

UNIVERSITY OF TECHNOLOGY SYDNEY

**STATISTICAL METHODS FOR INFERRING
RECOMBINATION IN BACTERIAL
GENOMES**

FATEMEH-NEHLEH KARGARFARD

PhD

September 2022

UNIVERSITY OF TECHNOLOGY SYDNEY

**STATISTICAL METHODS FOR INFERRING
RECOMBINATION IN BACTERIAL
GENOMES**

FATEMEH-NEHLEH KARGARFARD

Thesis submitted in fulfillment
of the requirements for the degree of
Doctor of Philosophy

Faculty of Science

September 2022

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Fatemeh-Nehleh Kargarfard, declare that this thesis is submitted to fulfil the requirements for the award of Doctor of Philosophy in the Faculty of Science at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of the requirements for a degree at any other academic institution except as fully acknowledged within the text. This thesis is the result of a Collaborative Doctoral Research Degree program with the NSW Department of Primary Industries and the University of Technology Sydney. The Australian Research Council supported this work; linkage grants LP180100593—the Australian Centre funded this project for Genomic Epidemiological Microbiology (AusGEM).

This research is supported by the Australian Government Research Training Program.

(Fatemeh Nehleh Kargarfard)

Date: 08/09/2022

Production Note:
Signature removed prior to publication.

ABSTRACT

Homologous recombination events in bacterial genomes have wide-ranging effects on public health in society. This phenomenon is a significant factor explaining the prevalence of antimicrobial resistance. When recombination occurs in bacteria, a segment of foreign DNA is introduced into its chromosome. This evolutionary mechanism can give rise to antibiotic resistance. On the other hand, reconstructing bacteria's evolutionary history in the presence of recombination is notoriously tricky. Phylogenetic trees investigate evolutionary history and relationships between organisms, which is essential for understanding and analyzing natural processes. These trees are required tools for numerous fundamental and practical research. Recombination detection in a bacterial genome is one application of phylogenetic trees. Advanced phylogenetic inference methods, such as maximum likelihood and Bayesian inference, use probabilistic models that are computationally expensive, especially for large datasets. Detecting the boundary of recombination events and reconstructing a global phylogenetic tree illustrating the underlying evolutionary pattern of biological sequences has never been a straightforward problem. At the same time, the rapidly growing number of bacterial whole-genomes has produced an extra challenge for the computational approaches to reconstructing fast and accurate phylogenetic trees with the presence of recombination.

In this thesis, we introduce PhiloBacter, a maximum likelihood-based tool to detect recombination in bacterial genomes and account for it during phylogenetic reconstruction. Specifically, it estimates the probability of each site in an alignment to be recombinant. We then presented two approaches to incorporate these probabilities to infer the clonal history of these genomes. The first borrows ideas from sequencing error estimation, and the other uses mixtures of matrices to account for uncertainty introduced through recombination.

We also present a new simulation tool, BaciSim, for bacterial genomes that undergo recombination.

Finally, we developed a software pipeline for the semi-automatic identification of recombination and reconstruction of a phylogenetic tree from an alignment bacterial genome. Using simulated datasets, we investigated the accuracy and reliability of our approach to detect recombination events and to get better estimates of the clonal history of a collection of genomes that underwent recombination. We benchmarked our methods with other widely used methods (Gubbins and ClonalFrameML). Our simulations show that PhiloBacter tends to outperform these two methods.

ACKNOWLEDGEMENT

First, I would like to express my sincere thankfulness to both of my supervisors, Professor Aaron Darling and Dr. Mathieu Fourment, for the scientific and non-scientific lessons they taught me. I'm proud of myself for working under their supervision. Their valuable advice, continuous encouragement, and constant and kind support were critical throughout the development of my research work. Without their help and support, this project would not have been possible.

I am also grateful to all team members for their technical and emotional support: Sid Krishnan, Frederick Jaya, Dr. Daniela Gaio, Dr. Barbara Brito-Rodriguez, Dr. Kay Anantanawat, Dr. Leigh Monahan, and Dr. Matthew Macaulay.

I am sincerely thankful to Dr. Mehrad Hamidian for all his valuable advice and support.

I also thank all the Faculty and institute staff for administratively helping me during these years of my study.

Beyond work, I would like to thank my parents and sister for their consistent support and encouragement during all moments of my life. Their love was always close to me, even if they were geographically thousands of kilometers away.

I am sincerely thankful to all my friends for being in my life since I came to Australia. They always encouraged me during difficult times; my life wasn't so colorful without their presence.

I must also sincerely and deeply thank my partner. His companionship and love were the best encouragement during the last months of my studies.

Lastly, I genuinely acknowledge the organizations that provided financial support for this research, including the Australia Research Council and the Australian Centre for Genomic Epidemiological Microbiology (AusGEM).

THESIS FORMAT

This is a conventional thesis consisting of 6 chapters. This thesis aims to develop statistical methods for inferring recombination in bacterial genomes.

Chapter 1 is an introduction chapter, including the research background, motivation, aim, and objectives.

Chapter 2 is a literature review of conventional methods for inferring phylogenetic trees and recombination detection methods.

Chapter 3 is about general methods in this research work which introduce PhiloBacter.

Chapter 4 presents a new simulation tool for bacterial genomes that undergo recombination.

Chapter 5 is Result and Discussion.

Chapter 6 summarizes the main research outcomes of this thesis and possible future research directions that may arise based on the advances made.

TABLE OF CONTENTS

	Page
CERTIFICATE OF ORIGINAL AUTHORSHIP	i
ABSTRACT	ii
ACKNOWLEDGEMENT	iii
Thesis Format	iv
TABLE OF CONTENTS	v
LIST OF TABLES	x
LIST OF FIGURES	xi
CHAPTER ONE: INTRODUCTION	1
1.1 Research Motivation	1
1.2 Recombination	3
1.2.1 Homologous recombination in bacteria	4
1.3 What is a phylogenetic tree?	8
1.3.1 Mathematical definition of phylogenetic tree	9
1.3.2 Computer format of phylogenetic tree	10
1.4 Impact of recombination on phylogenetic analyses	12

1.5	Research Objectives and Contributions	13
1.6	Thesis Structure	14
CHAPTER TWO: LITERATURE REVIEW		15
2.1	Methods for inferring phylogenetic trees	15
2.1.1	Distance-based Approaches:	15
2.1.1.1	UPGMA and WPGMA	16
2.1.1.2	Neighbor-Joining	16
2.1.1.3	Maximum parsimony (MP)	16
2.1.2	Modeling Nucleotide Evolution	17
2.1.2.1	JC69 model	19
2.1.2.2	Some other models	20
2.1.2.3	GTR model	20
2.1.3	Statistical Approaches	21
2.1.3.1	Maximum Likelihood Method	21
2.1.3.2	Bayesian analysis using MCMC	24
2.2	Detection of recombination events in bacterial genomes	26
2.2.1	Phylogenetic-based recombination detection methods	28
2.2.1.1	ClonalFrame and ClonalFrameML	29
2.2.1.2	ClonalOrigin	30
2.2.1.3	Gubbins	30
2.2.1.4	Bacter	31
2.3	Summary	31
CHAPTER THREE: RESEARCH METHODOLOGY: PhiloBacter		33

3.1	Introduction to HMM	33
3.1.1	Basic Understanding of Markov Model	34
3.1.2	Hidden Markov Model (HMM)	36
3.1.3	What kind of problems can be resolved through the Hidden Markov Model?	38
3.2	Incorporating Sequence Uncertainty in Phylogenetic Inference	40
3.2.1	Uncertainty models in DNA sequences	42
3.3	PhiloBacter: A new tool to infer phylogenetic trees from recombinant bacterial genomes	43
3.3.1	Overview	43
3.3.2	Step one: Recombination Estimation	44
3.3.2.1	Description of algorithm	44
3.3.2.2	Lesson learned	52
3.3.3	Step two: Clonal tree Inference using whole genomes	55
3.3.3.1	PhiloBacter: Maximum Likelihood Calculation	56
3.3.3.2	PhiloBacter: Uncertainty in clonal tree Inference	58
3.4	Summary	62
CHAPTER FOUR: RESEARCH METHODOLOGY: BaciSim		64
4.1	Introduction	64
4.2	Background	65
4.3	Methods	67
4.3.1	Overview of BaciSim Simulator	68
4.4	Discussion	70
CHAPTER FIVE: RESULTS AND DISCUSSIONS		73

5.1	Pipeline introduction	73
5.1.1	Installing the pipeline	73
5.1.1.1	Prerequisites:	73
5.1.1.2	Installation Steps:	74
5.1.2	Analysis mode	75
5.1.3	Simulation mode	76
5.2	Evaluation Metrics	78
5.2.1	Evaluating Recombination Detection	78
5.2.2	PhiloBacter as a regression model	79
5.2.3	PhiloBacter as a classification model	79
5.2.4	Evaluating Phylogenetic reconstruction	81
5.3	Performance Evaluation	82
5.3.1	BaciSim Simulator	82
5.3.1.1	Investigating different values of v	82
5.3.1.2	Investigating different recombination lengths	86
5.3.1.3	Investigating different recombination rate	88
5.3.1.4	Investigating different tMRCA	89
5.3.2	SimBac Simulator	91
5.3.3	FastSimBac Simulator	93
5.4	Comparison with fastGEAR	95
5.4.1	fastGEAR Simulated Data	96
5.4.2	BaciSim Simulator	98
5.4.2.1	Detecting Recent Recombinations	98
5.4.2.2	Detecting Ancestral Recombinations	99

5.4.2.3	Detecting All Recombinations	101
5.5	Application to empirical data	101
5.5.1	Application to <i>Streptococcus pneumoniae</i>	101
5.5.2	Application to <i>Staphylococcus aureus</i>	106
5.5.3	Application to <i>Bacillus Cereus</i>	108
5.6	Resource Usage	111
5.7	Discussion	116
CHAPTER SIX: Conclusion and future directions		119
6.1	Conclusion	119
6.2	Perspectives and Future Research	121
6.2.1	PhiloBacter as a fast and more efficient tool	121
6.2.2	Generalised uncertainty approach	122
6.2.3	Internal and external nodes	123
6.2.4	A dynamic and general simulator	124
6.2.5	PhiloBacter and quantum computing	125
REFERENCES		126
APPENDICES		145

LIST OF TABLES

Tables	Title	Page
Table 5.1	Summary of recent recombination (leaves) detection for three selected lengths (500, 1000, and 1500) using BaciSim, fastGEAR, Gubbins, CFML, and PhiloBacter. The simulated data were generated using the following settings: v : 0.03, Number of genomes: 10, Alignment length: 100K, and tMRCA: 0.01. The left column indicates the number of simulated recombination events, with the 'Anc' columns set to zero to represent the absence of ancestral recombination in the simulated data. Column 'Rec' shows the number of recent recombination events.	99
Table 5.2	Summary of ancestral recombination (internal nodes) detection for three selected lengths (500, 1000, and 1500) using BaciSim, fastGEAR, Gubbins, CFML, and PhiloBacter. The simulated data were generated using the following settings: v : 0.03, Number of genomes: 10, Alignment length: 100K, and tMRCA: 0.01. The left column indicates the number of simulated recombination events, with the 'Rec' columns set to zero to represent the absence of recent recombination in the simulated data. Column 'Anc' shows the number of ancestral recombination events.	100

LIST OF FIGURES

Figures	Title	Page
Figure 1.1	A schematic view of transduction in which external DNA is entered into a bacterial cell by a virus (Image modified from “Conjugation,” by Adenosine (CC BY-SA 3.0). The modified image is licensed under a CC BY-SA 3.0 license.)	5
Figure 1.2	A schematic view of transformation in bacteria in which some bacteria take up naked DNA from the environment (Image modified from “Conjugation,” by Adenosine (CC BY-SA 3.0). The modified image is licensed under a CC BY-SA 3.0 license.)	7
Figure 1.3	A schematic view of transformation in bacteria in which one bacterial cell transfers genetic material to the other cell through straight contact (Image modified from “Conjugation,” by Adenosine (CC BY-SA 3.0). The modified image is licensed under a CC BY-SA 3.0 license.)	8
Figure 1.4	The topology of all trees is the same. (a) Rooted phylogenetic tree - Cladogram (b) Rooted phylogenetic tree - Phylogram (c) Unrooted phylogenetic tree.	10
Figure 2.1	An example of unrooted tree topology for four taxa with two internal nodes [71].	22
Figure 3.1	Representation of a Markov Chain with two states: This shows a Markov Chain with states S_0 and S_1 , and the arrows indicate the probability of departing from one state to the other state or remaining the same [114]	35
Figure 3.2	A graphical representation of an HMM	38

Figure 3.3	Overview of the PhiloBacter process, illustrating (Step 1) estimation of recombination events and (Step 2) phylogenetic inference accounting for the presence of recombination.	44
Figure 3.4	Schematic representation of the PhiloBacter HMM framework. This model encompasses five core components: (1) Hidden State Sequence, (2) Start Probability, (3) Transition Probability, (4) Emission Probability, and (5) Observation Sequence,	45
Figure 3.5	Illustration of the bi-state HMM approach utilized in this study for recombination state identification. The red node exemplifies a target node, highlighting the hypothesis that recombination amplifies polymorphism within each branch, subsequently elongating it.	46
Figure 3.6	Example of the HMM model observation for three different nodes, including external and internal nodes.	47
Figure 3.7	Depiction of the Four-State HMM framework. The red node serves as a representative example of the target node within the model. State 1 shows ClonalFrame, which didn't undergo any recombination.	53
Figure 3.8	Eight trees indicate the hidden states of HMM model. The red node serves as a representative example of the target node within the model. State 1 shows ClonalFrame, which didn't undergo any recombination.	55
Figure 4.1	A collection of clonal and local trees which simulates various evolutionary schemes for every segment of the genomes might represent mosaic evolutionary histories.	68
Figure 4.2	Overview of simulation studies. A flow diagram of the steps in a simulation study [130].	69

Figure 4.3	An example of the graphic output of BaciSim shows recombination events, their location and length along the alignment.	72
Figure 5.1	A schematic view of the pipeline modes. a) shows the pipeline has two modes; b) shows the steps of the analysis mode.	75
Figure 5.2	A schematic view of the pipeline. Detailing its five primary stages: (1) Simulation, (2) Sequence Generation, (3) Initial Tree Construction, (4) Recombination Detection and Tree Inference, and (5) Analysis.	78
Figure 5.3	RMSE for different value of ν of three methods: ClonalFrameML, Gubbins, PhiloBacter - Number of genome:10 - Alignment length:100K - tMRCA:0.01 - Recombination length:500	83
Figure 5.4	Accuracy for different values of ν for three methods: ClonalFrameML, Gubbins, PhiloBacter - Number of genome:10 - Alignment length:100K - tMRCA:0.01 - Recombination length:500	83
Figure 5.5	F1-Score for different values of ν for three methods: ClonalFrameML, Gubbins, PhiloBacter - Number of genome:10 - Alignment length:100K - tMRCA:0.01 - Recombination length:500	84
Figure 5.6	BaciSim Simulator: Distances between true (clonal) tree and trees of three methods: ClonalFrameML, Gubbins, PhiloBacter - Number of genome:10 - Alignment length:100K - tMRCA:0.01- Recombination rate:0.01 - Recombination length 500	85
Figure 5.7	BaciSim Simulator: Investigating how accurate each method (PhiloBacter, Gubbins and CFML from left to right) can estimate different intervals of recombination length. Number of genome:10 - Alignment length:100K - nu:0.05- tMRCA:0.01	87

- Figure 5.8 **BaciSim Simulator:** Distances between true (clonal) tree and trees of three methods **for different recombination rate:** ClonalFrameML, Gubbins, PhiloBacter- Number of genome:10 - Alignment length:100K - nu:0.05- tM-RCA:0.01 - Recombination length 500 89
- Figure 5.9 **BaciSim Simulator:** Distances between true (clonal) tree and trees of three methods **for different values of tMRCA:** ClonalFrameML, Gubbins, PhiloBacter - Number of genome:10 - Alignment length:100K - nu:0.05- Recombination rate:0.01 - Recombination length 500 91
- Figure 5.10 **SimBac Simulator:** Distances between true (clonal) tree and trees of three methods: ClonalFrameML, Gubbins, PhiloBacter - Number of genome:10 - Alignment length:100K- Recombination rate:0.0005 - Recombination length:500 93
- Figure 5.11 **FastSimBac Simulator:** Distances between true (clonal) tree and trees of three methods: ClonalFrameML, Gubbins, PhiloBacter - Number of genome:10 - Alignment length:100K - Recombination rate:0.0005 - Recombination length: 500 95
- Figure 5.12 The figure visually represents the population's genetic structure inferred by fastGEAR. In this illustration, the rows are aligned with the sequences, the columns correspond to specific positions within the alignment, and various colors depict different populations. 97
- Figure 5.13 Analysis of the PMEN1 genome alignment with Gubbins employing different phylogeny construction strategies [87] 103
- Figure 5.14 Analysis of the PMEN1 genome alignment with PhiloBacter. a) PhiloBacter tree, b) CFML tree 104
- Figure 5.15 Analysis of the PMEN1 genome alignment with CFML. Vertical bars indicate recombination events detected by the CFML. 105

Figure 5.16 Analysis of the PMEN1 genome alignment with PhiloBacter. Vertical bars indicate recombination events detected by the analysis.	107
Figure 5.17 The analysis of <i>Bacillus cereus</i> strains isolated in Australia: a) Tree representation using Gubbins, b) CFML-based phylogenetic tree, and c) PhiloBacter tree analysis.	110
Figure 5.18 The analysis of <i>Bacillus cereus</i> strains isolated in Australia with PhiloBacter. Vertical bars indicate recombination events detected by the analysis.	112
Figure 5.19 Resource usage comparison chart showcasing CPU consumption across various tools: PhiloBacter, CFML, Gubbins (Gubbins-result is our custom script for output manipulation), and RAxML.	113
Figure 5.20 Resource usage comparison chart showcasing RAM consumption across various tools: PhiloBacter, CFML, Gubbins (Gubbins-result is our custom script for output manipulation), and RAxML.	114
Figure 5.21 Comparative chart of job duration, illustrating the processing times for PhiloBacter, CFML, Gubbins (enhanced with our tailored script -Gubbins-result- for output refinement), and RAxML.	115
Figure 5.22 I/O resource usage comparison chart, displaying results for PhiloBacter, CFML, Gubbins (modified using our custom script -Gubbins-result- for output optimization), and RAxML.	116
Figure A.1 BaciSim Simulator: This figure is the same as Figure 5.3. We removed Gubbins's data to clarify the differences between CFML and PhiloBacter(s).	147
Figure A.2 BaciSim Simulator: This figure is the same as Figure 5.4. We removed Gubbins's data to clarify the differences between CFML and PhiloBacter(s).	148

Figure A.3	BaciSim Simulator: This figure is the same as Figure 5.5. We removed Gubbins's data to clarify the differences between CFML and PhiloBacter(s).	149
Figure A.4	SimBac Simulator: This figure is the same as Figure 5.10. We removed Gubbins data to clarify the differences between CFML and PhiloBacter.	150
Figure A.5	SimBac Simulator: This figure is the same as Figure 5.11. We removed Gubbins data to clarify the differences between CFML and PhiloBacter.	151
Figure A.6	SimBac Simulator: This figure is the same as Figure 5.12. We removed Gubbins data to clarify the differences between CFML and PhiloBacter.	152
Figure A.7	Details of Australian genomes used in section 5.5.3 Application to <i>Bacillus Cereus</i> .	153

CHAPTER ONE

INTRODUCTION

1.1 Research Motivation

The importance of recombination as a widespread evolutionary force to make diversity among organisms is not deniable. While mutation serves as the primary source of genetic variation, recombination takes center stage in introducing substantial evolutionary leaps, yielding novel combinations of alleles. Numerous species have undergone significant recombination events throughout their evolutionary journey. Even in organisms like bacteria, which primarily reproduce clonally, recombination remains a pivotal factor in their molecular evolution.

Phylogenies are crucial in addressing a wide range of biological questions, such as unraveling the relationships between organisms and genes, investigating the origins of emerging infectious diseases, and comprehending population dynamics and species migration patterns. Nevertheless, recombination significantly influences the construction of phylogenetic trees, affecting almost all aspects of tree parameters [1]. Traditional phylogenetic methods overlook recombination since they assume all loci in a multiple sequence alignment share a uniform evolutionary history. However, bacteria frequently exchange genetic information through recombination, swapping genome segments among organisms and altering their evolutionary histories.

Many studies have shown that bacterial recombination is a major factor explaining the prevalence of Antimicrobial resistance (AMR) [2]. Antibiotics are the primary medicines prescribed to prevent and fight bacterial infections. AMR happens when bacteria no longer respond to these important medicines. This phenomenon is currently one of the most severe health is-

sues facing humans and all living creatures. Currently, AMR causes the deaths of 700,000 people per year, equaling the impact of cancer; it is predicted that the AMR death rate will rise to over 10 million annually by 2050. In the Western Pacific region alone, the economic impact of AMR over the next ten years is estimated to exceed 1.35 trillion US dollars [3]. Antibiotic resistance can impact anyone, of any age, in any country. This issue can potentially jeopardize public health even more than COVID-19, threatening to set back medicine to the 19th century [2]. When recombination occurs in bacteria, a segment of foreign DNA is introduced into the chromosome; sometimes, the little piece of DNA passed on through recombination gives bacteria the ability to fight off the effects of antibiotics. Once a resistance gene is joined to a bacterium's genome, the bacterium can influence other bacteria and give the resistance gene to all of its descendants. Bacteria multiply rapidly, and resistance is magnified. Unfortunately, no measures like social distancing can reduce the risk of AMR because bacteria live everywhere: in food, water, and air [4], [5].

Antibiotic resistance is not the only effect of recombination. Such gene acquisition, loss, and replacement which are driven by homologous recombination, frequently lead to new pathogenic strains [6], new serotypes [7], opportunistic pathogens [8], metabolic adaptations[9], [10], immunity evasion, colonization of new hosts [11], increased virulence [12] and, in general, threats to public health [13]–[15].

The capability to detect outbreaks of infectious diseases early and to track them is vital to maintaining public health [16]. The advent of Next Generation Sequencing (NGS) technology has decreased the general cost of Whole Genome Sequencing (WGS), increasing the speed of this process and facilitating the use of bacterial WGS alignment in evolutionary analyses and outbreak detection. Developing accurate and efficient statistical and computational tools from the inferences provided by the phylogenetic trees from whole-genome sequences has the poten-

tial to diminish the worldwide load of infectious diseases, building predictive tools that would upgrade our drug development, vaccines design, and treatment targetting capabilities [17].

Apart from all the mentioned phylogeny applications, another vital use is employing phylogenetic reconstruction to detect recombination events. This thesis reconstructs the phylogenetic tree by considering the presence of recombination in the bacterial genome: making a tree inference and recombination detection simultaneously. In the rest of this chapter, the basic concepts of recombination and phylogenetic trees are defined and contextualized.

1.2 Recombination

Recombination is an exchange of genetic material between different chromosomes or within a single chromosome. In practice, recombination occurs in almost all multicellular organisms and some unicellular organisms, breaking and repairing DNA strands to produce new combinations of different gene states and alleles. The effects of recombination can be advantageous when they enhance existing adaptations, but they can also be harmful when a combination of beneficial alleles is disrupted. Recombination rates differ not only amongst individuals of the same species but also in different species and populations [18].

The importance of recombination as an evolutionary force in creating diversity among organisms is undeniable. Many organisms naturally undergo substantial amounts of recombination in their population through various underlying molecular mechanisms. While mutation is the primary source of natural genetic divergence, recombination can quickly introduce large evolutionary jumps, introducing new combinations of alleles [1].

If the substituted donor sequence resembles the region in the receiver DNA strand without being identical, we say it is homologous — this procedure is named homologous recombination (HR).

1.2.1 Homologous recombination in bacteria

Recombination plays a key role in the molecular evolution of bacteria, as they are organisms that reproduce clonally. When recombination occurs in bacteria, a piece of foreign DNA is introduced into the chromosome; i.e., several nucleotides change simultaneously [19], [20]. The DNA fragments obtained this way can replace the previous homologous sequences in the genome. If the incoming DNA does not match the receptor (chromosome), it usually fails to integrate and is automatically destroyed due to non-replication. This homologous recombination in bacteria is functionally equivalent to gene conversion's mechanism and nonreciprocal nature in eukaryotic organisms [21].

Natural genetic transformation, a recombination mechanism, is frequently observed in bacterial species [22], [23]. Natural transformation is a mechanism of great significance for pathogenic bacterial categories; in some cases, an infectious bacterial pathogen's survival may rely on its ability to repair DNA through recombination [23]. Evolution in bacteria was previously believed to be based on mutation or genetic drift. Still, we now consider recombination or gene transfer a primary driving force in prokaryotes' development [24]. This driving force has been extensively investigated in organisms such as *E.coli* [22], [25].

Recombination could help spread functional genes that produce more adaptable organisms. Cells may spread functional genes by genetic recombination and help ensure the species' survival. A type of RecA enzyme typically catalyzes recombination in bacteria. These recombinase enzymes repair DNA damage by homologous recombination [26].

While bacteria do not reproduce sexually in the true sense, many or most of them can transfer DNA fragments from one bacteria cell to another by one of the following three mechanisms:

1) Transduction is the transfer of DNA from one cell to another bacteria cell through viruses [27], [28]. A virus infects a bacterium and injects a DNA fragment of the donor into the victim cell. Bacteriophage viruses hijack the molecular machinery of the bacterial cell to synthesize DNA, RNA, and proteins quickly [29]. The virus' genetic material occasionally fuses with the DNA of the host. Then, it extrudes its viral DNA from the bacterial chromosome, where the bacterial genes may be incorporated into the newly released viral DNA. The virus causes the bacterium to reproduce many virus genomes with its genes. Then, the virus ruptures the cell and releases new virions, each repeating the cycle. In this way, one host's genes are combined with another host's genes, perhaps from another species [30].

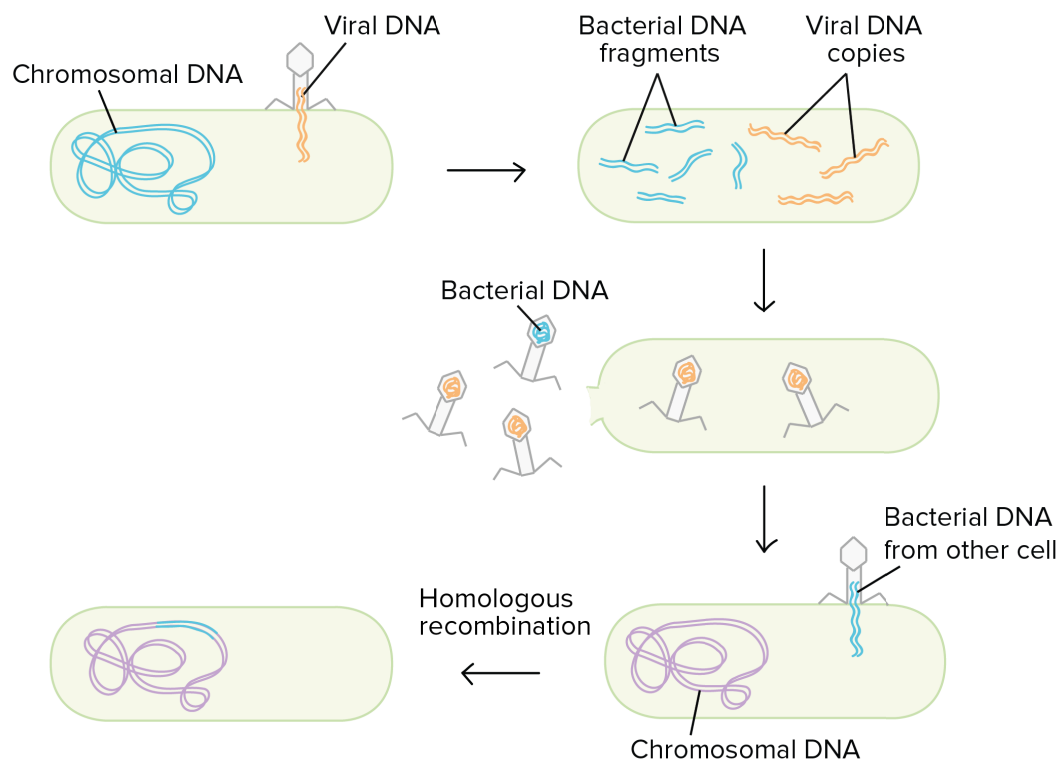


Figure 1.1 A schematic view of transduction in which external DNA is entered into a bacterial cell by a virus (Image modified from “Conjugation,” by Adenosine (CC BY-SA 3.0). The modified image is licensed under a CC BY-SA 3.0 license.)

Transduction is able to rapidly rearrange the genetic composition of bacterial populations, even if they breed asexually, with profound effects on bacteria and their habitats. This

type of gene transfer has contributed significantly to antibiotic resistance. Antibiotic resistance is a natural behavior of some bacterial cells' membranes, making it difficult for the antibiotic to bind there. Of course, this phenomenon can result from random mutation and does not affect the overall effectiveness of the antibiotic. Nevertheless, if a bacteriophage contaminates an antibiotic-resistant bacteria and then transfers the mutated gene to other cells, more bacteria will evolve resistant to the antibiotic and reproduce as with binary fission. Then the number of antibiotic-resistant cells in the population increases exponentially [30]. Transduction is summarized in Figure 1.1.

2) Transformation Some specific species of bacteria can take segments of foreign gene fragment DNA, known as plasmids, from their surroundings and insert these new segments into their chromosomes. First, the bacterium must go to a condition called "competence," which allows for "transformation." The bacterium must activate several genes expressing the required proteins to achieve this competence. Bacteria usually carry out transformations within a single species. Transformation occurs naturally and is also a feature of bacteria that scientists can use to their advantage: selected DNA can be inserted into a prokaryotic cell by placing the DNA in a culture medium, thus creating strains with desired characteristics [30]. Transformation is summarized in Figure 1.2.

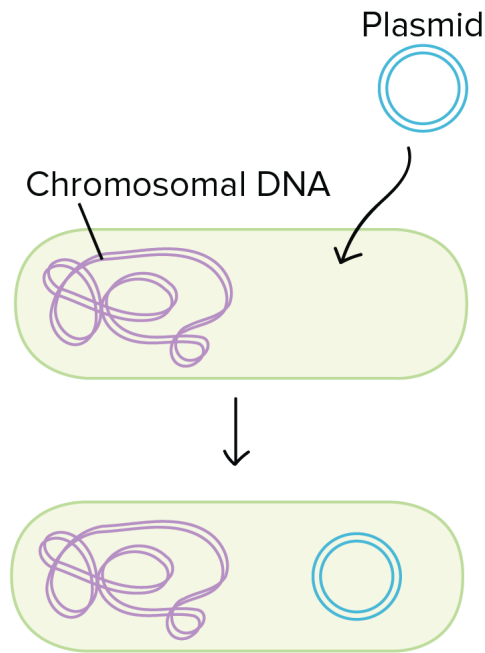


Figure 1.2 A schematic view of transformation in bacteria in which some bacteria take up naked DNA from the environment (Image modified from “Conjugation,” by Adenosine (CC BY-SA 3.0). The modified image is licensed under a CC BY-SA 3.0 license.)

3) Conjugation is the closest bacterial equivalent to intercourse. In conjugation, physical contact is made between two bacteria cells through a bridge-like structure called a pilus. The donor cell must have a small piece of DNA, named the F-plasmid, that the recipient cell lacks, which is then transferred to the recipient. The recipient’s DNA polymerase enzyme then produces a complementary strand to produce the typical double-stranded DNA structure and combines the recipient’s DNA with its genome [30]. Conjugation is summarized in Figure 1.3.

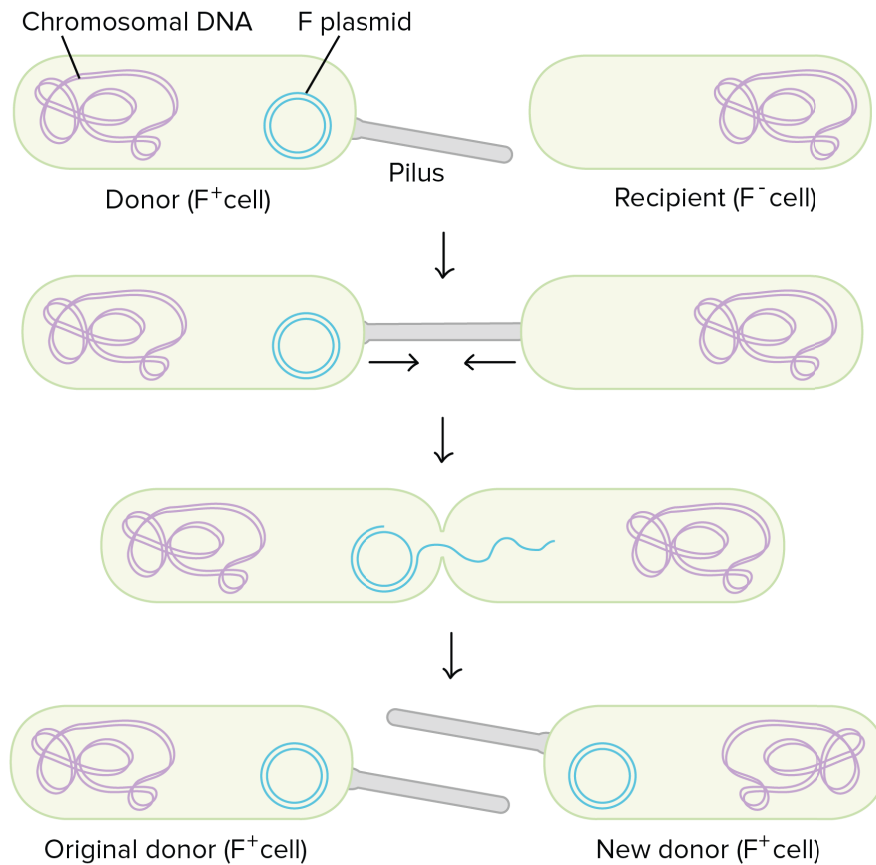


Figure 1.3 A schematic view of transformation in bacteria in which one bacterial cell transfers genetic material to the other cell through straight contact (Image modified from “Conjugation,” by Adenosine (CC BY-SA 3.0). The modified image is licensed under a CC BY-SA 3.0 license.)

1.3 What is a phylogenetic tree?

A phylogenetic tree is an elegant representation of evolutionary relationships among species, genes, populations, or even individuals similar to a pedigree. It illustrates which genes or organisms are the most related to each other.

Phylogenies are essential for working on biological problems such as determining the relationships between organisms and genes, identifying the origins of emerging infections and diseases, or modeling population transformations and migration between species. Before the emergence of DNA sequencing technologies, phylogenetics was only used to represent system-

atics and taxonomy species relationships, relying on inference methods from heritable traits and morphology. Molecular phylogeny is an exciting interdisciplinary field at the boundary of biology, statistics, and algorithmics. Today, most branches of biology use phylogenies [31], [32]. Here are just some examples of the practical application of phylogenetics: Understanding the outbreak of infectious diseases [33], like HIV and SARS [34], [35], forensics [36], [37], identifying the origin of pathogens [38], [39], drug discovery [40], [41], cancer identification and treatment [42], [43], investigation of human prehistory [44].

1.3.1 Mathematical definition of phylogenetic tree

From a mathematical perspective, phylogenetic trees are acyclic graphs composed by a set of vertexes connected by a set of edges. External nodes (tips, leaves, or terminal nodes) represent the current species (taxa); the branches are the edges. Internal nodes are also known as “hypothetical taxonomic units” (HTUs) to show that they are extinct ancestors of tips without sequence data available for them. All sequences’ most recent common ancestor is considered the tree’s root [1], [45].

A phylogenetic tree can be either rooted or unrooted. A rooted tree is a tree with the specified node as the root (Figure 1.4a), representing every taxon’s most recent common ancestor. An unrooted tree, the most recent common ancestor is unknown and therefore lacks a root node (Figure 1.4c) [45].

The degree of the node is the number of linked branches to a node. The degree of all the leaves is one. A binary tree, also known as a bifurcating tree or sometimes a fully resolved tree, is a tree that includes nodes with a degree of less than three. When non-root nodes have a degree greater than three, and root-node has a degree greater than two is called polytomy or multifurcation [45]. In this thesis, only binary trees are considered.

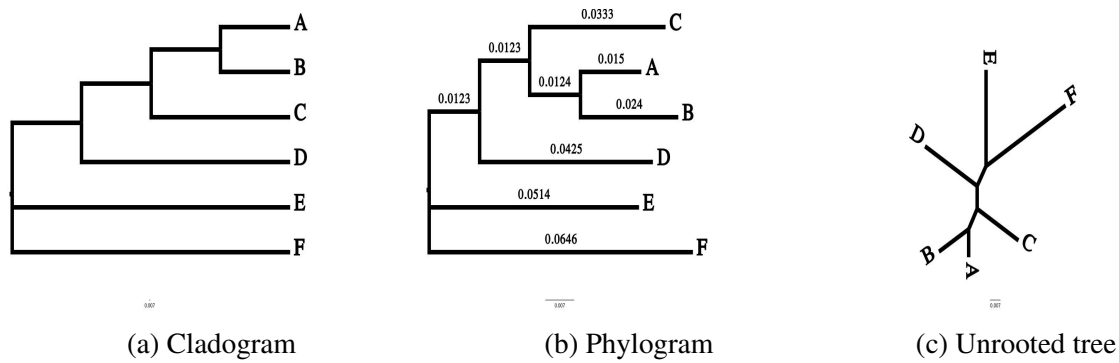


Figure 1.4 The topology of all trees is the same. (a) Rooted phylogenetic tree - Cladogram (b) Rooted phylogenetic tree - Phylogram (c) Unrooted phylogenetic tree.

Most tree reconstruction approaches cannot infer the placement of the root. However, rooted trees can be inferred using a molecular clock or non-reversible models [45]. Outgroup rooting is a method often used to root phylogenetic trees. Outgroups are designated from samples with the farthest evolutionary distance possible while still related to the nodes or species in the tree; the branch leading to the outgroup roots the tree. Midpoint rooting is another method for assigning a root to a phylogenetic tree, common in the analysis of population data [45].

Tree topology or tree structure describes the pattern of branches or the order of tree nodes [30]. A branch length represents the evolutionary distance between two nodes, where the unit can be the number substitution per sequence site or time (e.g. years). A cladogram (Figure 1.4a) is a tree topology miss-representing the branch length information, while a phylogram (Figure 1.4b) is a tree representation including both the topology and branch lengths [1], [45].

1.3.2 Computer format of phylogenetic tree

Newick tree format represents a phylogenetic tree in a computer-readable format using nested parentheses and commas. As an example, the trees in Figure 1.4 may be represented as:
a and b: (((((A, B), C), D), E), F);

b: (((A: 0.015, B: 0.024): 0.0124, C: 0.0333): 0.0123, D: 0.0425): 0.01234, E: 0.0514, F: 0.0646);

c: (((A, B), C), D, E, F);

In general, the number of branches of an unrooted tree with n species is $2n - 3$, and the total number of unrooted binary trees for n species is:

$$U_n = \frac{(2n-5)!!}{2^{n-3}(n-3)!}$$

In computer science, NP-hardness and NP-completeness are important concepts in computational complexity, which seeks to classify computational problems according to their inherent difficulty [46]. NP stands for "nondeterministic polynomial time"[47]. It represents the class of decision problems (problems with a yes/no answer) for which a solution can be verified in polynomial time given a certificate or witness (a solution). In other words, if you're given a potential solution to an NP problem, you can check if it's correct in a reasonable amount of time. NP-hard, on the other hand, stands for "nondeterministic polynomial-time hard". This class of problems is at least as hard as the hardest problems in NP. Informally, an NP-hard problem is one where it's at least as hard to find a solution as it is to verify one [48], [49].

Understanding the evolutionary history of different organisms or genes through phylogenetic reconstruction is not straightforward, and It is categorized as an NP-hard problem; indeed, the construction of phylogenetic trees falls into the category of combinatorial optimization problems, which necessitates the use of heuristics for tree-search [50], [51]. In practice, this makes the true answer unknowable unless the number of nodes is small; the sheer scale of the problem, especially when there are many species or sequences involved, means that all possibilities cannot be considered within a meaningful timeframe so that no individual solution can be verified as optimal. Despite the many methods proposed over the years, none have

yet ensured an estimated phylogenetic tree that is an exact “true” tree [1]. While finding the exact optimal solution may be challenging, efficient algorithms and techniques are available to approximate the solution and produce valuable results.

1.4 Impact of recombination on phylogenetic analyses

Conventional phylogenetic tree reconstruction methods assume that all loci of multiple sequence alignment share the same evolutionary history of substitution patterns, evolutionary rates, and tree topology. Still, the evolutionary behavior of bacteria typically violates this assumption, with trees made more complicated by recombination [52]. Recombination events produce a robust phylogenetic signal, so accounting for recombination in the phylogenetic tree inference method can simultaneously improve evolutionary analysis and recombination detection [53].

Recombination events can affect phylogenetic tree reconstruction in two ways. The first is tree topology [54]; while the branching structure of the phylogenetic tree usually remains unchanged, in some cases, it does change. The second is branch length, representing the evolutionary time between two nodes, significantly increased by a recombination event [54].

To better understand the impact of recombination on phylogenetic reconstruction, Hedge and Wilson [54] conducted simulations with different scenarios of recombination rates. Their scenario includes 1,000 populations, each consisting of 100 bacterial genomes, with a length of 1 Mb and a moderate mutation rate. They designed three scenarios: high, low, and no recombination. In each simulation, they recorded the clonal frame. They used different phylogenetic methods, namely neighbor-joining (NJ), unweighted-pair group method with arithmetic means (UPGMA), maximum likelihood (ML), and BEAST, to estimate the phylogeny. They found that the clonal frame’s topology was reconstructed with remarkable accuracy, even when

recombination was present, achieving a success rate of over 97%. However, the branch lengths of the clonal frame were not as robustly reconstructed, and their accuracy was affected by recombination. The distortion in branch lengths was more pronounced when recombining sites were removed. Therefore, In this study, we focused on branch length rather than topology for phylogenetic reconstruction.

Numerous tools for recombination detection likely point to the inherent difficulty of assessing recombination in molecular sequence data [1]. On the other hand, rapidly growing databases of bacterial whole genomes make detecting recombinations and accounting for them in bacterial phylogenetic tree inference more difficult.

1.5 Research Objectives and Contributions

Homologous recombination events lead most bacterial genomes to mosaic evolutionary histories [55]. The conventional phylogenetic tree inference can not illustrate the underlying evolutionary pattern of biological sequences. Still, a network that simulates various evolutionary schemes for every component of the genomes might infer them better.

The rapidly growing number of bacterial whole-genomes has produced a challenge for the computational approaches to reconstructing fast and accurate phylogenetic trees with the presence of recombination. Therefore, there is the main aim of the present thesis:

Proposing a new approach (*PhiloBacter*^{*}) that can tackle the problem of inaccurate phylogenetic reconstruction due to recombination by considering different evolutionary rates and substitution patterns for different genome segments.

*: *Philo* is a combining form appearing in loanwords from Greek, where it meant “love.” When studying *philosophy*, people want to understand how and why people do certain things and how to live a good life. When People use *PhiloBacter*, they want to understand

the history of the bacteria, how they evolve, and their specific characteristics (recombination) to make a better life for living creatures on this planet.

1.6 Thesis Structure

Chapter 1 introduced the background of this thesis, the research motivations, and the corresponding research objectives. Chapter 2 presents the related work of this research, including the classification of the existing methods. Chapter 3 explains the detail of the proposed method, which includes the recombination detection method and the new approach for tree inference in the presence of recombination. Chapter 4 presents a new simulator (BaciSim) for recombined bacterial genomes. Chapter 5 discusses evaluating simulated data and empirical datasets, comparison, and analysis with existing methods. Chapter 6 concludes this thesis and provides discussions of future work.

CHAPTER TWO

LITERATURE REVIEW

This chapter is an overview of the related work to this thesis topic. Section 2.1 reviews conventional methods for inferring phylogenetic trees. Then, the state-of-the-art methods for detecting recombination events in bacterial genomes are presented in Section 2.2. Finally, we briefly summarize the contents of this chapter.

2.1 Methods for inferring phylogenetic trees

Phylogenetic reconstruction methods can be categorized into two classes based on some overall features: distance-based and character-based. In the first set of approaches, a distance matrix is constructed by calculating a pairwise comparison of sequences. The subsequent analysis can use this matrix to infer the phylogenetic trees. Generally, a clustering algorithm plays the role of the converter of a distance matrix into a phylogenetic tree. UPGMA [56] (Unweighted Pair Group Method with Arithmetic mean) and NJ [57] (Neighbor-Joining) are the most popular methods in this category. Character-based approaches are another method of phylogenetic reconstruction and play a central role in the discourse of this thesis. Character-based methods directly use alignment to optimize a criterion to infer phylogeny. Maximum parsimony [58], Maximum-Likelihood [59], and Bayesian methods [60]–[62] are in this category.

2.1.1 Distance-based Approaches:

Taken together, distance-based methods can be summarised as algorithms that reduce tree size in a cyclical fashion, beginning by pairing the nearest nodes in the tree, then combining these pairs to a higher level as single nodes. In this manner, the size of the new tree is reduced,

and then the distance matrix with fewer entries is recalculated, with these steps repeated over the smaller datasets [31].

2.1.1.1 UPGMA and WPGMA

The UPGMA (Unweighted Pair Group Method with Arithmetic mean) and WPGMA (Weighted Pair Group Method with Arithmetic mean) algorithms construct rooted and ultrametric dendrograms. The latter represent equal distances from the root to every terminal node. At each step of this method, the closest two nodes are merged into a new node. In the next step, the average of two distances previously calculated is assigned as the distance between any two nodes. In molecular datasets (DNA, RNA, and protein), the ultrametricity assumption is equivalent to the molecular clock assumption that assumes contemporaneous taxa [1], [31].

2.1.1.2 Neighbor-Joining

The neighbor-joining algorithm requires an input distance matrix, including the distance between each pair of taxa. A full unresolved tree with a star network topology is the starting point for this algorithm; it calculates the Q matrix according to the initial matrix and then finds the least distant pair of nodes. These two closest nodes are converged, creating a new node on the tree; the distance of the other taxa to this new node is then recalculated, and this process is repeated until the tree topology ultimately resolves [57].

2.1.1.3 Maximum parsimony (MP)

The maximum parsimony (MP) approach looks for the tree topology with the least evolution, in other words, the shortest tree. The procedure for finding the parsimony tree contains two sub-problems:

1. Computing the number of substitutions, or tree length, for each tree topology
2. Searching for the tree topology that minimizes tree length over all possible tree topologies

On a given tree topology, the MP algorithm calculates the number of mutations for each site on each branch to reach the observed nucleotide at the tips to find the character length or site length. The sum of all branches on a tree, or the parsimony score for all nucleotides, is called a tree's parsimony length, which may be calculated for different tree topologies. After evaluating the scores of multiple topologies, the tree with the lowest parsimony score for all positions is chosen. This final tree is called the maximum parsimony tree or the most parsimonious tree[1], [45]. Computing the number of mutations on a fixed topology is straightforward and fast. On the other hand, finding the most parsimonious tree is an NP-hard problem, requiring heuristic processes since enumerating every topology is only feasible for very small datasets [1].

2.1.2 Modeling Nucleotide Evolution

Evolutionary forces affect DNA sequences leading to sequence changes over time, meaning that any two taxa with a common ancestor can evolve separately and eventually diverge; the scale measuring this divergence is known as a genetic distance. In phylogenetics, genetic distance can be represented by the length of branches between tree nodes. Therefore, if the exact genetic distance between all pairs of sequences is known, a basis for reconstructing the evolutionary tree of these sequences is available [1].

A probabilistic model can describe changes between nucleotides and is used to approximate the number of substitutions. Time-homogeneous, time-continuous stationary Markov models are typically employed for this objective [1], [63]; the Markov property is most commonly summarised in English as “given the present, the future does not depend on the past”.

This model has the usual transition matrices, parameterized by time, t . Specifically, if S_1, S_2, S_3, S_4 are the states, then the transition matrix:

$$P(t) = (P_{ij}(t)) \quad (2.1)$$

where each individual entry, $P_{ij}(t)$ refers to the probability that the state S_i will change to state S_j in time t . The state space for a nucleotide is $\Omega = \{A, C, G, T\}$. The corresponding transition matrices are based on this general matrix:

$$P(t) = \begin{pmatrix} p_{AA}(t) & p_{AG}(t) & p_{AC}(t) & p_{AT}(t) \\ p_{GA}(t) & p_{GG}(t) & p_{GC}(t) & p_{GT}(t) \\ p_{CA}(t) & p_{CG}(t) & p_{CC}(t) & p_{CT}(t) \\ p_{TA}(t) & p_{TG}(t) & p_{TC}(t) & p_{TT}(t) \end{pmatrix} \quad (2.2)$$

All of these models share the following assumptions:

- **Assumption 1:** Each nucleotide site in the sequence is usually evolving independently.

It is consistent with the main characteristic of a Markov chain that it has no memory;

- **Assumption 2:** Substitution rates are fixed over time; they have homogeneity.
- **Assumption 3:** A, C, G, and T have equilibrium (stationarity) frequencies, i.e., $(\pi_A, \pi_C, \pi_G, \pi_T)$ [1], [63].

Time reversibility A Markov process is deemed time reversible when the amount of change from state x to y and vice versa are equal; this applies even if the two states have different frequencies. This means that:

$$\pi_x \mu_{xy} = \pi_y \mu_{yx}$$

Most of the stationary processes typically used in DNA evolution are time reversible.

Nucleotide substitution models, meanwhile, are at the heart of the maximum likelihood estimation and Bayesian inference in phylogeny. These models are also required to simulate sequence data for organisms associated with a particular tree [64].

During the DNA replication process, it is likely that the polymerase incorporates a non-complementary nucleotide [1]. When a purine (A, G) replacement with a purine and a pyrimidine (C, T) with a pyrimidine occurs, it is called transition. In contrast, the change of purine to pyrimidine or vice versa is called a transversion [1]. Most nucleotide substitution models have been developed based on transition/transversion rate parameters. Here we briefly describe some of these models:

2.1.2.1 JC69 model

The simplest model for DNA sequence evolution is known as the Jukes-Cantor (JC69) [65]. JC69 is not considered a biologically realistic model, but it is a good place to start. Base frequencies ($\pi_A = \pi_G = \pi_C = \pi_T = \frac{1}{4}$) and mutation rates are assumed equal in this model. The only parameter of this model is the substitution rate μ .

$$Q = \begin{pmatrix} -\frac{3\mu}{4} & \frac{\mu}{4} & \frac{\mu}{4} & \frac{\mu}{4} \\ \frac{\mu}{4} & -\frac{3\mu}{4} & \frac{\mu}{4} & \frac{\mu}{4} \\ \frac{\mu}{4} & \frac{\mu}{4} & -\frac{3\mu}{4} & \frac{\mu}{4} \\ \frac{\mu}{4} & \frac{\mu}{4} & \frac{\mu}{4} & -\frac{3\mu}{4} \end{pmatrix}$$

2.1.2.2 Some other models

In 1980, a model with two parameters was first proposed by Kimura, called K2P or K80 [66]. The first parameter referred to the transition, and the second to the transversion rate. A year later, he introduced another model called K3ST, also known as K3P or K81 [67], which allows for three substitution types. In K81, the first parameter is the transition rate, as in K80. Still, now there are two parameters for the rate of transversions: the second parameter is designated to conserve the weak/strong properties of nucleotides, and the third is to preserve the amino/keto properties of nucleotides. In 1981, Felsenstein [59] introduced an elaboration with four parameters, calling it F81, in which the substitution rate corresponds to the stationary frequency of the target nucleotide. The HKY model [68] integrated the K81 and F81 models into a single five-parameter model in 1985.

2.1.2.3 GTR model

The general time-reversible (GTR) [69] model is the most complex "time-reversible" model where sites are modeled independently. It is a commonly selected model in phylogenetic inference. At the same time, some more complex models are not time-reversible or model dependencies between more than one site at a time. The parameters of the transition rate matrix include the base frequency vector at the equilibrium, $\Pi = (\pi_A, \pi_G, \pi_C, \pi_T)$, and 6 substitution rate parameters and the rate matrix is:

$$Q = \begin{pmatrix} -(\alpha\pi_G + \beta\pi_C + \gamma\pi_T) & \alpha\pi_G & \beta\pi_C & \gamma\pi_T \\ \alpha\pi_A & -(\alpha\pi_A + \delta\pi_C + \varepsilon\pi_T) & \delta\pi_C & \varepsilon\pi_T \\ \beta\pi_A & \delta\pi_G & -(\beta\pi_A + \delta\pi_G + \eta\pi_T) & \eta\pi_T \\ \gamma\pi_A & \varepsilon\pi_G & \eta\pi_C & -(\gamma\pi_A + \varepsilon\pi_G + \eta\pi_C) \end{pmatrix} \quad (2.3)$$

Where $\alpha, \beta, \gamma, \delta, \varepsilon$ and η are the transition rate parameters.

When the nucleotide evolution model and thus Q matrix is specified, calculating the probabilities of transition from one base to any other during the evolutionary time t , $P(t)$, by computing the matrix exponential:

$$P(t) = \exp(Qt)$$

As soon as the probabilities $P(t)$ are estimated, this equation can be employed to approximate the expected genetic distance between two sequences based on the evolutionary models specified by the Q matrix.

2.1.3 Statistical Approaches

2.1.3.1 Maximum Likelihood Method

Maximum likelihood (ML) is a method for approximating the parameters θ of a probability model (distribution) given some data Y and was introduced into molecular phylogenetics by Felsenstein in 1981 [70]. Parameters are estimated by maximizing a likelihood function L . In phylogenetics, the likelihood function is the conditional probability of the sequence alignment X given the parameters $Pr[X|\theta]$, which may include the branch lengths, the parameters of the substitution model (if any), and the tree topology:

$$\begin{aligned} L(\theta) &= Pr[X|\theta] \\ &= Pr(\text{aligned sequences}|\text{parameters}) \end{aligned}$$

Simplifying the problem requires some primary and idealistic assumptions. First, dif-

ferent sites of the sequence evolve independently and with a similar evolutionary rate. Second, evolution in one taxon is independent of another taxon [1], [45]. Based on these assumptions, calculating the probability of the input-aligned sequences turns into calculating the product of the probabilities of single sites [45]:

$$L(\theta) = f(X|\theta) = \prod_{i=1}^n f(X_i|\theta) \quad (2.4)$$

where X_i is the i th site in a sequence alignment.

In order to avoid numerical underflow during optimization of the likelihood function, phylogenetic programs usually maximize the log-likelihood instead:

$$l(\theta) = \log(L(\theta)) = \sum_{h=1}^i \log\{f(X_i|\theta)\} \quad (2.5)$$

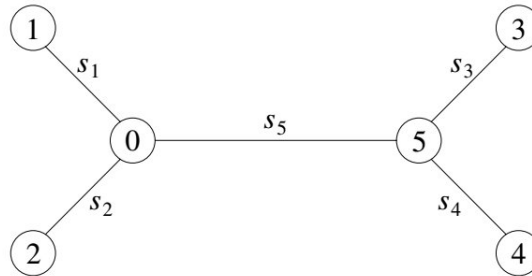


Figure 2.1 An example of unrooted tree topology for four taxa with two internal nodes [71].

Unfortunately, there is no simple and straightforward analytical formula to calculate the maximum likelihood parameters for phylogenetic trees, so the following explanation is based on a simplified example, from which the procedure for evaluating more complex trees can be simply extrapolated. Accordingly, Figure 2.2 represents a tree for $n = 4$ sequences 1, 2, 3, 4. Assuming that the model is time-reversible, the evolution can start at any node as a tree root for the computation of its likelihood (Pulley Principle [70]). So we choose node "0" as a starting

point and proceed along the branches to generate the sequence $x = (x_1x_2x_3x_4)$. If the state of internal nodes 0 and 5 are known (i.e., x_0, x_5), then

$$P(x|\theta, x_0, x_5) = P_{x_0x_1}(s_1)P_{x_0x_2}(s_2)P_{x_0x_5}(s_5)P_{x_5x_3}(s_3)P_{x_5x_4}(s_4)$$

Where $P_{z_u z_v}(s)$ gives the probability of substituting nucleotide z_u to nucleotide z_v while S substitution happens along the branch which starts at u and ends at v . Generally, the internal nucleotides are unknown; At the same time, it is necessary to know the ancestral states, the solution to overcome this problem is to consider all possible states for each internal node.

$$P(x|\theta) = \sum_{x_0 \in A} \sum_{x_5 \in A} \pi_{x_0} P(x|\theta, x_0, x_5)$$

where $A=\{A,C,T,G\}$ and $(\pi_A, \pi_C, \pi_G, \pi_T)$ is the stationary distribution of the evolutionary model.

Based on this example, now is the time to state the process of finding conditional likelihoods as a general formula by starting at a node whose only descendants are leaves. So for any node x , with two leaves y and z , we can compute $L_S^{(x)}$:

$$L_S^{(x)} = \sum_{s=1}^4 [(\sum_{s_y=1}^4 P_{S_x S_y}(v_y) L_{S_y}^{(y)}) (\sum_{s_z=1}^4 P_{S_x S_z}(v_z) L_{S_z}^{(z)})] \quad (2.6)$$

And for the leaves:

$$L_{S_y}^{(x)} = \begin{cases} 1 & \text{if } S_y = x_i \\ 0 & \text{otherwise} \end{cases}$$

Once conditional likelihoods are computed for all $n - 2$ internal nodes with two offspring, we can consider those nodes as new nodes and then use their conditional likelihoods

to calculate the likelihoods of their ancestor nodes. So, the process of calculating conditional likelihoods and replacing the new nodes to find conditional likelihoods of their ancestor nodes proceed upward until we reach the tree's root. At the root, node 0, of the tree, the conditional likelihood $L_{S_0}^{(0)}$ will be calculated. Then, the overall likelihood of the entire model is

$$L = \sum_{s_0 \in A} \pi_{s_0} L_{S_0}^{(0)} \quad (2.7)$$

Note that π_{s_0} is the base frequency for the root of the tree and gives us the probability that the root is in state S_0 [1], [31], [45], [71], [72].

2.1.3.2 Bayesian analysis using MCMC

Bayesian phylogenetic methods revolutionized genomic sequence data analysis when they were first presented in the 1990s. Bayesian statistics utilize probability distributions to quantify the uncertainty of all unknown parameters. The distribution of unknown parameters before observing the data is known as the prior distribution. When the prior has been updated with information provided by observed data, the resulting distribution is called "posterior distribution." In other words, the posterior distribution includes prior knowledge and data information, forming Bayesian inferences together[73].

Given X as a set of aligned DNA from n taxa. We aim to infer the phylogeny of these taxa and estimate the other parameters, such as the substitution model parameters. Let τ denote the tree topology, b represents the tree branch lengths, and θ contains the parameters in the evolutionary model, such as rate ratio. Let $f(\tau, b, \theta)$ define the prior and $f(X|\tau; b; \theta)$ the likelihood. Therefore, the posterior distribution based on the Bayesian formula is then:

$$f(\tau, b, \theta | X) = \frac{f(X | \tau, b, \theta) f(\tau, b, \theta)}{m(x)} \quad (2.8)$$

Where $m(x)$ the marginal probability distribution of the data can be calculated based on the summation over all parameters:

$$m(x) = \sum_{\tau} \int_{\beta} \int_{\phi} f(X | \tau, b, \theta) f(\tau, b, \theta) d(\theta) d(b) \quad (2.9)$$

In statistics, $m(x)$ is a normalizing constant that guarantees posterior distribution is a proper statistical density and integrates to 1. Typically, in Bayesian phylogenetics, θ is independent of τ and b whereas ϕ and β denote the parameter space for θ and b respectively [73].

In Bayesian phylogenetic inference, because of the curse of dimensionality of unknown parameters, it is not tractable to estimate $m(x)$ easily, except for the tiny number of taxa. MCMC can use Bayesian inferences without the computation of normalizing constant [74].

Markov chain Monte Carlo (MCMC) In molecular phylogenetic reconstruction, calculating prior and likelihood is not too hard. To compute the marginal probability, we must calculate the summation over all possible tree topologies, integration over branch lengths in those trees, and all variables in the evolutionary substitution model. Markov chain Monte Carlo (MCMC) algorithms present a robust approach to performing Bayesian computation to overcome infeasible calculation. The most significant property of the Markov chain is its convergence towards a stationary state irrespective of the starting point. The Markov chain Monte Carlo (MCMC) is applied in Bayesian statistics by generating a set of valid instances from the posterior parameter space to approximate unknown parameters. By setting up a Markov chain with the posterior distribution as its equilibrium distribution, samples of the target distribution can be obtained

from different chain states. The chain is launched at an arbitrary point and continues until it converges onto ideal distribution, with all chain steps including a proposal not far from the current state [1]. A schema of an MCMC algorithm can be described in the following steps:

1. Start with arbitrary topology, branch lengths, and evolutionary model parameters (τ , b , θ).
2. Iterate the following steps:
 - Using NNI, SPR, or TBR as tree rearrangement algorithms, make a change to the tree by generating new trees through the neighborhood of the current tree. Note that branch lengths b may also be changed in this step.
 - Make changes in branch lengths b .
 - Make changes in substitution parameters θ .
 - After any k iteration, record the values of τ , b , and θ to the disk to sample the chain.
3. After the run termination, make a summary of the results.

If the MCMC iteration were long enough, the equilibrium distribution of states in the chain would be an excellent approximation to the posterior distribution [63].

2.2 Detection of recombination events in bacterial genomes

Homologous recombination (HR) detection methods can be categorized according to what they accomplish [75]:

1. Revealing the presence of recombination in the alignment of sequences
2. Detecting the mosaic pattern behind the sequences

3. Identifying breakpoint positions

4. Estimating recombination rates [1]

Detecting the mosaic pattern behind sequences and identifying breakpoint positions are typically accomplished by uncovering distinguishable local similarities from a subset of aligned sequences [1]; these HR detection methods may also be advanced by finding particular sites responsible for phylogenetic incongruences. Population genetics fundamentals and phylogenetic analysis make it possible to estimate recombination rates [1]. All of these HR detection methods usually reveal the presence of recombination in the alignment of sequences.

When characterizing recombination detection methods statistically, we can divide them into parametric and non-parametric approaches [1]. Parametric approaches estimate population parameters through a sample based on a coalescent theory. The alternate approach relies on non-parametric statistics deduced from sequence alignments and/or tree topology [1].

Reconstructing ancestral recombination graphs (ARGs) form a distinct category that incorporates components from all the methods cited above and represents individual recombination events supported by population statistics.

Non-parametric approaches can be broken down into five algorithmic subsets as follows:

- **Similarity approaches:** To uncover gene conversion, these methods explore anomalous identity in variable pieces of the genome [76].
- **Distance approaches:** These methods use a sliding window technique to identify local differences between genomes [77]; for example, RDP4 [78]/RDP5 [79].

- **Compatibility approaches:** These methods employ phylogenetic signals of each alignment site and do not need the phylogeny itself [80], [81]; for example, ptACR [81].
- **Substitution distribution approaches:** These methods categorize sequences using similar substitution properties patterns and their comparison with the estimated model distribution parameters [82]; for example, BratNextGen [83], fastGEAR [84], HREfinder [85].
- **Phylogenetic approaches:** These methods are founded on differences between phylogenetic trees and are the class most relevant to this thesis [86], [87].

2.2.1 Phylogenetic-based recombination detection methods

One of the valuable strategies to detect recombination events is phylogenetic networks. As mentioned in Chapter 1, recombination events were conducted to intermix between evolutionally distant organisms, and therefore, a conventional inference of the tree could not reflect the true phylogenesis. Phylogenetic networks could suitably visualize genetic exchange and be divided into explicit and implicit [88]. Explicit networks are most attractive because of their interpretability, and they include information about ancestors and recombinants. Although, they are computationally expensive since many recombination events do not emit strong signals to differentiate them from mutations, especially when their targets are conservative genes [89]. Conversely, implicit networks depict the most inconsistent clades where tree topology is affected. These networks could illustrate alternative evolutionary scenarios which can verify with other procedures. When potential signals are discovered, detecting breakpoints and recombinant sequences become feasible.

The other exciting class of phylogenetic mechanisms is clonal models [20], [87], [90], [91]. These methods monitor for whole-genome sequences and then reconstruct phylogeny

employing conservative positions within housekeeping genes. Those genes define a clonal frame conducting genuine relationships between different clonal groups.

Currently, popular existing methods that consider recombination in phylogenetic tree construction include ClonalFrame [90] and ClonalFrameML [91], ClonalOrigin [20], Gubbins [87], and Bacter [53]. These methods have used the concept of a "clonal frame" [92] for each tree branch, the portion of the genome that has not changed due to recombination.

Since our new method in this thesis belongs to the clonal category, in the rest of this section, we include an overview of the most well-known tools of this group:

2.2.1.1 ClonalFrame and ClonalFrameML

ClonalFrame [90], implemented by a Bayesian Monte Carlo Markov chain (MCMC), performs phylogenetic inference based on the point mutations in the clonal frame. At the same time, identify recombinations as contiguous sites with a meaningfully high density of polymorphisms. This method assumes that recombination events occur with a novel, constant and unknown rate of substitutions in the contiguous region of the sequence; however, the origins of these events are not modeled, and for this reason, it often underestimates the number of recombination events that have happened.

ClonalFrameML [91] is the maximum likelihood version of ClonalFrame. At the same time, this new software has not been limited by MCMC convergence issues and is much faster in analyzing much larger genomic sequences. ClonalFrameML (CFML) employs an initial phylogenetic tree and computes the probability of encountering recombination for each loci utilizing maximum likelihood. The input of CFML is an alignment of bacterial sequences in FASTA format and an initial phylogenetic tree. This initial tree is usually inferred using fast maximum likelihood methods such as IQ-TREE [93], [94], which do not consider recombi-

nation. The first step in CFML is to reconstruct ancestral sequences for internal nodes of the initial tree and any missing base in the alignments. Recombination parameters and branch lengths of the clonal tree are estimated using the Baum-Welch [95] algorithm, a particular case of the expectation-maximization algorithm. Each site's clonal/recombination status is inferred using the Viterbi algorithm [96], [97]. The authors use the bootstrapping method to quantify the uncertainty of the parameters.

2.2.1.2 ClonalOrigin

ClonalOrigin [20] is similar to ClonalFrame, and the only difference is that the model explicitly incorporates each recombination's origin as a point on the clonal tree. It makes the recombination detection process in ClonalOrigin more accurate than ClonalFrame from the ancestral recombination graph (ARG) model. It assumes that the ARG is a tree-based network, and the clonal frame is its base tree. Similar to ClonalFrame and ClonalFrameML, the rate of mutation and recombination are considered constant throughout the genome that may not always be appropriate.

2.2.1.3 Gubbins

Gubbins [87] is an algorithm that uses an iterative process to detect horizontal sequence transfer in a bacterial genome and simultaneously reconstruct a maximum likelihood phylogeny based on the point mutations on the clonal frame. The regions containing high densities of base substitutions are considered recombinant regions. The underlying mechanism of recombination is not essential for Gubbins. It detects the potential recombination regions in large datasets of bacterial genomes and eliminates those SNPs (single nucleotide polymorphisms) from alignments to reconstruct a more accurate phylogenetic. Although, it was observed that removing

recombining sites exacerbates branch length distortion [54].

Gubbins demonstrates an increased substitution rate among ML-tree branches. The input is alignment in FASTA format. It can detect Recent and ancestral events [75].

2.2.1.4 Bacter

Bacter [53] is based on the model implemented in ClonalOrigin [20]. It also reconstructs ARGs and utilizes a novel Bayesian Monte Carlo Markov chain (MCMC) algorithm to simultaneously infer evolutionary relationships, homologous gene conversion, and overall conversion rate. Bacter is based on the single-step process that enhances detection accuracy and decreases the uncertainty if the phylogenetic signal is poor. The authors show that Bacter uncovers gene flow between pathogenic and non-pathogenic E-coli serotype O157 representatives, previously undetected [53]. Since Bacter is implemented in BEAST2, one can build complex models, although it tends to be too computationally demanding for large genomic datasets. The limitation of this tool is because of many parameters to be optimized and insufficient throughput [75].

2.3 Summary

Standard phylogenetic tree inference methods (regardless of their class, i.e, distance-based or statistical categories and maximum-likelihood or Bayesian approaches) assumed that all sites of multiple sequence alignment have a uniform evolutionary pattern. In contrast, the evolutionary history of bacteria has shown something else. HR and horizontal gene transfer have complicated their evolutionary history, so some standard assumptions in phylogenetic reconstruction are invalidated [52]. Regarding the evolutionary view, recombination events in the bacterial genome provide a rich phylogenetic signal that can be used in detecting and

modifying the phylogenetic tree inference [53].

We reviewed conventional approaches in phylogenetic reconstruction in this chapter. Many efficient and fast tools have been developed based on these methods, which can apply to extensive datasets, such as Raxml-NG [98], IQ-TREE [93], [94], and Physher [99] (ML approach), Beast [100], and MrBayes [101] (Bayesian approach), UShER [102] and MP-Boot [103] (maximum parsimony).

However, none of these methods can provide an authentic tree in bacterial sequences due to the recombination events. Therefore, studying alternative approaches that can tackle the problem of inaccurate phylogenetic reconstruction in this organism is exciting and essential. We reviewed the latest and most state-of-the-art tools in this scope. Although, some of them assume low recombination rates, such that a significant part of a genome is considered clonal, which has not been under recombination. But this is not true for some bacteria: e.g., in *E. coli* strains, very few contigs larger than a few kb have not been impacted by recombination [75]. Therefore, developing a new tool that can answer the different dimensions of the recombination problem in bacteria is necessary. This tool should discover recombination boundaries and still infer a reliable clonal tree even if the recombination rate is high. The next chapter introduces a new tool that addresses these issues.

CHAPTER THREE

RESEARCH METHODOLOGY: PHILOBACTER

The main goal of this chapter is to introduce a new tool, PhiloBacter, which detects homologous recombinations in the whole genome of bacteria and reconstructs an authentic tree. The organization of this chapter is as follows. Section 3.1 introduces Hidden Markov Models (HMM), and reviews critical problems that could be addressed through their practical application. In Section 3.2, Incorporating Sequence Uncertainty in Phylogenetic Inference is presented. In Section 3.3, PhiloBacter is explained in detail. Finally, we briefly summarize the contents of this chapter in section 3.4.

3.1 Introduction to HMM

Hidden Markov Models (HMM) are functional and well-known statistical models which first found application in speech recognition [104]. HMM is based on a graphical model, often applied to predict a sequence of unobservable (hidden or latent) states using a set of observable variables. So far, many research communities have applied this modeling technique to understand real-world data as diverse as the weather, text documents, and time series data in the stock market. Over the last two decades, bioinformatic problems have also been successfully approached using HMMs, including multiple sequence alignment [105], [106], trimming and gene annotation, searching in genomic databases and gene discovery [107], prediction of bacterial lipoproteins [108], prediction of cell-wall sorting signals [109], prediction of protein secondary structure [110] and prediction of transmembrane protein topology [111].

Felsenstein and Churchill have significantly contributed to phylogenetics by introducing Hidden Markov Models (HMM) [112]. They successfully constructed a robust theoret-

ical framework and devised computational methods tailored to estimate the likelihood of a phylogeny. Their unique approach incorporated evolutionary rates based on the principles of HMM, which proved instrumental in advancing our understanding of genetic evolution.

Furthermore, the work of Boussau and colleagues is noteworthy in the use of phylogenetic Hidden Markov Models (HMM) [113]. They ingeniously applied this tool to identify recombination breakpoints within alignments subject to homologous recombination. Their innovative method involved an exhaustive search for varying evolutionary histories within the alignment. This comprehensive approach illuminated the occurrence and nature of recombination events, thus providing crucial insights into the complex mechanisms underlying genetic variation and evolution.

To explain HMMs in detail, an overview of Markov models is essential, as the hidden states behave as a Markov process.

3.1.1 Basic Understanding of Markov Model

Markov processes or chains are random processes in discrete or continuous time; they are memoryless, which means the next event is influenced only by the present event and not by any other older events. Indeed, the behavior of the chain in the past does not play any role in the behavior of the process in the future. This is known as the Markov property. From a mathematical perspective, the probability of an event at step t will be determined by step $t - 1$. Expressed another way, the probability of $s(t)$ given $s(t - 1)$, that is $p(s(t)|s(t - 1))$. This model is called the first-order Markov model, which includes a sequence of stochastic variables X_1, X_2, X_3, \dots that have the Markov property, that is:

$$\Pr(X_{t+1} = x \mid X_1 = x_1, X_2 = x_2, \dots, X_n = x_t) = \Pr(X_{t+1} = x \mid X_t = x_t)$$

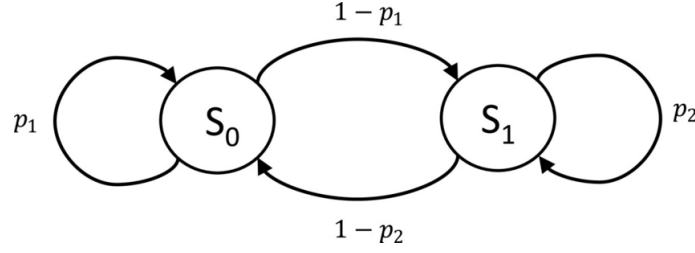


Figure 3.1 Representation of a Markov Chain with two states: This shows a Markov Chain with states S_0 and S_1 , and the arrows indicate the probability of departing from one state to the other state or remaining the same [114]

A Markov chain is characterized by a state space; initial states, which are the probability distribution of starting at each state; and a transition probability matrix. Each element P_{ij} of a transition probability matrix represents the probability of departing from state i to state j . The dimension of a transition matrix is $M \times M$ where M is the cardinality of the state space. A transition matrix or stochastic matrix satisfies the constraint $\sum_{j=1}^M P_{ij} = 1$. That is, the row sum of the probability matrix must equal 1.

$$P = \begin{pmatrix} P_{1,1} & P_{1,2} & \cdots & P_{1,j} & \cdots & P_{1,M} \\ P_{2,1} & P_{2,2} & \cdots & P_{2,j} & \cdots & P_{2,M} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ P_{i,1} & P_{i,2} & \cdots & P_{i,j} & \cdots & P_{i,M} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ P_{M,1} & P_{M,2} & \cdots & P_{M,j} & \cdots & P_{M,M} \end{pmatrix}$$

A Markov chain can be depicted graphically using a weighted directed graph. An example of a two-state Markov chain with state space $S = \{S_0, S_1\}$ is shown in Figure 3.1. The value of each edge of this graph shows the transition probability from one node (state) to another. The corresponding transition matrix is defined as:

$$P = \begin{pmatrix} P_{1,1} & P_{1,2} \\ P_{2,1} & P_{2,2} \end{pmatrix}$$

Markov models mainly illustrate stochastic systems, such as weather patterns. In a Markov process, every state is observable or visible.

3.1.2 Hidden Markov Model (HMM)

A Hidden Markov Model (HMM) is a Markov process, but the states are unknown here. An HMM is sometimes known as a "doubly-embedded stochastic process", having two stochastic processes: hidden and observed. The former process consists of hidden states that are not straightforwardly observable; the latter consists of the visible process of observable symbols. The underlying hidden state process drives the visible process. While observed states can be regarded as a noisy module of the system states of interest, in most cases they are insufficient to characterize the state accurately. Both visible and latent variables in an HMM may be discrete or continuous. A HMM consists of the following five components:

1. Hidden State Sequence:

A hidden state sequence $X = X_1, X_2, \dots, X_T$ where X_i is drawn from the state space $S = \{S_1, S_2, \dots, S_M\}$. These sets can be symbols, tags, or labels describing anything, like the gene type.

2. Observation sequence:

The data is a sequence of observations $O = O_1, O_2, \dots, O_T$ drawn from a set of possible observations N , that is $O_i \in N$ for all i . Each observation depends only on the hidden state that generated it. It should be noted that the identical observation sequence can be emitted from different hidden state sequences. In addition, the observation sequence

should have at least one symbol but could have any length; also, the observation sequence can't have any gaps and should be continuous.

3. Start Probability:

A start probability distribution $\pi = \pi_1, \pi_2, \dots, \pi_M$ where $\pi_i = p(X_1 = i)$ is the probability that the Markov chain will start in state i . Since it is a probability distribution we have $\sum_{i=1}^M \pi_i = 1$.

4. Transition Probability:

The transition probability is the probability of jumping from one state to another state. These probabilities are defined within a transition matrix $A = (a_{ij})$ of dimension $M \times M$. a_{ij} is the probability of moving from state i to j at time step $t + 1$. That is

$$a_{ij} = p(X_{t+1} = j | X_t = i).$$

5. Emission Probability:

The emission probabilities point to the association between the hidden states in the model and the input data, that is, the sequence of observations. Emission probabilities are the probability of an emission (observation) precisely expressing the latent state of the model for that particular state transition. In other words, a sequence of observation likelihoods that expresses the probability of an observation o_t being produced from a state s_i .

The observation probability matrix $B = b_j(k)$ has dimension $N \times M$ and each element represents the probability of emitting observation k given hidden state j . That is

$$b_j(k) = p(O_t = k | X_t = j).$$

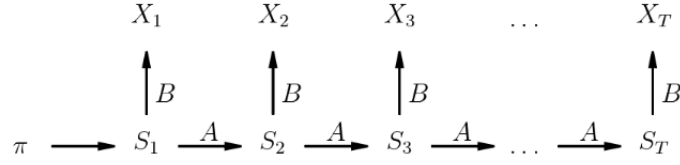


Figure 3.2 A graphical representation of an HMM

HMM, parameters are commonly described by λ , with $\lambda = (A, B, \pi) = (\text{Transition probability, Emission Probability, Start Probability})$.

The joint likelihood of a particular observation sequence $O = O_1, O_2, \dots, O_T$ and a state sequence $X = (X_1, X_2, \dots, X_T)$ can be directly computed as:

$$p(X, O | \pi, A, B) = \pi_{X_1} b_{X_1}(O_1) a_{X_1, X_2} b_{X_2}(O_2) \dots a_{X_{T-1}, X_T} b_{X_T}(O_T).$$

3.1.3 What kind of problems can be resolved through the Hidden Markov Model?

Three critical questions are often answered by HMM theory, given observation sequences $O = O_1, O_2, \dots, O_T$, and the model $\lambda = (A, B, \pi)$. These are:

Decoding Problem The aim of the decoding problem is to estimate the optimum hidden state for HMMs using the Viterbi and posterior algorithms. The Viterbi algorithm calculates the maximum probability path (with the output a path that is a sequence of states), known as the Viterbi path, using a dynamic programming algorithm. The desirable outcome is to calculate the hidden labels of a sequence with a maximum probability, but it is an NP-hard problem. The posterior algorithm calculates the most likely state or label of each sequence position and binds them as a single label. This is fast enough, but there is no guarantee that it will become consistent with the possible labeling of the sequence because it may not be able to meet the model's constraints.

Evaluation Problem Given a sequence of observations O and an HMM $\lambda = (A, B, \pi)$, the aim is to calculate the model likelihood $P(O | \lambda)$. This procedure is based on the summation of all potential state sequences:

$$P(O) = \sum_X P(O | X)P(X)$$

where the summing is over each possible hidden sequence $X = X_1, X_2, \dots, X_T$.

The number of possible hidden sequences for an HMM model where the dimension of hidden states is N , and the sequence of observations is T respectively, is N^T . In a real-world dataset, where these dimensions are both considerable, N^T is enormous, so it's impossible to gain the overall observation likelihood by summing an observation likelihood for all latent state sequences.

These enormous calculations are avoided using a well-organized $O(N^2T)$ algorithm known as the forward algorithm. Indeed, as the dynamic programming algorithm, the forward algorithm splits a complex problem into several simpler subproblems, keeping intermediate values while it accumulates the probability of the input sequence. In practice, it calculates the observation probability by adding the probabilities of every feasible hidden state path that could create the observation sequence. It's efficient because it implicitly folds each path into a single forward trellis.

Learning Problem When the goal is learning the HMM parameters or the best set of state transition and emission probabilities (a_{ij} and b_{ij} , respectively), one approach is to maximize $p(O|\lambda)$. Given a set of output sequences, the maximum likelihood estimation of these parameters can deliver the best sets. This problem has no tractable algorithm as the solution, but generally, a local maximum likelihood can be estimated using the Forward-Backward or Baum-Welch algorithm. The Baum-Welch method is a special type of expectation-maximization al-

gorithm.

The intrinsic uncertainty found in biological data, often attributed to factors such as sequence heterogeneity and various biological events, underscores the importance of acknowledging recombination events as potential sources of this uncertainty. Consequently, our primary objective is to directly face and model the uncertainty stemming from recombination in the context of phylogenetic analyses.

This process begins with comprehensively presenting uncertainty models tailored to DNA sequences. We aim to thoroughly understand these models and how they encapsulate the nuances of genetic data. After this, we integrate this uncertainty into the process of phylogenetic inference. This involves devising strategies and methods that can incorporate the varying degrees of uncertainty arising from recombination events, aiming to enrich the accuracy and reliability of phylogenetic reconstructions.

3.2 Incorporating Sequence Uncertainty in Phylogenetic Inference

Multiple sequence alignment (MSA) is the input of conventional phylogenetic inference methods. This alignment is deterministic: each position of the MSA holds a letter from a set of the alphabet representing a nucleotide or amino acid sequence; or a “gap” where there is missing data, insertion or deletion. This is based on a simplification assumption, whereas uncertainty usually presents in empirical sequence datasets because of the sequencing technology employed and related errors. This ambiguity usually arises from sequencing error, inaccurate read mapping, and alignment error but can also be from the natural level of heterogeneity of the sequences. Researchers typically use purifying methods to decrease sequence ambiguity, for example, by discarding low-quality reads or poor-confidence segments in alignment [115], [116]. Such refining could also dismiss helpful information, so it becomes controversial for

datasets with similar sequences or a high rate of erroneous sequences [117].

Phylogenetic inference methods and evolution models propose a simple way to integrate sequence data uncertainty. Incorporating IUPAC ambiguity code [118] is the simplest model of uncertainty specification, which can convert nucleotides to other symbols at the same alignment position. For example, character R corresponds to A or G. Most phylogenetic programs support IUPAC degenerate codes. Felsenstein [119] presented a more sophisticated way of modeling sequence uncertainty, which assumes a fixed error rate. Parker et al. [120] proposed a Bayesian Markov-Chain Monte Carlo to consider uncertainty emerging from phylogenetic error. Kuhner and McGill [121] investigated Felsenstein's model on simulated datasets. Assuming a constant error rate, they concluded that error rate models improve branch length estimates.

In DNA-based maximum likelihood algorithms, each tip at each site is associated with a probability vector of dimension 4, where each element represents the probability to be a particular base. This thesis assumes that the first element corresponds to A, followed by C, G and T. For example, when there is no error, and A (Adenine nucleotide acid) has been observed, the tip value would be (1,0,0,0), which means a probability of 1 (deterministic) for A and a probability of 0 for C, G, and T. These vectors are known as tip partial, and are needed to initialize the recursion in the Felsenstein's peeling algorithm [59], computing probabilities from the tips upward to the root. It should be noted that each of these four values illustrates the probability of the observation of different nucleotide acids A, C, G and T. Therefore, their sum shouldn't be equal to 1. To represent any completely uninformative position, such as gaps or missing data, all values in the tip partial vector are set to 1. These partial vectors can easily account for degenerate codes such as R, for which the partial vector would be (1,1,0,0).

3.2.1 Uncertainty models in DNA sequences

Three error models have been previously described.

1. UFE: UniForm Error

The UniForm error model (UFE) is the simplest sequencing error model, assuming a single error rate, ϵ , shared across sequences and positions [119], [121]. For example, where the true base is A, the probability of observing A would be $1 - \epsilon$, and the probability of observing each of C, G, or T, defined by alignment row $i \in [1, n]$ and column $j \in [1, m]$, given the observed state S_{ij} is $\epsilon/3$. For n sequences of length m , the likelihood of having an actual state $x \in \{A, C, G, T\}$ at the alignment row $i \in [1, n]$ and column $j \in [1, m]$, when the observed state S_{ij} is

$$L_{ij}(x) = P(S_{ij} = y|x, \epsilon) = \begin{cases} 1 - \epsilon & \text{if } x = y \\ \epsilon/3 & \text{otherwise} \end{cases}$$

2. PSE: Position-Specific Error

The position-specific error model (PSE) is an extension of the UFE model, in which error rates are variable across sequences and sites. In other words, every single site at alignment row $i \in [1, n]$ and column $j \in [1, m]$ is allocated a specific error rate ϵ_{ij} . Hence, the likelihood calculation is as follows [122]:

$$L_{ij}(x) = P(S_{ij} = y|x, \epsilon_{ij}) = \begin{cases} 1 - \epsilon_{ij} & \text{if } x = y \\ \epsilon_{ij}/3 & \text{otherwise} \end{cases}$$

3. PSL: explicit Per-State Likelihoods

In some frameworks, the upstream software provides per-state likelihoods at every align-

ment position (for example, variant-calling tools such as GATK [123]). As a result, per-position and per-state likelihoods can be initialized directly by the values presented in the corresponding input file [122]:

$$L_{ij}(x) = P(S_{ijx})$$

where S_{ijx} is the likelihood of state $x \in \{A, C, G, T\}$ at alignment row $i \in [1, n]$ and column $j \in [1, m]$.

RAxML-NG supports UFE and PSL error models [122].

3.3 PhiloBacter: A new tool to infer phylogenetic trees from recombinant bacterial genomes

3.3.1 Overview

When there is an alignment of the bacterial genome as a set of observed sequences $O = O_1, O_2, \dots, O_T$, it is known that there are hidden states behind the alignment. This presents two challenges: firstly, how to tune the HMM parameters (A, B, π) to find hidden states; and secondly, how to utilize the hidden patterns for tree inference. PhiloBacter resolves this problem in two steps:

1. Recombination Estimation
2. Phylogenetic inference in the presence of recombination

The only input required by PhiloBacter is an alignment of the bacterial DNA sequences. Detecting recombination is the first step and can be accomplished using one of the existing stan-

standard methods, such as IQ-TREE [93], [94] or RAXML [124]. This will reconstruct a phylogenetic tree based on the input alignment; this tree is called an uncorrected tree. An HMM is then employed to find the location and other properties of recombination events in the sequences.

The second step is reconstructing a new tree with approximately correct branch lengths. The output from step one is utilized to reconstruct a corrected phylogenetic tree, one which accounts for recombination. These two general steps of PhiloBacter are shown in Figure 3.3.

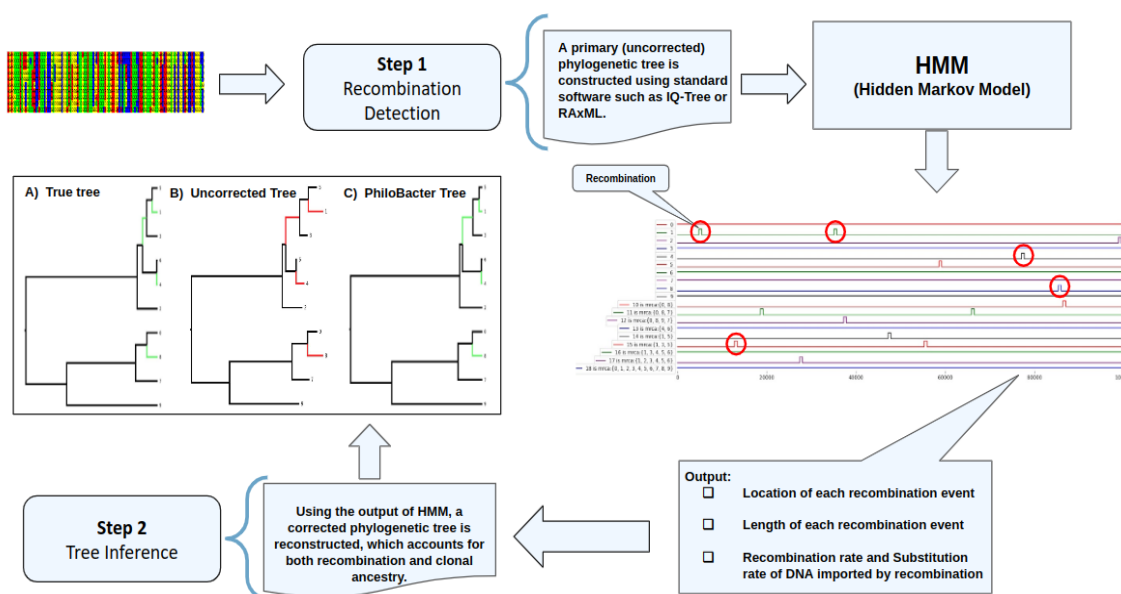


Figure 3.3 Overview of the PhiloBacter process, illustrating (Step 1) estimation of recombination events and (Step 2) phylogenetic inference accounting for the presence of recombination.

3.3.2 Step one: Recombination Estimation

3.3.2.1 Description of algorithm

Let O denote the aligned sequence data, which consists of the N genotypes with total length L and O_{nj} denote the nucleotide of j th site $j \in \{1, \dots, L\}$ in multiple alignments of n th lineage. The first goal is to extract hidden mosaic patterns in the alignment.

It is assumed that there are hidden states behind the observed alignment, expressing

whether each nucleotide is in the recombination part on the branch above each node of the genealogy or not. This combination of observed sequences and hidden states lends itself to the application of an HMM, to learn about the unobservable states which depend on the alignment. Therefore, each state O_{nj} is described as having two states: Recombination and Clonal Frame. Recombination refers to the area that contains foreign DNA; while the Clonal Frame is that part of the genome that has not undergone recombination, the core part of the bacterial genome sequence.

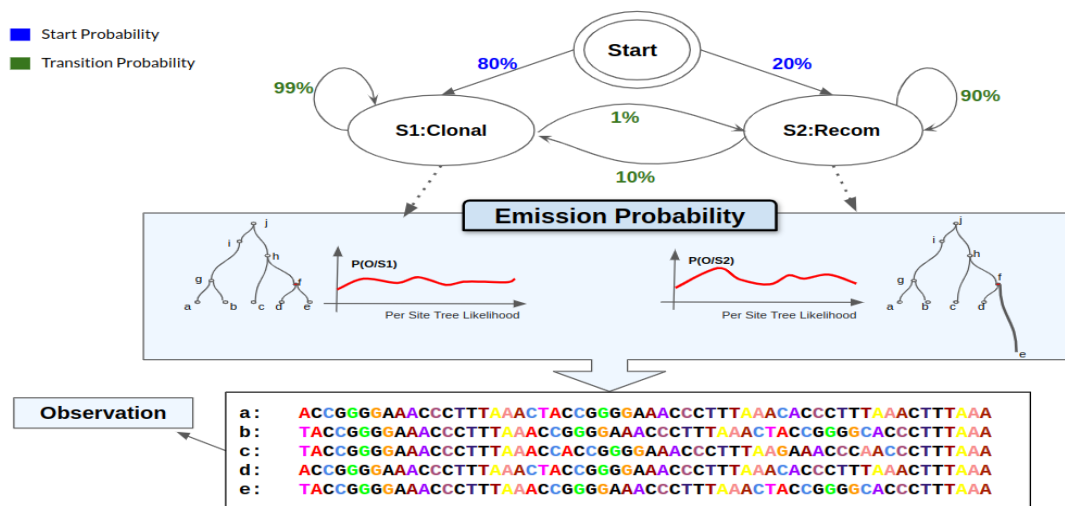


Figure 3.4 Schematic representation of the PhiloBacter HMM framework. This model encompasses five core components: (1) Hidden State Sequence, (2) Start Probability, (3) Transition Probability, (4) Emission Probability, and (5) Observation Sequence,

This novel parametric probabilistic framework activates HMMs to detect recombination and clonal frame segments in an alignment. A new HMM that benefits from having position-specific emission probabilities for modeling several trees with the various branch lengths underlying the genome is proposed. The Baum welch algorithm approximates the parameters of the HMM transition distributions. Figure 3.4 is a schematic view of HMM for PhiloBacter. The details of the different components of this HMM model are:

Two hidden states of HMM: It is assumed that recombination makes a high level of polymorphism at each branch, providing enough information to differentiate the recombined regions from the clonal ones. Therefore, this HMM has two distinct possible hidden states. The first state still represents the clonal tree, and the second indicates recombination events that happened on every branch, regardless of the effect of its neighbor branches. Figure 3.5 shows a graphical tree view of these two states, with the target node highlighted in red; the second tree displays a very long branch above the target node, indicating a recombination signal.

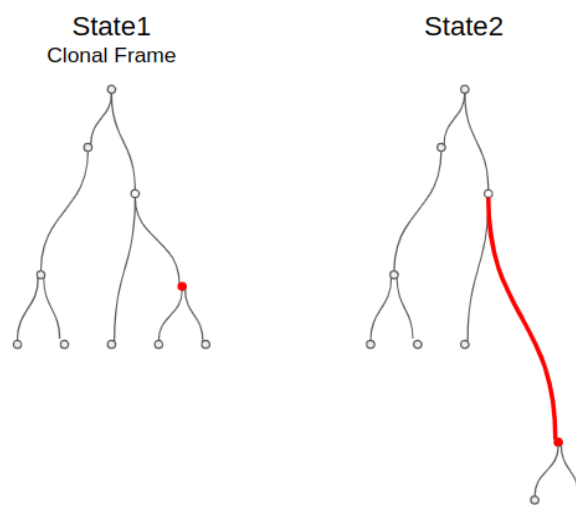


Figure 3.5 Illustration of the bi-state HMM approach utilized in this study for recombination state identification. The red node exemplifies a target node, highlighting the hypothesis that recombination amplifies polymorphism within each branch, subsequently elongating it.

Initial probabilities: Each state of the HMM requires the adoption of an initial probability. The influence of start probability is rendered insignificant by the sizable length of the bacterial DNA sequence, making it feasible to assign a uniform distribution.

Observations: As discussed in section 3.2, in a standard DNA-based maximum likelihood algorithm, the tip partial holds the probability of the observed sequence, and Felsenstein's peeling algorithm [59] uses the tip partials to calculate the partial likelihood for internal nodes.

It implies that we can define a partial probability for all tree nodes. So we have a vector with 1×4 dimensions for every node in each locus, and we have a matrix of $(2 \times N - 2) \times (L \times 4)$ size for a given alignment with length L for N taxa, where $(2 \times N - 2)$ is the total number of nodes of an unrooted tree. The elements of this matrix for leaves include 0 and 1. In contrast, for internal nodes, these values would be a decimal number between zero and one to show the probability of the estimated sequence for the extinct organisms. An individual HMM is employed for each tree branch (above a target node) to evaluate recombination loci, where the specific input (observation) for each HMM model is the partial likelihood for the target node. Therefore, the dimension of the observation matrix for each individual HMM model is $(L \times 4)$. Figure 3.6 illustrates an example of the HMM model observation matrix for three different nodes, including two external nodes (nodes a and b) and one internal node (node g).



Figure 3.6 Example of the HMM model observation for three different nodes, including external and internal nodes.

Emission probabilities: The emission probabilities outlined in section 3.1.2 point to the relationship between the hidden states in the model and the sequence of observations. These probabilities are the probability of observable data addressing the hidden state of the model for the particular state transition. As a result, calculating per-site likelihoods for both the clonal tree and recombination tree of the hidden state can be taken as the emission probability of the HMM model. As mentioned above, the recombinant tree has a long branch above the target

node, beginning with selecting a height parameter. The substitution rate of DNA imported by recombination, v , fits this criterion but needs to be estimated.

As we mentioned in section 3.1.2, the emission probabilities point to the relation between the hidden states in the model and the sequence of observations. These probabilities are the probability of observable data addressing the hidden state of the model for the particular state transition. As a result, if we calculate per-site likelihoods for our hidden state's trees (clonal tree and recombination), those would be taken as the emission probability of our HMM model. Since we mentioned above that the recombinant tree has a long branch above the target node, the main question is how I can construct the recombinant tree. To put it simply, what parameter reveals how height is enough for the chosen branch? The answer is v ; it's the substitution rate of DNA imported by recombination. We need to estimate it.

The likelihood function is the conditional probability of the sequence alignment O given the parameters $Pr[O|\theta]$, which θ may include the branch lengths (b), the parameters of the substitution model and the tree topology (τ):

$$L(O_j|\theta) = Pr[O_j|b, m, \tau]$$

= Pr(j th column of the alignment | branch lengths, substitution model parameters, tree topology)

Hence, the per-site likelihood for the descendant sequences is defined conditionally on the branch lengths. Because the parameters of the substitution model and tree topology would be similar for the two hidden states, they are omitted in the following equation:

$$Pr(O_j|\mathbf{b}, v_i, H_j) = \begin{cases} Pr(O_j|\mathbf{b}_{-i}, b_i + v_i), & \text{if } H_j = R \\ Pr(O_j | \mathbf{b}), & \text{otherwise} \end{cases}$$

where b_i is the branch under investigation and \mathbf{b}_{-i} is a vector of branches \mathbf{b} without element b_i . and v_i is the substitution rate of the imported segment as recombination. It should be noted that we assumed different values of v for each branch.

The optimal value for v must be estimated for each branch; approximating this value would convert the problem to a standard optimization problem, where the objective function is forward probability. Since the problem incorporates a scalar function and one variable to estimate, the SciPy [125] library offers an efficient tool to solve it. SciPy is a package of numerical procedures for the Python programming language that presents infrastructure blocks for modeling and solving scientific problems. SciPy provides algorithms for optimization, integration, eigenvalue problems, algebraic equations, differential equations, and so on. Some constraints can be tuned to avoid considering irrelevant values to v .

Transition probabilities: In general, when k are hidden states in the HMM model, a $K \times K$ transition probabilities matrix needs to be specified to represent the probability of moving from state i to state j . See the following transition probability definitions for two states of HMM:

$$Pr(H_i|H_j) = \begin{cases} \alpha & H_i = C \text{ and } H_j = R \\ 1 - \alpha & H_i = C \text{ and } H_j = C \\ \beta & H_i = R \text{ and } H_j = C \\ 1 - \beta & H_i = R \text{ and } H_j = R \end{cases} \quad (3.1)$$

Where H_i and H_j are shown as the hidden states, α is the probability of moving from the clonal state when the previous observation has been in the recombination state, and β is the probability of moving from recombination state to clonal one.

Since the recombination rate is unknown and varies between different datasets, it is necessary to learn about it based on the aligned sequences for each branch of the HMM model to be able to construct the transition probabilities matrix. The Baum-Welch algorithm [95] is an appropriate tool to approximate the transition probabilities. It utilizes the forward-backward algorithm to calculate the statistics for the expectation phase, facilitating the computation of the posterior marginals of all hidden states to calculate the transition probability. Procedures 1 and 2 are the pseudocode of the forward and backward algorithms. Emission probability has already been calculated another way (as described above); thus, there is no need to recalculate this probability. As Procedure 3 shows, the forward and backward algorithms have been used to compute the probability transition from state i to state j . Different transition probabilities are now offered for each tree branch.

Procedure 1: Forward procedure

```

1 Procedure Forward(trans, emission, InitialProbabilities) is
   |   /* trans is an initial guess for transition probability, emission
   |   |   is the emission probability matrix of HMM model          */
2   |    $\alpha(1) = \text{InitialProbabilities} * \text{emission}(1)$ 
3   |   for  $t = 2, 3, \dots, L$  do
   |   |   // L is the length of alignment
4   |   |
5   |   |    $\alpha(t) = \text{emission}(t) * \sum_{j=1}^n \alpha(t-1) * \text{trans}_j$  // n is number of hidden
   |   |   |   states
6   |   |
7   |   end
8   |   return  $\alpha$ 
9 end

```

In addition, It should be noted that based on this transition probability, it's possible to calculate the recombination rate based on the following equation, which has been presented in

the ClonalFrameML paper [91].

$$Pr(H_i|H_j) = \begin{cases} e^{-d_{jk}M\frac{R}{\theta}} & H_j = U \text{ and } H_k = U \\ 1 - e^{-d_{jk}M\frac{R}{\theta}} & H_j = U \text{ and } H_k = I \\ 1 - e^{-d_{jk}/\delta} & H_j = I \text{ and } H_k = U \\ e^{-d_{jk}/\delta} & H_j = I \text{ and } H_k = I \end{cases} \quad (3.2)$$

Having calculated transition probability, the first and fourth elements can be put equal to the first and fourth equations of matrix number 3.2, allowing those equations to be solved. In this way, the values of $\frac{R}{\theta}$ and δ can be calculated. Although the δ value does not need to be estimated in this way because PhiloBacter can detect recombination boundaries, the beautiful point is that the δ value computed in this manner is remarkably close to the length seen by PhiloBacter.

Procedure 2: Backward procedure

```

1 Procedure Backward(trans, emission) is
  /* trans is an initial guess for transition probability, emission
    is the emission probability matrix of HMM model */
  // L is the length of alignment
2
3   $\beta(L) = [1, 1]$ 
4  for  $t = L - 1, \dots, 2, 1$  do
5     $\beta(t) = \sum_{j=1}^n \beta(t+1) * trans_j * emission(t+1)$  // n is number of
      hidden states
6
7  end
8  return  $\beta$ 
9 end

```

Procedure 3: Update procedure

```
1 Procedure Update(trans, emission, InitialProbabilities, iter = n) is
2   for  $n = 1, \dots, iter$  do
3      $\alpha = \text{Forward}(\text{trans}, \text{emission}, \text{InitialProbabilities})$ 
4      $\beta = \text{Backward}(\text{trans}, \text{emission})$ 
5     for  $t = 1, \dots, L$  do
6        $\gamma(t) = \alpha_i(t) * \beta_i(t) / \sum_{j=1}^n \alpha_j(t) * \beta_j(t)$ 
7       // i and j represent the observation state at times t and
8       t+1.
9        $\text{numerator} = \alpha_i(t) * \text{trans}_{ij} * \beta_j(t+1) * \text{emission}(t+1)$ 
10       $\text{denominator} = \sum_{k=1}^n \sum_{w=1}^n \alpha_k(t) * \text{trans}_{kw} * \beta_w(t+1) * \text{emission}_w(t+1)$ 
11       $\xi_{ij} = \text{numerator} / \text{denominator}$ 
12    end
13     $\text{trans} = \sum_{t=1}^{L-1} \xi_{ij}(t) / \sum_{t=1}^{L-1} \gamma_i(t)$ 
14  end
```

3.3.2.2 Lesson learned

Like all problems in the research world, the described approaches have certainly not been developed as a first solution. Other methods that did not lead to the correct answer for this problem or were not efficient or optimal have also been examined. In the following, some of these approaches are outlined.

Four states HMM: There is an assumption that the level of polymorphism at the branches above and the bottom of each non-root node gives us some signal to detect the hidden states. Therefore, it is possible to consider each non-root node in the initial tree as a *target* node, with, c_r and c_l denoting the children (right child and left child) of the *target* and p denoting the parent. Under this assumption, a model with four states can be considered. State one indicates a clonal state, which means the current phylogenetic tree with current topology and branch lengths could reflect the evolutionary pattern of that alignment position. State two represents recombination events on the branch between the *target* node and c_r . State 3 and

state 4 also illustrate the recombined state on the branches between the *target* node and node c_l , *target* node, and node p , respectively. As I mentioned, the corresponding branch length in recombination events is much longer than in the clonal state.

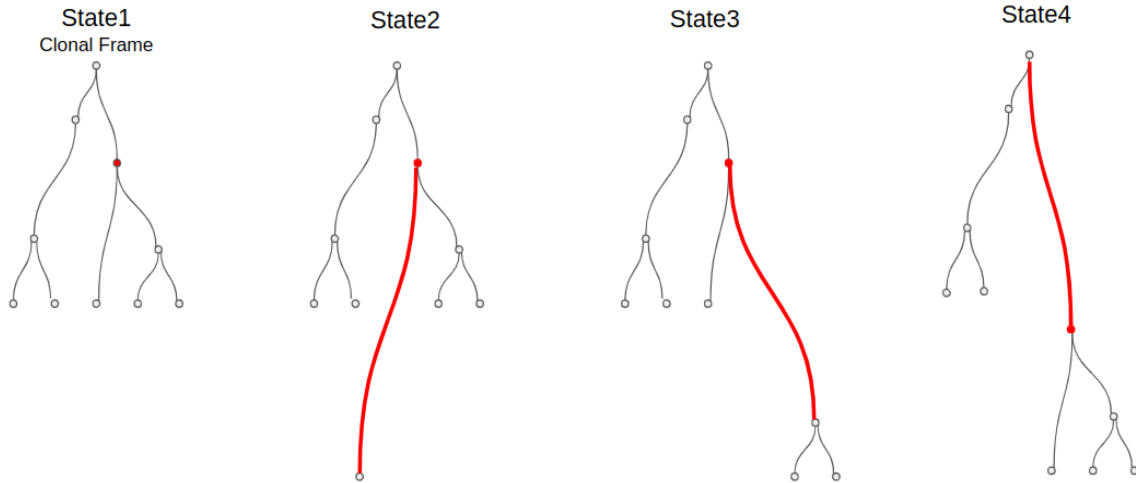


Figure 3.7 Depiction of the Four-State HMM framework. The red node serves as a representative example of the target node within the model. State 1 shows ClonalFrame, which didn't undergo any recombination.

Figure 3.7 shows a view of the four-state HMM model. As long as there are no simultaneous recombination events on the branches above the two sister nodes, this model successfully reveals recombination events with acceptable accuracy. But since there is no guarantee that concurrent events will not happen on branches above the two sister nodes or even one node and its parent, it is prudent to investigate models that can detect recombination without exceptions.

Observations: Consider an arbitrary internal node to be a target node. Re-root the tree based on this node. In such a case, the target node has three children, where it is possible to calculate the partial likelihood of each child being a tip or non-tip node. Thus, there are three partial vectors, each with 1×4 dimension. If these three vectors are merged to make a vector with size 1×12 , this vector plays the role of observation for each locus. As a result, the total observation vector for one sequence with length L would be $1 \times 12 \times L$, and the observation for the whole

alignment with N taxa has dimensions $(2 \times N - 2) * (L \times 12)$, where $(2 \times N - 2)$ is the total number of nodes of an unrooted tree. According to Figure 3.6, if the target node is node g, the observation vector for the first site is:

$$O_i = [1, 0, 0, 0, 0, 0, 0, 1, 0.00174, 0.00268, 0.00525, 0.00952]$$

Note that the first four values are related to node a, the second four belong to node b, and the last four represent the partial likelihood of node i.

Eight states HMM: An existing solution to the problem of simultaneous recombination events on branches connected to the target node is to use a permutation to examine all the states where the recombination event may occur in these three branches. In this way, eight states can cover all synchronous and non-synchronous recombination events around a target node.

The advantage of this approach is that it can identify all simultaneous recombination events occurring around a node. However, if the depth of the tree is considerable, traversing from the leaves to the root of the tree results in many branches being examined more than once to scan for the presence of recombination. So, although this approach can detect almost all recombination events, it is computationally inefficient and redundant.

Figure 3.8 shows a view of the eight clonal and recombinant trees. The observation matrix for eight-states-HMM is quite similar to the four-states-HMM. It can be concluded that the best approach to address recombination events is the two-state HMM model, which is capable of detecting synchronous and asynchronous events yet is not prone to redundancy.

So far, PhiloBacter has played the role of a recombination detection tool using phylogenetic tree signals. One of the exciting features of PhiloBacter is that it is not a deterministic approach; in contrast, it is based on uncertainty (probability). The output of the first phase is

probabilistic values. In other words, for each locus in the bacterial genome sequence, there is a grade indicating the likelihood that this site was recombined or is still clonal. However, the central part of the problem remains the inference of the tree whose branch lengths are least impacted by recombination.

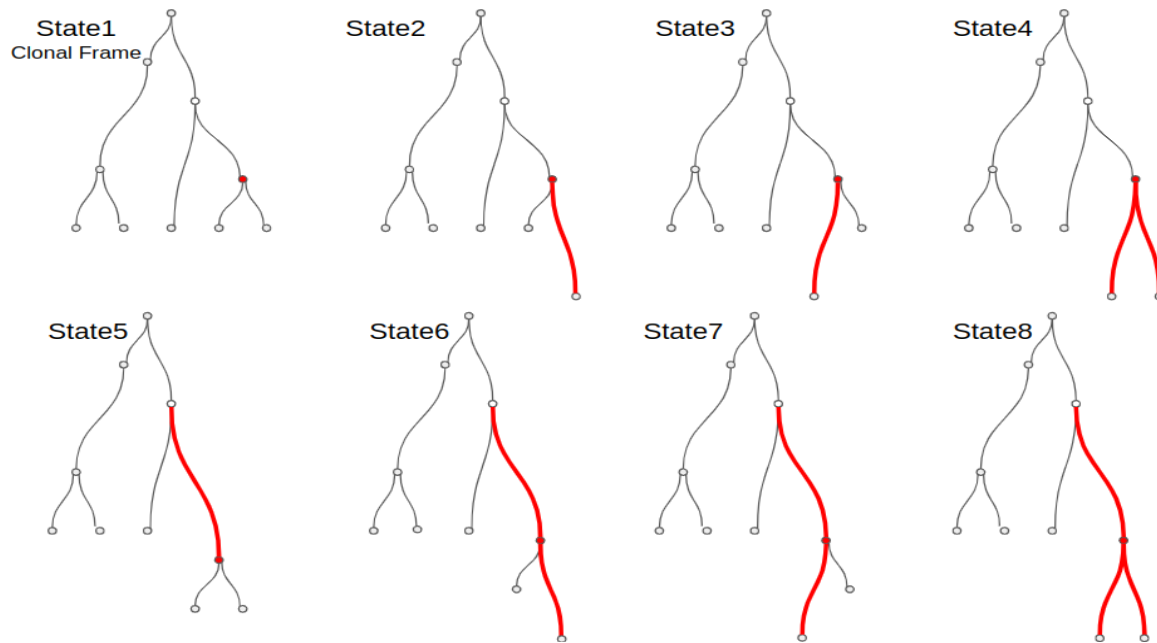


Figure 3.8 Eight trees indicate the hidden states of HMM model. The red node serves as a representative example of the target node within the model. State 1 shows ClonalFrame, which didn't undergo any recombination.

3.3.3 Step two: Clonal tree Inference using whole genomes

In the rest of this chapter, two novel strategies for tree inference of recombinant bacterial genomes are proposed. The probabilistic outputs of the HMM model described in the previous section are used to reconstruct a tree that accounts for the presence of recombination.

Regular Maximum Likelihood Calculation: As explained in Chapter 2, one of the conventional ways to infer a tree is the maximum likelihood. The details of the approach are described in the same chapter. Here, Algorithm 4 shows the pseudocode of this approach.

The input of this algorithm is an alignment of the desired organism, M , including evolutionary parameters such as the transition/transversion ratio and base frequencies $(\pi_A, \pi_C, \pi_G, \pi_T)$. τ shows the tree topology with its branch lengths as the number of substitutions per site, and P is the transition probability matrix of the evolutionary model. P can be employed to approximate the expected genetic distance, the branch length, between two sequences based on evolutionary models such as GTR or JC69. The alignment is converted to the tip-partials, and the partial likelihood for internal nodes is calculated with the help of P and tip-partials and M .

Algorithm 4: Regular ML estimation to Infer Phylogeny

```

Input: Alignment,  $M, \tau, P$ 
Output: A set of branches length
1 begin
2   Initialization:
3      $A = [1, 0, 0, 0]$ 
4      $C = [0, 1, 0, 0]$ 
5      $G = [0, 0, 1, 0]$ 
6      $T = [0, 0, 0, 1]$ 
7      $P = P_{GTR/JC69}(brs)$  //  $P_{GTR/JC69}$  is the transition probability
        matrix of the desired evolutionary model.  $brs$  points to the
        tree branches' length.
        // Calculate ML based on Felsenstein rules
8   Function  $ML\text{-}Calculation(Partial, \tau, M) : double$  is
9     for  $node$  in postorder-iteration do
10      if  $node$  is NOT leaf() then
11         $partial =$  employing Felesentian rules using regular tip-partial,  $P$ 
12      end
13    end
14    return Likelihood
15  end
16 end

```

3.3.3.1 PhiloBacter: Maximum Likelihood Calculation

So far, no biologically proven evidence suggests that both branches of the phylogenetic tree have the same evolutionary parameters at any particular locus [126]. This new proposed

approach involves a simple extension of the Felsenstein [59] pruning algorithm, which makes it feasible to include a mixture probability transition matrix. The main idea of this approach is inspired by Kosakovsky et al. [126].

To address the problem of inaccurate branch length due to recombination, the extension of Felsenstein's [59] pruning algorithm enables efficient likelihood computations for models that vary over sites and branches. This development provides an opportunity to model the process at every branch-site combination as a mixture of two Markov substitution models. It considers every branch in an individual position as an unobserved state chosen independently from any other branches.

Let O_j denote the nucleotides at site $j \in \{1, \dots, l\}$ in the alignment. The likelihood probability is clearly based on the model of sequence evolution, M , and the tree, τ , with the branch lengths. In theory, each locus could be allocated its model of evolution.

$$L(\tau, M) = Pr(O_j | \tau, M)$$

Given different categories of parameters for each branch site, this probability could be represented as:

$$Pr(O_j | M_i) = \sum_{i=1}^n P(M_i) P(O_j | M_i)$$

Where n is the number of different states, in this scenario, it is two: according to clonal and recombination states.

In summary, calculating the log-likelihood of a tree includes the following steps:

1. Diagonalise Q . (matrix 2.3 show the Q of the GTR model)

2. Take the exponential of Qt for all tree branches, where t is the branch length
3. For all loci, utilize equation (2.6) using a post-order traversal of the tree
4. Take the logarithm and sum over sites.

Due to recombination, two Markov models exist for each branch in the bacterial genome. The posterior probability of the HMM model of the previous section helps in the construction of those specific Markov models per branch site. Two P can be defined for each tree branch; then, using the posterior probability of each position of alignment to make the $P_{mixture}$:

$$P_i^j(t) = Pr(H_j = C | O_{ij})P(t) + Pr(H_j = R | O_{ij})P(t + v_i) \quad (3.3)$$

Where t is the initial branch length, v is the substitution rate of DNA imported by recombination, and P is the transition probability matrix of the Markov Model. The conditional probability $Pr(H_j = R \text{ or } C | O_{ij})$ gives us the posterior value that indicates the information that each locus on each tree branch can be in recombination or clonal region. In practice, the recombination probability for each nucleotide injects into the pruning algorithm. And thus, the final inferred tree has involved recombination regions, and the length of its branches is not incorrect due to the presence of recombination events. Algorithm 5 shows the pseudocode of this approach.

3.3.3.2 PhiloBacter: Uncertainty in clonal tree Inference

Section 3.2 was an overview of the uncertainty models in DNA sequences and the reasons for this phenomenon. The leading cause of the uncertainty was related to sequencing technology. Although, the natural level of heterogeneity of the sequences or other biological

Algorithm 5: PhiloBacter:ML estimation using $P_{mixture}$ to Infer Phylogeny

Input: Alignment, M, τ, P , Posterior Probability Matrix, v ,

Output: A set of corrected branches length accounts for the presence of recombination events.

```
1 Initialization:
2    $A = [1, 0, 0, 0]$ 
3    $C = [0, 1, 0, 0]$ 
4    $G = [0, 0, 1, 0]$ 
5    $T = [0, 0, 0, 1]$ 
   /*  $P$  is the transition probability matrix of the desired
   evolutionary model, which is time reversible.  $brs$  points to the
   tree branches' length. */
6    $P = P(brs)$ 
   /* The value of  $v$  is different for each branch. */
7    $P_v = P(brs + v)$ 
8 begin
   /* Calculate ML using  $P_{mixture}$  */
9   Function  $ML\text{-}Calculation(Partial, \tau, M, v) : double$  is
10    for  $node$  in  $postorder\text{-}iteration$  do
11      /* for all leaves and internal nodes we use  $P_{mixture}$  */
12       $P_{mixture} = (1 - posterior) * P + posterior * P_v$ 
13       $partial =$  employing  $P_{mixture}$  as a transition probability matrix
14    end
15    return Likelihood
16 end
```

events could be the other reasons for the uncertainty. This idea was a stimulus to develop a new approach that considered the recombination events could also model by uncertainty.

PhiloBacter uses a probabilistic approach to look at each alignment site at each branch by applying an HMM model to estimate the probability. This probability is the chance of being in recombination or clonal segment for each tree branch at every alignment site. This probability provides authentic information from which genealogy may be reconstructed in the presence of recombination. The new approach employs the obtained probability for each locus to update the partial likelihood of each taxon. It uses a PSE (Position-Specific Error) model to update the tip partials. In this model of uncertainty, every single site at alignment row $i \in [1, n]$ and column $j \in [1, m]$ is allocated a specific error rate ε_{ij} . PhiloBacter assumed ε is:

$$\epsilon_{ij} = Pr(H_{ij} = R \mid O_{ij}) * v_i \quad (3.4)$$

$Pr(H_{ij} = R \mid O_{ij})$ value indicates the likelihood of each tree branch locus belonging to recombination. For the clonal state, it becomes $Pr(H_{ij} = C \mid O_{ij})$, and v_i is the substitution rate of DNA imported by recombination. Under this model, the probability of reading A, when the underlying base is A, becomes $1 - \epsilon_{ij}$, and the probability of reading each of C, G, or T, when the underlying base is A, becomes $\epsilon_{ij}/3$.

By updating the tip-partials in this manner, it is possible to correct the branches that lead to the leaves when recombination has occurred in them. But if the recombination events have happened on the branches of the internal nodes, the updated tip-partials cannot correct the length of the related branches. So, to correct the middle branches, another procedure is still needed to incorporate the uncertainty caused by recombination.

Recombination has rarely been known to change the topology of a tree [127]. By inferring the tree based on uncertainty in partial, the possibility to correct the tree's topology is provided if the topology has also changed due to the recombination.

This advantage is not limited to the maximum likelihood method for reconstructing the tree. It is possible to pass the updated tip-partials as input to a tool like Beast [100], [128], which works using the Bayesian inference.

Equation 3.3 can be used to correct internal branches of the tree that have been affected by recombination but only for the partial likelihood calculation of the internal nodes. For the nodes whose children are leaves, it is possible to use the updated tip-partial. The algorithm shows six pseudocodes related to this approach.

Algorithm 6: PhiloBacter: Incorporating Uncertainty in ML estimation to Infer Phylogeny

Input: Alignment, M , τ , P , Posterior Probability Matrix, v ,

Output: A set of corrected branches length accounts for the presence of recombination events.

```
1 Initialization:
2    $A = [1, 0, 0, 0]$ 
3    $C = [0, 1, 0, 0]$ 
4    $G = [0, 0, 1, 0]$ 
5    $T = [0, 0, 0, 1]$ 
6 begin
7   /* Update tipsPartial using HMM posterior probability where
   it's different for each site of alignment */
   // The value of  $v$  is different for each branch.
8    $NewPartial \leftarrow$ 
    $[1 - (posterior * v), (posterior * v)/3, (posterior * v)/3, (posterior * v)/3]$ 
   // Example of NewPartial:
   //  $A = [0.97, 0.99, 0.99, 0.99]$ 
   //  $C = [0.94, 0.98, 0.98, 0.98]$ 
   //  $G = [0.97, 0.99, 0.99, 0.99]$ 
   //  $T = [0.94, 0.98, 0.98, 0.98]$ 
   // Calculate ML using Updated-tipsPartial and  $P_{mixture}$ 
9   Function  $ML\text{-}Calculation(NewPartial, \tau, M) : \text{double}$  is
10    for  $node$  in  $postorder\text{-}iteration$  do
11      if  $node$  is NOT  $leaf()$  then
12        if  $children$  of the  $node$  are  $leaf$  then
13           $partial =$  employing Felesentian rules but using  $NewPartial$ 
14        end
15        else
16           $P_{mixture} = (1 - posterior) * P + posterior * P_v$ 
17           $partial =$  employing  $P_{mixture}$  as a transition probability matrix
18        end
19      end
20    end
21    return Likelihood
22 end
```

3.4 Summary

This chapter started with an overview of HMM and continued investigating uncertainty models in nucleotide sequences. Subsequently, a new tool called PhiloBacter was introduced, addressing the issue of recombination in the bacterial genome from a probabilistic angle. PhiloBacter consists of two primary components or parts. The first is able to stand independently as a recombination detection tool and is based on the phylogeny signal in the data. The output of this part includes the boundary of recombination events in the genome, the length of each event, the recombination rate, and the substitution rate of the recombination segment of each branch. This part of PhiloBacter determines a score for each nucleotide in the sequence, indicating the probability that the nucleotide belongs to the recombination region. The second part of PhiloBacter deals with the inference of the phylogeny tree in the presence of recombination in the genome. For this purpose, two new approaches have been introduced. Both use the output of the first part of PhiloBacter, the probability values assigned to each nucleotide. In the first approach, for calculating the likelihood, the probability matrices are replaced by mixture probability matrices which point to the presence of recombination on each branch. In other words, different evolutionary parameters are used for various sites of one branch. In the second approach, it is assumed that the recombination can lead to some uncertainty in the genome. One of the available uncertainty models in DNA sequences has been used to model recombination events in bacterial genomes. Then the tree inference utilizing the uncertainty is made. As a result, the effect of recombination on the length of the branches as well as on the topology of the tree decrease. In chapter five, the results of PhiloBacter on simulated and experimental datasets will be examined, and the results will be compared with Gubbins and ClonalFrameML methods. For this reason, it is necessary to introduce a simulation tool, BaciSim, that produces

the desired simulated data in the next chapter.

Availability The code is available at the GitHub link
(<https://github.com/nehlehk/PhiloBacter>)

CHAPTER FOUR

RESEARCH METHODOLOGY: BACISIM

The main goal of this chapter is to introduce a new tool, BaciSim, which simulates the whole genome of organisms with homologous recombinations in their genome. In Section.4.1, we talk about simulators and their role in computational biology. In Section 4.2, we have a quick review of related work in this area. In Section.4.3, BaciSim's detail is explained. The discussion will be in Section 4.4.

4.1 Introduction

Computer programs for simulating sequences have traditionally been employed in population genetics. In particular, Simulators are fundamental and challenging tools in computational evolutionary biology because the evolutionary history of the studied organisms is generally unknown, and it's impossible to predict the genetic subsequent of most processes analytically [129], [130]. Simulators are suitable testbeds for validating newly developed methods and comparing different algorithms in silicon. The availability of sophisticated and customizable simulation software can make simulation an available option for researchers in many fields [129], [130].

The critical applications of simulations are divided into three groups: predictive, statistical inference and evaluation of statistical evolution algorithms [130].

The advent of new technologies for genome sequencing has led to the generation of large amounts of genomic data from bacterial populations, increasing the demand for flexible simulations by which models and hypotheses can be efficiently tested in light of experimental observations [131]. Nevertheless, very few bacterial population genetic simulators exist, and

those that do exist do not cover many possible scenarios.

This study needs a simulator to employ all its primary applications. First, to predict recombination events in the bacteria genome, to statistically infer the phylogenetic tree, and finally, to evaluate our inferred phylogenetic tree, we need to simulate data whose parameters are known to us. In practice, a simulator helps us assess an inference method's accuracy and efficiency to investigate its ability to extract predefined and known events. We evaluate the validity of our proposed method (PhiloBacter) using comprehensive simulation datasets and compare it to the other state-of-the-art related algorithms.

To this aim, we have developed a new simulation, BaciSim, that simulates bacterial genomes using a specified phylogenetic tree to offer both mutation and recombination events, like bacterial population structure. BaciSim is suitable for bacterial species whose homologous recombination plays a significant role in their genome arrangement. *Bacillus cereus* is an example of bacteria that can simulate via BaciSim.

Before we describe BaciSim's details, we review the existing methods and tools recently developed for bacterial genome simulation.

4.2 Background

In the last decade, some tools have been made to simulate the bacteria genome, each of which is relatively practical and useful. In general, simulators can be categorized into two types:

- **Coalescent simulation:** In the coalescent simulator, also known as backwards-in-time, the starting point is the current population. The simulation's direction is backward in time and coalesces individuals together until their most recent common ancestor is found [132], [133].

- **Forward simulation:** As the name of these simulators suggests (also named individual-based simulation), the direction of simulation is forward; i.e., they maintain a population of individuals and simulate the next generation by sampling the current one and moving forward in the same way over time [132], [133].

Coalescent simulators are generally faster than forwarding simulators because they only consider the current population, but the forward one provides more flexibility in the model definition.

SimBac [132] implements an efficient coalescent-based tool for simulating the whole bacterial genome and has a general model of bacterial recombination that allows the user to define the recombination rate.

FastSimBac [133] estimates the coalescent with the bacterial sequential Markov coalescent (BSMC). It is based on the SMC [134], [135] idea, models the clonal frame, and simulates the coalescent and recombination processes along the genome dependent on the clonal genealogy.

MSPro [136] is a coalescent-based simulator that is memory-efficient and fast enough but can only model a limited range of evolutionary models.

CoreSimul[137] uses a forward-in-time approach that simulates prokaryotes' genome with homologous recombination. Simulations are driven by a phylogenetic tree and include different substitution models and codons selection.

SLiM [138] includes two types: Wright-Fisher (WF) models and non-Wright-Fisher (nonWF) models. The Wright-Fisher model is founded on multiple simplifying hypotheses that are usually incompatible with realistic ones, like structured populations or overlapping generations. In contrast, The nonWF framework is individual-based and more realistic. It provides an opportunity to simulate more scenarios and estimate more parameters. Therefore,

it gives results for the same scenario for both models to ensure that they conduct similarly.

All these tools can simulate different evolutionary scenarios for bacterial genomes and are relatively effective and fast. But, to evaluate our method, we need more details about recombination events, such as: which internal node has experienced recombination, the exact location of recombination events, and local trees, especially in the case of external recombination. Since these tools couldn't provide all our requirements, we developed a new simulator to meet our goals. Here we present BaciSim, an efficient backward-in-time simulator to model bacterial genome evolution with homologous recombination along phylogenetic trees (<https://github.com/nehlekh/BaciSim>).

4.3 Methods

Regardless of different kinds of recombination (conjugation, transformation, and transduction), prokaryotes and especially bacteria could experience recombination in another aspect: Internal and external recombination. In the former case, homologous recombination happens with the lineage of the same species, while in the latter case, the lineage of external species imports to the genome [136].

This study concentrates on external recombination. Internal recombination is not a straightforward procedure to detect because it typically does not result in a high level of polymorphism but does lead to homoplasy and genetic incompatibility.

In contrast, the main effect of external recombination is increasing polymorphism. The recombined fragments contain a higher number of substitutions that are not observed in the clonal frame [91], [136].

The arrival of these external partial sequences to the genome substantially impacts branch lengths and can also change the tree topology [87]. Indeed, the external species should

ultimately coalesce with the common ancestor of the clonal frame (on the timescale of species divergence), which would be much older than the MRCA of the clonal species [136].

Homologous recombination events lead the bacterial genomes to have mosaic evolutionary histories where the single and global tree cannot illustrate the underlying evolutionary pattern of biological sequences. However, a collection of clonal and local trees which simulates various evolutionary schemes for every segment of the genomes might represent them better. Some parts of these local trees have overlapped with the clonal tree to show the vertical descent of DNA. At the same time, recombination events made the other parts of these trees utterly different in terms of topology or evolutionary rates [20]. Figure 4.1 show the schematic view of clonal and local trees.

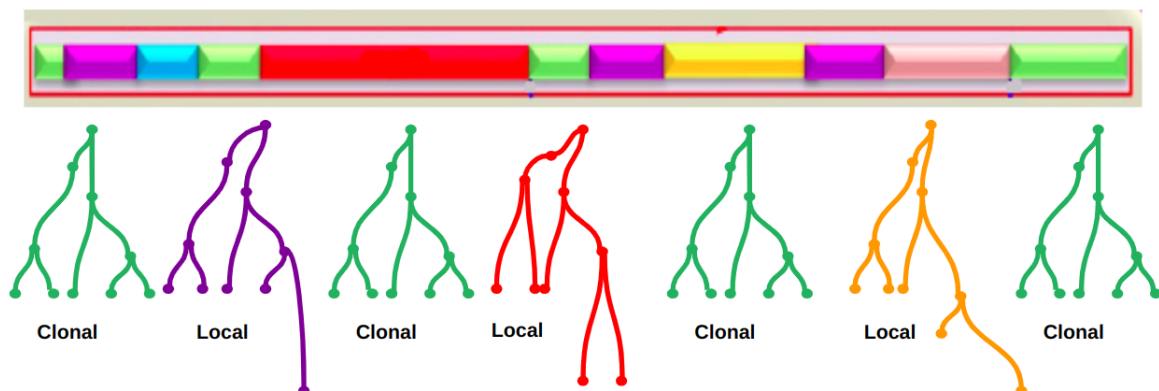


Figure 4.1 A collection of clonal and local trees which simulates various evolutionary schemes for every segment of the genomes might represent mosaic evolutionary histories.

4.3.1 Overview of BaciSim Simulator

Hoban et al. [130] suggested a flow diagram of the steps in a simulation study that is illustrated in Figure 4.2. Our simulator used the concept of clonal and local trees to generate the mosaic pattern of bacteria sequences. The simulated datasets were produced as described below:

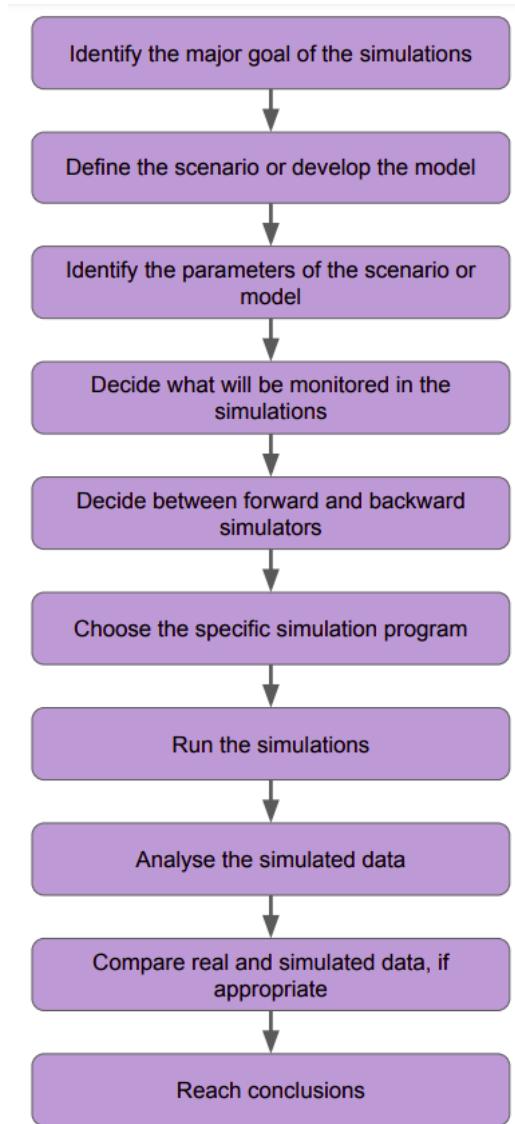


Figure 4.2 Overview of simulation studies. A flow diagram of the steps in a simulation study [130].

- We generate a random tree (τ) of N sample of bacteria under the Kingman's coalescent process [139] as a clonal frame tree. However, the user can also provide a custom phylogenetic tree to provide the option for simulation of scenarios inferred from real bacterial populations [140].
- We assume that the recombination rate is $\rho/2$ on each branch of the clonal tree, and the sum of branch lengths is T . Therefore the total number (R) of recombination events based on a Poisson distribution is [20] :

$$R|\tau, \rho \sim \text{Poisson}(\frac{\rho T}{2})$$

- We generate R numbers of local trees in which the length of one of the tree branches (the branch is chosen randomly) due to recombination is much greater than the same branch in the clonal tree to show the high level of polymorphism. We use a non-uniform normal random distribution to choose that branch. Indeed, the chances of deeper branches undergoing recombination are higher than the other branches.
- We also assume that the recombination events are uniformly distributed throughout the sequence, whereas their length is geometrically distributed with mean δ [20].
- To select the location of the recombination event along the genome, we start from the first site and randomly select the event's location and move toward the end of the alignment.
- We use the local and clonal genealogies created in the previous steps as an input to Seq-gen [141] software to generate the alignment. We used the GTR mutation model to simulate our target sequences.

4.4 Discussion

We developed a new tool to simulate bacterial genomes with external homologous recombination along a phylogenetic tree. This work is the first version of our simulator, which is somewhat limited, as we only focused on evolutionary parameters related to external recombination events. Even so, our tool has capabilities that other methods do not provide. These features are required to evaluate recombination detection tools:

1. Clonal tree

Procedure: BaciSim: Simulation of bacterial genomes with external homologous recombination

Input: N :genome number, L :genomelen, δ :recomlen, $tMRCA$, ρ :recomrate, v

Output: A Clonal tree

A collection of local trees represents different recombination events along the alignment.

An alignment

1 **Initialization:**

2 A random tree (τ) of N sample of bacteria under the Kingman's coalescent

3 $R = \text{Poisson}(\frac{\rho T}{2})$ // the number of recombination events

4 **begin**

5 **while** $R > 0$ **do**

6 Select a random edge of the tree and make it longer using v

7 Set Recombination Length using a geometric distribution with δ mean

8 $R = R - 1$

9 Forget this event and make the new one

10 **end**

11 Alignment = Seq-gen(Clonal tree, local trees)

12 **end**

2. Recombination trees

3. Recombination Location (which node and leaves or internal nodes)

4. Recombination Parameters:

a) Recombination rate

b) Substitution rate of DNA imported (v),

c) tMRCA (time of Most Recent Common Ancestors),

d) Recombination length

5. Graphic Output (Figure 4.3 is an example of the graphic output.)

Forthcoming studies could further investigate variation in the evolutionary scenarios and different kinds of recombinations. We hope that our simulator can continue to be used to benchmark novel methods for other organisms than bacteria as well.

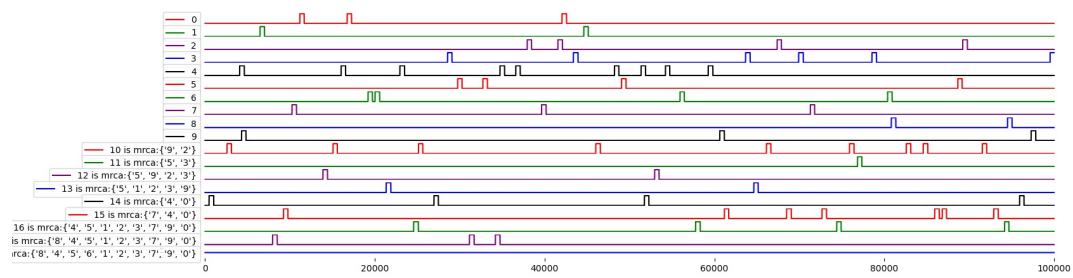


Figure 4.3 An example of the graphic output of BaciSim shows recombination events, their location and length along the alignment.

CHAPTER FIVE

RESULTS AND DISCUSSIONS

This chapter aims to demonstrate that our method -PhiloBacter- generates reliable estimates of the clonal tree and reasonably identifies recombination regions. In the rest of this chapter, we first introduce the pipeline, which automates our evaluation and comparison. In section 5.2, the evaluation metrics have been discussed. In section 5.3, different experiment result has been shown to evaluate the performance of our new method. Section 5.4 investigates a real dataset and discusses the results in section 5.5.

5.1 Pipeline introduction

We present two tools in this thesis: PhiloBacter and BaciSim. Although PhiloBacter consists of two tools, the first can be used independently as a recombination detection tool. To simplify the usage of these tools, validate their result, and compare them to the other state-of-the-art methods, a pipeline has been built using Nextflow [142] software.

5.1.1 Installing the pipeline

Here, walk through the steps to install and set up the PhiloBacter pipeline on your system. Please follow the instructions carefully to ensure successful installation.

5.1.1.1 Prerequisites:

- Make sure you have Git installed on your system.
- Ensure you have Conda installed.

5.1.1.2 Installation Steps:

- Clone the PhiloBacter repository:

```
$ git clone https://github.com/nehlekh/PhiloBacter.git
```

- Navigate to the cloned directory:

```
$ cd PhiloBacter
```

- Install Nextflow via Bioconda:

```
$ conda install -c bioconda nextflow
```

- Set Up the PhiloBacter Environment: The PhiloBacter pipeline should ideally create its Conda environment automatically. However, if you face issues or if Nextflow doesn't appear to create the environment as expected, you can manually set it up: a. Create the environment using the provided PhiloBacter.yml file:

```
$ conda env create -f PhiloBacter.yml
```

- b) b. Activate the environment:

```
$ conda activate PhiloBacter
```

The pipeline is now ready to run. The application of this pipeline can be classified into two modes (Figure 5.1-a). The first mode is for all users who have sequences of a group of bacteria and are interested to learn about recombination events or want to have an authentic and reliable phylogenetic tree of those bacteria not impacted by recombination. These users can choose the analysis mode to use the pipeline.

5.1.2 Analysis mode

In this case, the pipeline consists of two steps. The first step builds an initial tree from the sequence using RAxML [124]. In the second step, suppose the user is only looking for recombination events and their boundaries (by utilizing the step, HMM, mentioned in section 3.3.2) in the genome or, in other words, looks for the mosaic pattern of the genome. In that case, the pipeline can respond to this request accurately. In addition, the other option is also available if the user is interested in the phylogeny tree. However, the user does not need to set any parameters; if one runs the pipeline using the default settings, all the steps will be done automatically. The bonus capability is that this pipeline is not only specific to PhiloBacter. This is possible if the user also wants to use two other well-known tools in this field, such as ClonalFrameML [91] and Gubbins [87] (although Gubbins only needs alignment as input), and collect the final trees of all three methods. Here is the command to use the pipeline for real datasets (Figure 5.1-b).

```
$ ./nextflow main.nf --mode Analysis --seq genome.fasta --method  
pb, cfml, gub
```

"--seq" provides an option to introduce the desired aligned sequence (fast format) to the pipeline, and "--method" is used to specify the method for the analysis.

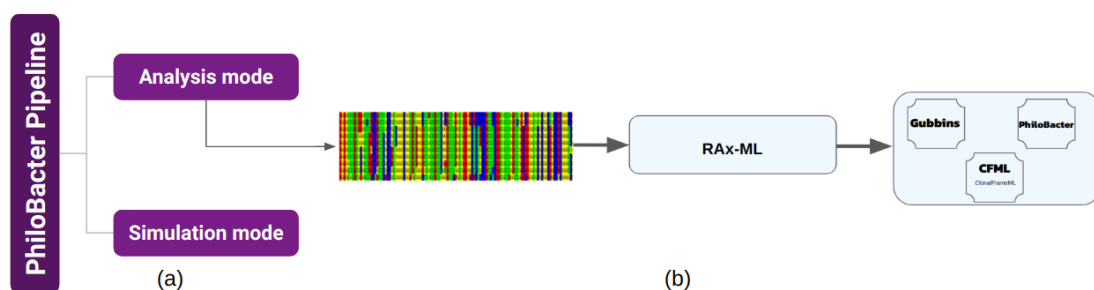


Figure 5.1 A schematic view of the pipeline modes. a) shows the pipeline has two modes; b) shows the steps of the analysis mode.

5.1.3 Simulation mode

The second mode of the pipeline is specified for experts and developers of recombination detection tools in bacterial genomes who want to compare different approaches. Also, this mode helped us to evaluate the performance of PhiloBacter in a structured manner. Users can choose the simulation mode to use this option. In this case, the pipeline consists of five main steps. Figure 5.2 shows the schematic view of the pipeline.

- **Step 1- Simulation:** The first step is to simulate the clonal and local trees. BaciSim was considered the main simulator. As mentioned in the previous chapter and also seen in Figure 5.2. The output of this step includes:
 - A clonal tree.
 - A set of local trees that implies recombination events.
 - A comprehensive report of recombination events (the start, end of each event, and the recombination host node).
 - A graphical representation of recombination events showing the length and location of events along the genome.
- **Step 2- Generating Sequences:** To generate the alignment, we use the local and clonal trees created in the last step as input to Seq-Gen [141] software. The default evolution model is GTR. The user can provide a scaling factor to shrink or expand the branch lengths of the tree. It's also possible to provide user-defined base frequencies and substitution rate parameters for the GTR substitution model.
- **Step 3- Constructing an initial tree:** We have used RAxML [124] to build the initial tree based on the sequences that are the output of Seq-Gen. However, as we mentioned

in Chapter 3, using another tool, such as IQ-Tree [93], is possible instead of RAxML for this step.

- **Step 4- Recombination detection and tree inference:** In this step, in addition to PhiloBacter, Gubbins and CFML tools can be run on the same simulated dataset. The output of this step is recombination events and tree inference.
- **Step 5- Analysis:** The last step of the pipeline consists in evaluating recombination events and phylogenetic tree estimates. Specifically, inferred trees are compared to the clonal (true) tree. For this purpose, some evaluation metrics are required, which will explain in the next section.

There are various parameters for data simulation that can be adjusted according to the user's requirement, such as genome and recombination length or recombination rate, and time to the most recent ancestor (tMRCA).

```
$ ./nextflow main.nf --mode sim --genome 10 --genomelen 100000  
--recomlen 500 --tMRCA 0.01 --recomrate 0.01 --nu_sim 0.05
```

To illustrate the functionality and robustness of our comparison, for this step, we used not only our simulator (BaciSim) but also we utilized two other well-known simulators: FastSimbac [133] and SimBac [132]. We used different simulation scenarios to quantify the accuracy of our method. In each scenario, we tried to consider the variation of one parameter while keeping the other parameters fixed. For instance, the effect of recombination length (δ) is investigated when recombination rate, ν and tMRCA are fixed.

The analysis mode is composed of the third and fourth steps of the simulation mode.

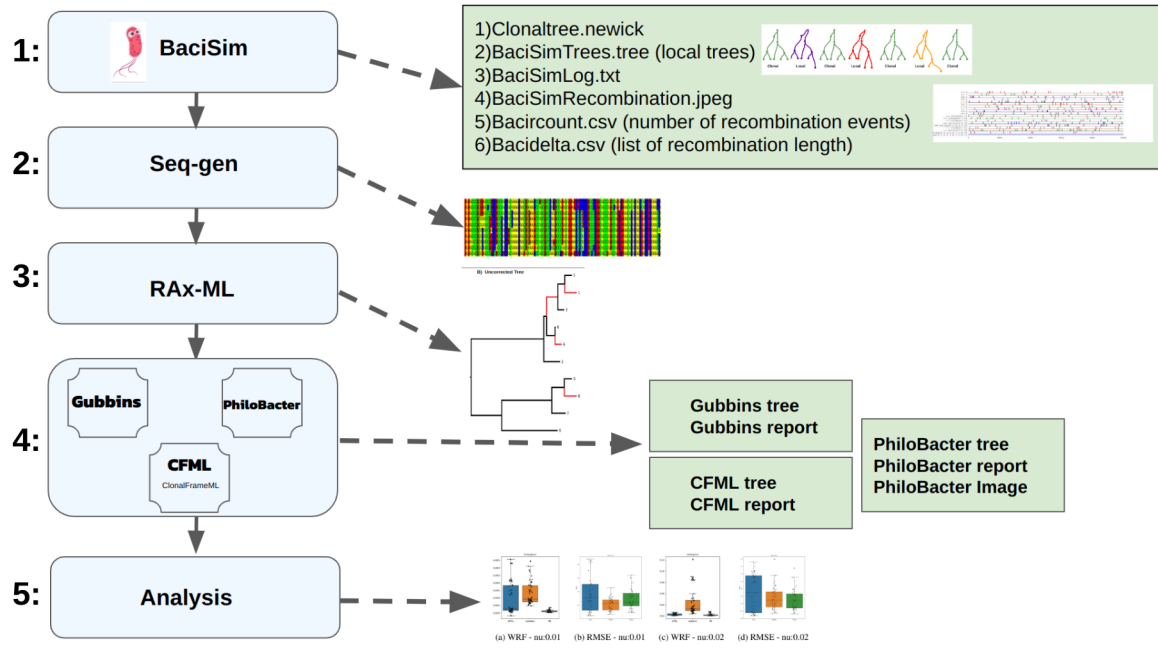


Figure 5.2 A schematic view of the pipeline. Detailing its five primary stages: (1) Simulation, (2) Sequence Generation, (3) Initial Tree Construction, (4) Recombination Detection and Tree Inference, and (5) Analysis.

5.2 Evaluation Metrics

Evaluation metrics play a crucial role in assessing the performance of a new method. There are a lot of metrics to measure the performance of different statistical models. Standard criteria often work well for most problems. However, the evaluation criteria should be chosen to best show the problem's different dimensions and what is most important about the model.

Since PhiloBacter consists of two relatively different problems, different evaluation metrics are also required to evaluate each part.

5.2.1 Evaluating Recombination Detection

PhiloBacter recombination detection can be considered either a classification model or a regression one (machine learning algorithms categories). The simulated data can be regarded as the actual labels (value), and the estimated results by PhiloBacter consider predictions. An

acceptable statistical learning method is one where the difference between the actual and estimated values in a given data set is slight. So, we must measure how accurately our estimation matches the correct data.

5.2.2 PhiloBacter as a regression model

PhiloBacter output for each alignment site is a continuous value, making it possible to consider it a regression problem. So, we need a metric based on calculating the distance between the estimated and ground truth. We believe the probability of recombination sites in simulated data is 1 and clonal sites 0 (our ground truth). There are different metrics to measure this distance, like Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) which RMSE somehow covers other metrics as well:

$$\text{RMSE} = \sqrt{\sum_{i=1}^n \frac{(z_i - y_i)^2}{n}}$$

In the current study, z_i would be the probability of recombination for the i th position of the alignment. Where y_i (actual value) indicates whether the i th site of the simulated dataset is recombination or not, also, It should be mentioned that n is the sequence alignment length.

Forasmuch as the RMSE indicates how close the actual data points are to the estimated values, the smaller value for RMSE is, the better the estimates are.

5.2.3 PhiloBacter as a classification model

As we mentioned in the previous section, the output of PhiloBacter for each alignment site is a continuous value, which can be converted to zero and one by considering a threshold. In such a case, we have a classification problem and can use classification metrics to evaluate the model. The positive point of converting PhiloBacter to classification is that comparing it

with Gubbins and CFML methods would be more reasonable because the output of those two methods can only be assigned zero values and one for the recombinant and clonal classes.

Classification models have discrete outputs, so a metric is needed to compare discrete classes.

Accuracy : The ratio of correct predictions to the total data.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP is the number of true positives, In this thesis, it means the number of identified recombination sites that are recombination in the simulated data. TN is the number of true negatives (the number of estimated clonal sites that are indeed clonal sites in the simulated data). FP is the number of false positives (the number of identified recombination sites that are clonal in the simulated data). FN is the number of false negatives (the number of identified clonal areas that are recombination in the simulated data.)

High accuracy often indicates good model performance, but not always, so other metrics are needed to evaluate the model.

Precision is the ratio of positive cases that were truly identified.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall is the ratio of real positive cases that are correctly detected.

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1-score : The F1-score metric combines precision and recall. In fact, the F1 score is the harmonic mean of the two. The formula of the two essentially is:

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The higher the F1 score, the higher the prediction power of the classification model. When F1 is close to 1 means, we have a perfect model.

5.2.4 Evaluating Phylogenetic reconstruction

The typical way to evaluate the performance of the phylogenetic reconstruction algorithm is by calculating the similarity against the correct tree generated by the simulator. Different metrics have been defined to compute the distance between a pair of trees. Metrics are divided into rooted or unrooted and topological or weighted (using branch lengths). However, the most critical issue in this thesis is the branch lengths of the inferred tree from the bacterial genome. Since the final tree in our method is unrooted, here we only describe metrics for the unrooted trees that also include branch length:

Weighted Robinson-Foulds :

Given two trees, T_1 and T_2 , the Robinson-Foulds (RF) adds the number of partitions that exist in tree T_1 (but not T_2) to the number of partitions present in tree T_2 (but not T_1). This distance is one of the most popular metrics in tree comparison [143].

The weighted Robinson-Foulds distance is the aggregate of rootward branch lengths for entire nodes in a clade appearing in only one of the trees. This metric adds the edge length to the RF metric [143].

Euclidean distance (Branch score) : The simplest branch score quantifies two trees' absolute differences in branch lengths [144].

Note: In computing the Euclidean and Robinson–Foulds distances, for the absent branch, that length would be set to zero; consequently, the difference in the length of that branch will be maximum.

5.3 Performance Evaluation

This section presents the results of employing PhiloBacter, Gubbins and CFML on several simulated datasets. The common point of all datasets is that they all consist of 10 genomes, the length of which is 100,000 base pairs. To show the robustness of PhiloBacter and make a better comparison, we first used BaciSim to simulate the data, and then two other simulators were utilized.

5.3.1 BaciSim Simulator

5.3.1.1 Investigating different values of v

In this series of experiments, we used different values of v between 0.01 and 0.1. The length of recombination events here is fixed and 500. tMRCA of the clonal genealogy and the recombination rate is 0.01. And each of these scenarios has been repeated ten times.

Figure 5.3 visually represents the RMSE values for the given datasets. Lower values of RMSE indicate better predictive power. Essentially, the closer an RMSE value is to zero, the more effective the predictive model is considered to be.

From the figure, it can be discerned that PhiloBacter relatively consistently yields the lowest RMSE values across a range of scenarios. This observation suggests that PhiloBacter exhibits superior predictive performance compared to the other methods tested. Specifically,

PhiloBacter's RMSE values remain consistently low for varying parameter ν values, signifying that it outperforms the competing methods under various circumstances.

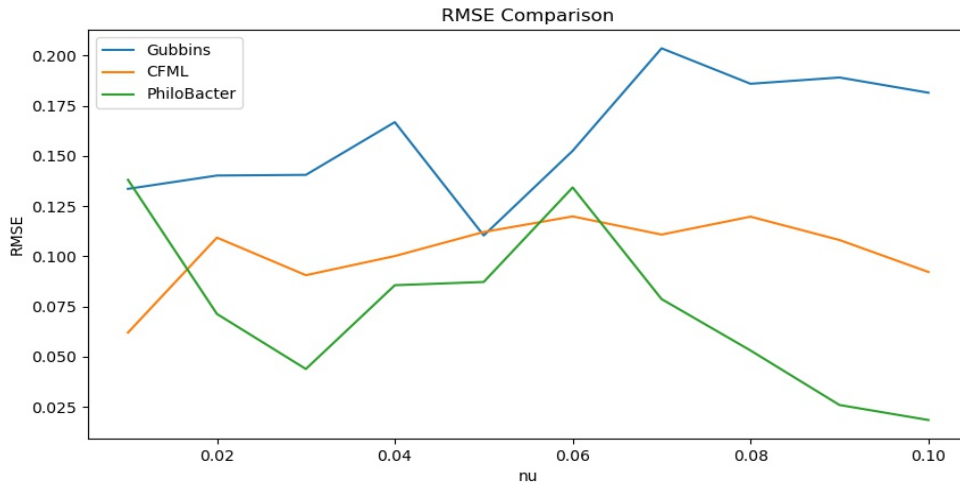


Figure 5.3 RMSE for different value of ν of three methods: ClonalFrameML, Gubbins, PhiloBacter - Number of genome:10 - Alignment length:100K - tMRCA:0.01 - Recombination length:500

Figures 5.4 and 5.5 present the computed Accuracy and F1-Score values for the datasets under consideration. These metrics are crucial indicators of a model's predictive power - the closer their values are to one, the stronger the model is at prediction.

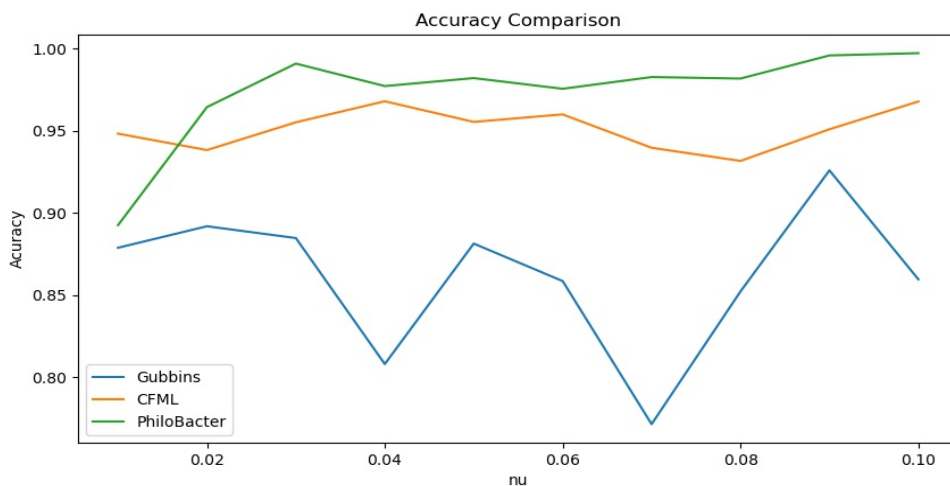


Figure 5.4 Accuracy for different values of ν for three methods: ClonalFrameML, Gubbins, PhiloBacter - Number of genome:10 - Alignment length:100K - tMRCA:0.01 - Recombination length:500

Upon analyzing the figures, it becomes evident that the performance of PhiloBacter stands out. Except for scenarios where ν equals 0.01, PhiloBacter consistently registers the highest Accuracy and F1-Score values in all other instances. This suggests that PhiloBacter outperforms the other models in predictive capability under various conditions.

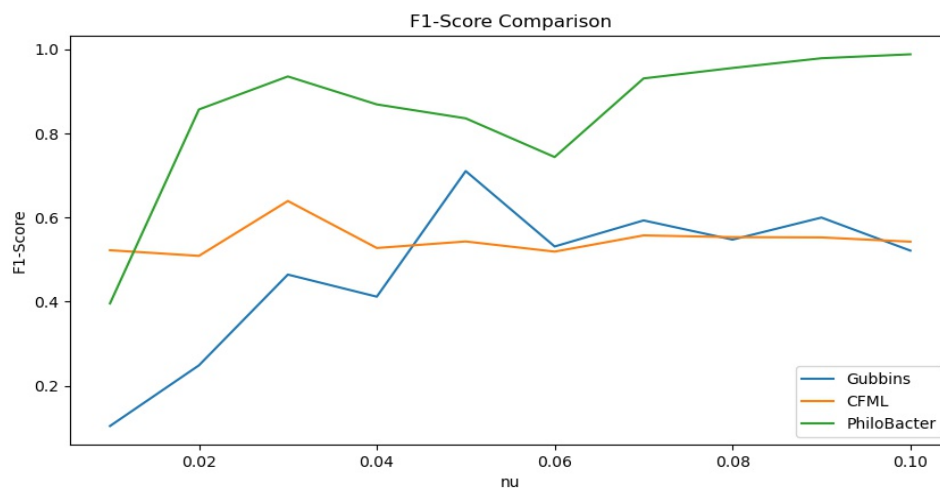
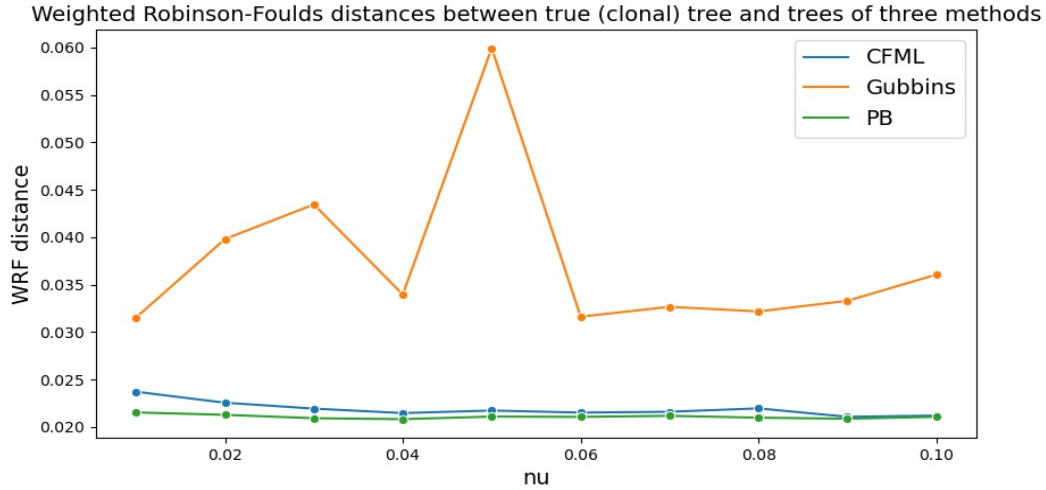


Figure 5.5 F1-Score for different values of ν for three methods: ClonalFrameML, Gubbins, PhiloBacter - Number of genome:10 - Alignment length:100K - tMRCA:0.01 - Recombination length:500

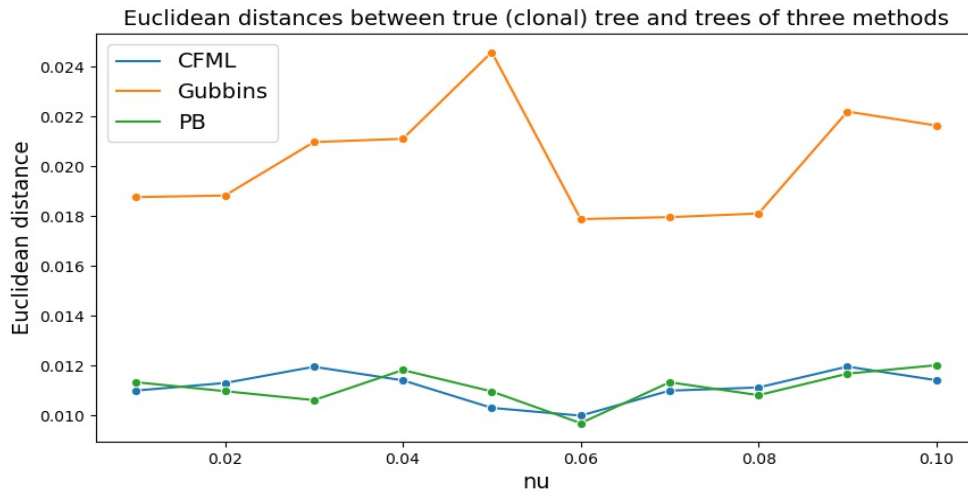
Figure 5.6 illustrates the divergence between the true clonal genealogy and the trees inferred by CFML, Gubbins, and PhiloBacter. This divergence, or distance, is evaluated based on two distinct metrics: the Weighted Robinson-Foulds metric and the Euclidean distance.

To assess the divergence of the Gubbins-inferred tree, we had to undertake a rescaling of its output tree. This process was necessary to establish a common ground for comparison among all the models.

The analysis reveals interesting insights. For all values of ν , the trees generated by PhiloBacter exhibit a smaller Weighted Robinson-Foulds distance from the true tree (derived from simulated data) than the CFML-inferred tree. This indicates that PhiloBacter more accurately captures the structure of the true tree under this particular metric.



(a) Weighted Robinson-Foulds distances between true (clonal) tree and trees of three methods



(b) Euclidean distances between true (clonal) tree and trees of three methods

Figure 5.6 **BaciSim Simulator**: Distances between true (clonal) tree and trees of three methods: ClonalFrameML, Gubbins, PhiloBacter - Number of genome:10 - Alignment length:100K - tMRCA:0.01- Recombination rate:0.01 - Recombination length 500

On the other hand, the situation becomes less clear-cut when evaluating the divergence based on Euclidean distance. The data shows variability: for some ν values, the CFML model exhibits a smaller distance from the true tree, signifying superior performance. For other ν values, both models display equal performance, and for the remainder, PhiloBacter outperforms with a smaller Euclidean distance. This variation underscores the importance of examining multiple evaluation metrics when comparing model performance in phylogenetic reconstruction.

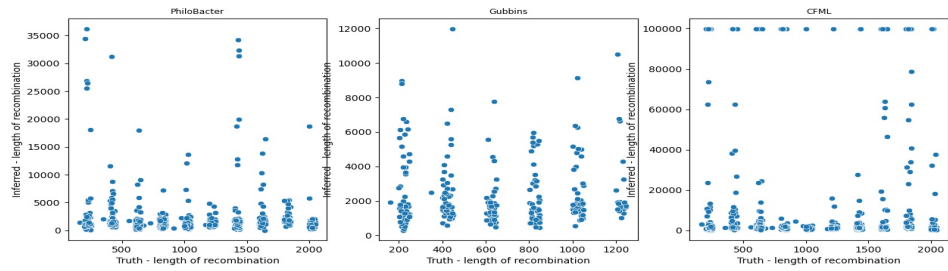
tion.

5.3.1.2 Investigating different recombination lengths

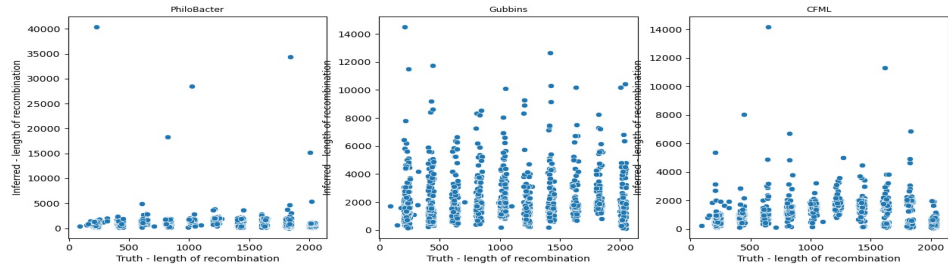
We conducted an additional experiment to examine the recombination length estimation. In this experiment, the recombination length in the simulated data varied within the range of [200,400,600,800,1000,1200,1400,1600,1800,2000]. Each scenario within this range was repeated five times. The outcomes of this experiment, considering the values of ν 0.01, 0.03, 0.05, 0.07, and 0.09, are visualized in Figure 5.7.

The first key observation drawn from this figure is that Gubbins tends to overestimate the length of the recombinant region considerably when compared to the actual size. Additionally, when the value of ν is set to 0.01, CFML predicts numerous recombination events that span the entire genome's length.

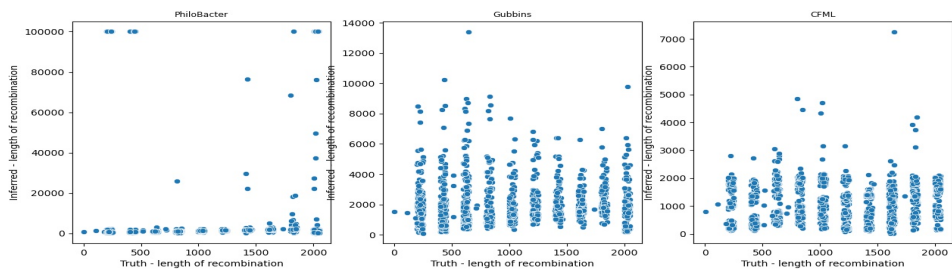
PhiloBacter, on the other hand, while not perfectly accurate in estimating the length of recombination events, generally exhibits a more measured approach. In specific scenarios, its estimates exceed the actual size by several multiples. However, a closer look at Figure 5.7 reveals that, on balance, PhiloBacter outperforms Gubbins and CFML in estimating the size of the recombination events. While there is room for improvement, PhiloBacter provides a more accurate representation of the recombinant region sizes than the other two methods in the context of this experiment.



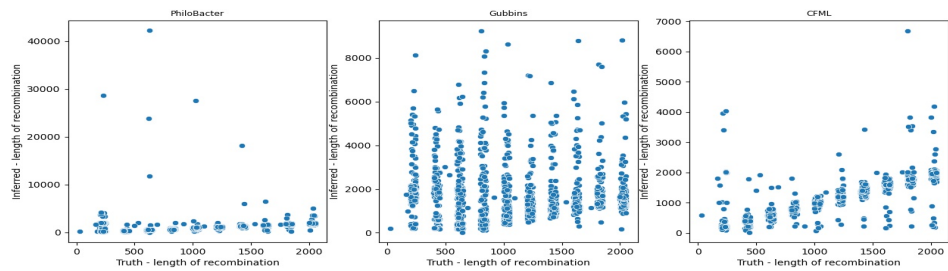
(a) $v : 0.01$



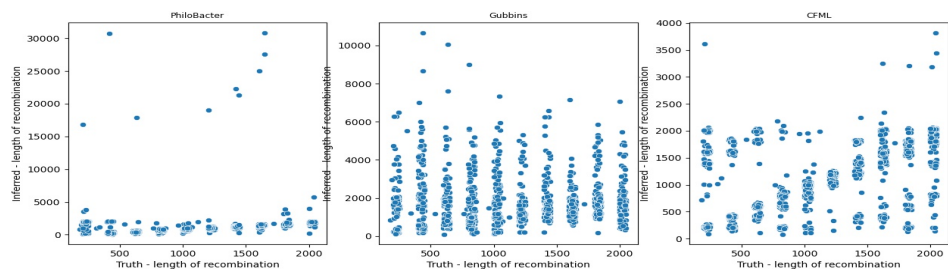
(b) $v : 0.03$



(c) $v : 0.05$



(d) $v : 0.07$



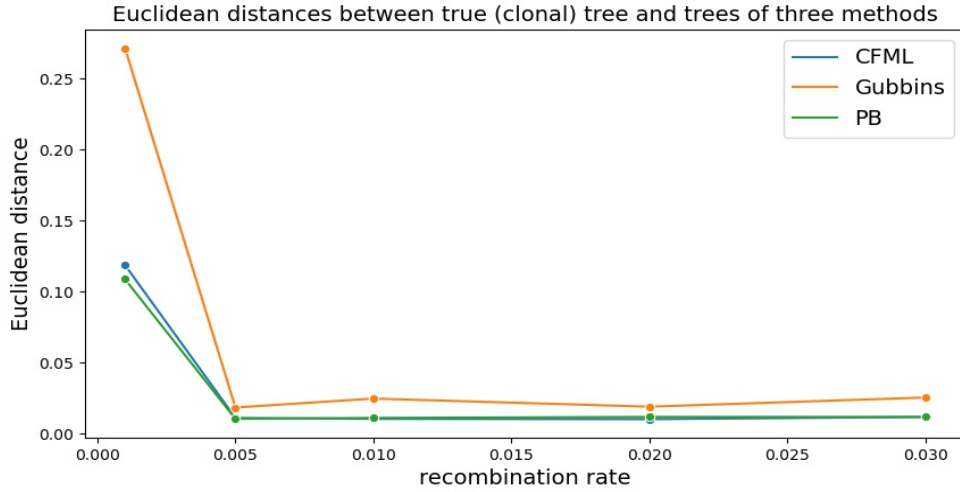
(e) $v : 0.09$

Figure 5.7 **BaciSim Simulator**: Investigating how accurate each method (PhiloBacter, Gubbins and CFML from left to right) can estimate different intervals of recombination length. Number of genome:10 - Alignment length:100K - nu:0.05- tMRCA:0.01

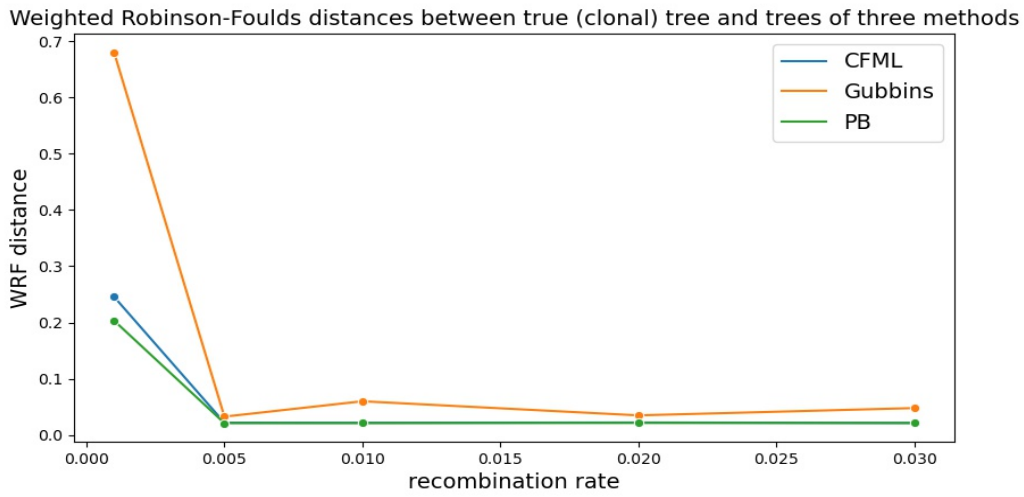
5.3.1.3 Investigating different recombination rate

In this set of experiments, we manipulated the recombination rate, selecting values from the set 0.005, 0.01, 0.02, 0.03. The length of the recombination events was kept constant at 500 nucleotides, while the time to the most recent common ancestor (tMRCA) and ν were fixed at 0.01 and 0.05, respectively. Each scenario was run through ten repeated trials. The outcomes of these comparative experiments are illustrated in Figure 5.8.

This figure highlights some intriguing trends. When the recombination rate is low, Gubbins' results significantly diverge from those produced by CFML and PhiloBacter, which demonstrate more consistency between each other. However, as the recombination rate increases, the disparity between the three methods narrows, indicating that all three perform comparably under high recombination rates. This comparison provides valuable insights into the performance variations of these phylogenetic methods across different recombination rates.



(a) Euclidean distances between true (clonal) tree and trees of three methods



(b) Weighted Robinson-Foulds distances between true (clonal) tree and trees of three methods

Figure 5.8 **BaciSim Simulator**: Distances between true (clonal) tree and trees of three methods **for different recombination rate**: ClonalFrameML, Gubbins, PhiloBacter- Number of genome:10 - Alignment length:100K - nu:0.05- tMRCA:0.01 - Recombination length 500

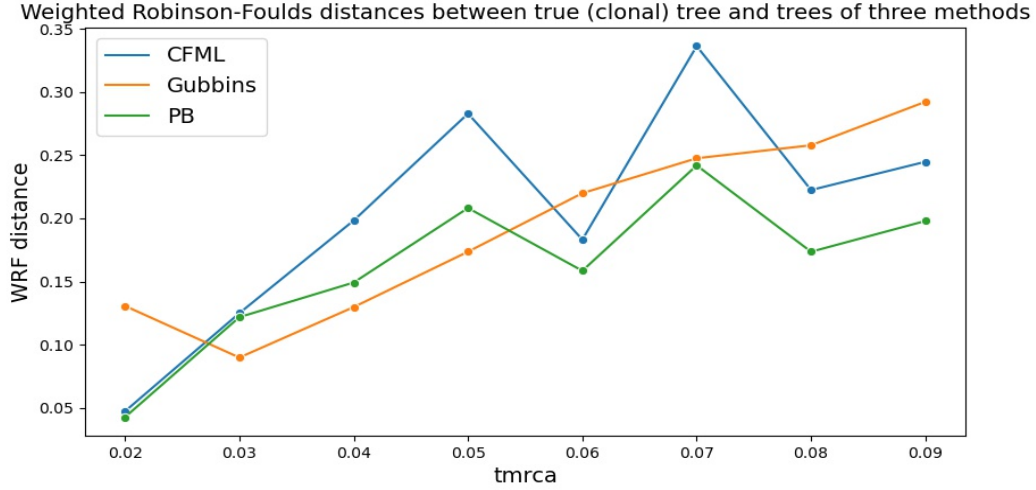
5.3.1.4 Investigating different tMRCA

The final parameter we explored for variation is tMRCA. We created a range of scenarios with tMRCA values spanning from 0.02 to 0.1. In these cases, the recombination event length was held constant at 500 nucleotides, while the recombination rate and v were set to 0.01 and 0.05, respectively. Each of these scenarios was replicated ten times to ensure robustness in

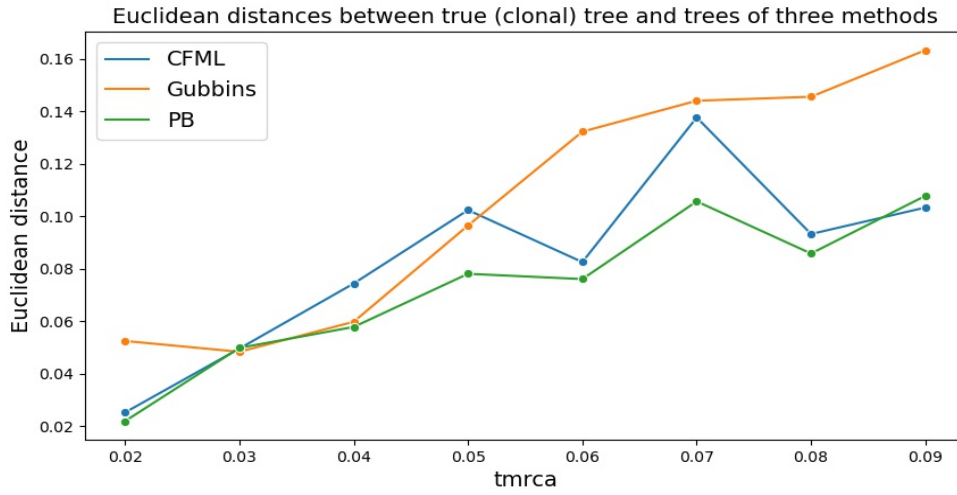
our findings. The results are visualized in Figure 5.9.

Upon analyzing the figure, several crucial findings are unveiled. For tMRCA values of 0.03, 0.04, and 0.05, the tree inferred by Gubbins illustrates a smaller Euclidean distance from the clonal tree than those inferred by the other two methods. This indicates that Gubbins displays higher accuracy in these particular scenarios.

However, the scenario alters when we explore different tMRCA values. In this case, the trees inferred by PhiloBacter generally outperform those from Gubbins and CFML regarding both types of distances. PhiloBacter's Weighted Robinson-Foulds (WRF) distance consistently remains the lowest across all the scenarios. This demonstrates PhiloBacter's enhanced capacity to accurately infer phylogenetic trees, maintaining a minimal distance from the true clonal tree regardless of the variability in tMRCA.



(a) Weighted Robinson-Foulds distances between true (clonal) tree and trees of three methods



(b) Euclidean distances between true (clonal) tree and trees of three methods

Figure 5.9 **BaciSim Simulator**: Distances between true (clonal) tree and trees of three methods **for different values of tMRCA**: ClonalFrameML, Gubbins, PhiloBacter - Number of genome:10 - Alignment length:100K - nu:0.05- Recombination rate:0.01 - Recombination length 500

5.3.2 SimBac Simulator

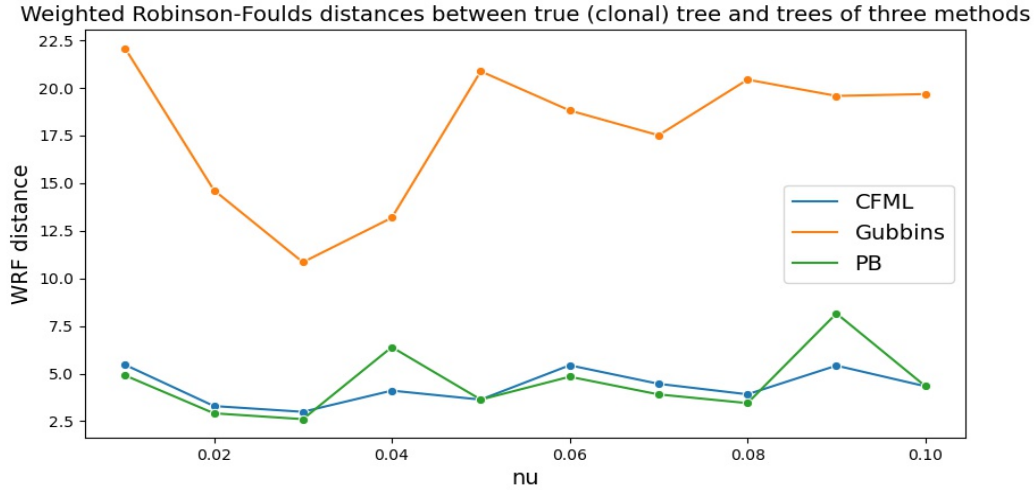
In this section, we utilized SimBac [132], a bacterial genome simulation software, to generate the data sets. One of the limitations of SimBac is that it does not explicitly specify the recombination host node, making it impossible to determine the tMRCA. Because of these constraints, the simulation data does not provide sufficient information to directly compare the

accuracy and lengths of recombination events.

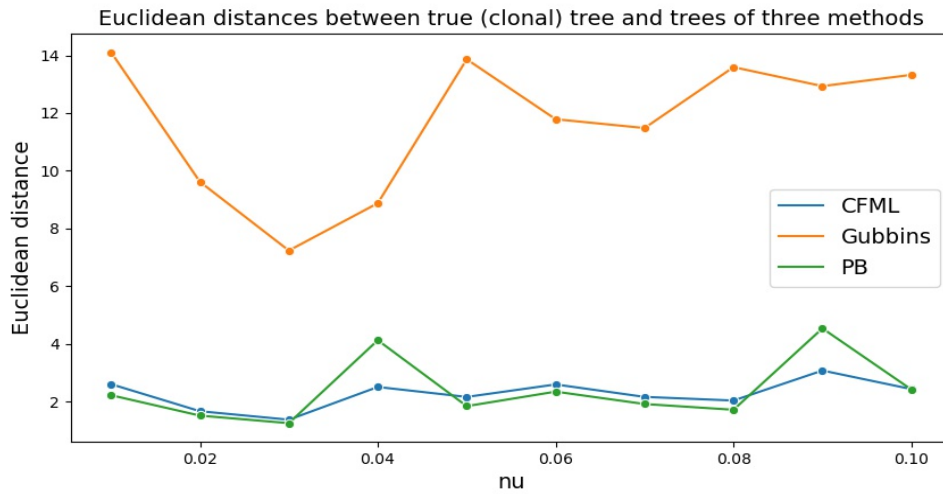
We just focused on computing the distance between the trees inferred by the models and the actual trees the simulator provided.

To ensure the robustness and reliability of our findings, we conducted this experiment ten times, each time with different values of v ranging from 0.01 to 0.1. The recombination length in these experiments was constant at 500 nucleotides, and the recombination rate was fixed at 0.0005.

The findings from this set of experiments are presented in Figure 5.10. The distances derived from the Gubbins method differ significantly from those of CFML and PhiloBacter, suggesting a considerable variation in its performance. On the other hand, CFML and PhiloBacter exhibit similar performance, as evidenced by their comparable distances. However, upon close examination, minor differences become apparent. In scenarios involving most values of v , the distances between the PhiloBacter-inferred trees and the clonal genealogy (measured using both Weighted Robinson-Foulds and Euclidean distances) are slightly less than the distances for trees inferred by CFML. Although the differences are subtle, they indicate that PhiloBacter offers a slight edge over CFML in accurately reconstructing phylogenetic trees under various conditions.



(a) Weighted Robinson-Foulds distances between true (clonal) tree and trees of three methods



(b) Euclidean distances between true (clonal) tree and trees of three methods

Figure 5.10 **SimBac Simulator**: Distances between true (clonal) tree and trees of three methods: ClonalFrameML, Gubbins, PhiloBacter - Number of genome:10 - Alignment length:100K- Recombination rate:0.0005 - Recombination length:500

5.3.3 FastSimBac Simulator

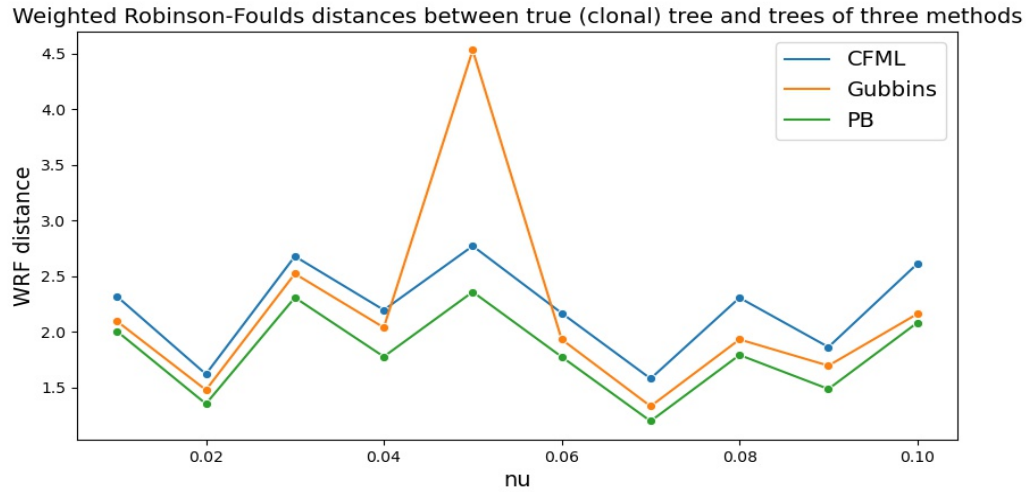
This segment is similar to the experiments conducted in the previous section, but this time using FastSimBac [133] as the simulator for generating the data. FastSimBac, a new version of SimBac, is known for its robust and rapid bacterial genome simulation capabilities. All the settings and parameters, including the v values, recombination length, and recombination

rate, remain identical to the ones employed in the previous experiment.

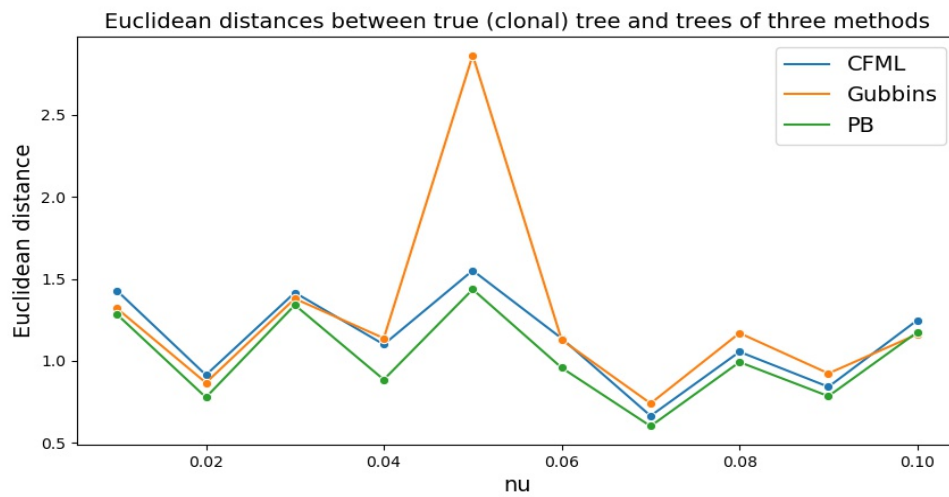
Several key insights emerge upon conducting the experiment and analyzing the results, as shown in Figure 5.11. For instance, when the value of v is set to 0.05, Gubbins, surprisingly, generates a tree that deviates considerably from the clonal tree. The distances between Gubbins' inferred tree and the clonal tree are notably higher than those derived from the other methods, indicating some abnormality in the tree generation process under this specific parameter setting.

On the contrary, for other values of v , the performances of all three methods, as indicated by their distance metrics, are relatively close.

However, despite the similarities, there is a subtle distinction when we delve deeper. The trees inferred by PhiloBacter for all tested scenarios demonstrate a slight edge over the other methods. Regardless of the value of v employed, PhiloBacter consistently generates more accurate trees that adhere more closely to the clonal genealogy. These findings indicate that PhiloBacter, despite the close competition, emerges as a more effective tool for accurate phylogenetic tree reconstruction under a range of conditions.



(a) Weighted Robinson-Foulds distances between true (clonal) tree and trees of three methods



(b) Euclidean distances between true (clonal) tree and trees of three methods

Figure 5.11 **FastSimBac Simulator**: Distances between true (clonal) tree and trees of three methods: ClonalFrameML, Gubbins, PhiloBacter - Number of genome:10 - Alignment length:100K - Recombination rate:0.0005 - Recombination length: 500

5.4 Comparison with fastGEAR

PhiloBacter has thus far been benchmarked against two distinguished tools: Gubbins and CFML. These tools were specifically chosen for their unique ability to detect recombination in bacterial genomes and create phylogenetic trees that take this recombination into account, a capability that closely mirrors the functions of PhiloBacter. However, various tools aim to

identify recombination or craft bacterial phylogenetic trees. However, a limited number of these tools are both highly cited by peers and capable of concurrently managing these complex tasks.

What sets PhiloBacter apart is its comprehensive approach, and in this context, we find it essential to include a comparison with another specialized tool, fastGEAR [84]. Developed exclusively for detecting recombination in bacteria, fastGEAR presents a different approach, and its output exhibits notable differences from those produced by PhiloBacter, Gubbins, and CFML. This section delves into this comparison, highlighting selected examples that emphasize the contrasts and distinctions between these tools.

5.4.1 fastGEAR Simulated Data

In our comparative analysis, we chose a subset of 15 sequences from the dataset that fastGEAR used for its simulations. This created a consistent benchmarking environment, ensuring all tools were evaluated under similar conditions. Subsequently, we subjected this curated subset of sequences to rigorous testing using all four tools: PhiloBacter, Gubbins, CFML, and fastGEAR itself. By doing so, we aimed to comprehensively understand each tool's performance, strengths, and potential areas of improvement when presented with the same dataset.

- FastGEAR identified 25 recent recombination events, indicating occurrences above the tree's external nodes. Furthermore, it pinpointed ten ancestral recombination events. The output from fastGEAR, illustrating these recent recombination events on the tree's tips, can be viewed in Figure 5.12.

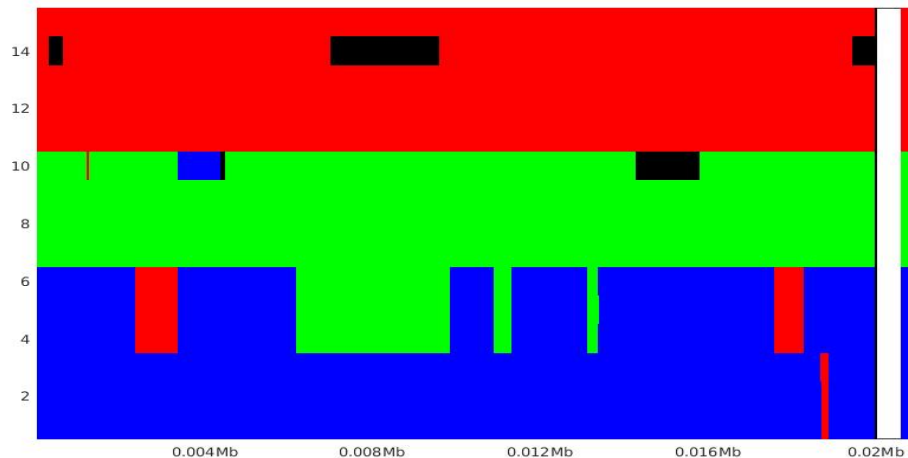


Figure 5.12 The figure visually represents the population's genetic structure inferred by fastGEAR. In this illustration, the rows are aligned with the sequences, the columns correspond to specific positions within the alignment, and various colors depict different populations.

- CFML detected a total of 39 recombination events within the analyzed data. Of these events, only six occurred above the tip nodes, representing recent recombination. At the same time, the remaining 33 took place above the internal nodes, indicative of ancestral recombination events.
- Gubbins identified ten recombination events in the analyzed dataset. Within this discovery, four events were detected above the tip nodes, signifying recent instances of recombination. The remaining six events were observed above the internal nodes.
- PhiloBacter identified 29 recombination events in the same dataset. Within this discovery, five events were detected above as recent recombination events. The other 24 events were observed as ancestral recombination events.

Some intriguing observations arose from this experiment, which merit discussion:

The three methods - PhiloBacter, Gubbins, and CFML - identified recombination events occurring above merely two external nodes. All other detected events were classified as ances-

tral recombination. Contrastingly, fastGEAR pinpointed these two nodes and identified six other external nodes as having recently been recombined. Regarding ancestral recombination events, PhiloBacter, Gubbins, and CFML reported a higher count than fastGEAR. Notably, events recognized as occurring in parents or grandparents of tip nodes by the former three methods were classified as recent recombination events by fastGEAR. The discrepancy between the methods may be attributable to differences in the definition and interpretation of 'recent' versus 'ancestral' recombination events and how the data was simulated. For a comprehensive examination of this experiment, including detailed insights into the sequences, recombination events, and phylogenetic trees as inferred by the three methods, please refer to the available data at the following repository: (https://github.com/nehlekh/PhiloBacter/tree/main/example_data/fastGEAR_example).

5.4.2 BaciSim Simulator

In this section, we engaged BaciSim, to craft scenarios using simulated data. This allowed us to generate controlled conditions, providing a robust framework for our analysis and enabling a more comprehensive understanding of the underlying processes and patterns.

5.4.2.1 Detecting Recent Recombinations

We've incorporated a specific feature in our simulator, BaciSim, allowing for the exclusive generation of recombination events above external nodes. To enable a more straightforward comparative analysis with fastGEAR, we executed a series of experiments focusing solely on these recent recombination events. Key settings were adjusted to specifications such as a v value of 0.03, a tMRCA for the clonal genealogy, and a recombination rate set at 0.01. Additionally, we varied the length of the recombination events within a defined range, spanning

from 500 to 1500 in increments of 100. Table 5.1 presents selected outputs from these experiments, specifically focusing on the 500, 1000, and 1500 recombination lengths. The table's left column lists the number of recombination events generated in the simulation data via BaciSim, reflecting our exclusive focus on recent recombination events above the leaves. Consequently, the 'Anc' columns for all rows are zero, clearly illustrating that the simulated data does not include any ancestral recombination.

An interesting observation from the table is the contrasting performance of the tools in detecting events, particularly concerning the length of recombination. FastGEAR and Gubbins demonstrate a limited ability to detect events when the recombination length is shorter, with their identification prowess improving as the recombination length increases. In contrast, CFML and PhiloBacter display a more consistent performance, working relatively uniformly regardless of the recombination size.

Table 5.1

Summary of recent recombination (leaves) detection for three selected lengths (500, 1000, and 1500) using BaciSim, fastGEAR, Gubbins, CFML, and PhiloBacter. The simulated data were generated using the following settings: v : 0.03, Number of genomes: 10, Alignment length: 100K, and tMRCA: 0.01. The left column indicates the number of simulated recombination events, with the 'Anc' columns set to zero to represent the absence of ancestral recombination in the simulated data. Column 'Rec' shows the number of recent recombination events.

	Simulator		fastGEAR		Gubbins		CFML		PhiloBacter	
Length	Rec	Anc	Rec	Anc	Rec	Anc	Rec	Anc	Rec	Anc
500	40	0	13	0	10	0	33	2	32	2
1000	38	0	30	1	27	0	35	4	32	4
1500	39	0	34	0	32	3	38	3	36	5

5.4.2.2 Detecting Ancestral Recombinations

In the preceding section, we focused on conducting experiments to assess the various tools' proficiency in detecting recent recombination events. Now, we have extended our analysis by repeating the experiments to contrast the tools' ability to detect ancestral recombination

events. Although the setting for simulating the data remained consistent, BaciSim was configured to simulate recombination exclusively above the internal nodes.

Table 5.2 outlines this investigation's results. It reveals a surprising observation: fastGEAR identified all recombination instances as recent events, failing to recognize even a single occurrence as an ancestral one. This trend appeared independent of the recombination length, showing no variation with size increase. Conversely, Gubbins displayed enhanced performance as the size of the recombination area in the simulated data grew. This demonstrated an ability to adapt to the specific attributes of the data. In line with the findings from the previous section, PhiloBacter and CFML continued to display a more consistent performance. Their ability to work relatively uniformly, regardless of the recombination size, extended to identifying ancestral events.

It is essential to clarify that the objective of this section was to provide a broad overview and comparison of fastGEAR's functionality relative to the other three tools—PhiloBacter, Gubbins, and CFML. We intentionally focused on detecting different types of recombination events and how the tools responded to variations in length. The detailed analysis of the accuracy in determining the exact size of detected events was out of the scope of this comparison.

Table 5.2

Summary of ancestral recombination (internal nodes) detection for three selected lengths (500, 1000, and 1500) using BaciSim, fastGEAR, Gubbins, CFML, and PhiloBacter. The simulated data were generated using the following settings: v : 0.03, Number of genomes: 10, Alignment length: 100K, and tMRCA: 0.01. The left column indicates the number of simulated recombination events, with the 'Rec' columns set to zero to represent the absence of recent recombination in the simulated data. Column 'Anc' shows the number of ancestral recombination events.

	Simulator		fastGEAR		Gubbins		CFML		PhiloBacter	
Length	Rec	Anc	Rec	Anc	Rec	Anc	Rec	Anc	Rec	Anc
500	0	25	13	0	0	9	1	21	0	22
1000	0	18	40	0	0	14	3	18	2	17
1500	0	32	34	0	0	25	1	32	5	30

5.4.2.3 Detecting All Recombinations

We embarked on another experiment wherein we simulated both types of recombination: Recent and Ancestral. The outcomes were consistent with our prior findings from the previous sections. Specifically, fastGear was unable to identify ancestral events. As previously discussed, this discrepancy might stem from variances in how 'recent' and 'ancestral' recombination events are defined and interpreted across these tools. The complex nature of recombination events and their detection variations might be tied to underlying differences in the algorithms or conceptual frameworks used by the tools. This highlights the need for careful consideration when choosing a method for specific applications and potentially opens avenues for further research to fully understand these discrepancies and work towards standardizing definitions and methodologies in this field.

5.5 Application to empirical data

In this section, to provide a more comprehensive evaluation of PhiloBacter's performance, we elected to utilize PhiloBacter on well-recognized, highly recombinogenic real datasets. The real datasets used in this section are available here: <https://zenodo.org/record/8260367>.

5.5.1 Application to *Streptococcus pneumoniae*

Streptococcus pneumoniae is a highly recombinogenic bacterium that resides primarily in the human nasopharynx, serving both as a commensal and a significant respiratory pathogen. This bacterium has been linked to nearly 15 million cases of invasive disease globally in 2000 [145]. Starting from the 1970s, there has been a notable decrease in the susceptibility

of the pneumococcal population to antibiotics, predominantly driven by the emergence and proliferation of various multidrug-resistant clones [146].

In alignment with Gubbins's study, we utilized a subset of 11 sequences from the original dataset, which encompassed a total of 240 PMEN1 sequences, for our evaluation.

Our discussion of this experiment spans four different figures. Figure 5.13 is an original image directly sourced from Gubbins' paper, providing us with a benchmark to compare our results. On the other hand, Figure 5.14 showcases the trees inferred from both CFML and PhiloBacter methods, offering a side-by-side visual comparison of the results yielded by these tools.

To delve deeper into the specifics of the recombination regions identified by these two methods, we focus on Figures 5.15 and 5.16. It's important to note here that Figure 5.15 is an output from the PhiloBacter pipeline, illustrating its capability in handling and processing real datasets.

However, there is an inherent ambiguity as we are dealing with real experimental data. Since a definite "correct answer" does not exist in such a scenario, it becomes challenging to determine which tool has outperformed the other. Despite this, certain observations can be made from the results.

Upon inspecting of Figure 5.16, it becomes apparent that PhiloBacter has successfully identified more recombination events than Gubbins and CFML. Vertical bars represent predicted recombination events occurring on both internal and terminal branches. The first 11 rows depict terminal branches, while the remaining rows illustrate internal branches. Consequently, the recombination events represented on these internal branches are shared by multiple isolates due to their common ancestry. This highlights PhiloBacter's sensitivity in detecting such critical evolutionary events.

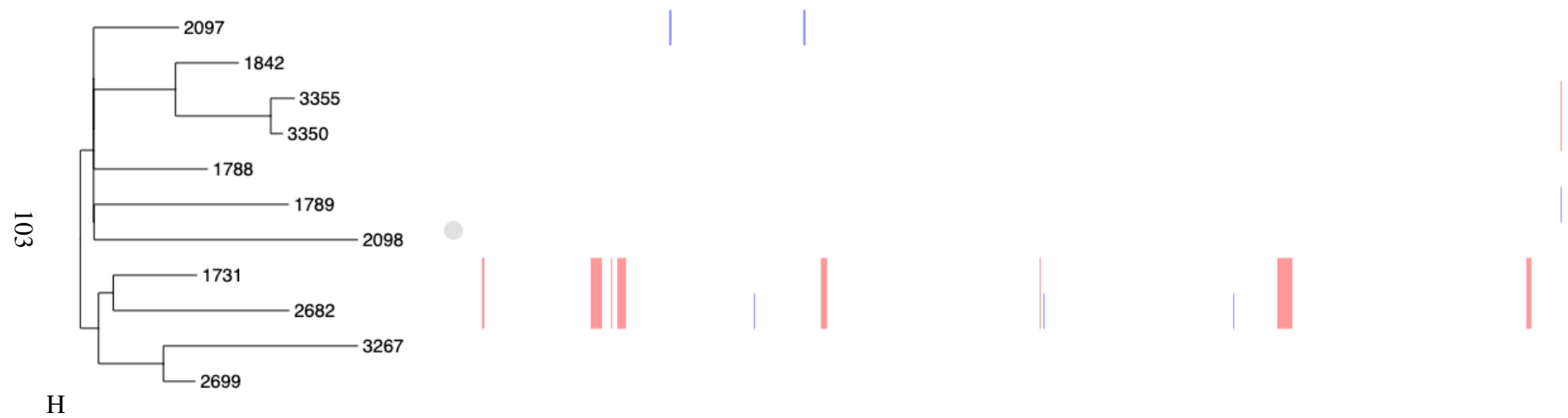


Figure 5.13 Analysis of the PMEN1 genome alignment with Gubbins employing different phylogeny construction strategies [87]

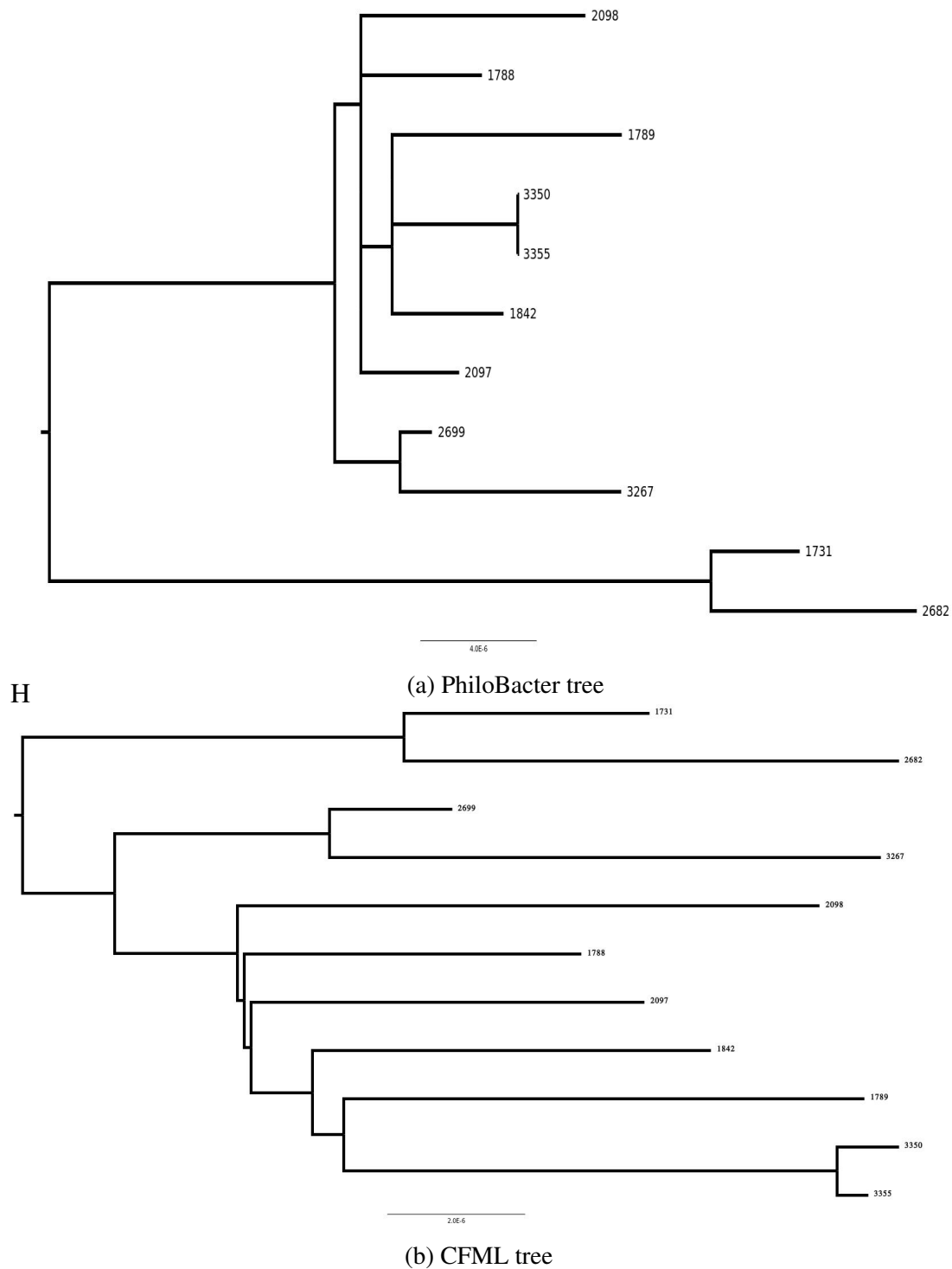


Figure 5.14 Analysis of the PMEN1 genome alignment with PhiloBacter. a) PhiloBacter tree, b) CFML tree

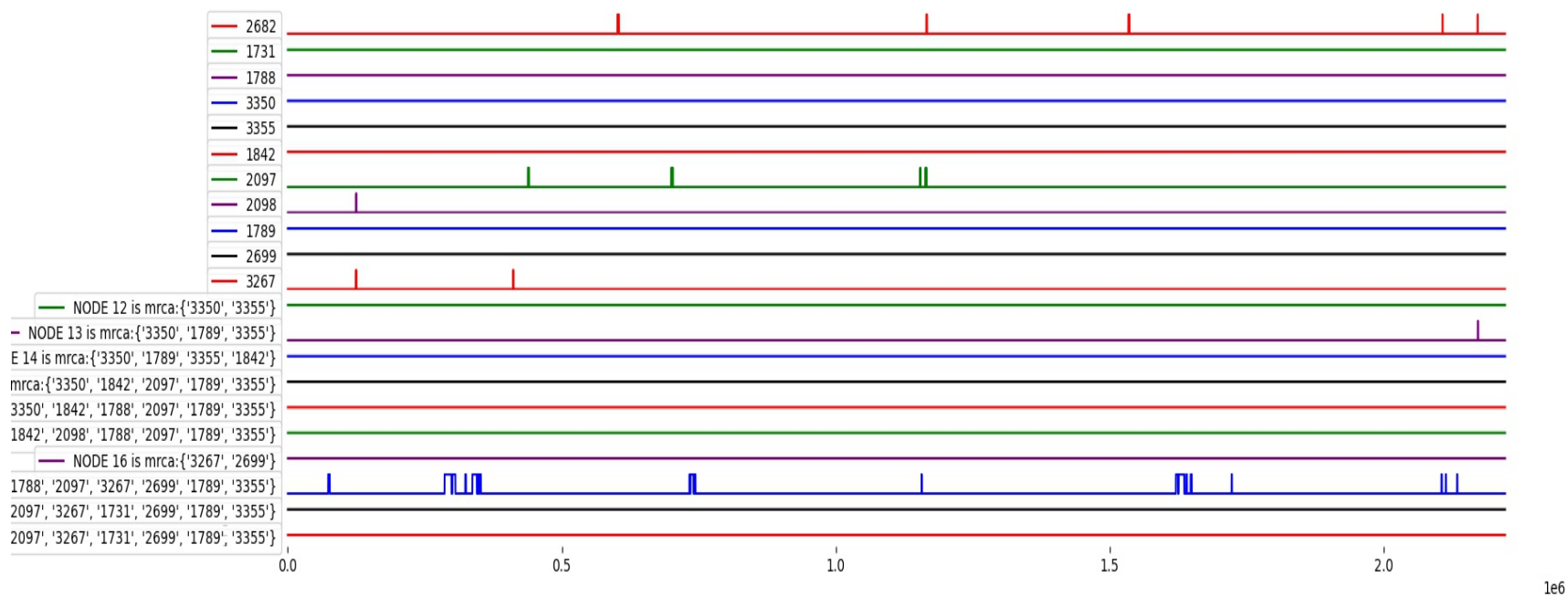


Figure 5.15 Analysis of the PMEN1 genome alignment with CFML. Vertical bars indicate recombination events detected by the CFML.

Additionally, when observing the branch lengths, PhiloBacter offers a more reasonable size for the phylogenetic tree. This suggests that the tree's structure has been less influenced by the recombination events, providing a more realistic depiction of the evolutionary history. This feature of PhiloBacter is significant as it improves the interpretability and applicability of the results, making it an increasingly promising tool for such complex phylogenetic analyses.

5.5.2 Application to *Staphylococcus aureus*

Staphylococcus aureus, including the Methicillin-resistant strain (MRSA), has evolved predominantly clonally [147]. The rarity of observed exchange between MLST loci and the low level of homoplasy in the *S. aureus* ST239 phylogeny supports this observation [148]. A specific strain of this species, MRSA, has become known for its resistance to multiple antibiotics [149] that are usually effective against standard staph infections.

In our exploration of *S. aureus* ST239, we strategically aligned our methodology with Gubbins. Drawing from the extensive alignment detailed in the study by Croucher et al. [87], we selected 14 representative sequences belonging to the Thailand clade. The chosen sequences, representative of key characteristics within the clade, further allowed for a nuanced understanding of the underlying patterns and relationships within *S. aureus* ST239.

In a study by PhiloBacter, 19 recombination events were identified within the clonal genealogy of the species. These included five predicted recombinations above leaves (S26, S38, S78, S85, and DEN907), and others involving internal nodes. Two significant recombination events were found to have longer lengths: the first included taxa S40, S130, S102, S85, S87, S93, and S71, while the second included taxa S2, S26, DEN907, and S78.

CFML identified 28 recombination events. Four took place above external nodes, with three appearing above taxa S38 and one above taxa S78. A notable internal recombination of

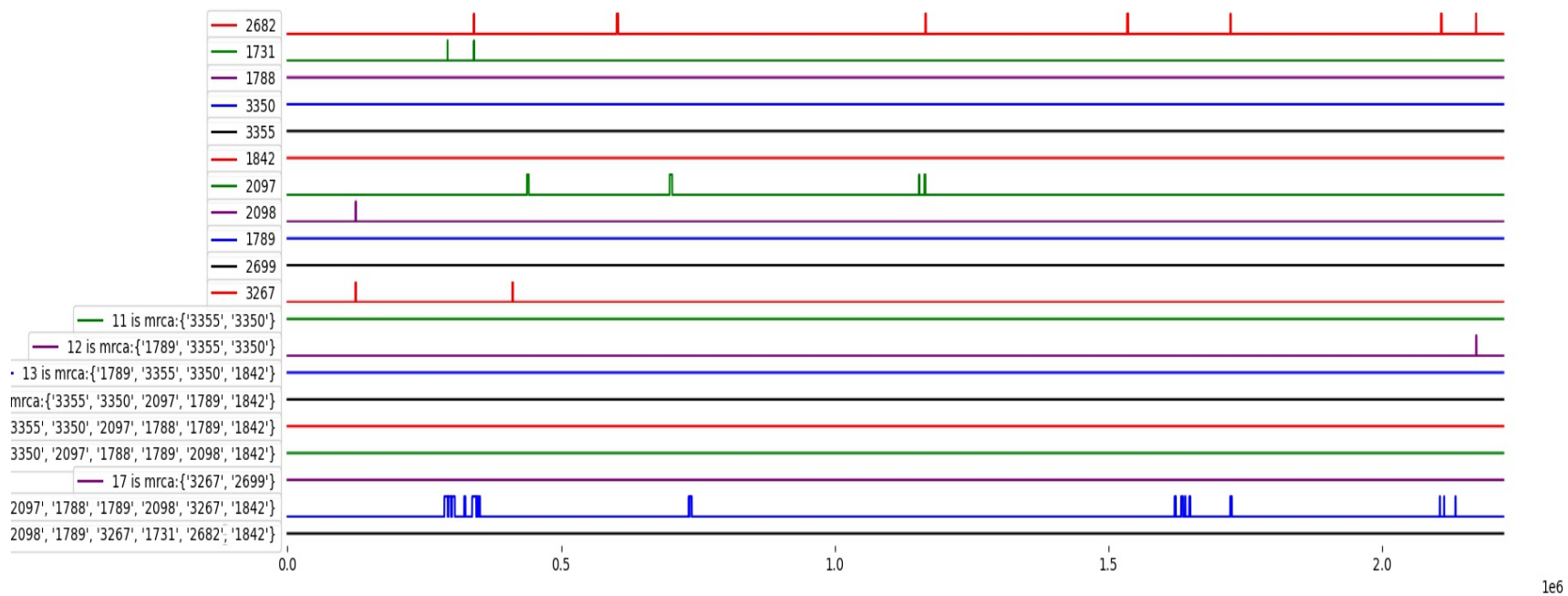


Figure 5.16 Analysis of the PMEN1 genome alignment with PhiloBacter. Vertical bars indicate recombination events detected by the analysis.

extended length occurred above the nodes comprising taxa S40, S130, S102, S85, S87, S93, and S71. Additionally, recombination events were detected above nodes S2, S78, and S26.

In this study, Gubbins detected a total of 51 recombination events. Forty of these events occurred above the leaves, specifically at the following taxa: S130, S87, S85, S40, S102, S78, S2, S26, DEN907, S7, S38, and TW20. Additionally, there were two internal recombination events. The first one involved nodes above taxa S130, S87, S85, S40, S102, S93, and S71, while the second included nodes TW20, S26, S78, S2, DEN907, S7, and S38.

Despite the differences in the number of recombination events detected, there is a commonality in some of the identified recombinations. For instance, an extended recombination event, including taxa S40, S130, S102, S85, S87, S93, and S71, was identified by both PhiloBacter and CFML. Some taxa like S78, S26, and DEN907 appeared in the results of all three methods. Also, The repeated occurrence of certain taxa across different methods might provide valuable insights into the specific recombination patterns or hotspots within the genealogy of the species.

For the detail of recombination events and phylogenetic trees as inferred by the three methods, please refer to the available data at the following repository: (https://github.com/nehlehk/PhiloBacter/tree/main/example_data/ST239).

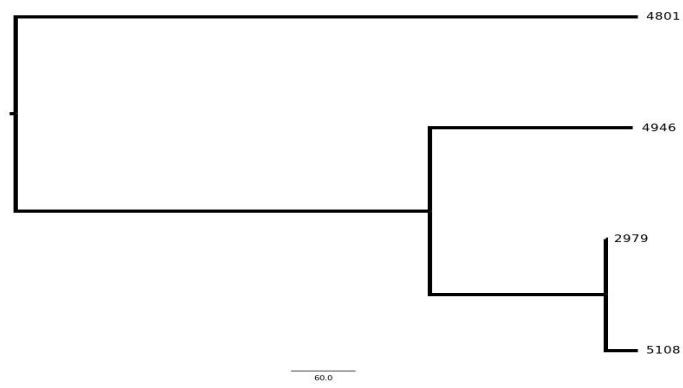
5.5.3 Application to *Bacillus Cereus*

Bacillus cereus, a gram-positive bacterium that can function in anaerobic and aerobic environments, is commonly found in soil, plants, and food. Often, it leads to intestinal issues such as nausea, vomiting, and diarrhea. In those with weakened immune systems, however, it may result in severe infections, including septicemia and endophthalmitis, potentially causing vision loss [150].

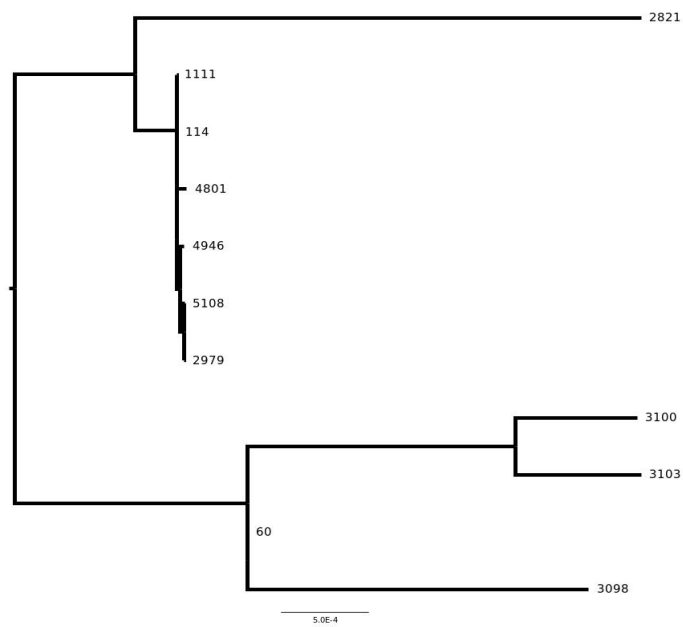
Evidence has been found for genetic exchange within the *B. cereus* subgroup through homologous or nonhomologous recombination or interaction between different lineages [151], [152].

The website PubMLST.org serves as a repository for open-access, meticulously curated databases. These databases combine sequence data from populations with origin and phenotype details for more than 130 microbial species and genres. Specifically, the cgMLST scheme for the *B. cereus sensu lato* group was pioneered by Nicolas Tourasse and his team [153]. For our analysis, we concentrated on sequences isolated in Australia, resulting in a selection of 11 records from the database. Figure A.7 shows a detailed overview of these records.

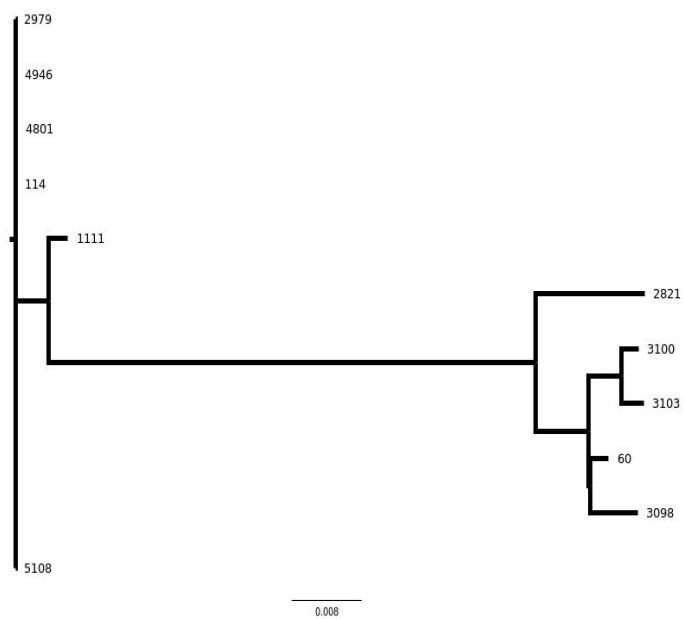
Similar to our previous experiments, we evaluated the database using three distinct methods in this segment of our thesis. The outcomes obtained from Gubbins deviated markedly from those of the other methods, leading to an unusually structured tree. As depicted in figure 5.17a, this tree incorporates only four taxa, neglecting the remaining seven. Moreover, Gubbins identified a mere eight recombination events within these sequences.



(a) Gubbins tree



(b) CFML tree



(c) PhiloBacter tree

Figure 5.17 The analysis of *Bacillus cereus* strains isolated in Australia: a) Tree representation using Gubbins, b) CFML-based phylogenetic tree, and c) PhiloBacter tree analysis.

Conversely, CFML predicted 2,760 recombination events, encompassing internal and external events. However, 300 events had lengths under 100, indicating that CFML fragmented some events into multiple shorter segments. The tree inferred by CFML can be viewed in figure 5.17b.

PhiloBacter reported 391 recombination events. Its corresponding tree is showcased in figure 5.17c. A schematic representation of the events predicted by PhiloBacter can be observed in figure 5.18. This figure shows that taxa '2979', '4801', and '4946' underwent only a single recombination. This finding mirrors CFML's results. Moreover, both methods predicted zero events for taxa '60' and '114'.

The branch lengths determined by CFML are considerably shorter than those of the PhiloBacter tree. It's commonly assumed that branches undergoing recombination tend to have longer lengths than those without recombination events. Given this premise, PhiloBacter appears to provide a more accurate estimation of branch lengths than CFML.

For the detail of recombination events and phylogenetic trees as inferred by the three methods, please refer to the available data at the following repository: (https://github.com/nehlekh/PhiloBacter/tree/main/example_data/Bacillus%20Cereus).

5.6 Resource Usage

As we transition from theory to practice, it's crucial to consider the computational resources required to implement these methodologies. We've compared PhiloBacter with Gubbins and CFML across four parameters: CPU usage, memory (RAM) usage, job duration, and input/output (I/O) operations.

The figures 5.19, 5.20, 5.21, and 5.22 indeed illustrate that PhiloBacter's resource uti-

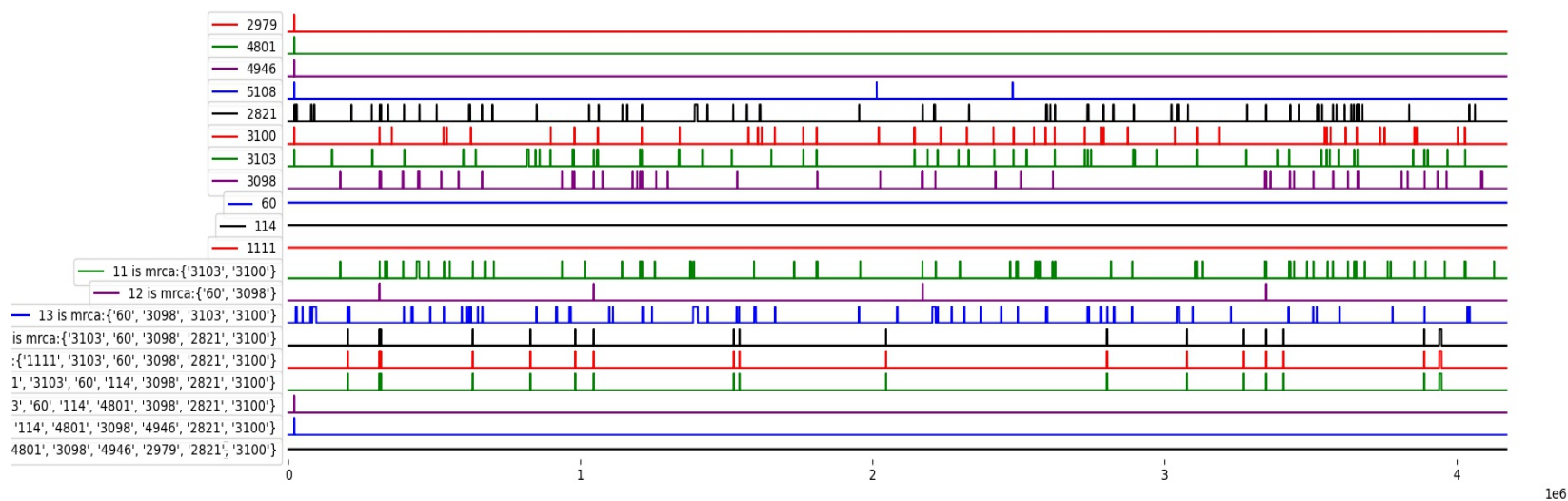


Figure 5.18 The analysis of *Bacillus cereus* strains isolated in Australia with PhiloBacter. Vertical bars indicate recombination events detected by the analysis.

lization currently exceeds that of CFML and Gubbins. It should be noted that the software testing was carried out on a system with the following specifications:

- Operating System: 64-bit
- Memory: 16 GB
- Processor: Intel® Core™ i5-8365U CPU, with a base clock speed of 1.60 GHz and 8 cores.

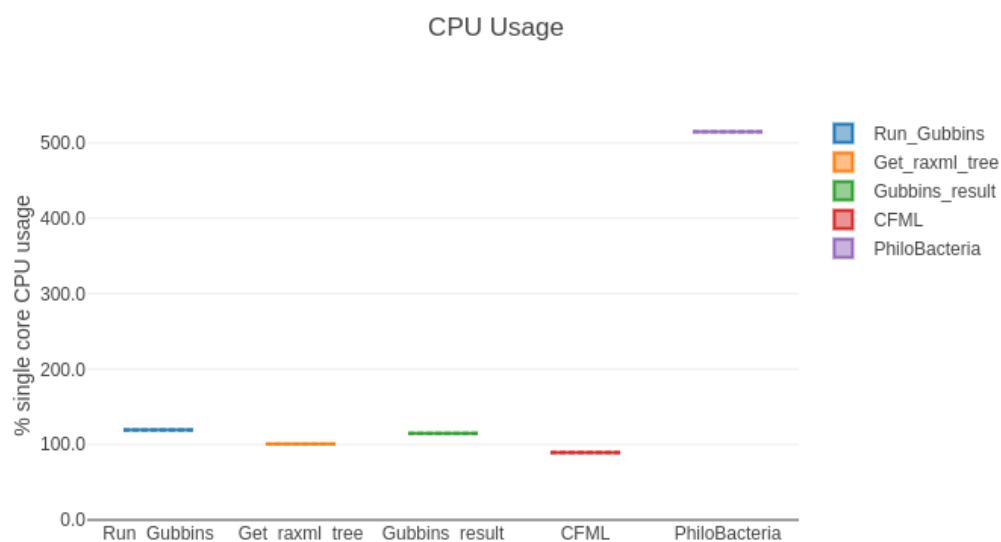


Figure 5.19 Resource usage comparison chart showcasing CPU consumption across various tools: PhiloBacter, CFML, Gubbins (Gubbins-result is our custom script for output manipulation), and RAxML.

We attribute this to two main factors. First, PhiloBacter is still in its initial version and, like many first-generation tools, is yet to be optimized for efficiency. Second, the fact that it was developed using Python, could add to the computational overhead.

Python is an interpreted language, which means it is not compiled into machine code before execution. Instead, the Python interpreter reads and executes each program line sequen-

tially, translating each statement into a sequence of one or more subroutines already compiled into machine code. This contrasts with compiled languages, such as C or C++, where source code is transformed into machine code before it's executed.

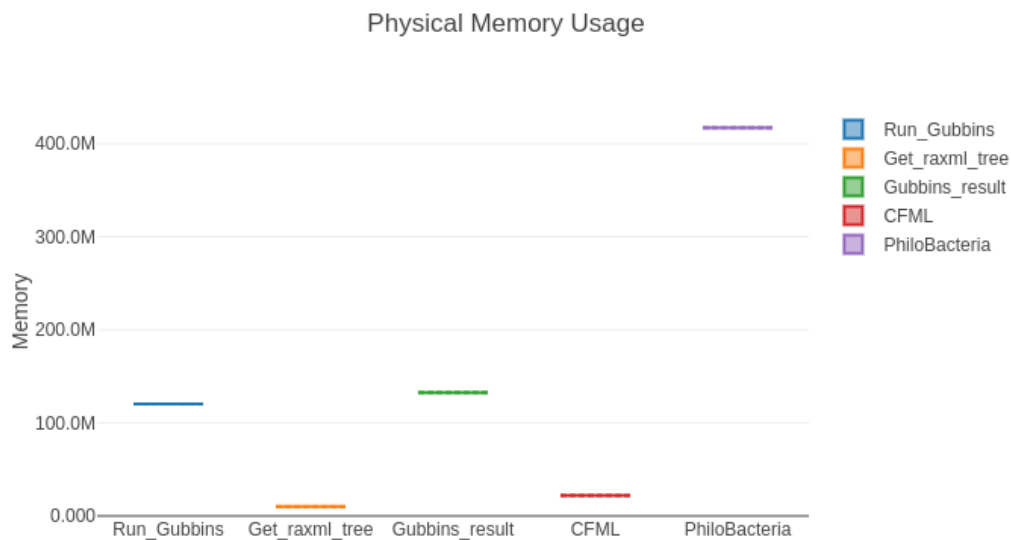


Figure 5.20 Resource usage comparison chart showcasing RAM consumption across various tools: PhiloBacter, CFML, Gubbins (Gubbins-result is our custom script for output manipulation), and RAxML.

This characteristic of Python generally leads to slower execution times compared to compiled languages. This is mainly because the interpretation process—reading and translating the code—happens every time the program runs, introducing an overhead absent in precompiled code.

Furthermore, Python's dynamic typing and automatic memory management also contribute to its relative slowness. Dynamic typing means the variable type can change during runtime, and automatic memory management means the language is responsible for allocating and deallocating memory as needed. While these features make Python flexible and easy to use, they also contribute to slower execution speeds because the interpreter must continuously

ensure type compatibility and manage memory.

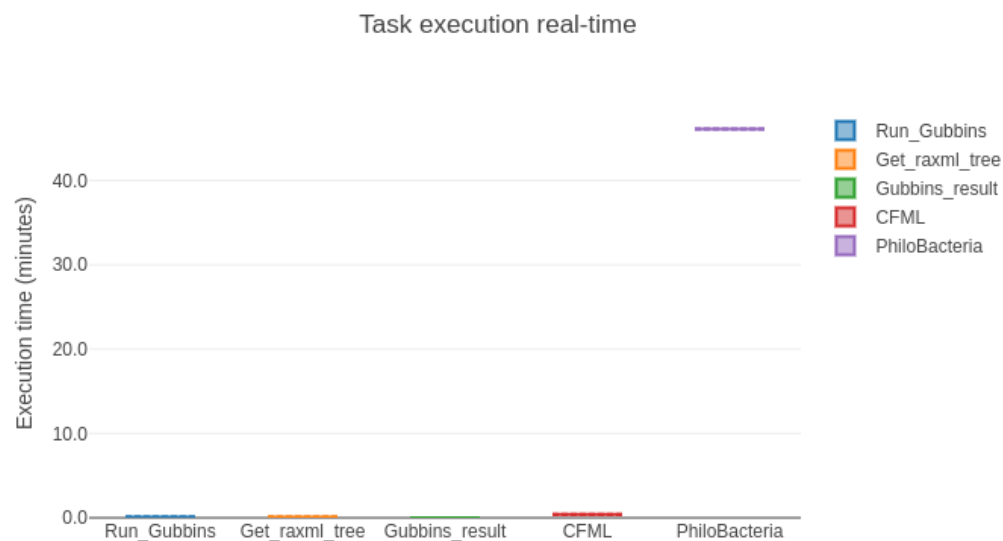


Figure 5.21 Comparative chart of job duration, illustrating the processing times for PhiloBacter, CFML, Gubbins (enhanced with our tailored script -Gubbins-result- for output refinement), and RAxML.

While the current version of PhiloBacter might require higher computational resources, we are acutely aware of these constraints. Looking forward, we are focused on improving the efficiency of PhiloBacter in future iterations. For the next version of PhiloBacter, efforts could be made to rewrite some critical parts of the code in a compiled language such as C or Rust to improve performance and significantly reduce resource usage. Parallelization of the code could also be explored to take advantage of modern multi-core processors.

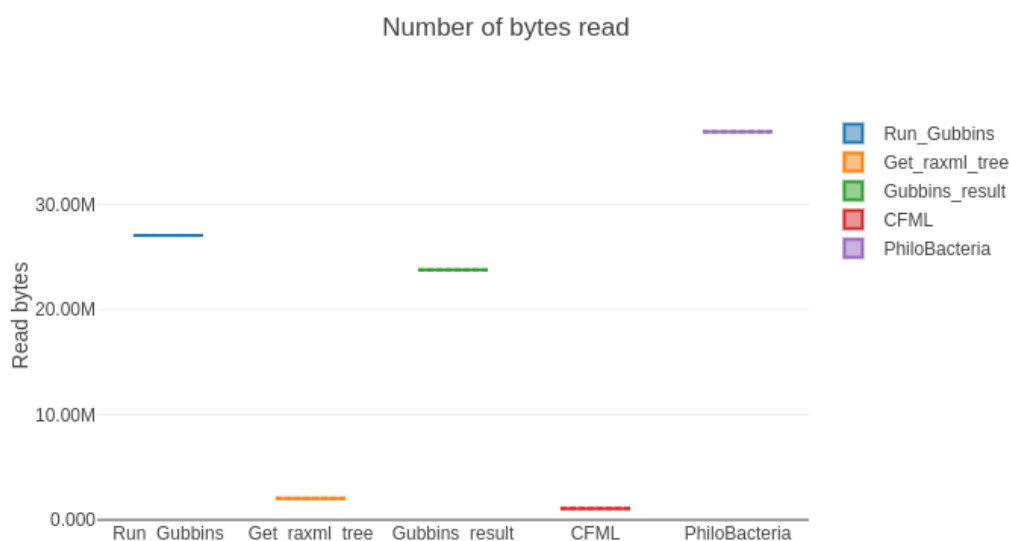


Figure 5.22 I/O resource usage comparison chart, displaying results for PhiloBacter, CFML, Gubbins (modified using our custom script -Gubbins-result- for output optimization), and RAxML.

5.7 Discussion

Detecting recombination and understanding its impact on the construction of phylogenetic trees have been longstanding pursuits within genomics research. However, these efforts have been met with considerable challenges due to the complex processes involved. Available methodologies have addressed some aspects of the problem, but the puzzle remains incomplete, leaving this an intriguing area for further exploration.

Recombination events introduce an asymmetry within bacterial genomes about both polymorphism and phylogenetics. This unevenness presents an opportunity to detect recombination segments and construct more accurate phylogenetic trees in the presence of recombination. Inspired by this potential, we developed a new method to infer recombination from the alignment of whole bacterial genome sequences. In this thesis, we present our novel tool, PhiloBacter.

PhiloBacter operates on a probabilistic model, scrutinizing each alignment site at each tree branch. Leveraging the Hidden Markov Model (HMM) method, it estimates the associated probabilities and, subsequently, utilizes these values to reconstruct the phylogenetic tree. PhiloBacter offers two strategies for this reconstruction, both employing the Maximum Likelihood (ML) method.

Our work with PhiloBacter shares similarities with ClonalFrameML (CFML), a tool that uses a related approach to infer the ML tree from the bacterial alignment sequence. However, several distinctive characteristics and innovations set PhiloBacter apart:

1. **Simplicity:** PhiloBacter’s design emphasizes simplicity and ease of understanding. Unlike other tools, it does not entangle users with complicated and obscure parameters.
2. **Unique Perspective:** As far as we know, PhiloBacter is the first method that investigates bacterial recombination from a sequence uncertainty perspective. This innovative approach may pave the way for more fascinating discoveries about bacterial genomes.
3. **Different Angles, Same Tools:** While CFML and PhiloBacter utilize the HMM and Baum-Welch algorithm, they do so for different purposes. CFML estimates various model parameters, including point mutation rate, recombination rate, recombination length, recombination substitution probability, and branch length. In contrast, PhiloBacter focuses on estimating only the recombination and imported substitution rates. This exemplifies how the same tools can tackle the problem from different angles.
4. **Handling High Recombination Rates:** Some recombination detection tools struggle with high recombination rates as these tools often remove the recombinant regions. With high recombination rates, these tools can potentially discard significant portions of the sequence, leaving little useful information for tree inference. PhiloBacter overcomes

this challenge by incorporating all regions into its model and employing the calculated probabilities in tree reconstruction.

5. **Performance and Accuracy:** Despite not attempting to estimate all model parameters, PhiloBacter performs remarkably in inferring a clonal tree closer to the actual tree. It also exhibits superior accuracy in estimating recombination lengths compared to other methods, as evidenced by various Figures in this chapter.
6. **Robustness:** Massive experimentation on various simulated datasets has demonstrated that PhiloBacter is robust against changes in multiple parameters that might influence the data. This robustness makes PhiloBacter an attractive option for complex genomic analyses.

PhiloBacter presents a compelling new approach to bacterial recombination analysis, pushing the boundaries of our current understanding and capabilities. With its unique perspective and features, PhiloBacter holds great promise for future genomics research.

CHAPTER SIX

CONCLUSION AND FUTURE DIRECTIONS

6.1 Conclusion

The thesis discussed here is centred around a significant challenge in bacterial genetics and bioinformatics - the issue of detecting recombination events in bacterial genomes and the subsequent construction of their phylogenetic trees. Phylogenetic trees are diagrammatic representations showing the inferred evolutionary relationships among biological entities. They are based on similarities and differences in their physical or genetic characteristics.

In bacteria, recombination is an integral part of the evolutionary process, wherein one bacterial organism exchanges genetic material with another. These recombination events, however, can complicate the inference of phylogenetic trees, which are generally meant to represent a branching, tree-like pattern of evolution.

Chapter 3 of the thesis proposes innovative methodologies to address this issue. Instead of taking a traditional approach to phylogenetic reconstruction, these methods are designed to detect instances of recombination in bacterial genomes and appropriately factor these events into the inference of phylogenetic trees. By doing so, they provide a more accurate picture of bacterial evolution, which recognizes and accounts for the significant role of recombination.

Chapter 4 introduces a new tool for simulation that has been developed specifically to test the effectiveness of the proposed solution. In this context, simulation tools are computer programs designed to emulate the processes of bacterial recombination and phylogenetic tree construction. These tools enable the researchers to create a variety of virtual scenarios in which the performance of their proposed methods can be scrutinized.

Chapter 5 involves a rigorous series of simulation scenarios to test the proposed solution's effectiveness. These simulations focus on evaluating the performance of a tool named PhiloBacter. This tool, designed as a central part of the proposed methodologies, is put through its paces and compared against two well-known methods used in the field. Its performance is evaluated regarding its ability to accurately reconstruct phylogenetic trees in scenarios where recombination events have occurred. The findings from these tests indicate that PhiloBacter consistently outperforms the other methods across most scenarios, thus effectively fulfilling the central aim of the thesis.

A significant outcome of this research was the development of a novel tool for recombination detection. This tool employs a probabilistic approach, assigning a unique probability to each alignment site in the genome, effectively mapping out the likelihood of recombination events occurring at each location. Probabilistic values and this approach's underlying mathematical and statistical framework make it a robust and reliable solution. This methodology opens up new possibilities for similar bioinformatics applications, paving the way for future advancements in the field.

The thesis also presents two innovative approaches to integrating these probabilistic values into inferring phylogenetic trees. One of these approaches has been explicitly designed to deal with uncertainty in the tree inference process, a significant leap forward considering the complexity of bacterial evolution. Introducing these methodologies represents a considerable contribution to the field, offering novel solutions beyond addressing the immediate problem of bacterial recombination. They present fresh perspectives and provide new avenues for future research.

One of the most critical contributions of the thesis is its unique perspective on bacterial recombination. The research examines the issue of bacterial recombination from the view-

point of uncertainty. This perspective is novel and can be applied to studying recombination in various other organisms, thus broadening its potential impact on the broader field of genetic research.

Finally, the thesis introduces BaciSim, a new simulation tool developed to generate specific simulated datasets. BaciSim is designed to be a simple and fast simulator that can produce data according to the particular requirements of the research at hand. This tool further exemplifies the innovative spirit of the study, highlighting its commitment to developing new tools and methodologies to advance the field of bacterial genetics and phylogenetics.

6.2 Perspectives and Future Research

While we have seen promising results and findings in our research, we also acknowledge some problems and challenges we must conquer in our future endeavours. Though these challenges may present difficulties, they offer exciting opportunities for further growth and development.

6.2.1 PhiloBacter as a fast and more efficient tool

A key area of focus for our future work revolves around optimizing PhiloBacter, our computational tool. To make PhiloBacter a faster and more efficient tool, we aim to optimize its application to large bacterial whole-genome datasets, drastically reducing the computational time from days to hours. The PhiloBacter software was initially implemented in Python, and we have since incorporated Cython [154], a Python-C hybrid compiler. This implementation has led to an impressive increase in computational speed - by a factor of five times to be precise.

However, we also acknowledge that the current method of PhiloBacter computation, which consists of a matrix with dimensions equivalent to the total number of tree nodes mul-

multiplied by the alignment length, poses specific issues. Current libraries, such as Beagle [155], [156], which we had previously implemented, have shown limitations in responsiveness. Thus, it becomes evident that there is a need to develop a more effective solution from scratch. Our proposed approach to this issue is to create the desired code in C++, a language renowned for efficiently handling complex computational tasks.

6.2.2 Generalised uncertainty approach

In the third chapter of our research, we delved into the intricate world of computational biology, specifically focusing on the impact of recombination and its potentially misleading effects on the topology of phylogenetic trees. We highlighted an innovative method, namely the 'uncertainty approach', that we theorized might be instrumental in correcting such distortions. This approach essentially incorporates a probabilistic perspective into the analysis of recombination events, accounting for the intrinsic uncertainty and potential errors within the recombination process.

In the future, our research will seek to validate the efficacy of the uncertainty approach in mitigating the skewness of tree topology due to recombination events. To comprehensively evaluate the impact of the uncertainty approach, we will need to establish various simulation scenarios. These scenarios will mimic different biological situations, from simple to complex and multifaceted.

By varying the parameters within these simulation scenarios, we can observe the changes in tree topology and thus assess how the uncertainty approach affects it under different conditions. For instance, one scenario could involve high recombination rates and low mutation rates, while another might involve the inverse. The effects of the uncertainty approach can then be monitored under each unique set of circumstances.

Subsequently, we will analyze the results of these simulations with a critical eye, looking for evidence of topology correction by the uncertainty approach. This involves comparing the simulated tree topology before and after applying the uncertainty approach. After using the uncertainty approach, a successful modification would entail the tree topology better reflecting the authentic evolutionary relationships between species.

We hope to enhance our understanding of the uncertainty approach and its effects on tree topology by systematically implementing and examining these different simulation scenarios. Moreover, this rigorous assessment will allow us to refine the uncertainty approach, improving its capacity to correct any misleading effects caused by recombination and thus advance the field of computational biology.

6.2.3 Internal and external nodes

In the course of this thesis, we've ventured into a relatively unexplored domain of computational biology by modelling recombination with an emphasis on uncertainty. This represents a pioneering attempt at an innovative approach, focusing on the mechanical aspect of recombination and considering the probabilistic uncertainties intrinsic to this biological phenomenon.

While our initial efforts have centred on a more generalized form of uncertainty, we recognize that further refinements could be made. There is significant potential in tailoring different approaches to accommodate uncertainty at different levels of the phylogenetic tree. These levels could include both external nodes, which represent present-day organisms or taxa, and internal nodes, which represent their common ancestors.

External nodes are directly observable and thus usually less uncertain, but they can still be influenced by factors such as measurement errors or sequencing inaccuracies. Internal

nodes, on the other hand, are inherently more uncertain due to their inferred nature. A comprehensive uncertainty approach would need to address these different sources of uncertainty nuancedly, considering their varying levels and impacts.

Looking ahead, we believe that developing a specific uncertainty model for recombination could yield significant benefits for the accuracy and reliability of tree inference. Such a model could systematically incorporate the uncertainty from recombination events into phylogenetic analyses, allowing for more realistic and robust reconstructions of evolutionary histories.

A more refined model might also explain that recombination events are not uniformly distributed but tend to occur more frequently in certain "hotspots" within the genome. It could also consider how the recombination rate varies among organisms or parts of the same genome.

In conclusion, while we have made a promising start in incorporating uncertainty into recombination modelling, there remains much potential for further exploration and refinement in this field. Our continued work in this area promises to enhance our understanding of evolutionary processes and relationships greatly.

6.2.4 A dynamic and general simulator

BaciSim, our computational simulator, has thus far been a precious tool in our research. Its core function has been to simulate bacterial evolution, providing a framework within which we can generate and examine various scenarios of bacterial genetic diversity. However, the potential for BaciSim extends beyond its current capabilities, and our plans include expanding its scope and versatility.

Our vision for BaciSim is to transform it into a more comprehensive simulator capable of modelling various biological scenarios. Instead of solely focusing on bacteria, we plan to

broaden its purview to include other types of organisms as well. This would involve incorporating additional parameters and models into BaciSim to accurately reflect various organisms' different genetic mechanisms and evolutionary processes.

Moreover, by considering other population structures BaciSim could offer insights into how different population dynamics influence evolutionary outcomes.

Finally, to enhance the reliability and accuracy of BaciSim, the outcomes generated by the simulator could be meticulously compared and aligned with actual datasets. This cross-referencing ensures that the simulation reflects theoretical expectations and aligns closely with real-world observations and results.

6.2.5 PhiloBacter and quantum computing

Suppose we look at the problem of inferring phylogenetic trees from the perspective of the optimization problem, where the goal is to identify the most probable tree configuration, similar to finding a system's minimum energy state. However, such a task introduces significant computational challenges, as it demands considerable computational power to traverse the large space of possible tree configurations and risks becoming trapped in a local minimum.

To circumvent these challenges, we are contemplating the utilization of quantum annealing, a technique that employs quantum mechanics to solve optimization problems. Quantum annealing can potentially navigate the solution space more efficiently and effectively by leveraging quantum phenomena such as superposition and quantum tunnelling. Thus, it is expected to locate the global minimum with greater accuracy and speed. By adopting quantum annealing, we aim to cut down considerably on the time and computational resources required to analyze phylogenetic trees using PhiloBacter, leading to more accurate evolutionary relationship inferences and contributing to the broader development of computational biology.

REFERENCES

- [1] P. Lemey, M. Salemi, and A.-M. Vandamme, *The phylogenetic handbook: a practical approach to phylogenetic analysis and hypothesis testing*. Cambridge University Press, 2009.
- [2] L. Taylor. “Superbugs a far greater risk than covid in pacific, scientist warns.” (), [Online]. Available: <https://www.theguardian.com/world/2020/sep/10/superbugs-a-far-greater-risk-than-covid-in-pacific-scientist-warns>. (accessed: 10 Sep 2020).
- [3] WHO. “Antimicrobial resistance.” (), [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/antimicrobial-resistance>. (accessed: 17 November 2021).
- [4] R. Cave, J. Cole, and H. V. Mkrtchyan, “Surveillance and prevalence of antimicrobial resistant bacteria from public settings within urban built environments: Challenges and opportunities for hygiene and infection control,” *Environment international*, vol. 157, p. 106 836, 2021.
- [5] G. G. Perron, A. E. Lee, Y. Wang, W. E. Huang, and T. G. Barraclough, “Bacterial recombination promotes the evolution of multi-drug-resistance in functionally diverse populations,” *Proceedings of the Royal Society B: Biological Sciences*, vol. 279, no. 1733, pp. 1477–1484, 2012.
- [6] B. R. Levin and O. E. Cornejo, “The population and evolutionary dynamics of homologous gene recombination in bacteria,” *PLoS genetics*, vol. 5, no. 8, e1000601, 2009.

- [7] V. Gürtler and B. C. Mayall, “Genomic approaches to typing, taxonomy and evolution of bacterial isolates.,” *International Journal of Systematic and Evolutionary Microbiology*, vol. 51, no. 1, pp. 3–16, 2001.
- [8] F. Aujoulat, S. Romano-Bertrand, A. Masnou, H. Marchandin, and E. Jumas-Bilak, “Niches, population structure and genome reduction in *ochrobactrum intermedium*: Clues to technology-driven emergence of pathogens,” *PloS one*, vol. 9, no. 1, e83376, 2014.
- [9] R. Bosch, E. Garcíea-Valdés, and E. R. Moore, “Complete nucleotide sequence and evolutionary significance of a chromosomally encoded naphthalene-degradation lower pathway from *pseudomonas stutzeri* an10,” *Gene*, vol. 245, no. 1, pp. 65–74, 2000.
- [10] T. B. Rounge, T. Rohrlack, T. Kristensen, and K. S. Jakobsen, “Recombination and selectional forces in cyanopeptolin nrps operons from highly similar, but geographically remote planktothrix strains,” *BMC microbiology*, vol. 8, pp. 1–10, 2008.
- [11] N. Potnis, P. P. Kandel, M. V. Merfa, *et al.*, “Patterns of inter-and intrasubspecific homologous recombination inform eco-evolutionary dynamics of *xylella fastidiosa*,” *The ISME journal*, vol. 13, no. 9, pp. 2319–2333, 2019.
- [12] L. Hao, M. T. Holden, X. Wang, *et al.*, “Distinct evolutionary patterns of *neisseria meningitidis* serogroup b disease outbreaks at two universities in the usa,” *Microbial Genomics*, vol. 4, no. 4, 2018.
- [13] E. J. Feil, “Small change: Keeping pace with microevolution,” *Nature Reviews Microbiology*, vol. 2, no. 6, pp. 483–495, 2004.

- [14] J. L. Martiénez and F. Baquero, “Interactions among strategies associated with bacterial infection: Pathogenicity, epidemicity, and antibiotic resistance,” *Clinical microbiology reviews*, vol. 15, no. 4, pp. 647–679, 2002.
- [15] R. B. G. Pessoa, W. F. de Oliveira, D. S. C. Marques, M. T. dos Santos Correia, E. V. M. M. de Carvalho, and L. C. B. B. Coelho, “The genus aeromonas: A general approach,” *Microbial pathogenesis*, vol. 130, pp. 81–94, 2019.
- [16] J. Yassif, A. Santhakumar, and N. Lightfoot, “Enhancing global security through infectious disease threat reduction,” *Global Health Security*, vol. 1, 2013.
- [17] C. Collins and X. Didelot, “A phylogenetic method to perform genome-wide association studies in microbes that accounts for population structure and recombination,” *PLoS computational biology*, vol. 14, no. 2, e1005958, 2018.
- [18] J. Stapley, P. G. Feulner, S. E. Johnston, A. W. Santure, and C. M. Smadja, *Recombination: The good, the bad and the variable*, 2017.
- [19] M. H. Schierup and J. Hein, “Consequences of recombination on traditional phylogenetic analysis,” *Genetics*, vol. 156, no. 2, pp. 879–891, 2000.
- [20] X. Didelot, D. Lawson, A. Darling, and D. Falush, “Inference of homologous recombination in bacteria using whole-genome sequences,” *Genetics*, vol. 186, no. 4, pp. 1435–1449, 2010.
- [21] M. Spies and R. Fishel, “Mismatch repair during homologous and homeologous recombination,” *Cold Spring Harbor perspectives in biology*, vol. 7, no. 3, a022657, 2015.
- [22] O. Johnsborg, V. Eldholm, and L. S. Håvarstein, “Natural genetic transformation: Prevalence, mechanisms and function,” *Research in microbiology*, vol. 158, no. 10, pp. 767–778, 2007.

- [23] H. Bernstein, C. Bernstein, and R. E. Michod, "Sex in microbial pathogens," *Infection, Genetics and Evolution*, vol. 57, pp. 8–25, 2018.
- [24] X. Didelot and M. C. Maiden, "Impact of recombination on bacterial evolution," *Trends in microbiology*, vol. 18, no. 7, pp. 315–322, 2010.
- [25] N. Matthey and M. Blokesch, "The dna-uptake process of naturally competent vibrio cholerae," *Trends in microbiology*, vol. 24, no. 2, pp. 98–110, 2016.
- [26] P. G. Hofstatter, A. K. Tice, S. Kang, M. W. Brown, and D. J. Lahr, "Evolution of bacterial recombinase a (reca) in eukaryotes explained by addition of genomic data of key microbial lineages," *Proceedings of the Royal Society B: Biological Sciences*, vol. 283, no. 1840, p. 20161453, 2016.
- [27] N. D. Zinder, """ transduction" in bacteria," *Scientific American*, vol. 199, no. 5, pp. 38–43, 1958.
- [28] H. Ozeki and H. Ikeda, "Transduction mechanisms," *Annual Review of Genetics*, vol. 2, no. 1, pp. 245–278, 1968.
- [29] K. B. Low and D. D. Porter, "Modes of gene transfer and recombination in bacteria," *Annual Review of Genetics*, vol. 12, no. 1, pp. 249–287, 1978.
- [30] G. Kaiser. "Horizontal gene transfer in bacteria." (), [Online]. Available: [https://bio.libretexts.org/Bookshelves/Microbiology/Book%3AMicrobiology_\(Kaiser\)/Unit_2%3ABacterial_Genetics_and_the_Chemical_Control_of_Bacteria/3%3ABacterial_Genetics/3.1%3A_Horizontal_Gene_Transfer_in_Bacteria](https://bio.libretexts.org/Bookshelves/Microbiology/Book%3AMicrobiology_(Kaiser)/Unit_2%3ABacterial_Genetics_and_the_Chemical_Control_of_Bacteria/3%3ABacterial_Genetics/3.1%3A_Horizontal_Gene_Transfer_in_Bacteria). (accessed: Apr 10, 2022).
- [31] O. Gascuel, *Mathematics of evolution and phylogeny*. OUP Oxford, 2005.

- [32] Z. Yang and B. Rannala, “Molecular phylogenetics: Principles and practice,” *Nature reviews genetics*, vol. 13, no. 5, pp. 303–314, 2012.
- [33] G. Munjal, M. Hanmandlu, and S. Srivastava, “Phylogenetics algorithms and applications,” in *Ambient Communications and Computer Systems*, Springer, 2019, pp. 187–194.
- [34] T. T.-Y. Lam, C.-C. Hon, and J. W. Tang, “Use of phylogenetics in the molecular epidemiology and evolutionary studies of viral infections,” *Critical reviews in clinical laboratory sciences*, vol. 47, no. 1, pp. 5–49, 2010.
- [35] A.-M. Vandamme and O. G. Pybus, “Viral phylogeny in court: The unusual case of the valencian anesthetist,” *BMC biology*, vol. 11, no. 1, pp. 1–3, 2013.
- [36] E. J. Bernard, Y. Azad, A.-M. Vandamme, M. Weait, and A. M. Geretti, “Hiv forensics: Pitfalls and acceptable standards in the use of phylogenetic analysis as evidence in criminal investigations of hiv transmission,” *HIV medicine*, vol. 8, no. 6, pp. 382–387, 2007.
- [37] J. Wise, “Providing the csi treatment: Criminal justice practitioners and the csi effect,” *Current Issues in Criminal Justice*, vol. 21, no. 3, pp. 383–399, 2010.
- [38] M. A. Spyrou, K. I. Bos, A. Herbig, and J. Krause, “Ancient pathogen genomics as an emerging tool for infectious disease research,” *Nature Reviews Genetics*, vol. 20, no. 6, pp. 323–340, 2019.
- [39] B. Choi, C. Wyss, and U. Göbel, “Phylogenetic analysis of pathogen-related oral spirochetes,” *Journal of Clinical Microbiology*, vol. 34, no. 8, pp. 1922–1925, 1996.
- [40] J. C Ashton, “Phylogenetic methods in drug discovery,” *Current Drug Discovery Technologies*, vol. 10, no. 4, pp. 255–262, 2013.

- [41] S. M. Mawalagedera, D. L. Callahan, A. C. Gaskett, N. Rønsted, and M. R. Symonds, “Combining evolutionary inference and metabolomics to identify plants with medicinal potential,” *Frontiers In Ecology And Evolution*, vol. 7, p. 267, 2019.
- [42] J. A. Somarelli, K. E. Ware, R. Kostadinov, *et al.*, “Phylooncology: Understanding cancer through phylogenetic analysis,” *Biochimica et Biophysica Acta (BBA)-Reviews on Cancer*, vol. 1867, no. 2, pp. 101–108, 2017.
- [43] R. Schwartz and A. A. Schäffer, “The evolution of tumour phylogenetics: Principles and practice,” *Nature Reviews Genetics*, vol. 18, no. 4, pp. 213–229, 2017.
- [44] T. A. Brown, “Molecular phylogenetics,” in *Genomes. 2nd edition*, Wiley-Liss, 2002.
- [45] Z. Yang, *Molecular evolution: a statistical approach*. Oxford University Press, 2014.
- [46] H. Izadkhah, “P, np, np-complete, and np-hard problems,” in *Problems on Algorithms: A Comprehensive Exercise Book for Students in Software Engineering*, Springer, 2022, pp. 497–511.
- [47] A. Máté, “Nondeterministic polynomial-time computations and models of arithmetic,” *Journal of the ACM (JACM)*, vol. 37, no. 1, pp. 175–193, 1990.
- [48] D. S. Johnson, “The np-completeness column: An ongoing guide,” *Journal of Algorithms*, vol. 5, no. 2, pp. 284–299, 1984.
- [49] A. Wigderson, “P, np and mathematics—a computational complexity perspective,” in *Proceedings of the ICM*, vol. 6, 2006, pp. 665–712.
- [50] D. Money and S. Whelan, “Characterizing the phylogenetic tree-search problem,” *Systematic biology*, vol. 61, no. 2, p. 228, 2012.

- [51] S. Roch, “A short proof that phylogenetic tree reconstruction by maximum likelihood is hard,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 3, no. 1, pp. 92–94, 2006.
- [52] M. C. Maiden, “Multilocus sequence typing of bacteria,” *Annu. Rev. Microbiol.*, vol. 60, pp. 561–588, 2006.
- [53] T. G. Vaughan, D. Welch, A. J. Drummond, P. J. Biggs, T. George, and N. P. French, “Inferring ancestral recombination graphs from bacterial genomic data,” *Genetics*, vol. 205, no. 2, pp. 857–870, 2017.
- [54] J. Hedge and D. J. Wilson, “Bacterial phylogenetic reconstruction from whole genomes is robust to recombination but demographic inference is not,” *MBio*, vol. 5, no. 6, e02158–14, 2014.
- [55] A. Boc and V. Makarenkov, “Towards an accurate identification of mosaic genes and partial horizontal gene transfers,” *Nucleic acids research*, vol. 39, no. 21, e144–e144, 2011.
- [56] P. Sneath and R. Sokal, “Unweighted pair group method with arithmetic mean,” *Numerical Taxonomy*, pp. 230–234, 1973.
- [57] N. Saitou and M. Nei, “The neighbor-joining method: A new method for reconstructing phylogenetic trees.,” *Molecular biology and evolution*, vol. 4, no. 4, pp. 406–425, 1987.
- [58] M. Goodman and J.-F. Pechère, “The evolution of muscular parvalbumins investigated by the maximum parsimony method,” *Journal of molecular evolution*, vol. 9, no. 2, pp. 131–158, 1977.
- [59] J. Felsenstein, “Evolutionary trees from dna sequences: A maximum likelihood approach,” *Journal of molecular evolution*, vol. 17, no. 6, pp. 368–376, 1981.

- [60] B. Rannala and Z. Yang, “Probability distribution of molecular evolutionary trees: A new method of phylogenetic inference,” *Journal of molecular evolution*, vol. 43, no. 3, pp. 304–311, 1996.
- [61] S. Höhna, M. J. Landis, T. A. Heath, *et al.*, “Revbayes: Bayesian phylogenetic inference using graphical models and an interactive model-specification language,” *Systematic biology*, vol. 65, no. 4, pp. 726–736, 2016.
- [62] S. Li, D. K. Pearl, and H. Doss, “Phylogenetic tree construction using markov chain monte carlo,” *Journal of the American statistical Association*, vol. 95, no. 450, pp. 493–508, 2000.
- [63] Z. Yang *et al.*, *Computational molecular evolution*. Oxford University Press, 2006.
- [64] R. Del Amparo and M. Arenas, “Consequences of substitution model selection on protein ancestral sequence reconstruction,” *Molecular biology and evolution*, vol. 39, no. 7, msac144, 2022.
- [65] T. H. Jukes, C. R. Cantor, *et al.*, “Evolution of protein molecules,” *Mammalian protein metabolism*, vol. 3, no. 21, p. 132, 1969.
- [66] M. Kimura, “A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences,” *Journal of molecular evolution*, vol. 16, no. 2, pp. 111–120, 1980.
- [67] M. Kimura, “Estimation of evolutionary distances between homologous nucleotide sequences,” *Proceedings of the National Academy of Sciences*, vol. 78, no. 1, pp. 454–458, 1981.

- [68] M. Hasegawa, H. Kishino, and T.-a. Yano, “Dating of the human-ape splitting by a molecular clock of mitochondrial dna,” *Journal of molecular evolution*, vol. 22, no. 2, pp. 160–174, 1985.
- [69] S. Tavaré, “Some probabilistic and statistical problems in the analysis of dna sequences,” *Lectures on mathematics in the life sciences*, vol. 17, no. 2, pp. 57–86, 1986.
- [70] J. Felsenstein, “Maximum-likelihood estimation of evolutionary trees from continuous characters,” *American journal of human genetics*, vol. 25, no. 5, p. 471, 1973.
- [71] R. Nielsen, *Statistical methods in molecular evolution*. Springer, 2006.
- [72] A. Cho, “Constructing phylogenetic trees using maximum likelihood,” 2012.
- [73] M.-H. Chen, L. Kuo, and P. O. Lewis, *Bayesian Phylogenetics: methods, algorithms, and applications*. CRC Press, 2014.
- [74] J. P. Huelsenbeck and F. Ronquist, “Bayesian analysis of molecular evolution using mrbayes,” in *Statistical methods in molecular evolution*, Springer, 2005, pp. 183–226.
- [75] A. E. Shikov, Y. V. Malovichko, A. A. Nizhnikov, and K. S. Antonets, “Current methods for recombination detection in bacteria,” *International Journal of Molecular Sciences*, vol. 23, no. 11, p. 6257, 2022.
- [76] T. Ohta and C. J. Basten, “Gene conversion generates hypervariability at the variable regions of kallikreins and their inhibitors,” *Molecular Phylogenetics and Evolution*, vol. 1, no. 2, pp. 87–90, 1992.
- [77] G. F. Weiller, “Phylogenetic profiles: A graphical method for detecting genetic recombinations in homologous sequences,” *Molecular Biology and Evolution*, vol. 15, no. 3, pp. 326–335, 1998.

- [78] D. P. Martin, B. Murrell, M. Golden, A. Khoosal, and B. Muhire, “Rdp4: Detection and analysis of recombination patterns in virus genomes,” *Virus evolution*, vol. 1, no. 1, 2015.
- [79] D. P. Martin, A. Varsani, P. Roumagnac, *et al.*, “Rdp5: A computer program for analyzing recombination in, and removing signals of recombination from, nucleotide sequence datasets,” *Virus Evolution*, vol. 7, no. 1, veaa087, 2021.
- [80] T. C. Bruen, H. Philippe, and D. Bryant, “A simple and robust statistical test for detecting the presence of recombination,” *Genetics*, vol. 172, no. 4, pp. 2665–2681, 2006.
- [81] Y.-P. Lai and T. R. Ioerger, “A statistical method to identify recombination in bacterial genomes based on snp incompatibility,” *BMC bioinformatics*, vol. 19, no. 1, pp. 1–15, 2018.
- [82] M. J. Gibbs, J. S. Armstrong, and A. J. Gibbs, “Sister-scanning: A monte carlo procedure for assessing signals in recombinant sequences,” *Bioinformatics*, vol. 16, no. 7, pp. 573–582, 2000.
- [83] M. de Been, W. van Schaik, L. Cheng, J. Corander, and R. J. Willems, “Recent recombination events in the core genome are associated with adaptive evolution in enterococcus faecium,” *Genome biology and evolution*, vol. 5, no. 8, pp. 1524–1535, 2013.
- [84] R. Mostowy, N. J. Croucher, C. P. Andam, J. Corander, W. P. Hanage, and P. Marttinen, “Efficient inference of recent and ancestral recombination within bacterial populations,” *Molecular biology and evolution*, vol. 34, no. 5, pp. 1167–1182, 2017.
- [85] W.-B. Wang, T. Jiang, and S. Gardner, “Detection of homologous recombination events in bacterial genomes,” *PloS one*, vol. 8, no. 10, e75230, 2013.

- [86] J. Maynard Smith and N. H. Smith, “Detecting recombination from gene trees.,” *Molecular biology and evolution*, vol. 15, no. 5, pp. 590–599, 1998.
- [87] N. J. Croucher, A. J. Page, T. R. Connor, *et al.*, “Rapid phylogenetic analysis of large samples of recombinant bacterial whole genome sequences using gubbins,” *Nucleic acids research*, vol. 43, no. 3, e15–e15, 2015.
- [88] M. Tan, H. Long, B. Liao, *et al.*, “Qs-net: Reconstructing phylogenetic networks based on quartet and sextet,” *Frontiers in genetics*, vol. 10, p. 607, 2019.
- [89] D. P. Martin, P. Lemey, and D. Posada, “Analysing recombination in nucleotide sequences,” *Molecular Ecology Resources*, vol. 11, no. 6, pp. 943–955, 2011.
- [90] X. Didelot and D. Falush, “Inference of bacterial microevolution using multilocus sequence data,” *Genetics*, vol. 175, no. 3, pp. 1251–1266, 2007.
- [91] X. Didelot and D. J. Wilson, “Clonalframeml: Efficient inference of recombination in whole bacterial genomes,” *PLoS computational biology*, vol. 11, no. 2, e1004041, 2015.
- [92] R. Milkman and M. Bridges, “Molecular evolution of the escherichia coli chromosome. iv. sequence comparisons.,” *Genetics*, vol. 133, no. 3, pp. 455–468, 1993.
- [93] B. Q. Minh, H. A. Schmidt, O. Chernomor, *et al.*, “Iq-tree 2: New models and efficient methods for phylogenetic inference in the genomic era,” *Molecular biology and evolution*, vol. 37, no. 5, pp. 1530–1534, 2020.
- [94] L.-T. Nguyen, H. A. Schmidt, A. Von Haeseler, and B. Q. Minh, “Iq-tree: A fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies,” *Molecular biology and evolution*, vol. 32, no. 1, pp. 268–274, 2015.

- [95] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.
- [96] A. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” *IEEE transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.
- [97] G. D. Forney, “The viterbi algorithm,” *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.
- [98] A. M. Kozlov, D. Darriba, T. Flouri, B. Morel, and A. Stamatakis, “Raxml-ng: A fast, scalable and user-friendly tool for maximum likelihood phylogenetic inference,” *Bioinformatics*, vol. 35, no. 21, pp. 4453–4455, 2019.
- [99] M. Fourment. “Physher.” (), [Online]. Available: <https://github.com/4ment/physher>. (accessed: 18 Aug 2022).
- [100] R. Bouckaert, J. Heled, D. Kühnert, *et al.*, “Beast 2: A software platform for bayesian evolutionary analysis,” *PLoS computational biology*, vol. 10, no. 4, e1003537, 2014.
- [101] F. Ronquist, M. Teslenko, P. Van Der Mark, *et al.*, “Mrbayes 3.2: Efficient bayesian phylogenetic inference and model choice across a large model space,” *Systematic biology*, vol. 61, no. 3, pp. 539–542, 2012.
- [102] Y. Turakhia, B. Thornlow, A. S. Hinrichs, *et al.*, “Ultrafast sample placement on existing trees (usher) enables real-time phylogenetics for the sars-cov-2 pandemic,” *Nature Genetics*, vol. 53, no. 6, pp. 809–816, 2021.

- [103] D. T. Hoang, L. S. Vinh, T. Flouri, A. Stamatakis, A. von Haeseler, and B. Q. Minh, "Mpbboot: Fast phylogenetic maximum parsimony tree inference and bootstrap approximation," *BMC evolutionary biology*, vol. 18, no. 1, pp. 1–11, 2018.
- [104] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [105] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.
- [106] L. Pachter, M. Alexandersson, and S. Cawley, "Applications of generalized pair hidden markov models to alignment and gene finding problems," in *Proceedings of the fifth annual international conference on Computational biology*, 2001, pp. 241–248.
- [107] K. Munch and A. Krogh, "Automatic generation of gene finders for eukaryotic species," *BMC bioinformatics*, vol. 7, no. 1, pp. 1–12, 2006.
- [108] J. Razmara, S. B. Deris, R. B. M. Illias, and S. Parvizpour, "Artificial signal peptide prediction by a hidden markov model to improve protein secretion via lactococcus lactis bacteria," *Bioinformation*, vol. 9, no. 7, p. 345, 2013.
- [109] Z. I. Litou, P. G. Bagos, K. D. Tsirigos, T. D. Liakopoulos, and S. J. Hamodrakas, "Prediction of cell wall sorting signals in gram-positive bacteria with a hidden markov model: Application to complete genomes," *Journal of bioinformatics and computational biology*, vol. 6, no. 02, pp. 387–401, 2008.
- [110] K. Asai, S. Hayamizu, and K. Handa, "Prediction of protein secondary structure by the hidden markov model," *Bioinformatics*, vol. 9, no. 2, pp. 141–146, 1993.

- [111] A. Krogh, B. Larsson, G. Von Heijne, and E. L. Sonnhammer, “Predicting transmembrane protein topology with a hidden markov model: Application to complete genomes,” *Journal of molecular biology*, vol. 305, no. 3, pp. 567–580, 2001.
- [112] J. Felsenstein and G. A. Churchill, “A hidden markov model approach to variation among sites in rate of evolution.,” *Molecular biology and evolution*, vol. 13, no. 1, pp. 93–104, 1996.
- [113] B. Boussau, L. Guéguen, and M. Gouy, “A mixture model and a hidden markov model to simultaneously detect recombination breakpoints and reconstruct phylogenies,” *Evolutionary Bioinformatics*, vol. 5, EBO–S2242, 2009.
- [114] B. Wang, J. Deng, Y. Sun, W. Guo, and G. Feng, “Secrecy capacity of a class of erasure wiretap channels in wban,” *Sensors*, vol. 18, no. 12, p. 4135, 2018.
- [115] S. Capella-Gutiérrez, J. M. Silla-Martínez, and T. Gabaldón, “Trimal: A tool for automated alignment trimming in large-scale phylogenetic analyses,” *Bioinformatics*, vol. 25, no. 15, pp. 1972–1973, 2009.
- [116] G. Talavera and J. Castresana, “Improvement of phylogenies after removing divergent and ambiguously aligned blocks from protein sequence alignments,” *Systematic biology*, vol. 56, no. 4, pp. 564–577, 2007.
- [117] G. Tan, M. Muffato, C. Ledergerber, *et al.*, “Current methods for automated filtering of multiple sequence alignments frequently worsen single-gene phylogenetic inference,” *Systematic biology*, vol. 64, no. 5, pp. 778–791, 2015.
- [118] A. Cornish-Bowden, “Nomenclature for incompletely specified bases in nucleic acid sequences: Recommendations 1984.,” *Nucleic acids research*, vol. 13, no. 9, p. 3021, 1985.

- [119] J. Felsenstein and J. Felsenstein, *Inferring phylogenies*. Sinauer associates Sunderland, MA, 2004, vol. 2, ch. Likelihood methods, pg. 256.
- [120] J. Parker, A. Rambaut, and O. G. Pybus, “Correlating viral phenotypes with phylogeny: Accounting for phylogenetic uncertainty,” *Infection, Genetics and Evolution*, vol. 8, no. 3, pp. 239–246, 2008.
- [121] M. K. Kuhner and J. McGill, “Correcting for sequencing error in maximum likelihood phylogeny inference,” *G3: Genes, Genomes, Genetics*, vol. 4, no. 12, pp. 2545–2552, 2014.
- [122] O. Kozlov, “Models, optimizations, and tools for large-scale phylogenetic inference, handling sequence uncertainty, and taxonomic validation,” Ph.D. dissertation, KIT-Bibliothek, 2018.
- [123] A. McKenna, M. Hanna, E. Banks, *et al.*, “The genome analysis toolkit: A mapreduce framework for analyzing next-generation dna sequencing data,” *Genome research*, vol. 20, no. 9, pp. 1297–1303, 2010.
- [124] A. Stamatakis, “Raxml version 8: A tool for phylogenetic analysis and post-analysis of large phylogenies,” *Bioinformatics*, vol. 30, no. 9, pp. 1312–1313, 2014.
- [125] M. Fourment. “Scipy.” (), [Online]. Available: <https://docs.scipy.org/doc/>. (accessed: 18 Aug 2022).
- [126] S. L. Kosakovsky Pond, B. Murrell, M. Fourment, S. D. Frost, W. Delport, and K. Scheffler, “A random effects branch-site model for detecting episodic diversifying selection,” *Molecular biology and evolution*, vol. 28, no. 11, pp. 3033–3043, 2011.
- [127] M. Arenas and D. Posada, “The effect of recombination on the reconstruction of ancestral sequences,” *Genetics*, vol. 184, no. 4, pp. 1133–1139, 2010.

- [128] M. A. Suchard, P. Lemey, G. Baele, D. L. Ayres, A. J. Drummond, and A. Rambaut, “Bayesian phylogenetic and phylodynamic data integration using beast 1.10,” *Virus evolution*, vol. 4, no. 1, vey016, 2018.
- [129] D. A. Dalquen, M. Anisimova, G. H. Gonnet, and C. Dessimoz, “Alf—a simulation framework for genome evolution,” *Molecular biology and evolution*, vol. 29, no. 4, pp. 1115–1123, 2012.
- [130] S. Hoban, G. Bertorelle, and O. E. Gaggiotti, “Computer simulations: Tools for population and evolutionary genetics,” *Nature Reviews Genetics*, vol. 13, no. 2, pp. 110–122, 2012.
- [131] A. Sipola, P. Marttinen, and J. Corander, “Bacmeta: Simulator for genomic evolution in bacterial metapopulations,” *Bioinformatics*, vol. 34, no. 13, pp. 2308–2310, 2018.
- [132] T. Brown, X. Didelot, D. J. Wilson, and N. De Maio, “Simbac: Simulation of whole bacterial genomes with homologous recombination,” *Microbial genomics*, vol. 2, no. 1, 2016.
- [133] N. De Maio and D. J. Wilson, “The bacterial sequential markov coalescent,” *Genetics*, vol. 206, no. 1, pp. 333–343, 2017.
- [134] G. A. McVean and N. J. Cardin, “Approximating the coalescent with recombination,” *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 360, no. 1459, pp. 1387–1393, 2005.
- [135] P. Marjoram and J. D. Wall, “Fast" coalescent" simulation,” *BMC genetics*, vol. 7, no. 1, pp. 1–9, 2006.
- [136] T. Akita, S. Takuno, and H. Innan, “Coalescent framework for prokaryotes undergoing interspecific homologous recombination,” *Heredity*, vol. 120, no. 5, pp. 474–484, 2018.

- [137] L.-M. Bobay, “Coresimul: A forward-in-time simulator of genome evolution for prokaryotes modeling homologous recombination,” *BMC bioinformatics*, vol. 21, no. 1, pp. 1–7, 2020.
- [138] J. Cury, B. C. Haller, G. Achaz, and F. Jay, “Simulation of bacterial populations with slim,” *Peer Community Journal*, vol. 2, 2022.
- [139] J. F. C. Kingman, “The coalescent,” *Stochastic processes and their applications*, vol. 13, no. 3, pp. 235–248, 1982.
- [140] M. M. Saber and B. J. Shapiro, “Benchmarking bacterial genome-wide association study methods using simulated genomes and phenotypes,” *Microbial genomics*, vol. 6, no. 3, 2020.
- [141] A. Rambaut and N. C. Grass, “Seq-gen: An application for the monte carlo simulation of dna sequence evolution along phylogenetic trees,” *Bioinformatics*, vol. 13, no. 3, pp. 235–238, 1997.
- [142] P. Di Tommaso, M. Chatzou, E. W. Floden, P. P. Barja, E. Palumbo, and C. Notredame, “Nextflow enables reproducible computational workflows,” *Nature biotechnology*, vol. 35, no. 4, pp. 316–319, 2017.
- [143] D. F. Robinson and L. R. Foulds, “Comparison of phylogenetic trees,” *Mathematical biosciences*, vol. 53, no. 1-2, pp. 131–147, 1981.
- [144] M. K. Kuhner and J. Felsenstein, “A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates,” *Molecular biology and evolution*, vol. 11, no. 3, pp. 459–468, 1994.

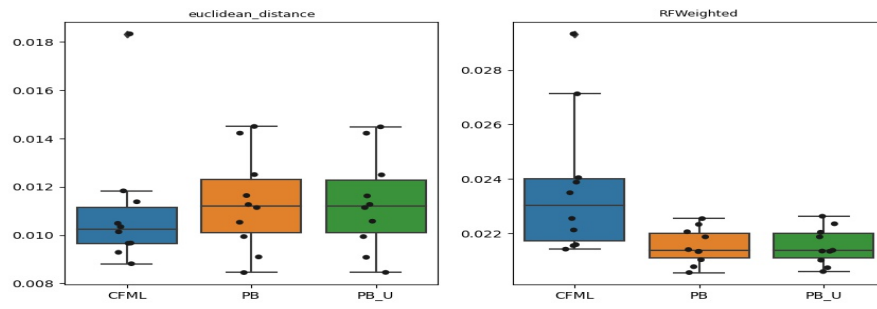
- [145] K. L. O'brien, L. J. Wolfson, J. P. Watt, *et al.*, "Burden of disease caused by streptococcus pneumoniae in children younger than 5 years: Global estimates," *The Lancet*, vol. 374, no. 9693, pp. 893–902, 2009.
- [146] S. Castillo-Ramírez, J. Corander, P. Marttinen, *et al.*, "Phylogeographic variation in recombination rates within a global clone of methicillin-resistant staphylococcus aureus," *Genome biology*, vol. 13, no. 12, pp. 1–13, 2012.
- [147] E. J. Feil, J. E. Cooper, H. Grundmann, *et al.*, "How clonal is staphylococcus aureus?" *Journal of bacteriology*, vol. 185, no. 11, pp. 3307–3316, 2003.
- [148] S. R. Harris, E. J. Feil, M. T. Holden, *et al.*, "Evolution of mrsa during hospital transmission and intercontinental spread," *Science*, vol. 327, no. 5964, pp. 469–474, 2010.
- [149] A. S. Lee, H. De Lencastre, J. Garau, *et al.*, "Methicillin-resistant staphylococcus aureus," *Nature reviews Disease primers*, vol. 4, no. 1, pp. 1–23, 2018.
- [150] R. H. McDowell, E. M. Sands, and H. Friedman, "Bacillus cereus," in *StatPearls [Internet]*, StatPearls Publishing, 2022.
- [151] R. T. Okinaka and P. Keim, "The phylogeny of bacillus cereus sensu lato," *The bacterial spore: from molecules to systems*, pp. 237–251, 2016.
- [152] F. G. Priest, M. Barker, L. W. Baillie, E. C. Holmes, and M. C. Maiden, "Population structure and evolution of the bacillus cereus group," *Journal of bacteriology*, vol. 186, no. 23, pp. 7959–7970, 2004.
- [153] N. J. Tourasse, K. A. Jolley, A.-B. Kolstø, and O. A. Økstad, "Core genome multilocus sequence typing scheme for bacillus cereus group bacteria," *Research in Microbiology*, p. 104 050, 2023.

- [154] “Cython c extension for python.” (), [Online]. Available: <https://cython.org/>. (accessed: 1 June 2022).
- [155] D. L. Ayres, A. Darling, D. J. Zwickl, *et al.*, “Beagle: An application programming interface and high-performance computing library for statistical phylogenetics,” *Systematic biology*, vol. 61, no. 1, pp. 170–173, 2012.
- [156] M. A. Suchard and A. Rambaut, “Many-core algorithms for statistical phylogenetics,” *Bioinformatics*, vol. 25, no. 11, pp. 1370–1376, 2009.

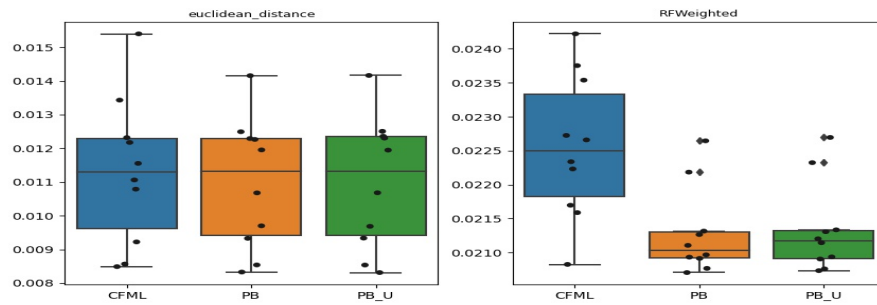
APPENDICES

APPENDIX A

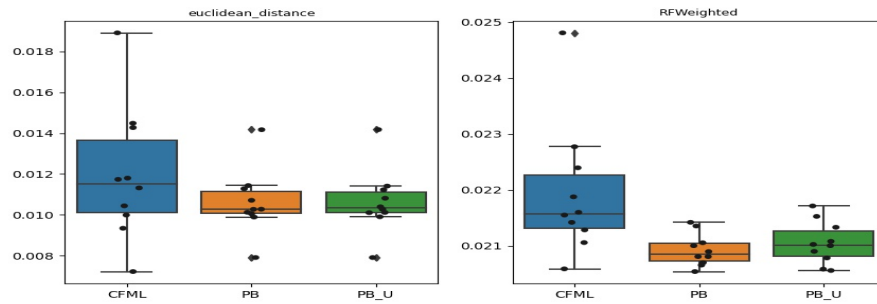
MORE EXPERIMENT



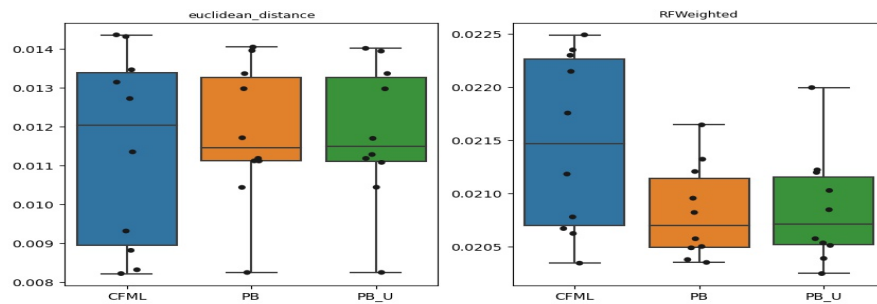
(a) nu:0.01



(b) nu:0.02

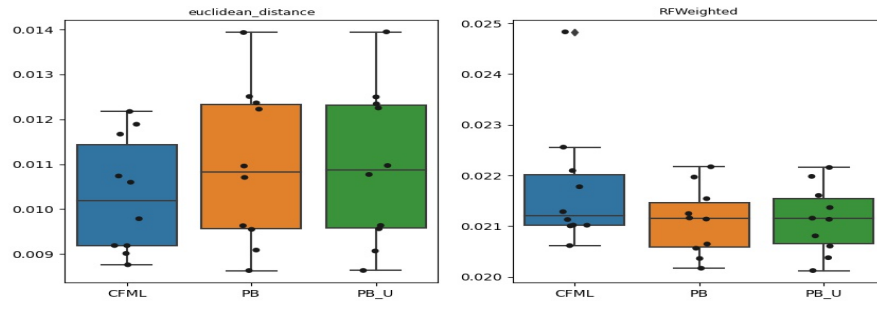


(c) nu:0.03

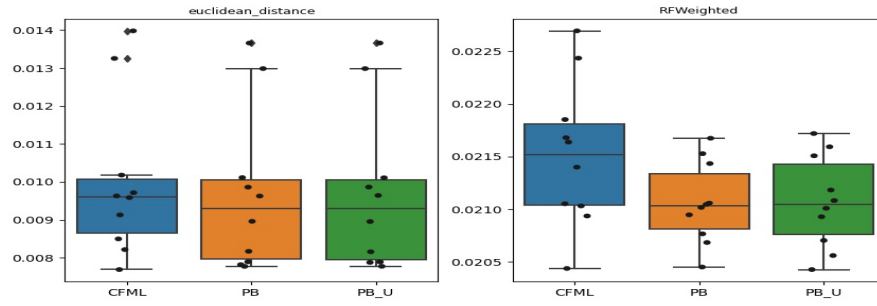


(d) nu:0.04

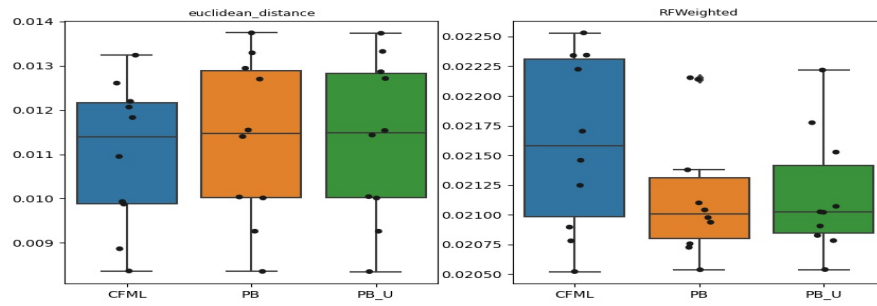
Figure A.1 **BaciSim Simulator:** This figure is the same as Figure 5.3. We removed Gubbins's data to clarify the differences between CFML and PhiloBacter(s).



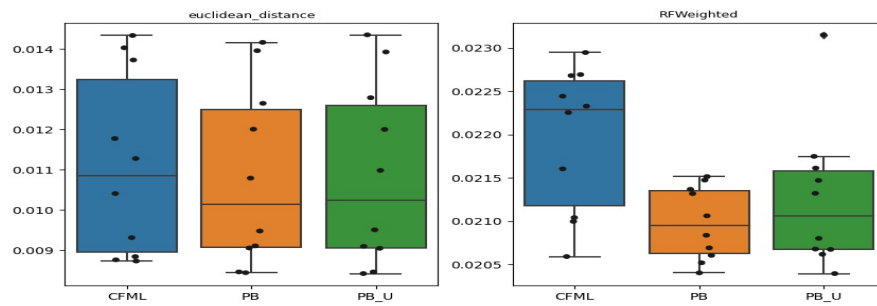
(a) nu:0.05



(b) nu:0.06

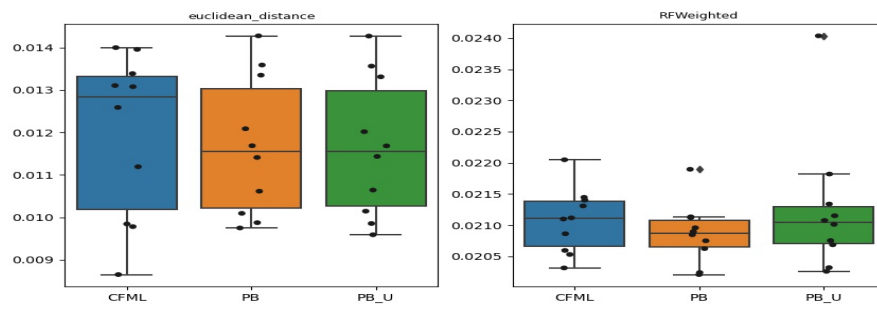


(c) nu:0.07

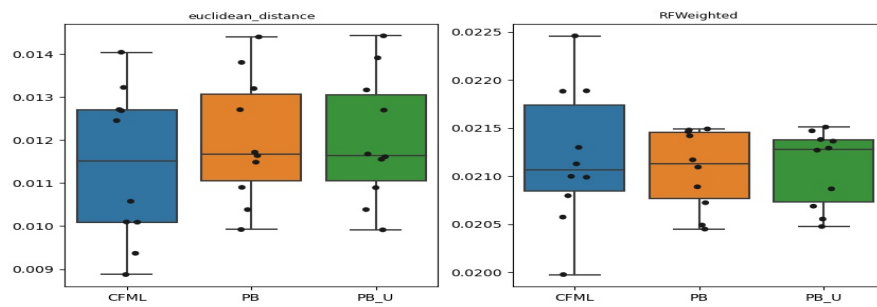


(d) nu:0.08

Figure A.2 **BaciSim Simulator:** This figure is the same as Figure 5.4. We removed Gubbins's data to clarify the differences between CFML and PhiloBacter(s).

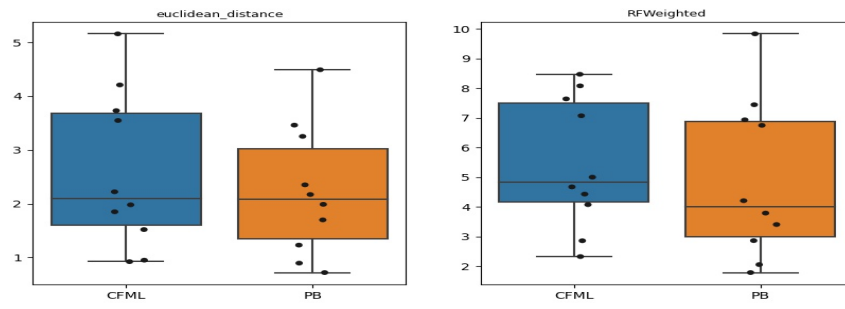


(a) $\nu:0.09$

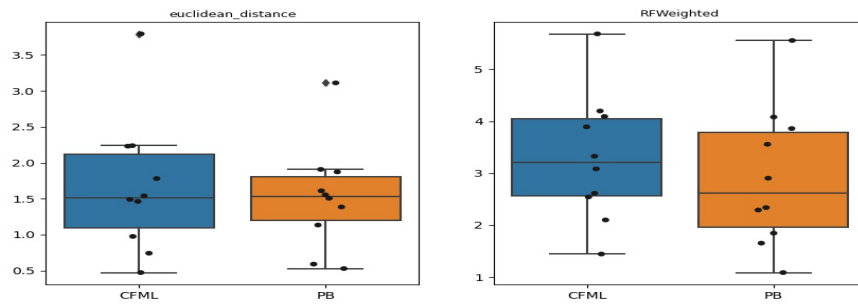


(b) $\nu:0.1$

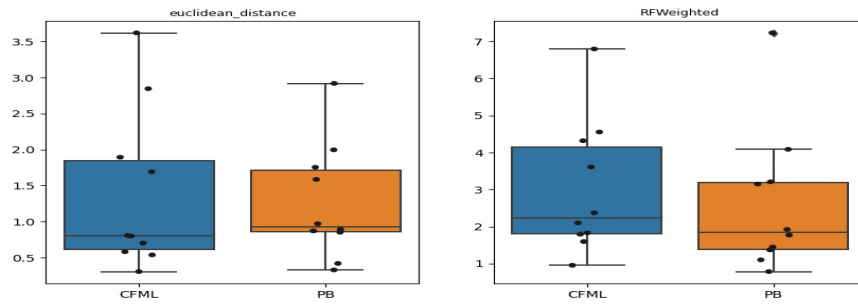
Figure A.3 **BaciSim Simulator:** This figure is the same as Figure 5.5. We removed Gubbins's data to clarify the differences between CFML and PhiloBacter(s).



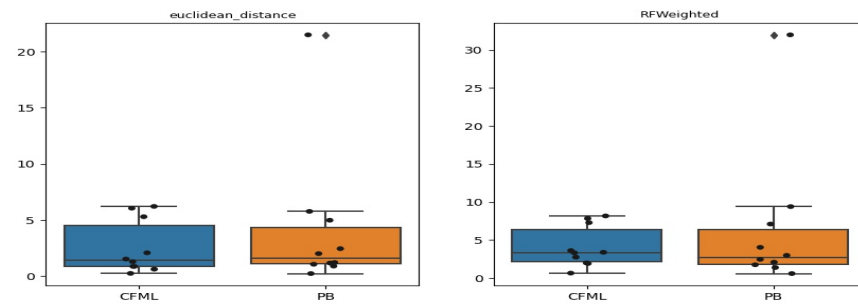
(a) $\nu:0.01$



(b) $\nu:0.02$

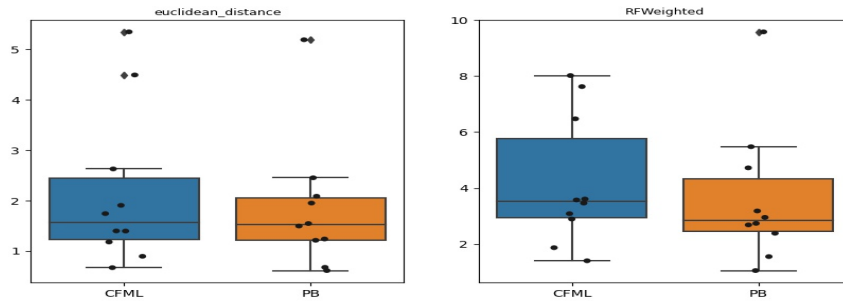


(c) $\nu:0.03$

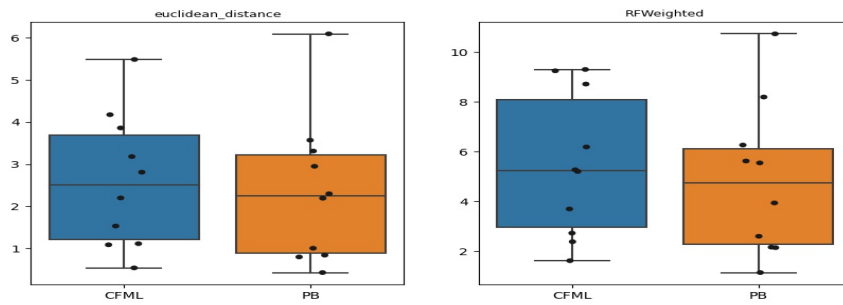


(d) $\nu:0.04$

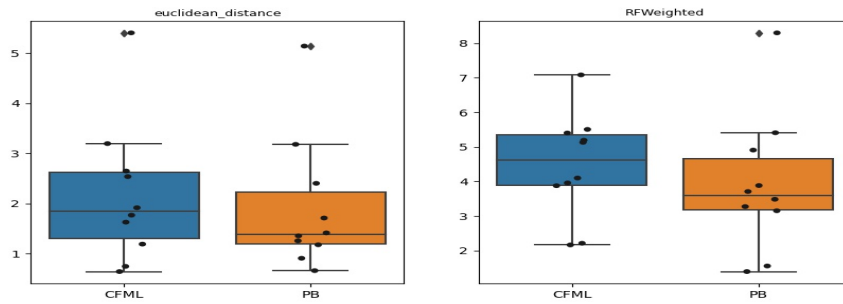
Figure A.4 **SimBac Simulator:** This figure is the same as Figure 5.10. We removed Gubbins data to clarify the differences between CFML and PhiloBacter.



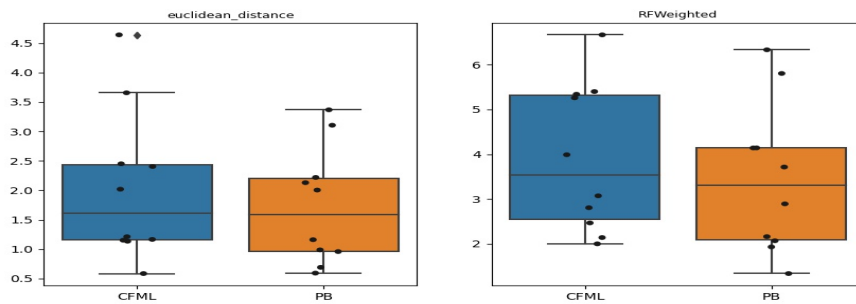
(a) $\nu:0.05$



(b) $\nu:0.06$

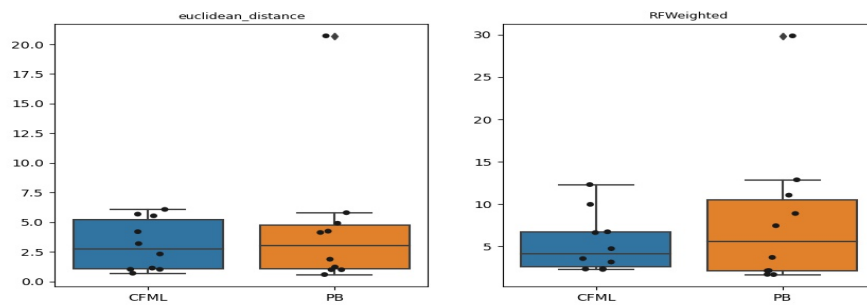


(c) $\nu:0.07$

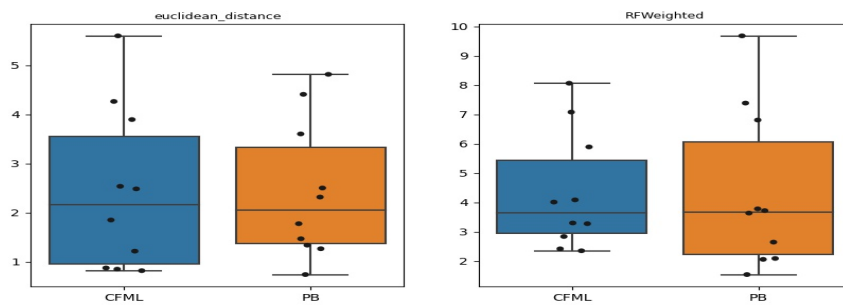


(d) $\nu:0.08$

Figure A.5 SimBac Simulator: This figure is the same as Figure 5.11. We removed Gubbins data to clarify the differences between CFML and PhiloBacter.



(a) $\nu:0.01$



(b) $\nu:0.02$

Figure A.6 **SimBac Simulator:** This figure is the same as Figure 5.12. We removed Gubbins data to clarify the differences between CFML and PhiloBacter.


Isolate fields 							MLST							
Id	Isolate	aliases	variety serovar	country	year	source	glp	gmk	ilv	pta	pur	pyc	tpl	ST
60	T03a361	Dru 4	kurstaki	Australia	1990		7	8	16	13	2	16	7	8
114	K4834/A0039	ASC_383; LSU39	anthracis	Australia	1994		1	1	1	1	1	1	1	1
1111	B11/0221		cereus	Australia	2011	blood	145	1	83	1	1	37	88	595
2821	DAR 81934	GCA_000342025.1; NZ_CM001804.1; PRJNA179326; PRJNA198418; SAMN02469463	thuringiensis	Australia			15	7	7	2	7	8	13	197
2979	K4834	GCA_001273065.1; LFYJ01; PRJNA257008; SAMN03757458; SRS966994	anthracis	Australia	1997		1	1	1	1	1	1	1	1
3098	A50	GCA_001729445.1; MAID01; PRJNA324744; SAMN05231870	cereus	Australia	2014		33	8		19	2	17	17	
3100	A9	GCA_001729295.1; LZPN01; PRJNA324744; SAMN05225334	cereus	Australia	2014		50	8	14	12	12	36	7	139
3103	LCR12	GCA_001699805.1; MCAX01; PRJNA331062; SAMN05436732	cereus	Australia	2014		11	9	14	12	12	14	7	18
4801	A0006		anthracis	Australia			1	1	1	1	1	1	1	1
4946	A0224		anthracis	Australia			1	1	1	1	1	1	1	1
5108	BANT009		anthracis	Australia	1997		1	1	1	1	1	1	1	1

Figure A.7 Details of Australian genomes used in section 5.5.3 Application to *Bacillus Cereus*.