_join(metadf, Bayesians$VRE_736_MASCOT.csv, by = c("SpecimenID" = "V1"))

# merge linkage columns
metadf <- metadf %>%
  mutate(Average = coalesce(V2.x, V2.y),
         Ward = coalesce(V2.x.x, V2.y.y),
         Bayes = coalesce(V2.x.x.x, V2.y.y.y)) %>%
  select(SpecimenID, ST, Truth, Average, Ward, Bayes)

# Create list of AllPossibleRoutes for each cluster
GT <- lapply(list.files("~/School/Pitt/VRE Thesis/data_for_arvon/736 and 1471/736 and 1471"),
             function(x) readxl::read_xlsx(path = paste0("~/School/Pitt/VRE Thesis/data_for_arvon/736 and 1471/736 and 1471/",
                                                         x),
                                           sheet = "AllPossibleRoutes"))
names(GT) <- list.files("~/School/Pitt/VRE Thesis/data_for_arvon/736 and 1471/736 and 1471")

AVG <- lapply(unlist(stringr::str_extract_all(list.files("~/School/Pitt/VRE Thesis/EDS-HAT_VRE_Clusters_Results_06182021"),
                                              "VRE_[[:digit:]]*_Average_SNP.*")),
              function(x) readxl::read_xlsx(path = paste0("~/School/Pitt/VRE Thesis/EDS-HAT_VRE_Clusters_Results_06182021/",x),
                                            sheet = "AllPossibleRoutes"))
names(AVG) <- unlist(stringr::str_extract_all(list.files("~/School/Pitt/VRE Thesis/EDS-HAT_VRE_Clusters_Results_06182021"),
                                              "VRE_[[:digit:]]*_Average_SNP.*"))

WARD <- lapply(unlist(stringr::str_extract_all(list.files("~/School/Pitt/VRE Thesis/EDS-HAT_VRE_Clusters_Results_06182021"),
                                               "VRE_[[:digit:]]*_Ward_SNP.*")),
               function(x) readxl::read_xlsx(path = paste0("~/School/Pitt/VRE Thesis/EDS-HAT_VRE_Clusters_Results_06182021/",x),
                                             sheet = "AllPossibleRoutes"))
names(WARD) <- unlist(stringr::str_extract_all(list.files("~/School/Pitt/VRE Thesis/EDS-HAT_VRE_Clusters_Results_06182021"),
                                               "VRE_[[:digit:]]*_Ward_SNP.*"))

BAYES <- lapply(unlist(stringr::str_extract_all(list.files("~/School/Pitt/VRE Thesis/EDS-HAT_VRE_Clusters_12042021"),
                                                "VRE_[[:digit:]]*_MASCOT.*")),
                function(x) readxl::read_xlsx(path = paste0("~/School/Pitt/VRE Thesis/EDS-HAT_VRE_Clusters_12042021/",x),
                                              sheet = "AllPossibleRoutes"))
```

```
names(BAYES) <- unlist(stringr::str_extract_all(list.files("~/School/Pitt/VRE Thesis/EDS-HAT_VRE_Clusters_12042021"),
                                                "VRE_[[:digit:]]*_MASCOT.*"))

# Filter list to include only 'UNIT' Type and then arrange by Rank
GT <- lapply(GT, function(x) dplyr::filter(x, Type == 'UNIT') %>%
               dplyr::arrange(Rank) %>%
               dplyr::select(Route, Rank))
AVG <- lapply(AVG, function(x) dplyr::filter(x, Type == 'UNIT') %>%
                dplyr::arrange(Rank) %>%
                dplyr::select(Route, Rank))
WARD <- lapply(WARD, function(x) dplyr::filter(x, Type == 'UNIT') %>%
                 dplyr::arrange(Rank) %>%
                 dplyr::select(Route, Rank))
BAYES <- lapply(BAYES, function(x) dplyr::filter(x, Type == 'UNIT') %>%
                  dplyr::arrange(Rank) %>%
                  dplyr::select(Route, Rank))

# Identify Geographic Locations for Ground Truth Clusters

metadf$Truth[which(metadf$Truth == "A" & metadf$ST == 1471)] <- NA_character_ # VRE33105 and VRE33096 are from same patient
metadf$Truth[which(metadf$Truth == "D" & metadf$ST == 1471)] <- "12D"
metadf$Truth[which(metadf$Truth == "G" & metadf$ST == 1471)] <- "5S"
metadf$Truth[which(metadf$Truth == "I" & metadf$ST == 1471)] <- "5D"
metadf$Truth[which(metadf$Truth == "K" & metadf$ST == 1471)] <- NA_character_
metadf$Truth[which(metadf$Truth == "L" & metadf$ST == 1471)] <- "CT11"
metadf$Truth[which(metadf$Truth == "M" & metadf$ST == 1471)] <- "RLTA"
metadf$Truth[which(metadf$Truth == "N" & metadf$ST == 1471)] <- NA_character_
metadf$Truth[which(metadf$Truth == "O" & metadf$ST == 1471)] <- "5S"
metadf$Truth[which(metadf$Truth == "P" & metadf$ST == 1471)] <- "10G"
metadf$Truth[which(metadf$Truth == "Q" & metadf$ST == 1471)] <- NA_character_
metadf$Truth[which(metadf$Truth == "A" & metadf$ST == 736)] <- "12S"
metadf$Truth[which(metadf$Truth == "B" & metadf$ST == 736)] <- "7G"
metadf$Truth[which(metadf$Truth == "C" & metadf$ST == 736)] <- "3E"
metadf$Truth[which(metadf$Truth == "D" & metadf$ST == 736)] <- "12N"
metadf$Truth[which(metadf$Truth == "E" & metadf$ST == 736)] <- NA_character_
metadf$Truth[which(metadf$Truth == "F" & metadf$ST == 736)] <- "CT10"
metadf$Truth[which(metadf$Truth == "I" & metadf$ST == 736)] <- "10G"
metadf$Truth[which(metadf$Truth == "J" & metadf$ST == 736)] <- "11N"
metadf$Truth[which(metadf$Truth == "K" & metadf$ST == 736)] <- NA_character_
metadf$Truth[which(metadf$Truth == "L" & metadf$ST == 736)] <- "CT11"

# Identify Geographic Locations for Average Clusters
metadf$Average[which(metadf$Average == "A" & metadf$ST == 1471)] <- "RHAB"
metadf$Average[which(metadf$Average == "B" & metadf$ST == 1471)] <- "5S"
```

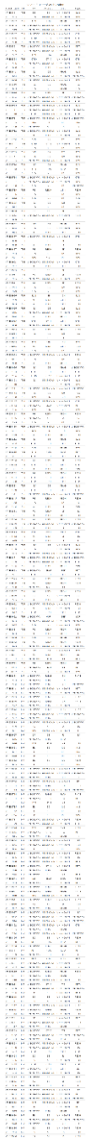
```
metadf$Average[which(metadf$Average == "C" & metadf$ST == 1471)] <- "SICU"
metadf$Average[which(metadf$Average == "D" & metadf$ST == 1471)] <- "5D"
metadf$Average[which(metadf$Average == "E" & metadf$ST == 1471)] <- NA_character_
metadf$Average[which(metadf$Average == "F" & metadf$ST == 1471)] <- "CT10"
metadf$Average[which(metadf$Average == "G" & metadf$ST == 1471)] <- NA_character_
metadf$Average[which(metadf$Average == "H" & metadf$ST == 1471)] <- NA_character_
metadf$Average[which(metadf$Average == "I" & metadf$ST == 1471)] <- NA_character_
metadf$Average[which(metadf$Average == "J" & metadf$ST == 1471)] <- "5D"
metadf$Average[which(metadf$Average == "K" & metadf$ST == 1471)] <- NA_character_
metadf$Average[which(metadf$Average == "L" & metadf$ST == 1471)] <- "5S"
metadf$Average[which(metadf$Average == "M" & metadf$ST == 1471)] <- "RLTA"
metadf$Average[which(metadf$Average == "N" & metadf$ST == 1471)] <- "CT11"
metadf$Average[which(metadf$Average == "O" & metadf$ST == 1471)] <- NA_character_
metadf$Average[which(metadf$Average == "P" & metadf$ST == 1471)] <- "CT11"
metadf$Average[which(metadf$Average == "A" & metadf$ST == 736)] <- "5D"
metadf$Average[which(metadf$Average == "B" & metadf$ST == 736)] <- "RLTA"
metadf$Average[which(metadf$Average == "C" & metadf$ST == 736)] <- NA_character_
metadf$Average[which(metadf$Average == "D" & metadf$ST == 736)] <- "CT10"
metadf$Average[which(metadf$Average == "E" & metadf$ST == 736)] <- NA_character_
metadf$Average[which(metadf$Average == "F" & metadf$ST == 736)] <- "REDH"
metadf$Average[which(metadf$Average == "G" & metadf$ST == 736)] <- "12S"
metadf$Average[which(metadf$Average == "H" & metadf$ST == 736)] <- "10G"
metadf$Average[which(metadf$Average == "I" & metadf$ST == 736)] <- "5S"
metadf$Average[which(metadf$Average == "J" & metadf$ST == 736)] <- NA_character_
metadf$Average[which(metadf$Average == "K" & metadf$ST == 736)] <- NA_character_
metadf$Average[which(metadf$Average == "L" & metadf$ST == 736)] <- "12N"
metadf$Average[which(metadf$Average == "M" & metadf$ST == 736)] <- "CT11"
metadf$Average[which(metadf$Average == "N" & metadf$ST == 736)] <- "SICU"
metadf$Average[which(metadf$Average == "O" & metadf$ST == 736)] <- "6GF"
metadf$Average[which(metadf$Average == "P" & metadf$ST == 736)] <- NA_character_

# Identify Geographic Locations for Ward Clusters
metadf$Ward[which(metadf$Ward == "A" & metadf$ST == 1471)] <- "12N"
metadf$Ward[which(metadf$Ward == "B" & metadf$ST == 1471)] <- "11N"
metadf$Ward[which(metadf$Ward == "C" & metadf$ST == 1471)] <- NA_character_
metadf$Ward[which(metadf$Ward == "D" & metadf$ST == 1471)] <- "RLTA"
metadf$Ward[which(metadf$Ward == "E" & metadf$ST == 1471)] <- "5S"
metadf$Ward[which(metadf$Ward == "F" & metadf$ST == 1471)] <- "SICU"
metadf$Ward[which(metadf$Ward == "G" & metadf$ST == 1471)] <- "10G"
metadf$Ward[which(metadf$Ward == "H" & metadf$ST == 1471)] <- NA_character_
metadf$Ward[which(metadf$Ward == "I" & metadf$ST == 1471)] <- "7G"
metadf$Ward[which(metadf$Ward == "J" & metadf$ST == 1471)] <- "RHAB"
metadf$Ward[which(metadf$Ward == "K" & metadf$ST == 1471)] <- NA_character_
metadf$Ward[which(metadf$Ward == "L" & metadf$ST == 1471)] <- NA_character_
metadf$Ward[which(metadf$Ward == "M" & metadf$ST == 1471)] <- NA_character_
metadf$Ward[which(metadf$Ward == "N" & metadf$ST == 1471)] <- NA_character_
```

```
metadf$Ward[which(metadf$Ward == "O" & metadf$ST == 1471)] <- "5D"
metadf$Ward[which(metadf$Ward == "P" & metadf$ST == 1471)] <- NA_character_
metadf$Ward[which(metadf$Ward == "Q" & metadf$ST == 1471)] <- "5S"
metadf$Ward[which(metadf$Ward == "R" & metadf$ST == 1471)] <- NA_character_
metadf$Ward[which(metadf$Ward == "S" & metadf$ST == 1471)] <- "RLTA"
metadf$Ward[which(metadf$Ward == "T" & metadf$ST == 1471)] <- "CT11"
metadf$Ward[which(metadf$Ward == "U" & metadf$ST == 1471)] <- NA_character_
metadf$Ward[which(metadf$Ward == "V" & metadf$ST == 1471)] <- NA_character_
metadf$Ward[which(metadf$Ward == "W" & metadf$ST == 1471)] <- "CT11"
metadf$Ward[which(metadf$Ward == "X" & metadf$ST == 1471)] <- NA_character_
metadf$Ward[which(metadf$Ward == "A" & metadf$ST == 736)] <- NA_character_
metadf$Ward[which(metadf$Ward == "B" & metadf$ST == 736)] <- "RHAB"
metadf$Ward[which(metadf$Ward == "C" & metadf$ST == 736)] <- NA_character_
metadf$Ward[which(metadf$Ward == "D" & metadf$ST == 736)] <- NA_character_
metadf$Ward[which(metadf$Ward == "E" & metadf$ST == 736)] <- NA_character_
metadf$Ward[which(metadf$Ward == "F" & metadf$ST == 736)] <- "RHAB"
metadf$Ward[which(metadf$Ward == "G" & metadf$ST == 736)] <- "9F"
metadf$Ward[which(metadf$Ward == "H" & metadf$ST == 736)] <- "10G"
metadf$Ward[which(metadf$Ward == "I" & metadf$ST == 736)] <- "5S"
metadf$Ward[which(metadf$Ward == "J" & metadf$ST == 736)] <- NA_character_
metadf$Ward[which(metadf$Ward == "K" & metadf$ST == 736)] <- "12S"
metadf$Ward[which(metadf$Ward == "L" & metadf$ST == 736)] <- "RLTA"
metadf$Ward[which(metadf$Ward == "M" & metadf$ST == 736)] <- NA_character_
metadf$Ward[which(metadf$Ward == "N" & metadf$ST == 736)] <- "CT11"
metadf$Ward[which(metadf$Ward == "O" & metadf$ST == 736)] <- "3E"
metadf$Ward[which(metadf$Ward == "P" & metadf$ST == 736)] <- NA_character_
metadf$Ward[which(metadf$Ward == "Q" & metadf$ST == 736)] <- NA_character_
metadf$Ward[which(metadf$Ward == "R" & metadf$ST == 736)] <- "12N"
metadf$Ward[which(metadf$Ward == "S" & metadf$ST == 736)] <- "REDH"
metadf$Ward[which(metadf$Ward == "T" & metadf$ST == 736)] <- "CT11"
metadf$Ward[which(metadf$Ward == "U" & metadf$ST == 736)] <- NA_character_
metadf$Ward[which(metadf$Ward == "V" & metadf$ST == 736)] <- "6GF"
metadf$Ward[which(metadf$Ward == "W" & metadf$ST == 736)] <- "6GF"
metadf$Ward[which(metadf$Ward == "X" & metadf$ST == 736)] <- NA_character_

# Identify Geographic Locations for Bayesian Clusters
metadf$Bayes[which(metadf$Bayes == "A" & metadf$ST == 1471)] <- "RHAB"
metadf$Bayes[which(metadf$Bayes == "B" & metadf$ST == 1471)] <- "RHAB"
metadf$Bayes[which(metadf$Bayes == "C" & metadf$ST == 1471)] <- "REDH"
metadf$Bayes[which(metadf$Bayes == "D" & metadf$ST == 1471)] <- NA_character_
metadf$Bayes[which(metadf$Bayes == "E" & metadf$ST == 1471)] <- NA_character_
metadf$Bayes[which(metadf$Bayes == "F" & metadf$ST == 1471)] <- "CT11"
metadf$Bayes[which(metadf$Bayes == "G" & metadf$ST == 1471)] <- "RLTA"
metadf$Bayes[which(metadf$Bayes == "H" & metadf$ST == 1471)] <- "REDH"
metadf$Bayes[which(metadf$Bayes == "I" & metadf$ST == 1471)] <- "10G"
metadf$Bayes[which(metadf$Bayes == "J" & metadf$ST == 1471)] <- "RLTA"
```

```
metadf$Bayes[which(metadf$Bayes == "K" & metadf$ST == 1471)] <- "11S"
metadf$Bayes[which(metadf$Bayes == "L" & metadf$ST == 1471)] <- "RLTA"
metadf$Bayes[which(metadf$Bayes == "M" & metadf$ST == 1471)] <- NA_character_
metadf$Bayes[which(metadf$Bayes == "N" & metadf$ST == 1471)] <- NA_character_
metadf$Bayes[which(metadf$Bayes == "O" & metadf$ST == 1471)] <- NA_character_
metadf$Bayes[which(metadf$Bayes == "P" & metadf$ST == 1471)] <- NA_character_
metadf$Bayes[which(metadf$Bayes == "Q" & metadf$ST == 1471)] <- NA_character_
metadf$Bayes[which(metadf$Bayes == "R" & metadf$ST == 1471)] <- NA_character_
metadf$Bayes[which(metadf$Bayes == "S" & metadf$ST == 1471)] <- NA_character_
metadf$Bayes[which(metadf$Bayes == "T" & metadf$ST == 1471)] <- "9D"
metadf$Bayes[which(metadf$Bayes == "U" & metadf$ST == 1471)] <- NA_character_
metadf$Bayes[which(metadf$Bayes == "V" & metadf$ST == 1471)] <- NA_character_
metadf$Bayes[which(metadf$Bayes == "A" & metadf$ST == 736)] <- "12N"
metadf$Bayes[which(metadf$Bayes == "B" & metadf$ST == 736)] <- "3E"
metadf$Bayes[which(metadf$Bayes == "C" & metadf$ST == 736)] <- "12S"
metadf$Bayes[which(metadf$Bayes == "D" & metadf$ST == 736)] <- "12S"
metadf$Bayes[which(metadf$Bayes == "E" & metadf$ST == 736)] <- "10C"
metadf$Bayes[which(metadf$Bayes == "F" & metadf$ST == 736)] <- NA_character_
metadf$Bayes[which(metadf$Bayes == "G" & metadf$ST == 736)] <- "RLTA"
metadf$Bayes[which(metadf$Bayes == "H" & metadf$ST == 736)] <- "5S"
metadf$Bayes[which(metadf$Bayes == "I" & metadf$ST == 736)] <- "CT11"
metadf$Bayes[which(metadf$Bayes == "J" & metadf$ST == 736)] <- "6GF"
metadf$Bayes[which(metadf$Bayes == "K" & metadf$ST == 736)] <- "REDH"
metadf$Bayes[which(metadf$Bayes == "L" & metadf$ST == 736)] <- "REDH"
metadf$Bayes[which(metadf$Bayes == "M" & metadf$ST == 736)] <- "11S"
metadf$Bayes[which(metadf$Bayes == "N" & metadf$ST == 736)] <- "9D"
metadf$Bayes[which(metadf$Bayes == "O" & metadf$ST == 736)] <- "10G"

# 0.4010779
mclust::adjustedRandIndex(metadf_ST736$Truth, metadf_ST736$Bayes)
# 0.5694458
mclust::adjustedRandIndex(metadf_ST1471$Truth, metadf_ST1471$Average)
# 1
mclust::adjustedRandIndex(metadf_ST1471$Truth, metadf_ST1471$Ward)
# 0.4808493
mclust::adjustedRandIndex(metadf_ST1471$Truth, metadf_ST1471$Bayes)
# 0.3017723

# Calculation proportion of of matching classes
length(which(metadf_ST736$Truth == metadf_ST736$Average)) / nrow(metadf_ST736)
# 0.5132743
length(which(metadf_ST736$Truth == metadf_ST736$Ward)) / nrow(metadf_ST736)
# 0.380531
length(which(metadf_ST736$Truth == metadf_ST736$Bayes)) / nrow(metadf_ST736)
# 0.2566372
length(which(metadf_ST1471$Truth == metadf_ST1471$Average)) / nrow(metadf_ST1471)
```

```
# 0.8108108
length(which(metadf_ST1471$Truth == metadf_ST1471$Ward)) / nrow(metadf_ST1471)
# 0.3243243
length(which(metadf_ST1471$Truth == metadf_ST1471$Bayes)) / nrow(metadf_ST1471)
# 0.1351351
metadf$Bayes[which(metadf$Bayes == "P" & metadf$ST == 736)] <- NA_character_

# Write new sheet to .xlsx file
# xlsx::write.xlsx(metadf,
#                  file = "~/School/Pitt/VRE Thesis/data_for_arvon/VRE_Metadata_Filtered.xlsx",
#                  sheetName = "SNP_Cluster_Results",
#                  append = T,
#                  row.names = F)

# Tidy-up 'metadf' labels
metadf <- metadf %>% mutate(across(Truth:Bayes, stringr::str_replace_na, "UNKNOWN"))
metadf <- metadf %>% mutate(across(ST:Bayes, as.factor))
Labels <- c(levels(metadf$Truth), levels(metadf$Average), levels(metadf$Ward),
levels(metadf$Bayes)) %>% unique()
metadf <- metadf %>% mutate(across(Truth:Bayes, forcats::fct_expand, Labels))
metadf <- metadf %>% mutate(across(Truth:Bayes, factor, Labels))

# Split 'metadf' by ST type
metadf_ST736 <- metadf %>% filter(ST == 736)
metadf_ST1471 <- metadf %>% filter(ST == 1471)

# Remove isolates w/ "UNKNOWN" geographic location in GT
metadf_ST736 <- metadf_ST736 %>% filter(Truth != "UNKNOWN")
metadf_ST1471 <- metadf_ST1471 %>% filter(Truth != "UNKNOWN")

# Calculate Adjusted Rand Index
mclust::adjustedRandIndex(metadf_ST736$Truth, metadf_ST736$Average)
# 0.963843
mclust::adjustedRandIndex(metadf_ST736$Truth, metadf_ST736$Ward)
```

Appendix E Cluster Classification



Appendix Figure 5 https://d-scholarship.pitt.edu/42864/1/Cluster_Classification.png

Bibliography

1. Lance GN, Williams WT. A General Theory of Classificatory Sorting Strategies: 1. Hierarchical Systems. The Computer Journal. 1967;9(4):373-380. doi:10.1093/comjnl/9.4.373

2. Gordon AD. A Review of Hierarchical Classification. Journal of the Royal Statistical Society Series A (General). 1987;150(2):119-137. doi:10.2307/2981629

3. Murtagh F. A Survey of Recent Advances in Hierarchical Clustering Algorithms. The Computer Journal. 1983;26(4):354-359. doi:10.1093/comjnl/26.4.354

4. Bouckaert R, Vaughan TG, Barido-Sottani J, et al. BEAST 2.5: An advanced software platform for Bayesian evolutionary analysis. PLOS Computational Biology. 2019;15(4):e1006650. doi:10.1371/journal.pcbi.1006650

5. Bouckaert R, Heled J, Kühnert D, et al. BEAST 2: A Software Platform for Bayesian Evolutionary Analysis. PLOS Computational Biology. 2014;10(4):e1003537. doi:10.1371/journal.pcbi.1003537

6. Bouckaert RR, Drummond AJ. bModelTest: Bayesian phylogenetic site model averaging and model comparison. BMC Evolutionary Biology. 2017;17(1):42. doi:10.1186/s12862-017-0890-6

7. Nascimento FF, Reis M dos, Yang Z. A biologist's guide to Bayesian phylogenetic analysis. Nature Ecology & Evolution. 2017;1(10):1446-1454. doi:10.1038/s41559-017-0280-x

8. Magill SS, O'Leary E, Janelle SJ, et al. Changes in Prevalence of Health Care–Associated Infections in U.S. Hospitals. N Engl J Med. 2018;379(18):1732-1744. doi:10.1056/NEJMoa1801550

9. Sundermann AJ, Miller JK, Marsh JW, et al. Automated data mining of the electronic health record for investigation of healthcare-associated outbreaks. Infection Control & Hospital Epidemiology. 2019;40(3):314-319. doi:10.1017/ice.2018.343

10. Miller JK, Chen J, Sundermann A, et al. Statistical outbreak detection by joining medical records and pathogen similarity. Journal of Biomedical Informatics. 2019;91:103126. doi:10.1016/j.jbi.2019.103126

11. Kingman JFC. On the Genealogy of Large Populations. Journal of Applied Probability. 1982;19:27-43. doi:10.2307/3213548

12. Hubert L, Arabie P. Comparing partitions. Journal of Classification. 1985;2(1):193-218. doi:10.1007/BF01908075

13. Sokal RR, Rohlf FJ. The Comparison of Dendrograms by Objective Methods. Taxon. 1962;11(2):33-40. doi:10.2307/1217208

14. Healthcare Infection Control Practices Advisory Committee. Core Infection Prevention and Control Practices for Safe Healthcare Delivery in All Settings –Recommendations of the HICPAC. Published online December 27, 2018. Accessed November 12, 2021. https://www.cdc.gov/hicpac/recommendations/core-practices.html#anchor_1556561973

15. Sundermann AJ, Babiker A, Marsh JW, et al. Outbreak of Vancomycin-resistant Enterococcus faecium in Interventional Radiology: Detection Through Whole-genome Sequencing-based Surveillance. Clinical Infectious Diseases. 2020;70(11):2336-2343. doi:10.1093/cid/ciz666

16. Sundermann AJ, Chen J, Kumar P, et al. Whole-Genome Sequencing Surveillance and Machine Learning of the Electronic Health Record for Enhanced Healthcare Outbreak Detection. Clinical Infectious Diseases. Published online November 12, 2021:ciab946. doi:10.1093/cid/ciab946

17. Atreja A, Gordon SM, Pollock DA, Olmsted RN, Brennan PJ. Opportunities and challenges in utilizing electronic health records for infection surveillance, prevention, and control. Am J Infect Control. 2008;36(3 Suppl):S37-46. doi:10.1016/j.ajic.2008.01.002

18. Magill SS, Edwards JR, Bamberg W, et al. Multistate Point-Prevalence Survey of Health Care–Associated Infections. N Engl J Med. 2014;370(13):1198-1208. doi:10.1056/NEJMoa1306801

19. Bouckaert Remco, Lemey Philippe, Dunn Michael, et al. Mapping the Origins and Expansion of the Indo-European Language Family. Science. 2012;337(6097):957-960. doi:10.1126/science.1219669

20. Müller NF, Rasmussen D, Stadler T. MASCOT: parameter and state inference under the marginal structured coalescent approximation. Bioinformatics. 2018;34(22):3843-3848. doi:10.1093/bioinformatics/bty406

21. Yun S, Yun S. Masking as an effective quality control method for next-generation sequencing data analysis. BMC Bioinformatics. 2014;15(1):382. doi:10.1186/s12859-014-0382-2

22. O'Reilly JE, Donoghue PCJ. The Efficacy of Consensus Tree Methods for Summarizing Phylogenetic Relationships from a Posterior Sample of Trees Estimated from Morphological Data. Systematic Biology. 2018;67(2):354-362. doi:10.1093/sysbio/syx086

23. Murtagh F, Contreras P. Methods of Hierarchical Clustering. CoRR. 2011;abs/1105.0121. http://arxiv.org/abs/1105.0121

24. Slayton RB, Toth D, Lee BY. Vital Signs: Estimated Effects of a Coordinated Approach for Action to Reduce Antibiotic-Resistant Infections in Health Care Facilities — United States. MMWR. 2015;64(30):826-831.

25. Hastings WK. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. Biometrika. 1970;57(1):97-109. doi:10.2307/2334940

26. Hall BG, Barlow M. Phylogenetic analysis as a tool in molecular epidemiology of infectious diseases. Ann Epidemiol. 2006;16(3):157-169. doi:10.1016/j.annepidem.2005.04.010

27. Rand WM. Objective Criteria for the Evaluation of Clustering Methods. Journal of the American Statistical Association. 1971;66(336):846-850. doi:10.2307/2284239

28. Douglas II SR. The Direct Medical Costs of Healthcare-Associated Infections in U.S. Hospitals and the Benefits of Prevention. Centers for Disease Control and Prevention; 2009.

29. Müller NF, Rasmussen DA, Stadler T. The Structured Coalescent and Its Approximations. Molecular Biology and Evolution. 2017;34(11):2970-2981. doi:10.1093/molbev/msx186

30. Barido-Sottani J, Bošková V, Plessis LD, et al. Taming the BEAST—A Community Teaching Material Resource for BEAST 2. Systematic Biology. 2018;67(1):170-174. doi:10.1093/sysbio/syx060

31. Rambaut A, Drummond AJ, Xie D, Baele G, Suchard MA. Posterior Summarization in Bayesian Phylogenetics Using Tracer 1.7. Systematic Biology. 2018;67(5):901-904. doi:10.1093/sysbio/syy032

32. Sokal RR, Michener CD. A statistical method for evaluating systematic relationships. University of Kansas science bulletin. 1958;38:1409-1438.

33. Centers for Disease Control and Prevention (U.S.), National Center for Emerging Zoonotic and Infectious Diseases (U.S.). Division of Healthcare Quality Promotion. Antibiotic Resistance Coordination and Strategy Unit ., eds. Antibiotic resistance threats in the United States, 2019. Published online 2019. https://stacks.cdc.gov/view/cdc/82532

34. Ward JH. Hierarchical Grouping to Optimize an Objective Function. null. 1963;58(301):236-244. doi:10.1080/01621459.1963.10500845

35. Nabil B, Sabrina B, Abdelhakim B. Transmission route and introduction of pandemic SARS-CoV-2 between China, Italy, and Spain. J Med Virol. Published online July 27, 2020:10.1002/jmv.26333. doi:10.1002/jmv.26333

36. Kühnert D, Stadler T, Vaughan TG, Drummond AJ. Phylodynamics with Migration: A Computational Framework to Quantify Population Structure from Genomic Data. Molecular Biology and Evolution. 2016;33(8):2102-2116. doi:10.1093/molbev/msw064

37. Pečerska J, Kühnert D, Meehan CJ, et al. Quantifying transmission fitness costs of multi-drug resistant tuberculosis. Epidemics. 2021;36:100471. doi:10.1016/j.epidem.2021.100471

38. Ruppitsch W. Defining and Evaluating a Core Genome Multilocus Sequence Typing Scheme for Whole-Genome Sequence-Based Typing of Listeria monocytogenes. doi:10.1128/JCM.01193-15

39. Mellmann A, Harmsen D, Cummings CA, et al. Prospective Genomic Characterization of the German Enterohemorrhagic Escherichia coli O104:H4 Outbreak by Rapid Next Generation Sequencing Technology. PLOS ONE. 2011;6(7):e22751. doi:10.1371/journal.pone.0022751

40. Neumann B, Prior K, Bender JK, et al. A Core Genome Multilocus Sequence Typing Scheme for Enterococcus faecalis. Journal of Clinical Microbiology. 57(3):e01686-18. doi:10.1128/JCM.01686-18

41. Henri C, Leekitcharoenphon P, Carleton HA, et al. An Assessment of Different Genomic Approaches for Inferring Phylogeny of Listeria monocytogenes. Frontiers in Microbiology. 2017;8. Accessed April 4, 2022. https://www.frontiersin.org/article/10.3389/fmicb.2017.02351