ESTABLISHING A TRUSTWORTHY FIRST APPROXIMATION FOR EVOLUTIONARY DISTANCES

APPROVED BY SUPERVISORY COMMITTEE

Zbyszek Otwinowski
Nick V. Grishin
Yuh Min Chook
Lora Hooper
Khuloud Jaqaman

DEDICATION

I am thankful to everyone who helped and supported me during my time performing this work.

ESTABLISHING A TRUSTWORTHY FIRST APPROXIMATION FOR EVOLUTIONARY DISTANCES

by

RAQUEL BROMBERG

DISSERTATION

Presented to the Faculty of the Graduate School of Biomedical Sciences

The University of Texas Southwestern Medical Center at Dallas

In Partial Fulfillment of the Requirements

For the Degree of

DOCTOR OF PHILOSOPHY

The University of Texas Southwestern Medical Center at Dallas

Dallas, Texas

May, 2016

ESTABLISHING A TRUSTWORTHY FIRST APPROXIMATION FOR EVOLUTIONARY DISTANCES

RAQUEL BROMBERG, Ph.D.

The University of Texas Southwestern Medical Center at Dallas, 2016

ZBYSZEK OTWINOWSKI, Ph.D.
NICK V. GRISHIN, Ph.D.

Advances in sequencing have generated a large number of complete genomes. Traditionally, phylogenetic analysis relies on alignments of orthologs, but defining orthologs and separating them from paralogs is a complex task that may not always be suited to the large datasets of the future. An alternative to traditional, alignment-based approaches are whole-genome, alignment-free methods. These methods are scalable and require minimal manual intervention. I developed SlopeTree, a new alignment-free method that estimates evolutionary distances by measuring the decay of exact sub-sequence matches as a function of match length. SlopeTree corrects for horizontal gene transfer, for composition variation and low complexity sequences, and for branch-length nonlinearity caused by multiple

mutations at the same site. SlopeTree also includes several optional features for removing mobile elements from proteomes, for reducing proteomes to their conserved core, for automatically identifying poor quality proteomes in large inputs, and for explicitly identifying pairs of organisms that have horizontally transferred genes and then identifying those genes.

I tested SlopeTree on large and diverse sets of bacteria and archaea, and I also applied it at the strain level. I compared the SlopeTree trees to the NCBI taxonomy, to trees based on concatenated alignments, and to trees produced by other alignment-free methods. The results were consistent with current knowledge about prokaryotic evolution. I assessed differences in tree topology over different methods and settings and found that the majority of bacteria and archaea have a core set of proteins that evolves by descent. In trees built from complete genomes rather than from sets of core genes, I observed some grouping by phenotype rather than phylogeny.

In general, SlopeTree generates sensible topologies which are relatively stable between whole proteome and reduced proteome inputs, which validates the concept of species and phyla as having a core proteome evolving by descent, but not necessarily coevolving with the ribosome and its proteins.

.

TABLE OF CONTENTS

ABSTRACT v

PRIOR PUBLICATIONS xiv

LIST OF TABLES xv

LIST OF FIGURES xvi

LIST OF APPENDICES xviii

CHAPTER ONE: INTRODUCTION 20

1.1 DIVERSITY OF CELLULAR LIFE 20

Domains of cellular life 20

Horizontal gene transfer 21

1.2 HISTORY OF PHYLOGENETICS FOR PROKARYOTES 22

Prokaryotic phylogeny before molecular phylogenetics 22

Molecular phylogenetics 23

Problems with traditional methods: different genes tell different stories 25

1.3 THE GENOMIC DATA FLOOD 26

Advances in sequencing technology 26

From highly curated data to Big Data 27

How many species are there? 28

1.4 ALIGNMENT-FREE WHOLE-GENOME METHODS 29

Types of methods 29

Survey of current alignment-free methods 30

Composition Vector Trees (CVTree) 30
Feature Frequency Profiles (FFP) 31
D2 statistics 32
Co-phylog 33
Spaced Words (SW) 35
Average Common Substring (ACS) 36
Kmacs 36
ALFRED-G 37
The Kr method 38
1.5 WHY IS PHYLOGENETICS IMPORTANT? 38
Understanding how life evolves 39
Phylogenetics underlies taxonomy 39
Microbiome characterization 40
1.6 DISSERTATION OVERVIEW 40
SlopeTree overview 41
Horizontal gene transfer and alignment-free methods 42
The SlopeTree package 43
Assessing SlopeTree 44
CHAPTER TWO 48
2.1 MOTIVATION 48
2.2 SLOPETREE OVERVIEW 49
The main SlopeTree algorithm 49
2.3 IMPLEMENTATION 50
Assigning unique ordinals to proteomes and proteins 50
Assembling the k-mer lists 51

Removing low complexity sequences 52
Counting unique matches....53
Match-counting algorithm 1....53
Match-counting algorithm 2....54
Scoring matches 55
The SlopeTree match-count histogram....57
Background subtraction 58
Identifying left and right bounds for the SlopeTree data....60
Estimating evolutionary distances....62
Fitting the data....62
Constructing the distance matrices....64
2.3 RESULTS FOR SLOPETREE V1....64
137 Archaea 65
Bacteria 67
Comparison to other methods and distance to the 16S rRNA trees....71
2.4 DISCUSSION AND CONCLUSIONS 72
2.5 MATERIALS AND METHODS 73
Downloading proteomes, selecting input sets, and reference trees....73
Neighbor Joining 74
Pruning trees 74
Building SlopeTree Trees and other trees for comparison 75
Tree visualization 75

CHAPTER THREE 94
3.1 MOTIVATION 94
3.2 ADDRESSING MISPLACED ORGANISMS IN SLOPETREE TOPOLOGIES 95
Misplacement of *Petrotoga mobilis* 95
3.3 IMPLEMENTATION REFINEMENTS 97
Correcting for binning artifacts. 98
Improved bounds selection 99
Introducing a weighted fit 100
Converting slopes to evolutionary distances and correcting for revertants 100
Applying a Tikhonov positive restraint 102
Replacing SlopeTree's linear fit with a quadratic fit 103
3.4 INTRODUCING SLOPETREE FILTERS FOR PRE-PROCESSING INPUT DATA 105
Filtering mobile elements 106
Algorithm 1: Mobile Element Filter 107
Filtering by conservation 109
Algorithm 2: Conservation and Stability Filter 110
Selecting a reference 112
Flagging potentially problematic inputs 112
3.5 RESULTS 114
Filtering for mobile elements and by stability and conservation 115
Strain-level analysis 117
SlopeTree filtering benefits other methods 119

3.6 MATERIALS AND METHODS 119
Downloading proteomes, selecting input sets, and building Eisen-trees 119
Pruning trees 120
Building SlopeTree Trees 121
Commands for constructing the raw SlopeTree trees for the sets of bacteria, archaea and *E.coli* 121
Selecting the reference sets for bacteria and archaea 121
Building ST-trees with mobile elements removed 121
Building trees filtered by conservation 123
Building Alternative Trees 124
Average Common Substring 124
Composition Vector Tree (CVTree) 124
D2 Method 125
kmacs 125
Spaced Words 125
ALFRED-G 125
Comparing Trees 125
CHAPTER FOUR 145
4.1 INTRODUCTION AND MOTIVATION 145
4.2 AUTOMATIC IDENTIFICATION AND CORRECTION FOR SPECIFIC TYPES OF HORIZONTAL GENE TRANSFER 146
4.3 IMPLEMENTATION 147

Implementing a new fit: a sum of two exponentials 147
Problems with the fit 148
4.4 CORRECTING FOR HGT EXPLICITLY 150
Flagging organism pairs exhibiting signs of HGT 150
Two passes through the main SlopeTree match-counting algorithm 150
Algorithm 4: Pair-Wise Horizontal Gene Transfer (HGT) Correction 151
Examples of HGT, identified by the SlopeTree HGT correction 153
4.5 FINAL RESULTS ACROSS ALL CORRECTIONS AND FILTERS 153
SlopeTree applied to 73 archaea 154
SlopeTree applied to 495 bacteria 155
Bacteria that diverge from the Eisen-495 tree or the NCBI classification 158
Coprothermobacter proteolyticus, Dictyoglomi, Thermotogae and Synergistetes. 158
A sulfur-reducing thermophilic cluster. 158
Acidithiobacillus ferrooxidans ATCC 23270 and *Acidithiobacillus caldus.* 160
Dehalogenimonas lykanthroporepellens and *Dehalococcoides mccartyi 195.* 160
Rhodothermus marinus and *Salinibacter ruber* 161
Distances to Eisen-trees and other whole-proteome or alignment-free methods 161
SlopeTree trees using the HGT correction 162
4.6 DISCUSSION AND CONCLUSIONS 163
SlopeTree filtering benefits other methods 163
CHAPTER FIVE: CONCLUSIONS AND FUTURE DIRECTIONS 171
5.1 FUTURE DIRECTIONS 172
SlopeTree future development 172
Generating fast, high-quality, automatic alignments 174

Web-server for SlopeTree with selection for automatically generated, diverse taxa.....175

Bibliography 209

PRIOR PUBLICATIONS

Ayaz, P., S. Munyoki, E. A. Geyer, F. A. Piedra, E. S. Vu, R. Bromberg, Z. Otwinowski, N. V. Grishin, C. A. Brautigam and L. M. Rice (2014). "A tethered delivery mechanism explains the catalytic action of a microtubule polymerase." Elife **3**: e03069.

LIST OF TABLES

Table 2-1. Seven misplaced bacteria for early version of SlopeTree.....91
Table 2-2. Comparison to other methods (distance to the 16S rRNA trees). 92
Table 4-1. Symmetric difference distance to Eisen trees for SlopeTree and for six other whole-genome methods, over different levels of mobile-element and conservation filtering. 170

LIST OF FIGURES

Figure 1-1. Phylogeny reconstruction flowchart for SlopeTree. 47
Figure 2-1. Final k-mer list. 76
Figure 2-2. Rejecting low-complexity sequences. 77
Figure 2-3. The correlation matrix. 78
Figure 2-4. SlopeTree match-counting algorithm 1. 79
Figure 2-5. SlopeTree match-counting algorithm 2 (pseudocode). 80
Figure 2-6. SlopeTree match-counting algorithm 2 (visual example). 81
Figure 2-7. Calculating nit-scores for a sequence match between two organisms. 82
Figure 2-8. SlopeTree plot. 83
Figure 2-9. Subtracting the background. 84
Figure 2-10. Original bounds selection. 85
Figure 2-11. The meaning of SlopeTree slopes. 86
Figure 2-12. SlopeTree (v1) applied to 2001 bacteria. 87
Figure 2-13. Phylogenetic tree constructed by SlopeTree (v1). 88
Figure 2-14. SlopeTree (v1) applied to 137 archaea. 89
Figure 2-15. Phylogenetic Trees for SlopeTree (v1), 16S rRNA tree, and NCBI over 137 archaea. 90
Figure 3-1. Extracted evolutionary signal from a SlopeTree plot 127
Figure 3-2. SlopeTree plot for HGT instance. 128
Figure 3-3. SlopeTree plot for pair sharing a transfer from a single copy phage. 129
Figure 3-4. Binning artifacts 130
Figure 3-5. Calculating the effective amino acid population size 131
Figure 3-6. Positive restraint on SlopeTree distances. 132
Figure 3-7. Conserved protein identification. 133
Figure 3-8. Bacterial reference set. 134
Figure 3-9. Archaeal reference set. 135
Figure 3-11. SlopeTree tree of 72 *Escherichia coli* and *Shigella* using 20-mers. 137

Figure 3-12. Outgroups used in strain level SlopeTree tree. 138
Figure 3-13. 40-mer tree of *Escherichia coli/Shigella*. 139
Figure 3-14. SlopeTree and other alignment-free methods. 140
Figure 3-15. Eisen-495 trees for bacteria. 141
Figure 3-16. Eisen-73 trees for archaea. 142
Figure 3-17. Eisen-445 trees for archaea. 143
Figure 3-18. Eisen-71 trees for archaea. 144
Figure 4-1. Instability of the fit from the sum of two exponentials. 165
Figure 4-2. Correcting the 2 main classes of large-scale HGT. 166
Figure 4-3. Phylogenetic trees for 73 Archaea. 167
Figure 4-4. ST-tree of 495 Bacteria. 168

LIST OF APPENDICES

APPENDIX A 177
APPENDIX B 189
APPENDIX C 191
APPENDIX D 193
APPENDIX E 197

LIST OF DEFINITIONS

AF – Alignment free

ACS – Average Common Substring

ME – Mobile element

HGT – Horizontal gene transfer

MSA – Multiple sequence alignment

ST – SlopeTree

CHAPTER ONE

INTRODUCTION

1.1 DIVERSITY OF CELLULAR LIFE

Domains of cellular life

The three domains of cellular life—the Archaea, Eubacteria, and Eukaryota—were first resolved in 1977 by Carl Woese (1), who used an alignment of the 16S small subunit (SSU) rRNA from a diverse group of species. The concept of these three domains of cellular life has persisted, relatively undisputed, despite the dramatic increase in genomic data that we have seen in recent years. This clear, unambiguous separation of the domains exists both at the level of gene analysis and also at the level of phenotypic traits such as lipid content (2) and the complexity of ultrastructure. Without a doubt, these distinctions represent critical events in biological history, and they separate the characterization and classification of members from each domain into three different problems. Eukaryotes have one specific set of characteristics, which cause specific problems during analysis. Some of these characteristics are genome size, which can vary over an enormous range, introns, poor genomic coverage, poor genome annotation, and the prevalence of non-coding DNA. Bacteria and archaea, which form two separate domains, nevertheless have similar types of problems in classification. These genomes, if one ignores non-free-living organisms, cover a range consisting of approximately 1 order of magnitude (~1-13 million base pairs). Since the existence of the three domains of life has already been established and methods for resolving

them already exist, classification methods now should be tuned for each domain's characteristics specifically.

One problem common to all three domains is the problem of defining the root. Nevertheless, the root zone is well-defined and equivalent to the taxonomic root. With respect to bacteria and archaea, we have the concept of phyla, which are early-diverging branches. These branches are, by definition, close to the root zone.

The methods I developed here can be applied to all three domains but were tuned for prokaryotic domains, in particular bacteria, and take advantage of specific genomic and evolutionary features of prokaryotes. The methods give particular attention to domain-specific problems such as horizontally transferred mobile elements.

Horizontal gene transfer

Evolution by descent, also called vertical evolution, is essentially descent with modification, in which small-scale modifications between generations can mean, in the longer timescales, distinct species descending from a common ancestor. In contrast, horizontal gene transfer involves the swapping of groups of genes (i.e. operons) between organisms, opposed to the vertical transmission of genes from a parent to its offspring. For eukaryotes, there is also the issue of genome combination by endo-assimilation (intracellular assimilation) of other species, which may eventually become organelles. This was the case both for chloroplasts in the case of plants and mitochondria in the case of animals. These are not contradictory concepts in evolution but rather different, coexisting evolutionary modes that must somehow be separated when performing evolutionary analysis. When considering such analyses, the

combination of these different gene flows creates a multitude of problems, introducing basic questions regarding whether or not phylogenetic classification is meaningful for prokaryotic organisms (3, 4).

My thesis addresses this last question in particular, regarding a meaningful classification for prokaryotes, and offers some nuanced answers.

1.2 HISTORY OF PHYLOGENETICS FOR PROKARYOTES

Prokaryotic phylogeny before molecular phylogenetics

We have been aware of the existence of microorganisms since the 17th century, when advances in microscope technology by Antonie van Leeuwenhoek allowed for the observation of organisms not visible to the naked eye. Even before the 17th century, there were arguments for the existence of transferable, microbe-like entities that caused disease, with the experiments that clearly established the germ theory of disease being performed in the 19th century by researchers such as Koch and Pasteur on *anthrax bacillus* and *cholera*, respectively. Around this same time and then subsequently, many advances were made in techniques to isolate microorganisms, obtain pure cultures and to stain cells, including Gram staining, thus gradually enabling the study of microorganisms. This was an extremely productive period which marked the beginnings of molecular genetics and established many critical concepts in modern science and medicine. However, one area in which researchers were unable to progress, despite some effort, was that of microbial classification. The study of microbial evolution was mostly the domain of botanists and microorganisms were loosely classified as plants.

Prokaryotes have virtually no fossil evidence and their limited morphological features are not evolutionarily relevant. An early subdivision, based on the physical appearance of cells, was that of eukaryotes and prokaryotes. Gram staining was also useful in identifying subdivisions within the group, and to this day, a major split in the domain of Bacteria is between the gram positive bacteria and gram negative bacteria. However, prokaryotes vs. eukaryotes, and gram positive vs. gram negative, are very broad categories. Some early attempts at classification relied on morphology, and to this scheme was eventually added biochemical and physiological differences. One problem with this type of system is that two different species of bacteria can still look identical. Another problem is that that many physiological traits are adaptations of different species to the same environmental conditions. For many years, up until the 1970s, a sensible classification of the prokaryotes that reflected their evolutionary history and relationships in detail proved impossible (5, 6).

Molecular phylogenetics

Molecular phylogenetics enabled the classification of prokaryotic organisms. In 1963, Zuckerkandl and Pauling began to work on the molecular anthropology of hemoglobin (7). In this same year, the paper by Margoliash on mutations in cytochrome c was also released (8). And in 1965, Zuckerkandl and Pauling first began to discuss the molecular clock (9). Then, in 1977, a multiple sequence alignment (MSA) of the small subunit (SSU) 16S rRNA gene revealed the existence of the three domains of life (1). This rendered the SSU rRNA the gold standard for phylogenetics (10-12), which persisted for several years.

16S rRNA is in many ways ideal for phylogenetic work; it is abundant in the cell, is present in all living organisms, and is highly conserved, with both slow-evolving and fast-evolving portions. Making its application to phylogeny even more straightforward are multiple databases, such as the Ribosomal Database Project (RDP) (13) and Silva (14), where millions of 16S rRNA sequences are available for download.

There are two major problems with phylogeny based on 16 rRNA. Firstly, for short distances—between members of the same species, for instance—there are not enough mutations to resolve relationships between organisms. Secondly, if the interest is not in pure cladistics but rather we wish to know evolutionary distances, for instance to define horizontal transfer versus phylogenetic noise, we must consider that ribosomal genes can be subject to positive selection from antibiotics. This is a serious concern because ribosomes are a main target of antibiotics. There are also additional concerns that are not specific to 16S rRNA but rather apply to any method of phylogenetic inference that depends on a single gene. As was already mentioned, most MSA-dependent approaches do not scale well. Horizontal gene transfer (HGT) is perhaps the most obvious problem; a single instance of HGT, which is known to be widespread in prokaryotes (15), can completely misplace an organism in any phylogenetic scheme based on a single gene. Although in general, “core” elements such as ribosomal proteins and housekeeping genes are not thought to be as mobile as genes involved in metabolism or antibiotic resistance, ultimately any gene can be horizontally transferred, including rRNA genes (16), and these instances of HGT have been known to skew phylogenetic analyses (17). There is also the issue of *which* 16S rRNA sequence to choose from each organism; they frequently have multiple, heterogeneous rRNA operons (16).

Problems with traditional methods: different genes tell different stories

Organismal phylogeny based on single genes became standard in a time when obtaining whole genomes was prohibitively difficult, but the evolutionary history of a species is different from the history of any one of its genes (although the history of the one would be expected to be often reflected in the other). As more sequences became available, additional genes were used as phylogenetic markers, including protein elongation factors EF-Tu and EF-2 (18-20), chaperones Hsp60 and Hsp70 (21, 22), the largest subunits of the RNA polymerase (23, 24), RecA (25), a variety of aminoacyl-tRNA synthetases (26) and others. Approaches using single genes originally generated a wealth of phylogenetic insight, but these trees were frequently incongruent with one another (27, 28). To improve the accuracy of phylogenetic methods, phylogeneticists began to concatenate multiple conserved genes to produce larger MSAs and therefore better resolved trees (28-32).

One problem with both trees built from single genes and even more so for phylogenies based on alignments of concatenations of highly conserved genes (33) is that they are not scalable. Another problem is that the size and functional diversity of these gene groups is largely dependent on the number and diversity of taxa (34). For instance, in the recent work of Lang and Eisen (28), an analysis of ~900 diverse prokaryotes from both bacteria and archaea identified only 24 suitable (i.e. paralog-free) genes. These consisted of a subset of ribosomal proteins, two translation factors that both interact with the ribosome, and the alpha subunit of a phenylalanyl-tRNA synthetase which was the only protein in the set not interacting with the ribosome and which contributed only ~5% of the overall alignment

used to generate phylogeny. A similar situation was seen by Ciccarelli et al., in which for a group of 191 organisms, the set of 31 genes used in the final alignment included 23 ribosomal proteins (35). Therefore, an additional challenge to studying evolution at the organism level is to not fall into the trap of analyzing how the ribosomal complex evolves.

Alignment-based methods require a high level of expertise, but when dealing with very large inputs, manual intervention is not possible. These methods are also subject to the challenge of horizontal transfer. Even when traditional methods work, because they focus on coding regions of the genome, they may (for instance, for some eukaryotes) be estimating the evolution of an organism by means of less than 1% of its genome content.

1.3 THE GENOMIC DATA FLOOD

Advances in sequencing technology

Learning how to obtain complete genomes was a critical step to understanding biology and was achieved as early as 1977 for the genome of bacteriophage X174 (36). Sequencing technology was refined over subsequent decades and the first bacterial genomes sequenced—*Haemophilus influenza* (37) and *Mycoplasma genitalium*—in 1995, the first single-celled eukaryote—*Saccharomyces cerevisiae*—in 1996, and the first multicellular eukaryote—*Caenorhabditis elegans*—in 1998. In the past few years, methods for obtaining full genome sequences have advanced tremendously (38-40), leading to a second critical transition, when the number of genome sequences became too large for traditional, alignment-based phylogenetics (41-44). Even during the time of Sanger sequencing, the number of bacterial genomes began to cross this threshold (45). With the development of next generation

sequencing technology, we are experiencing a flood of complete genomes and metagenomes (46).

Between September 2001 and January 2012, the cost per Mb of DNA sequence went down by five orders of magnitude (www.dnasequencing.org/history-of-dna). As of January 2008, the rate of decrease has out-paced Moore's Law (www.genome.gov/sequencingcosts/). The first sequencing of the human genome (The Human Genome Project) cost approximately $2.7 billion dollars and took ~13 years. As of November 2012, a human genome at 30x coverage costs $5,495, an amount that is exceedingly close to what many consider the ultimate goal in advancing sequencing technology—the $1,000 human genome. Compared to sequencing a full human genome, which consists of 6 billion base pairs, obtaining full genomes of bacteria, whose genomes are in the range of 1.3 Mbp to 13 Mbp (47), is practically trivial; today, sequencing a bacteria costs under $1,000 and takes under a day.

From highly curated data to Big Data

It requires a substantial effort to obtain a genome. This was particularly true in the past. When the number of such difficult-to-acquire items is small, they tend to be highly curated. As the number of such items grows, however, they begin to look like data, which essentially means a big pileup of data. Due to their diversity, these pileups can be much more informative. However, this informativity comes at the cost of the contributors being of much more variable quality. What then need to be developed for big data approaches are initial filters. This need to filter data did not exist in the time of early genome deposits. Another problem with such pileups is ascertainment bias (also called sampling bias). Some related

groups of organisms (e.g. model organisms such as *Escherichia coli*, pathogenic organisms, etc.) have many more representatives than others. For most analyses, this requires pruning of the data into categories with more uniform sampling. One such category is the species category, where currently we have a large number of species for which a large number of strain genomes has been deposited. Even naming schemes are not foolproof here; *Shigella* and *Escherichia coli*, which are in fact the same species when analyzed, are one example of why relying on metadata such as organism names can be problematic.

How many species are there?

While next-generation technologies still require refinement, in general—and in particular for the smaller genomes of prokaryotes and viruses—obtaining full genome sequences is no longer the main challenge. The new bottleneck is the data analysis, because traditional, alignment-based methods are not scalable. This problem is compounded by the fact that the total number of species is vast. According to the DSMZ (the German Collection of Microorganisms and Cell Cultures), as of January 2013 approximately 11,500 prokaryotic species have been defined (http://www.dsmz.de/bacterialdiversity/prokaryotic-nomenclature-up-to-date.html). The total number of species of bacteria and archaea is controversial and even a decisive definition of species is still lacking. However, ~8 million species of bacteria have been detected in a gram of soil (48) and 20,000 species in a liter of seawater (49). Considering that alignment-based methods can perform poorly even when only applied to a few hundred organisms, it is clear that a new class of methods, even if their results are less

accurate than those from alignment-based methods, is needed. It is also clear that we will not be "saved" from this reality by eventually running out of species.

1.4 ALIGNMENT-FREE WHOLE-GENOME METHODS

In contrast to the majority of traditional MSA approaches, which cannot handle large inputs and often require extensive curation to produce high quality alignments of orthologs, alignment-free methods are scalable and require minimum manual intervention (50-53). The idea of using complete genomes to perform phylogeny has a long history (54), but lay dormant until enough complete genomes became available. The rate at which these methods are now appearing reflects the pressing need for unsupervised, scalable methods. These methods are insensitive to many issues that are problematic to traditional approaches, including differences in protein lengths and differences in gene content. They also avoid some data quality issues, for instance they are somewhat robust to data incompleteness. Because they use complete genomes, they may also provide a more sound approximation for organismal phylogeny (55).

Types of methods

Alignment-free methods compute similarity or distance metrics using a variety of statistical properties belonging to k-mers (fixed-length substrings or subsequences, also sometimes called n-grams, n-mers, k-tuples, and k-words) in genomes. These methods can use word counts or can use match lengths. Matches can also be either exact or inexact. Some exact, word count methods are Composition Vector Trees (CVTrees) (56-58), Feature Frequency

Profiles (FFP) (59-61), and D2 statistics (62-64). Alternatively, some word count methods that employ inexact matches are Co-phylog (65) and Spaced Word Frequencies (SWF) (66). Some match length methods are Average Common Substring (ACS) (67), kmacs (68), Kr (69), ALFRED-G (70, 71), and Underlying Approach (UA) (72).

Each of these methods relies on different properties of sequence similarity between two organisms, with some approximating evolutionary distance better than others. All methods are applicable to sequences at both the nucleotide and amino acid levels and most have been tested on both alphabets, both for real and simulated data. It is not my goal to cover every alignment-free method ever investigated, especially considering that the number of such methods is growing rapidly, both in terms of new methods and also new flavors of older methods (e.g. ACS as kmacs' predecessor). However, I describe some available methods briefly below.

Survey of current alignment-free methods

Composition Vector Trees (CVTree)

CVTree is a word count method using fixed-length k-mers. For every organism in a given analysis, a separate composition vector is generated. Each of these composition vectors is of the size 4^k in the case of nucleotides and 20^k in the case of amino acids (*k* is the k-mer length) so that every possible nucleotide or amino acid sequence of that length has an entry in the vector. For example, the first 5 entries for k=3 using nucleotides would correspond to the sequences AAA, AAC, AAG, AAT, CAA. The value of *k* is the only parameter the user is responsible for. The lower bound on this value is 5 for amino acids; the authors observed

that if they took all substrings (overlapping) from all the proteins of a single organism, the original sequences could be almost completely reconstructed if the length of the substrings were greater to or equal to 5 (73). For every possible sequence of the designated k-mer length, the number of times it appears in a given proteome is counted and then the frequency or probability of that k-mer is calculated by dividing the count by the total number of k-mers present in the proteome. The composition vector consists of these probabilities for each sequence minus the probability of the sequence appearing by chance, divided by the probability of the sequence appearing by chance. The probability of the sequence appearing by chance is estimated by means of a Markov model, and corrects for random neutral mutations. The correlation is then calculated for any pair of organisms by taking the cosine of their two vectors in 4^k or 20^k space, and distances, originally in the range of [-1,1] but normalized to the range of (0,1), are the written to a distance matrix from which a tree is constructed using a neighbor joining routine.

CVTree was the first alignment-free method to offer a web-server (74), which has been recently updated (75). The method has been shown to work for archaea (57), bacteria at the strain level (76), viruses (77), fungi (78) bacteria (79) and chloroplasts (80).

Feature Frequency Profiles (FFP)

FFP is a word count method using fixed length k-mers. For every proteome in the input, FFP creates a vector of size 4^k or 20^k for nucleotides or amino acids, respectively, where *k* is the k-mer length. Each position in these vectors corresponds to a specific nucleotide or amino acid sequence, and the vector is updated with the number of times each sequence appears in

the genome or proteome. Because for longer values of *k*, the vectors can quickly begin exceed the bounds of computer memory, FFP offers an option to use a reduced alphabet for nucleotides: R for the purines (A and G) and Y for the pyrimidines (C and T). A probability distribution is then generated from the vector of unprocessed counts, in which each count is divided by the total number of k-mers in the genome or proteome. For any two pairs of the input, the divergence between their two probability distributions is calculated using Kullback-Leibler Divergence (81, 82). This value for the divergence is then used in calculating the Jensen-Shannon Divergence for the pair (83), which is the final distance. Several corrections are applied, including for high frequency features, low complexity features, and proteomes or genomes whose sizes differ by more than 4 times. Trees are built using neighbor joining.

FFP has been shown to work on mammalian intronic sequences, prokaryotes, unicellular eukaryotes (59), bacteria at the strain level (61), and the text from a diverse assortment of English language books (59).

D2 statistics

D2 is a word count method which uses fixed-length k-mers. It is, conceptually, one of the older methods for performing whole-genome phylogeny, first explored in the 1980s (62). For any alphabet A, for some k-mer length *k*, each word is counted in a genome or proteome *X* and is also separately counted in a genome or proteome *Y*. The counting of every k-mer is straightforward, such that a k-mer which appears only once is given a count of one and absent k-mers have by definition counts of zero, etc. The D2 statistic, at its simplest, is a

sum of the product of these counts from two genomes or proteomes over every word present in either sequence.

$$D_2 = \sum_{w \in A^k} X_w Y_w$$

This simple metric has been extended in various way, mainly for the purpose of normalizing the counts, e.g. to take into account compositional bias and the probability of seeing a specific k-mer in a genome or proteome, or using a Poissonian model for how many times a k-mer appears. The final distance used is then the logarithm of the ratio between conserved and non-conserved k-mers (62, 64), essentially a measure of sequence dissimilarity using the raw D2 scores (S in the following equation).

$$D_{XY} = \left| \ln \left(\frac{S_{XY}}{S_{XX} \times S_{YY}} \right) \right|$$

The recently released D2 software has been applied to simulated data for both amino acids and nucleotides. It was also applied to empirical data taken from TreeBASE (64).

Co-phylog

Co-phylog is a word count method which uses a fixed length seed which can then be longer alignments of variable length. The method is applicable to both amino acid sequences and nucleotide sequences, but with a current focus for the latter. One unique aspect is that it has been successfully applied not only to assembled genomes but also unassembled genomes. Co-phylog is as efficient as alignment-free methods, but may obtain accuracy close to or comparable to alignment-based methods by using micro-alignments. The method identifies seed alignments (exact or approximate matches) between the query and subject sequences

and then extends them into longer alignments (i.e. 'micro-alignments') by means of dynamic programming. A seed could be described as 11001, where 1 indicates *required match* and 0 indicates *don't care* (84, 85). Thus, for 11001, a seed match might be AACCT and AATTT, but not AACCG and ATCCG, due to the underlined mismatch at the second *required* match position (i.e. 11001). Co-phylog defines a structure S of any seed as $\mathbf{C}_{a1,a2,\ldots,an}\mathbf{O}_{b1,b2,\ldots,bn-1}$ where the *a* and *b* values correspond to the lengths of the 1 and 0 sequences of the seed, respectively. Thus, for the example 11001 above, the value of S would be $\mathbf{C}_{2,1}\mathbf{O}_{2}$. C-grams and O-grams are then defined as the concatenations of the **C** sequences and **O** sequences, respectively. For a seed of 11001 and a sequence of AACCT, therefore, the C-gram would be AAT and the O-gram would be CC. k-mers are defined as in other methods, as overlapping substrings taken from a genome, and of the same length as the seed. From this list of k-mers, a list of C-grams and corresponding O-grams is generated for any two genomes being compared. Co-phylog reduces this list to a set of contexts, where a context is any group of identical C-grams from a genome that have only one unique O-gram. A final list is generated, of size R, consisting of the intersection of contexts for two genomes (irrespective of whether the corresponding O-grams are identical or not). The value I in the equation below is 0 if the context's O-grams from the two genomes are not identical, and 1 otherwise.

$$D = \frac{\sum_{i=1}^{|R|} I_i}{|R|}$$

Co-phylog has been applied to *Escherichia* and *Shigella* genomes, Enterobacteriaceae and Gammaproteobacteria, simulated data, and tested on various next generation sequencing

(NGS) data sets. The method resolves distances between closely related organisms well. Co-phylog has an additional advantage in that it runs on raw next-generation sequencing data (65).

Spaced Words (SW)

In general, SW is similar to Co-phylog, using a mask consisting of positions that are either *required match,* represented by 1, or *don't care,* represented by 0 (or sometimes X in the literature). One possible mask, or pattern, might be X0X0XX, which for a nucleotide sequence CTGCCG would correspond to the *word* CGCG; the method demands that the pattern both begin and end with an X, and that the number of *required matches* be equal to *k.* The given pattern is sometimes referred to as a *spaced k-mer*. For any two genomes being compared, a given pattern such as the one above is used to calculate frequencies of all possible spaced k-mers. For instance, for some the pattern given above, CGCG may appear ten times in one genome and five in the other. Frequencies are calculated relative to the sequence length. Two vectors large enough to store all possible words (i.e. the alphabet size to the power of *k*) store all of these relative frequencies. The distance is then defined as the distance between these frequency vectors (86). An extension of this approach uses multiple patterns and then averages the distances (66, 87, 88). These methods have been applied to both nucleotides and also amino acids, and have both software and a web-server available. Implementations have been tested on simulated data, plant genomes, and primate mitochondrial genomes.

Average Common Substring (ACS)

The ACS method calculates its distance metric by means of variable length, exact matches between genomes or proteomes. These lengths are then averaged to obtain an intermediate, value which is subsequently normalized to account for differences in genome or proteome length. This normalized value is a similarity measure rather than a distance measure. Several simple operations are performed on this similarity measure to convert it to a distance measure. To ensure that the more distant organisms have the larger distances, the inverse of the value is taken. To ensure that a genome's distance to itself is zero, a correction term is subtracted. Finally, to ensure that the distances are symmetric (i.e. distance from organism A to organism B is the same as that from organism B to A), the average of the two non-symmetric distances is computed (67). This method has been applied to archaea, bacteria, and eukaryotes as a single set, yielding a tree of life exhibiting the same 3-domain split produced by alignment-based methods. ACS performed well on viruses and mitochondrial genomes from 34 mammals. However, the method has largely been superseded by related methods, two of which are described below, which allow for mismatches.

Kmacs

Kmacs is a generalization of the ACS method that allows for *k* mismatches. The method does not actually calculate the exact number of such strings for every position due to the computational complexity of the problem, but rather approximates the value by means of a greedy heuristic. This reduces the complexity from $O(kn^2)$ to linear time, where n is the length of the sequences being considered. This is achieved first by calculating the longest

common substring, as described above, for each position in one genome to a query genome. For each of these longest common substrings, kmacs continues to match the two sequences until the end of the sequences is reached or else k+1 mismatches are reached. These operations are implemented efficiently by means of enhanced suffix arrays (68, 89). In the original kmacs publication (68), kmacs was tested on primate mitochondrial genomes, *Roseobacter* genomes, simulated DNA sequences, protein sets from BAliBASE, and on simulated protein sequences. Kmacs is available as source code as well as a web-server (66).

ALFRED-G

ALFRED-G is another generalization of the ACS method. Like kmacs, it estimates the ACS distance metric, allowing for a bounded number of mismatches, and employs a greedy algorithm to do so. ALFRED-G extends a method developed previously for efficiently calculating ACS distances allowing for a single mismatch (71). From the case of a single mismatch, they then apply the same method as in kmacs (68). Their approach relies on generalized suffix trees, which allow for the rapid calculation of the longest common prefix from any two sequences and also fast calculation of a longest common prefix which allows for k mismatches. The source code for this method is available and the method was tested on a small number of Primate mitochondrial genomes, a small number of Roseobacter genomes, and a set of protein sequences from BAliBASE. ALFRED-G is notable in that there was exact concordance between the mitochondrial tree it produced and the reference tree (alignment-based) (70). However, I found that ALFRED-G, at least in its current implementation, was also the slowest method of any method I tested.

The Kr method

Like kmacs and ALFRED-G, the Kr method is another method that is closely related to the ACS method. Kr takes two unaligned DNA sequences and estimates the number (Kr) of substitutions per site between the two sequences. The purpose of the method is to provide not only evolutionary distances or evolutionary trees that are in close accordance with the accepted taxonomy (as other alignment-free methods do), but to specifically provide an alignment-free method that uses a distance measure that is directly related to evolutionary events, e.g. is linear with evolutionary time. To calculate the value Kr for any pair of sequences (typically genomes), for every suffix in the one sequence, the shortest prefix absent from the other sequence is determined. These shortest prefixes are known as shustrings and have been described elsewhere (90, 91). The method establishes a probability density function that makes it possible to estimate the expected length of a shustring as a function of the number of substitutions between the two sequences. The derivation for the final distance is available in the original paper (69), but includes a final Jukes Cantor correction to obtain a proper evolutionary distance. The Kr value is asymmetric depending on which sequence is the query and which the subject; the method uses the smaller final value to correct for this. In the original paper, the method was tested on primate mitochondrial genomes, complete genomes from *Streptococcus agalactiae* strains, and complete genomes from *Drosophila* species. The method was also tested on simulated data.

1.5 WHY IS PHYLOGENETICS IMPORTANT?

Evolutionary adaptation by genetic change is the essence of any broader understanding of biology. But this requires starting from a reconstruction framework of evolutionary history and then mapping the observed phenotypes on it. Phylogenetic reconstruction both at the organismal or the gene level is the starting point for such reasoning, so it is pervasive either explicitly or implicitly in a very large number of discussions of the subject. However, it is not obvious how to perform such reconstruction and to what extent the uncertainty of it affects subsequent analyses. The problems are not only at the level of mathematically or technically devising reconstruction strategies, but they also arise from the assumptions of these procedures. So developing robust and scalable phylogenetic approaches is an important objective.

Understanding how life evolves

From the time of Darwin, the ability to reconstruct the evolutionary history across all the domains of life has been pursued. There was no possibility of such a reconstruction until we acquired the ability obtain genomic sequences. Without phylogenetics, we cannot say anything about the temporal evolution of life. Apart from any practical motivations, society considers how life evolved to be an important question to understand.

Phylogenetics underlies taxonomy

Another aspect of phylogenetics is that it underlies the communication language when describing a group of organisms. There is strong preference for taxonomy to follow

evolutionary divergence, and it is also preferable that new organisms, as they are discovered, can be classified accurately enough that they will not require multiple rounds of reclassification and renaming. Especially for prokaryotes, there is still lacking a clear definition of species that is clearly distinct for instance from strain, and in general there is an arbitrariness in how organisms are currently grouped, with some phyla for instance having gigantic diversity (e.g. the Proteobacteria) and often not being monophyletic.

Despite this, there will be occasional necessary exceptions, either due to uncertainties, preserving the previous terms (e.g. “fishes” with exclusion to land vertebrates), or to use phenotypic characteristics which may be considered more important than some aspects of phylogeny.

Microbiome characterization

As the genotypes encountered in microbiomes are mostly uncharacterized and belong to new species, they can only be described in terms of belonging to broader taxonomic categories, so phylogenetics is essential to informatively classify microbiomes. Often this information is used to define medical properties of microbiomes, for example potential sensitivity or resistance to antibiotics. k-mer methods, such as the one I developed here, are ideal for such analysis, because they can perform metagenome classification using short fragments, and they can do so efficiently, even for a very large input, as is often the case with the sequencing data typically produced for metagenomic studies (92).

1.6 DISSERTATION OVERVIEW

To describe what is common or different in biology, it is necessary to put broader labels on groups of organisms. In practical terms, for instance with infectious organisms, classification is necessary so as to better know the enemy. But any type of statement discussing the properties of life or of a domain of life in general terms must be able to assess how representative that property is if it is not a universal character. Currently, we do not have methods that will be free of ascertainment bias. Therefore, the work I present here represents real progress toward a more objective way of defining the universality of biological properties. In this is time when data coverage has much improved and problems have evolved from acquisition bias to analysis bias, this lack of a standard requires more attention. This is the area in which I contributed.

SlopeTree overview

I developed SlopeTree, a new alignment-free method which measures evolutionary distance by quantifying how quickly the number of matching sequences between two proteomes decays as a function of sequence length. The goal of SlopeTree was to develop a method for phylogenetic reconstruction that could take arbitrarily large inputs and produce trustworthy evolutionary distances. For these distances to be trustworthy, the method had to be robust to the many challenges one encounters in evolutionary analysis. While several current alignment-free methods include some corrections for background and composition, SlopeTree is the first to consider them all: the uneven composition of amino acids, the

possibility of backwards mutations, a background of coincidental matches over short k-mer lengths, and the issue of horizontal gene transfer.

By subtracting a background of short length, coincidental matches and restricting itself to a range of longer lengths (~7 or more amino acids), SlopeTree is able to follow the evolution of the highly conserved segments of proteins, using approximately 10,000 to 40,000 amino acids per genome pair. The highly conserved regions that SlopeTree targets correspond to the alignable regions in a multiple sequence alignment.

Horizontal gene transfer and alignment-free methods

Horizontal gene transfer is highly relevant for alignment-free methods because it adds a spurious contribution of similarity between genomes (15, 93, 94). There are multiple possible signatures of horizontally transferred proteins, for instance unusual codon usage (95-97). We identified a novel signature based on analysis of multiple copies of almost identical protein sequences in a genome, and those multiple copies almost invariably belonged to one of two categories: one category was of EF-Tu translation factor, which is frequently present in multiple copies; and the second was of mobile elements, as inferred from a very narrow or scattered phylogenetic footprint, even within a single species. When annotated, these mobile elements consisted primarily of parasitic proteins resulting from phage infections. Another level of filtering is by means of a dual evolutionary stability index indicating conservation and lack of stability or paralogy score, with large instability value representing very likely cases of HGT. A mobile element (ME) filter and a separate, conservation filter were built into SlopeTree using these two general concepts. Finally, a third HGT correction is based on

the curvature of the slope. Therefore, SlopeTree is unique in that it is not only robust to HGT, but it explicitly identifies and corrects for HGT at multiple stages of the analysis.

The SlopeTree package

The SlopeTree package includes both the main SlopeTree algorithm, which estimates evolutionary distance by quantifying how quickly the number of matching sequences between two proteomes decays as a function of sequence length, and several independent modules for filtering mobile elements and less-conserved proteins out of the data and recalculating distances for pairs still exhibiting significant HGT even after the earlier filtering steps. Altogether, the method consists of the following four modules: (1) a Mobile Element Filter, (2) a Conservation and Stability Filter, (3) the SlopeTree Main Algorithm and (4) a Pair-Wise Horizontal Gene Transfer (HGT) Correction. A flowchart is provided in Figure 1-1. In this dissertation, I present these modules not in the order in which they are typically applied, but rather in the order in which I developed them, with Chapter 2 focusing on the SlopeTree Main Algorithm, Chapter 3 focusing on the Mobile Element Filter and Conservation and Stability Filter, and Chapter 4 focusing on the Pair-wise Horizontal Gene Transfer Correction.

The Mobile Element Filter exploits a novel signature which is based on analysis of multiple copies of almost identical protein sequences in a genome. These highly repetitive proteins proved almost always to be mobile elements. The Conservation and Stability Filter calculates for each protein a value, which we call a *paralogy* score, from the ratio of the sum of how many genes each of the protein's k-mers has a match with in other genomes to the

sum of the total number of genomes the protein's k-mers have matches with. This ratio effectively separated orthologous proteins evolving by descent, which typically have a gene to genome ratio of one and therefore had paralogy scores of approximately one. Mobile elements on the other hand, have paralogy scores frequently much greater than one because their presence, absence, and copy number are much more unstable, while unconserved proteins which simply have no k-mer matches with any other proteins in the input have scores of 0.

The SlopeTree Main Algorithm estimates a distance for every pair of organisms from the decay in the number of exact sequence matches as a function of match length.

The Pair-Wise HGT Correction assesses the slopes produced by the SlopeTree Main Algorithm and identifies pairs of organisms that appear to have shared significant horizontal transfers; it runs the SlopeTree Main Algorithm on these pairs combined with a reference set to identify proteins that the pair shares but that are absent from the reference, and then it re-runs the SlopeTree Main Algorithm on just the pair, with the flagged proteins removed.

The four modules are not necessarily run together; for instance, the SlopeTree Main Algorithm can be run on unfiltered data or data passed through only one of the filters.

Assessing SlopeTree

I applied SlopeTree to bacteria, archaea, and sets of strains. For these sets, I built 'raw' trees using no data-cleaning or corrections and trees using different degrees of correction and different combinations of the corrections. I compared SlopeTree to simple 16S rRNA trees, and then to trees based on phylogenetically broad concatenated alignments from the literature

(28), in which supermatrices were constructed from 24 single-copy, ubiquitous genes and then passed to a Maximum Likelihood (ML) routine for tree-building. These comparisons were performed to assess the accuracy of the method and to identify potential biological sources for differences.

The SlopeTree strain-level trees were remarkably stable for different inputs. Even when only mobile elements together with proteins not part of the core were considered, the tree topology was highly similar. The archaeal trees were more fluid upon restricting the method to the most conserved proteins, but the majority of clades and relationships between deep branches remained the same. The deep, short branches in the bacterial trees were the most unstable, which is related to a generic problem of defining phylogenetic relationships in evolutionary radiation. For archaea and bacteria, I calculated topological and branch-length distances to the trees built from supermatrices for trees built by SlopeTree, ACS, CVTree, D2 and kmacs. By applying the mobile-element filter and conservation filter to the data prior to running the main SlopeTree routines, I was able to significantly reduce the distances to the trees built from supermatrices, not only for SlopeTree but for all other alignment-free methods.

I also observed approximately 20 bacteria whose placement on the phylogenetic trees frequently disagreed between alignment-free methods and the current NCBI classification. The consistency of these alternative placements for these bacteria when applying alignment-free methods suggests that their classification may require revision, or at the very least have complex histories. This is further supported by the fact that several of these bacteria had

similar disagreements between the trees built from supermatrices and the NCBI classification.

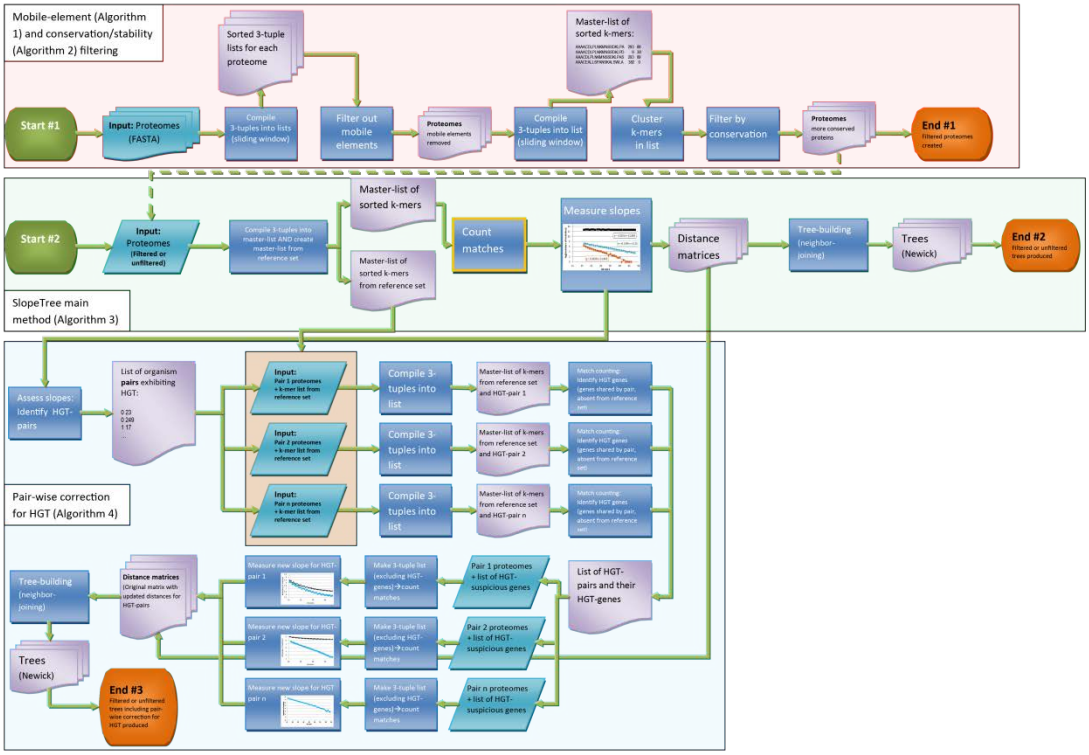


Figure 1-1. Phylogeny reconstruction flowchart for SlopeTree.

SlopeTree has 3 main parts: The mobile-element filtering (Algorithm 1) and the conservation/stability filtering (Algorithm 2); the SlopeTree main method (Algorithm 3) which produces a distance matrix and tree; and the pair-wise HGT correction (Algorithm 4) which reprocesses pairs that were flagged as showing signs of HGT. When not using mobile-element filtering or conservation filtering, Start #2 is the original starting point. Three pairs are shown for the pair-wise HGT correction code; this number can be in the 100s or 1000s depending on the input set. All proteomes are in FASTA format.

CHAPTER TWO

DEVELOPING A NEW ALIGNMENT-FREE METHOD FOR PHYLOGENY

2.1 MOTIVATION

The goal of this work was to develop a phylogenetic method whose metric was related to the evolutionary rate of some subset of conserved sites in a genome or proteome. Ideally, this metric would have uniform branch lengths. For example, branch lengths would not be skewed for fast-evolving (long branch lengths) or slow-evolving (short branch lengths) organisms. The method was also intended to have minimal sensitivity to the various analysis problems that exist for prokaryotic evolution, such as horizontal gene transfer (HGT), sequence bias, low complexity sequences, and missing data.

The decay of initially identical sequences represents the accumulation of mutations. Radioactive decay is analogous to this. A more precise analogy is to a generalization of the Jukes-Cantor method, with some number of states depending on the input (20 states if all amino acids are equally likely). SlopeTree is as a method that quantifies this decay between proteomes and from this decay estimates evolutionary distance estimate.

This chapter focuses on the early design of SlopeTree and although the main algorithm remains the same, many important aspects of the *current* SlopeTree package are not mentioned here.

2.2 SLOPETREE OVERVIEW

SlopeTree hinges on the idea that genomes between highly similar organisms are expected to share a large number of sequences, while genomes between distant organisms are expected to share only the sequences from the most highly conserved genes that are the most critical for life. For closely related organisms, both the number of identical sequences and also their maximal length is expected to be high. **SlopeTree estimates evolutionary distance as a function of the rate at which the number of exact, unique matches falls off as a function of match length.**

The main SlopeTree algorithm

This algorithm corresponds to Algorithm 3 in Figure 1-1.

Input: A set H of *n* proteomes $\langle H_1, H_2, \ldots, H_n \rangle$.

Output: A distance matrix D of SlopeTree evolutionary distances between all pairs in H, such that D_{ij} is the SlopeTree distance between proteomes H_i and H_j.

Algorithm: Let p_{ij} be the j^{th} protein in H_i, and let $p_k^{ij}[h]$ be a k-mer from p_{ij} of length k, starting at index h, where $0 \leq h < f$ given that p_{ij} has length f. For those k-mers at the end of each protein where $h+k>f$, the suffix is expanded by the necessary number of empty characters to fill the remainder of the k-mer. Each k-mer is stored as a 3-tuple consisting of

the k-mer, the proteome ID (*i*), and the gene ID (*j*). Let L be the alphabetically sorted list of all 3-tuples.

Let m_r^{xy} be an exact sequence match of length *r,* where $1 \leq r \leq k$ for proteomes P_x and P_y, where each match is counted exactly once. Let M_r^{xy} be count of all m_r^{xy}, where the same sequence is only counted once. For all *r* in the evolutionarily relevant range, ~r>8 amino acids, we define D_{xy} as an estimate of the evolutionary distance between proteomes P_x and P_y, where D_{xy} is the decay in the histogram of $\ln(M_r^{xy})$ as a function of *r*.

Computational Complexity: For *n* organisms and *m* amino acids, let $m=m_1+m_2+\ldots+m_n$. The compilation of L is done in $O(m)$, and the sort within all organisms is equal to $\Sigma O(m_i \log m_i)$ which is equal to $O(m \log m)$. The match-counting algorithm then requires $O(m \log m + n^2)$ time. Thus, the time complexity is $O(m \log m + n^2)$, with $m >> n$. We treat the alphabet size as a constant here.

2.3 IMPLEMENTATION

Here I discuss some important details regarding the implementation of this main module of SlopeTree, including how the method addresses uneven composition of amino acids and the background of coincidental matches over short k-mer lengths.

Assigning unique ordinals to proteomes and proteins

The first operation of SlopeTree is to detect all organisms in the input (a source directory containing FASTA files is provided by the user), alphabetically sort them by name, and assign them a unique integer, which we refer to as a genome ID, starting from 0.

Assembling the k-mer lists

SlopeTree generates a list of all k-mers (default=20-mers) from all proteomes in the input set by means of a sliding window. Those k-mers shorter than 20 (i.e. k-mers from the end of each protein) are buffered a '^', signifying 'no character', and k-mers containing non-standard amino acids (e.g. U) are ignored. In the same way that each proteome is given an ID (described above), each protein is given an integer ID which is unique within (but not between) proteomes. Each k-mer then is associated with a proteome ID and a protein ID as a 3-tuple, and these 3-tuples are sorted alphabetically into a final list (Figure 2-1). To facilitate various operations embedded in the SlopeTree code, and to facilitate development, SlopeTree uses its own procedures for k-mer counting and sorting and relies heavily on the Standard Template Library (STL).

The gene IDs are not required for the early version of SlopeTree that will be discussed in this chapter, but they are included in the description because they have become essential to many of SlopeTree's newer routines and including them in the discussion from the start may avoid some confusion later. They are also referred to in the formal description of the main SlopeTree algorithm above, and so could not be left out. However, I consider it important to note that the original SlopeTree implementation, which built the trees shown in this chapter, was originally quite simple. Essentially all it had to do was generate a sorted list

of k-mers and genome IDs from all inputs to calculate distances that were already quite close to an approximation for the accumulation of mutations among homologous sites.

Removing low complexity sequences

Originally, all length-20 substrings, assuming they consisted only of the 20 standard amino acids, were included in the k-mer list and subsequent analysis. However, when assessing longer length matches between proteomes, I frequently observed repetitive sequences, for instance k-mers consisting entirely or almost entirely of A or S. These low-complexity sequences could in some cases contribute long-length matches between relatively distant organisms for which such long length matches would not be expected. And this in turn could slightly skew the distance and make the pair appear closer than they were.

k-mers with significantly reduced amino acid alphabets (i.e. low complexity sequences) are not included in the sorted list. For each k-mer, SlopeTree counts the total number of times each amino acid is present (c_n). The low-complexity score (S) of the k-mer is calculated as the sum of the squares for these counts.

$$S = \sum_{n=1}^{20} c_n^2 \quad (1)$$

The k-mers with scores above a given cutoff (C) are discarded. Originally, I hard-coded this cutoff to 130; I had manually inspected various low-complexity k-mers and their typical range of low complexity scores. However, to allow for different values of k, C is now calculated by SlopeTree as $6.5k$. Figure 2-2 presents a list of k-mers of k=20 that are excluded for having scores about the cutoff just described.

Counting unique matches

Two algorithms were used for the purpose of counting unique sequence matches of lengths 1 to *k* between all pairs in an input set. Both algorithms used a 3-dimensional array, of dimensions *n* x *n* x *m*, where *n* was the total number of organisms in the input and *m* was the maximum possible match-length (or, as described below, the maximum value for a function of the match-length). I refer to this 3-dimensional array as the correlation matrix (Figure 2-3). The correlation matrix essentially stores histograms of the total number of unique sequence matches between every pair in the input, over every possible k-mer length up to the max k-mer length, which by default is set to 20 in the current SlopeTree code. At the beginning of the match-counting process, the correlation matrix is initialized to 0. Then, for each unique k-mer match of score *s* between any pair of organisms *p* and *q*, array *A* has the entry at $A[p][q][s]$ incremented.

Match-counting algorithm 1

The original match-counting algorithm compared every adjacent pair of k-mers in the final list, over every position. For this method, two additional arrays were held in memory and were used for keeping track of the matches: one array, which I refer to as the match vector, was a vector that was the same length as the maximum k-mer length (e.g. 20 by default); the other array, which I refer to as the genome vector, was a 2-dimensional array of dimensions maximum k-mer length x number of organisms. All arrays were initially set to zero. The algorithm then compared all adjacent k-mers, from the top of the list to the bottom. If there was a full-length sequence match, then every position of the match vector was incremented,

and for every length (1 through *k)*, the positions in the genome array corresponding to the two organism IDs were also incremented. This was in fact unnecessary; merely setting them to 1 would have been adequate, because I was only counting unique matches, but originally I stored the actual counts in case they would be useful later. For a partial-length match, first the information corresponding to the previous matches over the mismatched area was retrieved from the genome array; then the appropriate positions of the genome array and match vector were set to zero, after which the new matches over all lengths were recorded. A snapshot of this algorithm for a reduced alphabet and 3-mers is provided in Figure 2-4. This algorithm was adequate for smaller sets and could easily have been parallelized to run faster. It was however only used for the early proof-of-concept work on SlopeTree, generating the first plots, distances, and phylogenetic trees, and was applied to larger and larger input sets, up to a size of 140 bacteria. At this size, the algorithm proved so slow that a new implementation became necessary.

Match-counting algorithm 2

The algorithm described above was replaced by a recursive algorithm that is still used in the current SlopeTree implementation. This match-counting routine recursively partitions the sorted list of 3-tuples into blocks having the same leading amino acid, with three base cases for the recursion: the end of the k-mers has been reached, with the match reaching the last character in the block; the current block consists of only one k-mer, meaning that the current k-mer has no matches; and the end of the k-mer list has been reached. The pseudocode for

this algorithm is provided in Figure 2-5. An example run, over a reduced alphabet of 3 characters and for 5-mers, is shown in Figure 2-6.

Scoring matches

The match-scoring scheme for SlopeTree went through several iterations to reach its current version. Originally, I used match length as the score, with a match of a single amino acid giving a score of 1 and a match of 20 giving a score of 20. I extended this scoring scheme to nit scores, which increased by a factor of 2-3 the range of possible scores; 20-mers, which originally were limited to scores from 1 to 20, had a range of 1 to ~60. The maximum possible nit score is calculated in the code as the product of the k-mer length (default=20) and the largest nit score for any given single amino acid, which is typically ~3.

After assessing the frequencies of amino acids between all of my proteomes, I concluded that there was not enough variation between the frequencies to justify using a different set of nit-scores for every proteome; therefore, I calculated a single set of nit-scores by counting the number of instances of amino acids over *all* proteomes and dividing it by the total number of amino acids in this set. The motivation for using a single set of nit-scores was to keep SlopeTree fast and simple. For example, when using a single set of nit-scores, the nit-score for a match present in all proteomes would only have to be calculated once before updating the correlation matrix. On the other hand, if using each proteomes specific amino acid frequencies, the score for the match would have to be recalculated for every pair separately. *This was the scoring scheme used in SlopeTree v1 and was used to build all trees shown in this chapter.*

This reasoning, unfortunately, somewhat underestimated the variation in amino acid frequencies, which is frequently discussed in the literature (98). Therefore, in the current version of SlopeTree, for every sequence match, SlopeTree currently uses an average of the amino acid frequencies from either organism sharing the sequence to calculate its score (Figure 2-7). I explain the details of the current nit-scoring scheme here for clarity, so that all details regarding nit-scores may be found in one place. The two methods for calculating nit-scores (averaging over the entire input, as I did in this early version of Slopetree, vs. averaging between each pair separately, as is done now) are equivalent given an input of two organisms.

Before running the match-counting algorithm, for each proteome, the number of instances of each amino acid (c_a) and the total number of amino acids (T) are counted, and amino acid frequencies (f_a) of each proteome are then calculated:

$$f_a = \frac{c_a}{T} \tag{2}$$

For each proteome, for each amino acid, a nit score (s_a) is then calculated:

$$s_a = -\ln(f_a) \tag{3}$$

This can be rewritten to take into account that each proteome will have its own set of nit scores. Therefore, for a specific organism, e.g. organism p, this could be written:

$$s_{p,a} = -\ln\left(f_{p,a}\right) \tag{4}$$

For a match of length l between two organisms, p and q, where m[i] is the amino acid at match position i, the score m_{pq} for the match, m, would be:

$$m_{pq} = \sum_{i=1}^{l} \frac{1}{2}\left(s_{p,m[i]} + s_{q,m[i]}\right) \tag{5}$$

There were two motivations for using nit scores. One was to improve the rejection of coincidental matches. Coincidences of more frequent amino acids were more likely, so relying on a nit score provided better rejection, with stretches of frequent amino acids having to be longer to contribute to the evolutionarily relevant range of the data. The second consideration was to obtain a more fine-grained sampling than number of amino acids, which for 20-mers would have defined just 20 bins. However, the slope expressed in nits also had a composition-dependent relationship to the slope expressed in mutations. Because the target was a slope expressed in units of mutation, there was a need for a correction factor that was composition dependent. I discuss this correction factor in the subsequent chapter.

The SlopeTree match-count histogram

The SlopeTree algorithm produces a histogram for every pair of organisms. Each histogram consists of the number of unique k-mer matches shared by the pair, for a range of nit scores (rounded to integers) from 0 to the maximum possible nit score for the chosen k-mer length (t_i). These histograms, when plotted in natural log, generally have the same shape (Figure 2-8). They consist of a spike in the low nit score range corresponding to short-length, coincidental matches, although this spike can be absent for plots between extremely similar (e.g. same species) organisms; a linear dependence in the middle of the nit score range corresponding to the decay of evolutionarily conserved sequences; and a final drop to zero corresponding to the cap imposed on the matches by the k-mer length, assuming that the pair are evolutionarily close enough to have matches longer than the maximum k-mer length.

Background subtraction

Figure 2-8 shows an example of the exact-sequence-match histograms SlopeTree calculates for each pair. The range of the histogram is for all possible nit-scores (rounded to integers), which goes from 0 to the maximum possible nit-score for the chosen k-mer length (t_i). The spike on the left, apparent in Figure 2-8B, corresponds to the background of coincidental matches, e.g. sequences expected to match by chance due to short length of frequent amino acids.

In Figure 2-9, a SlopeTree plot for real data is shown alongside a SlopeTree plot built from randomized data. The process of randomizing or scrambling the genomic sequences removes the evolutionarily conserved sequences that contribute to the main slope of the SlopeTree plots. Originally, this scrambling was performed so as to better understand the significance of the main slope in the SlopeTree plots—that it corresponded to the decay of evolutionarily conserved sequences. However, it quickly became apparent that passing randomized sequences through SlopeTree generates a background that can then be subtracted, isolating the main signal. This background subtraction identifies the gray zone where evolutionarily significant sequences and coincidental matches coexist in the plots, and potentially makes it possible to clean some of the coincidences out of the real data of this gray zone.

I tested numerous routines for generating the best random proteomes. For every proteome, a new proteome was generated, with the same number of proteins and each protein being the same length as the template protein, but all protein sequences being randomly generated. In the beginning, these proteins were built by randomly sampling amino acids,

with the probability of getting a particular amino acid being consistent with its frequency in the original proteome. For the early version of SlopeTree discussed in this chapter, this was the routine used to generate the background.

In the current SlopeTree implementation, the sequences are generated at the same time that the main set of k-mers is extracted, prior to the final k-mer sort; these scrambled sequences come from the original proteins, which have randomly selected fragments of the original protein (fragments are of length 1-4, also chosen randomly for each fragment) reordered prior to applying the sliding window. I chose this final method so that the background I calculated would mirror as closely as possible the amino acid frequencies of real proteins, not just frequencies for single amino acids but also more complex compositional patterns.

I attempted several such schemes using different possible ranges of fragment lengths (e.g. fragment length 1-5 or 2-4, etc.) in an attempt to correct for a small problem in generating a background. The number of k-mers for the real data and random data was identical. However, because many sequences in the real data are evolutionarily conserved and contribute to actual matches, the number of coincidental background sequences in the *real* data is slightly smaller than the same number in the randomized data. This made for a slightly higher spike in the random background than in the real data. However, some relatively rough tests using different parameters and different scrambling schemes gave no real improvement, and the problem caused such a minor effect in the actual data analysis that ultimately I did not pursue it.

The sorted, merged k-mer list derived from the scrambled proteins is also passed through the SlopeTree match-counting algorithm (Algorithm 3), generating its own set of histograms in which the evolutionarily conserved sequences have been completely erased (blue plots in Figure 2-9). SlopeTree's background correction consists of subtracting the counts from the histograms obtained from randomized sequences from the histograms obtained from real data. Eventually, an additional constraint was implemented in the current version of SlopeTree is that for the nit scores in which the counts for the scrambled data are more than 25% the counts for the real data, the real data values are set to 0, and the left bound set to the nit score with the maximum count.

Identifying left and right bounds for the SlopeTree data

One of the most consistently troublesome tasks I had to implement as an automatic feature within SlopeTree was the selection of left and right bounds for each plot, prior to measuring the slope. SlopeTree uses the area of the histogram corresponding to the decay of evolutionarily conserved sequences. This requires that for each plot, the lower and upper bounds of this area be automatically selected. This task appeared straightforward, given that for all data, the coincidental matches disappeared at approximately the same nit-score (~30), and the right bound should have been easily identifiable as the first nit-score to have zero counts for a pair. However, repeatedly, the rules I selected for the bounds selection were unacceptable for some small set of pairs in the input, and the larger the input, the more problems I encountered in defining these rules.

Originally, the left bound was hard-coded at nit-score=30; this was reduced to 25 when the method was applied to larger datasets where and I observed how sparse matches could be between very distant organisms. Ultimately, this bound could not be hard-coded and had to be calculated individually for each plot. I eventually used the random background (described above) to provide a left bound for each pair, such that the left bound was defined by the value at which the random background went down to zero matches.

Defining rules for selecting a good right bound was equally challenging. As matches become sparser for the higher nit-scores (i.e. fewer long-length sequence matches), the data become noisier. I tried several approaches that made use of this, measuring the Chi2 over different ranges in an attempt to find the most reliable stretch of data. In the last scheme I used before abandoning this approach entirely (Figure 2-10), I measured the Chi2 for all sets of 6 data points starting at nit-score=24 and going up until the first value of 0. If Chi2-1 was greater than 0.01 for the range of points starting at nit-score 24, then the final range of points was from 24 to either 52, assuming that there were hits at that nit-score, or else the last nit-score with a non-zero value. The reasoning behind this scheme was that the data was noisy for high nit-scores but relatively smooth for the lower nit-scores; if even for the low nit-scores, it was noisy, then it could only be expected to be noisier as the number of matches became more sparse, and the safest approach would be to use all the points in the range. On the other hand, if Chi2-1 was less than 0.01, then all Chi2 values for all groups of 6 points were assessed, until the first with a Chi2-1 greater than 0.01. The final range was then nit score 24 to the final point in that final range, or to 52 if the range went beyond 52. The cutoff of 52 was implemented in this approach on the assumption that 20-mers were being

used, and because it was in the vicinity of nit-score 52 that the cap on k-mer length began to cause the number of matches to drop. That is, even if a pair had an exact match of 45, this was not reflected in the plots; for a match of 45, 25 20-mers would be observed, with values in the range of 40-60 depending on their composition.

Estimating evolutionary distances

The steepness of the slope, over the evolutionarily relevant range, is an approximation for evolutionary distance. Steeper slopes indicate greater distances, while shallower slopes indicate smaller distances. Plots for strains of the same species have slopes that are virtually zero (Figure 2-11). This simple observation is the foundation of the SlopeTree distance metric.

Fitting the data

In the original design of SlopeTree, and also during the early development of the program, it appeared that a simple linear fit would be more than adequate to measure the slope for the range of points identified by the primitive bounds-selection process I originally implemented and described above. These ranges of points, except in rare cases, almost always fell on a straight line and were very easy to fit to a simple linear equation. This was especially true for very similar organisms, but generally true for all pairs, assuming they came from the same domain of life. When the decay did not appear to be quite straight, it still appeared *relatively* straight, and I generally ascribed it to noise in the data as the data became sparser for higher nit-scores.

Originally, I used Numerical Recipes to calculate the fits. Eventually, I replaced this with my own least squares regression routine, coded as a module in the SlopeTree package. This fit still exists in the implementation, but has been replaced by another fit as described in the next chapter.

A linear fit:

$$y = ax + b$$

$$d = -a$$

Least squares regression for a linear equation

$$\sum_{i=0}^{n-1} (ax + b - y)^2$$

When multiplied out and simplified, this is equal to

$$(ax + b - y)^2 = a^2x^2 + 2abx - 2axy + b^2 - 2by + y^2$$

I then plugged this back into the summation and split the sum:

$$a^2 \sum_{i=0}^{n-1} x_i^2 + 2ab \sum_{i=0}^{n-1} x_i - 2\mathrm{a} \sum_{\mathrm{i}=0}^{\mathrm{n}-1} \mathrm{x_i\, y_i} + \mathrm{b^2 xn} - 2\mathrm{b} \sum_{\mathrm{i}=0}^{\mathrm{n}-1} \mathrm{y_i} + \sum_{\mathrm{i}=0}^{\mathrm{n}-1} \mathrm{y_i^2}$$

Notation:

$$S_{jk} = \sum_{i=0}^{n-1} x_i^j y_i^k$$

Using the new notation:

$$\sum_{i=0}^{n-1} (ax + b - y)^2 = a^2 S_{20} + 2abS_{10} - 2aS_{11} + b^2 S_{00} - 2bS_{01} + S_{02}$$

Derivatives in terms of a and b:

a: $2aS_{20} + 2bS_{10} - 2S_{11} = 0$

b: $2aS_{10} + 2bS_{00} - 2S_{01} = 0$

This is a system of linear equations:

$$[S_{20} \quad S_{10}][a] = [2\ S_{11}]$$

$$[S_{10} \quad S_{00}][b] = [2\ S_{01}]$$

Solve for a and b:

$$a = \begin{bmatrix} S_{11} & S_{10} \\ S_{01} & S_{00} \end{bmatrix} \Big/ \begin{bmatrix} S_{20} & S_{10} \\ S_{01} & S_{00} \end{bmatrix} = (S_{11}S_{00} - S_{10}S_{01})/(S_{40}S_{00} - S_{10}S_{10})$$

$$b = \begin{bmatrix} S_{20} & S_{11} \\ S_{10} & S_{01} \end{bmatrix} \Big/ \begin{bmatrix} S_{20} & S_{10} \\ S_{01} & S_{00} \end{bmatrix} = (S_{20}S_{01} - S_{10}S_{11})/(S_{40}S_{00} - S_{10}S_{10})$$

The data for each plot separately was used to calculate the various values of $S_{ij,}$ which were then plugged into the equations for *a* and *b*.

Constructing the distance matrices

Slopes were always either negative or in the case of extremely similar organisms, approximately 0. This is because if there is a match of length 20, then matches within that match are also counted—all unique matches of length 1,2,…,19. For this reason, the counts in SlopeTree must always decrease monotonically, with the exception of occasional binning artifacts. I reversed the sign for all slopes, making distances positive with larger values corresponding to larger distances. Distance matrices were passed to RapidNJ, a neighbor-joining program (99), to build the final trees.

2.3 RESULTS FOR SLOPETREE V1

The first version of SlopeTree consisted of the barebones algorithm described above—an unweighted linear fit of a range of points (selected using an assessment of Chi2 values for different ranges) of each plot, generated from all k-mers. This first version, in its most ambitious application, was run on a set of 2001 bacteria (Figures 2-12 and 2-13) and a set of 137 archaea (Figures 2-14 and 2-15). These were all the bacteria and archaea available in the NCBI database at that time. Beyond dividing the domains, no subset selection process took place; SlopeTree was intended to run unsupervised for large inputs in which all types of data issues might exist, and so this was how I tested the robustness of this early version of SlopeTree. For these sets, 16S rRNA trees were also built for comparison, using the RDP database (11). When multiple 16S rRNA genes were available for an organism in my set, I chose the one with higher quality, or the one that was longer. When I finally downloaded the sequences and built the tree, I used corrected Jukes-Cantor distances offered by the RDP.

137 Archaea

Even the early version of SlopeTree was highly successful when it came to resolving relationships between archaea. SlopeTree's classification of the 137 archaea was consistent with the NCBI taxonomy and also with the rRNA 16S tree generated. In the majority of the branching order and in many of the finer details, the three trees agreed (Figure 2-15). All phyla were distinct from one another, with no organisms misplaced at the phylum or class levels, and in nearly all cases, SlopeTree separated the archaea correctly down to the species level. For instance, the *Sulfolobaceae* family was correctly divided into the three genera

present: *Acidianus, Metallosphaera*, and *Sulfolobus,* and the 9 strains of *Sulfolobus islandicus* and two strains of *Solfolobus solfataricus* were sister groups.

Consistent between all three trees, but most notable in SlopeTree and the rRNA tree, was the proximity of *Thaumarchaeota* and *Korarchaeota*.

The only case of a clade being split up in SlopeTree's topology was a split in the order *Desulfurococcales,* and within that, the *Desulfurococcaceae* family: in one clade were the *Ignisphaera* and *Aeropyrum,* both members of the *Desulfurococcaceae,* and in the other the *Staphylothermus, Thermogladius, Thermosphaera,* and *Desulfurococcus*.

The classification *Methanothermococcus okinawensis* also differed with the NCBI classification. *M. okinawensis* is currently classified within the genus *Methanothermococcus* but SlopeTree positioned it within the *Methanococcus* clade. SlopeTree did not distinguish between the *Methanothermococcus* and *Methanococcus* genera, although it cleanly separated all other genera present. In the SlopeTree topology, *M. okinawensis IH1* is closest to *Methanococcus aeolicus Nankai-3;* their closeness was previously been observed (100).

One final difference between the SlopeTree archaeal tree and the NCBI taxonomy was the placement of the *Thermococci* class, which is currently classified as being within the phylum *Euryarchaeota,* whereas in our tree, it is monophyletic with the single representative of the *Nanoarchaeota*. The closeness of *Thermococcales* and *Nanoarchaeaota* is in the literature (101), based on phylogenies calculated from a subset of ribosomal proteins and also using unrelated molecular markers. However, members of the *Nanoarchaeota* have significantly reduced genomes, with the one representative in our set consisting of only ~150,000 amino acids. Whole-proteome methods have difficulty with such extremely

reduced genomes because they rely almost by definition on data-richness for their statistics and reduced genomes may lack enough sequence information for proper phylogenetic inference.

Slopetree v1, like the 16S rRNA tree, put the unclassified *halophilic archaeon DL31* deep within the Halobacteria class.

Bacteria

The 2001 bacteria were reduced to 1718 bacteria by removing organisms with reduced genomes (cutoff=450,000 amino acids), organisms of the category *Candidatus,* and organisms with branch lengths very different from those of their immediate neighbors. This final rule was my attempt to compensate for a problem I did not yet understand in the SlopeTree topology; I thought immediate neighbors with very different branch lengths might be caused by poor data quality. I observed several of such neighbors in the tree, with often one of them being a highly misplaced organism. I discarded this rule when I better understood the cause for these misplacements (horizontal gene transfer, discussed in subsequent chapters).

For the full tree of 2001 bacteria, forty-two bacterial genomes did not agree with NCBI at the phylum level. Most these discrepancies were easily explained. Seven of these forty-two were bacteria with reduced genomes, most notably various strains of *Carsonella ruddii*. The strains of *Carsonella ruddii* present in our input had approximately 50,000 amino acids per proteome. This is three times smaller than *Nanoarchaeota equitans* which may already have been too small for SlopeTree. *Carsonella ruddii*, which some argue has

nearly attained “organelle status” (102) may not belong in a dataset of bacteria to begin with. Twelve of these forty-two were in the *Candidatus* category. Two of these forty-two had branch lengths that were significantly longer than those of their immediate neighbors, which at the time I believed indicated an issue with the data quality. Three additional bacteria with mismatching branch lengths were also removed, despite being correctly classified: *Paenibacillus polymyxa M1, Acholeplasma laidlawii PG-8A,* and *Orientia tsutsugamushi Ikeda.* These three were correctly classified but nevertheless removed as a precaution. SlopeTree correctly classified another four bacteria (*Aster yellow witches broom, Onion yellows phytoplasma, Buchnera aphidicola str JF98,* and *Mycoplasma penetrans HF-2)* that had incongruent branch lengths, but these were already eliminated due to having reduced genomes. Ten were negligibly misclassified. These included four bacteria currently classified within the Bacteroidetes/Chlorobi group, which were closest to the Chlorobi. All four Deferribacteres present in our set were grouped together within the Deltaproteobacteria. The closeness of Deferribacteres and Deltaproteobacteria has been observed previously in the literature (103). The two Nitrospirae were grouped together and also proximal to the Deltaproteobacteria and close to the single representative of the *Chrysiogenetes* phylum. This may also be an acceptable classification; studies of the magnetotactic properties of members of *Nitrospirae* and *Deltaproteobacteria* indicate that these two phyla may be very close (104). Eight were classified correctly in the reduced tree of 1,718, indicating that problematic organisms present in the larger tree were skewing their placement.

Three were classified in contradiction to the current taxonomy, but with some indication that SlopeTree’s classification of them may have been the correct one.

Coprothermobacter proteolyticus DSM 5265 is classified as a Firmicute, while SlopeTree v1 placed it adjacent to the phylum Caldiserica. These two phyla were closest to the Dictyoglomus and then to the Thermotogae. It has been suggested in the literature before that *C. proteolyticus* has been misclassified as a Firmicute and that it is actually closely related to the Dictyoglomus (28, 105). *Thermodesulfobium narugense DSM 14796* and *Thermodesulfovibrio yellowstonii DSM 11347* are classified as a Firmicute and Nitrospirae, respectively; SlopeTree put them both closer to the Thermodesulfobacteria, which from the naming scheme alone appears plausible.

Seven bacteria were classified incorrectly at the phylum level by SlopeTree for no reason that I could see at the time of assessing these trees. I thought that their misclassification could be due to data-quality issues of the specific proteomes or other proteomes in the input set that skewed their placement (Table 2-1).

Petrotoga mobilis and *Dehalogenimonas lykanthroporepellens,* whose misplacement is discussed at length in later chapters, are not in this table because they were eliminated due to having incongruent branch lengths with their direct neighbors.

In order to explore the details of the branching order for this large set of bacteria, I created consensus trees, using SlopeTree and the 16S rRNA sequences with corrected Jukes-Cantor distances from the RDP. The input sets for the two trees differed slightly; the whole set of 2001 was the input for our consensus tree, but only 1894 were used for the rRNA tree due to not all bacteria in our set having 16S rRNA sequences in the RDP.

The two consensus trees agreed in many of their details, with several major clades in common. The Bacteroidetes, Chlorobi, and Ignavibacteria formed clades in both trees. This

grouping is extensively supported in the literature (106, 107). The Verrucomicrobia, Chlamydiae, and Planctomycetes also formed clades in both trees. This clade is frequently referred to as the 'PVC superphylum' and is supported both at the sequence level (28, 108) and also by similarity in cell compartmentalization between Planctomycetes and Verrucomicrobia (109). The Chloroflexi together with the single representative of the Thermobaculum phylum, *Thermobaculum terrenum ATCC BAA 798,* formed a clade in both trees. This is supported by gene order comparisons (103) as well as standard 16S rRNA analysis (110). Both trees suggested that the Chrysiogenetes and Deferribacteres are closely related, in agreement with the All Species Living Tree which puts them as sister groups (111, 112). Both trees also had the Deltaproteobacteria proximal to this clade, which is also supported (113). In the SlopeTree consensus tree, the Nitrospirae were close this group and also to the remaining Proteobacteria; gene order comparisons indicate that Deferribacteres, the Proteobacteria, and the Nitrospirae are proximal (103). In contrast, the rRNA tree placed the Nitrospirae in a completely different group. While both trees put the Thermodesulfobacteria and Aquificae as sister groups, there is little support for this in the literature. The phylogeny of the Thermodesulfobacteria is not clear.

In addition to the major clades of both trees being in agreement with the accepted taxonomy, the majority of the phyla from both consensus trees are monophyletic in terms of their families. On the other hand, in both trees, the Firmicutes and also the Proteobacteria are polyphyletic, with the Epsilonproteobacteria in particular at a distance from the other Proteobacteria.

There were several significant differences between the two trees. The Actinobacteria are monophyletic in the SlopeTree consensus tree, whereas in the rRNA tree they were polyphyletic. As already mentioned, the placement of the Nitrospirae differed between the trees, with SlopeTree's placement more consistent with previous observations. In both trees, the Synergistetes, Deinococcus-Thermus, Thermodesulfobacteria, Aquificae, Thermotogae and Dictyoglomi were placed close to one another; however, the Caldiserica were included in this group by SlopeTree, whereas the 16S rRNA tree puts the Caldiserica in a clade with the Epsilonproteobacteria. Finally, SlopeTree put the Fusobacteria closest to the Epsilonproteobacteria, wheras the 16S rRNA tree put them within the Firmicutes.

Comparison to other methods and distance to the 16S rRNA trees

I compared the results from this version of SlopeTree to trees also generated by CVTree, ACS, and FFP by calculating distances to the 16S rRNA reference trees I built from the RDP. SlopeTree produced trees closer to the 16S rRNA tree for both archaea and bacteria, with one exception where it was outperformed by the ACS method for bacteria (Table 2-2). I was forced to use reduced trees (only 1480 organisms for bacteria when I had built a tree of ~2000) because some of the other methods were unable to run beyond ~1500 organisms.

Despite SlopeTree's being competitive with other methods released at that time, I observed significant phylogenetic scattering in the bacterial SlopeTree tree. This was actually true for all methods. This scattering was in addition to the specifically misplaced organisms discussed above. Several clades, in particular the Proteobacteria, were not monophyletic and were in fact spread out across the entire tree in groups. It was not clear to

me at this time how much of this scattering was due to issues with SlopeTree and how much due to problems with the current accepted taxonomy. This is still a difficult question to address.

2.4 DISCUSSION AND CONCLUSIONS

Much of the work presented here was intended for proof of concept—could this novel distance metric produce high quality evolutionary distances. I found that it could, at least insofar as outperforming several other, similar alignment-free methods. The trees I built also demonstrated that the SlopeTree measure or distance possesses the very important property of having relatively uniform branch lengths from the root. This is in contrast to methods such as Average Common Substring or CVTree, where these lengths are severely linear with evolutionary time.

Different measures of evolution, for instance different alignment-free methods, will produce different trees. Generally, these measures are correlated, generating highly concordant trees. Each alignment-free method defines similarity between organisms in its own units, but it still needs to be established how each of these measures can be transformed into units of accumulation-of-mutations and with what level of accuracy. SlopeTree was designed to provide a measure with a close relationship to the accumulation of mutations. In the absence of selection, this relationship would be given by a simple formula, but at larger evolutionary distances, the slope is defined by slowly evolving protein segments subject to strong negative selection. At the domain level, the relationship becomes nonlinear and requires calibration between the slope and the number of accumulated mutations. At very

large distances, such as those between domains, the slope loses its relationship to evolutionary distance entirely. However, this is only significant for rooting archaeal and bacterial phylogenies.

The uniformity of the branch lengths from the "root" to the tips in the SlopeTree trees is not an artifact of the distance measure being nonlinear or saturating at some value. It may be a consequence of looking at a large number of conserved sites and if a particular locus evolved faster for a particular genome pair, its contribution becomes much smaller. Heterotachy, which is variable between positions in an alignment, has very different consequences in terms of branch length estimation for alignment-based methods and current alignment-free methods. Considering that there is much larger variability in branch lengths by alignment-based methods, it appears that more uniform branch lengths are a consequence of two factors: averaging between more proteins and potentially smaller sensitivity to heterotachy which is variable between positions in an alignment.

2.5 MATERIALS AND METHODS

Downloading proteomes, selecting input sets, and reference trees

The archive all.faa.tar.gz downloaded from the NCBI ftp website (ftp://ftp.ncbi.nih.gov/genomes/Bacteria/) contains proteomes in FASTA format of both archaea and bacteria. Because SlopeTree does not resolve organisms well at inter-domain distances, the archaea and bacteria in the archive were first identified and separated. Originally, this was done manually. Eventually, I wrote a script using the information contained in the NCBI-downloaded taxdump file (ftp://ftp.ncbi.nlm.nih.gov/pub/taxonomy/)

to identify each organism's domain. These archives were downloaded many times during the course of the SlopeTree development, with the final downloads taking place in May 2015. In the NCBI taxonomy, the root nodes for bacteria and archaea are 2 and 2157, respectively. Of the 2774 organisms in the FASTA archive, 165 were identified as archaea, 2607 as bacteria, and 2 as neither archaea nor bacteria (multiisoloate_uid216090 and multispecies_uid212977). However, for this early version of SlopeTree, I used a much earlier version of this archive; at this time, there were approximately 2000 bacterial proteomes and approximately 130 archaeal proteomes available.

The reference tree used to assess the results from SlopeTree was rudimentary, built from an alignment of 16S rRNA genes taken from the Ribosomal Database Project (11, 114-116). This tree (not included here) was not considered to be acceptable by at least one reviewer when the first SlopeTree manuscript was sent out.

Neighbor Joining

For all distance matrices produced by SlopeTree and the other methods discussed in this work, for both SlopeTree v1 and the current version, I used rapidNJ rapidNJ version 2.0.1 (99) to construct the trees.

```
./rapidnj distance_matrix.txt > distance_matrix_tree.txt
```

I briefly diverged from this tool and tried the tree-building tools of PHYLIP (fitch and kitsch) (117). These however had significantly longer run-times and did not appear to produce better trees, at least in terms of producing a smaller distance to the reference trees.

Pruning trees

This version of SlopeTree pruned from the input: 1) organisms with fewer than 140,000 amino acids; 2) organisms with *Candidatus* in their names; and 3) organisms having highly incongruent branch lengths with their immediate neighbor.

Building SlopeTree Trees and other trees for comparison

The code for SlopeTree v1 no longer exists, but the current implementation can approximate the results by using the linear fit which is currently commented out of the code. For the other methods, the commands I used over the years have not changed; examples of these can be found at the end of Chapter 3.

Tree visualization

The figures for all of the trees in this manuscript were generated using the ITOL web-server (118, 119).

Figure 2-1. Final k-mer list.

A) Example of sliding window (shown in red). Sequences at end of protein buffered with ^ to fulfill the length 20 requirement. B) Example subsection of final master list of sorted 20-mers, genome IDs, and gene IDs.

Rejected 20-mer	Score
YAQAEAAQADAQAGQAGAQA	132
AQAEAAQADAQAGQAGAQAG	136
EEKNSSKKKEKKEEKNKEEE	136
KEKAKNKKQEKKEVKKEPKK	142
KAKNKKQEKKEVKKEPKKDK	136
KNKKQEKKEVKKEPKKDKEK	142
KKQEKKEVKKEPKKDKEKDK	144
KQEKKEVKKEPKKDKEKDKE	132
QEKKEVKKEPKKDKEKDKEK	132
EKKEVKKEPKKDKEKDKEKD	136
KKEVKKEPKKDKEKDKEKDD	134
EEIPEKEIEEEEMAEKVEDE	134
SNGGGMPAGMPGGMGGMGGM	132
NGGGMPAGMPGGMGGMGGMG	152
GGGMPAGMPGGMGGMGGMGG	174
GGMPAGMPGGMGGMGGMGGM	162
GMPAGMPGGMGGMGGMGGMG	162
MPAGMPGGMGGMGGMGGMGG	162
PAGMPGGMGGMGGMGGMGGM	162
AGMPGGMGGMGGMGGMGGMM	172
GMPGGMGGMGGMGGMGGMM	171
MPGGMGGMGGMGGMGGMM	150
PGGMGGMGGMGGMGGMM	137
GGMGGMGGMGGMGGMM	136
GVSAAAPVAVAGGAAGAGAA	136
VSAAAPVAVAGGAAGAGAAA	148
SAAAPVAVAGGAAGAGAAAE	144
AAAPVAVAGGAAGAGAAAEE	146
INKVNKIIIIISIITILIAI	134
NKVNKIIIIISIITILIAII	134
KVNKIIIIISIITILIAIIL	134
VNKIIIIISIITILIAIILS	134
NKIIIIISIITILIAIILSI	156
KIIIIISIITILIAIILSIL	160
IIIIISIITILIAIILSILI	184
IIIISIITILIAIILSILIS	164
IIISIITILIAIILSILISN	142
KANNFAAAPEAAAAAGKAFA	136
ANNFAAAPEAAAAAGKAFAT	134
SQPQTQPQTQPQTQPQTQSQ	136
QPQTQPQTQPQTQPQTQSQT	142
NKNKNSNNNQGQYQNNNQQN	132
KNKNSNNNQGQYQNNNQQNN	132

Figure 2-2. Rejecting low-complexity sequences.

All 20-mers shown above were rejected from *Halanaerobium praevalens DSM 2228* due to their scoring above the low-complexity cutoff.

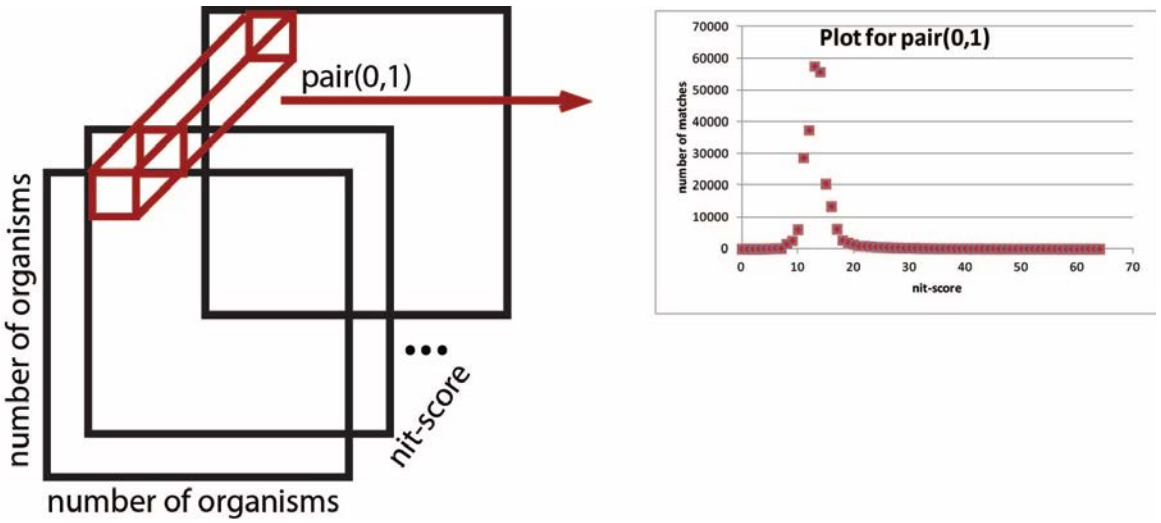


Figure 2-3. The correlation matrix.

Both implementations of SlopeTree used a 3-dimensional matrix to simultaneously count the number of unique sequence matches between all pairs of organisms in the input. Shown above, the depth, or layers, of the matrix correspond to the length of the match (nit-score is a function of match length, described later). The 'wedge' on the left that is shown in red corresponds to the number of unique matches for a pair, in this case the organism pair with IDs 0 and 1, over the entire range of lengths (1 to maximum k-mer length, default=20) or nit-scores, which is plotted on the right.

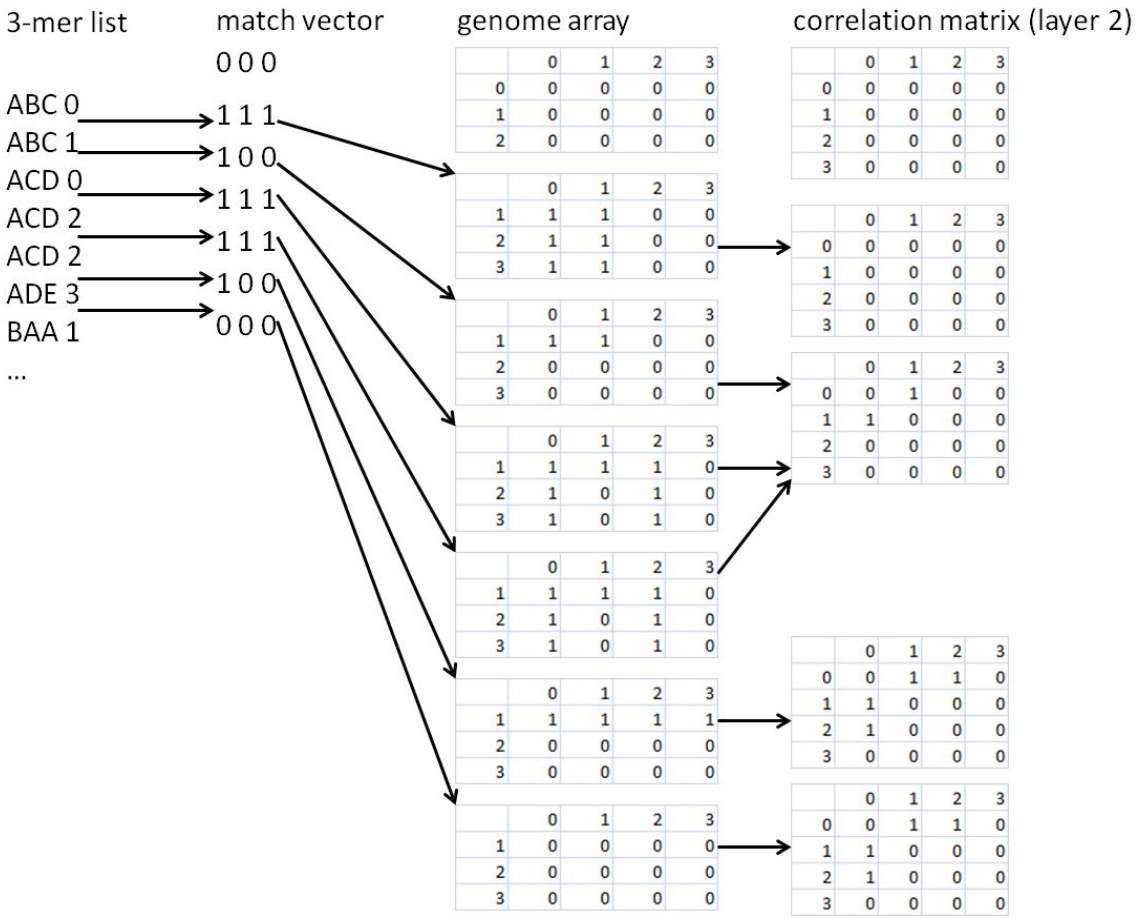


Figure 2-4. SlopeTree match-counting algorithm 1.

Displayed above is an representation of the original SlopeTree algorithm which compared adjacent k-mers in the list to count all sequence matches between pairs. For simplicity, this example was made using a reduced alphabet and 3-mers. The starting arrays are initialized to 0 and updated as the list is scanned *from top to bottom.*

```
find_matches(integer Column, integer first_row, integer last_row, double score, string sequence)
{
     if(first_row==last_row) //if we only have one sequence
          //do nothing
     else if(column==tag_length) //if we are past the last column
          //do nothing
     else
     {
          if entire block between first_row and last_row has the same leading character in column Column
          {
               char a = (leading amino acid of the block);
               if(a is valid amino acid) //a is not buffering character ^
               {
                    find_matches(Column+1,first_row,last_row,score+bitscore(a),sequence+a);
                    toggle_bits(first_row,last_row,score+bitscore(a),sequence+a);
               }
          }
          else //block between first_row and last_row have different amino acids in column Column
          {
               integer start = first_row;
               integer end = first_row;
               while(start<last_row)
               {
                    char a = (amino acid in Column for sequence at start);
                    while( (end<=last_row) && (amino acid in column for sequence at end==a) )
                    {
                         end+=1; //increment end;
                    }
                    if(end==start+1)
                         //do nothing; no matches at this depth
                    else if(a is valid amino acid)
                    {
                         find_matches(Column+1,start,end-1,score+bitscore(a),sequence+a);
                         toggle_bits(start,end-1,score+bitscore(a),sequence+a);
                    }
                    start = end;
               }
          }
     }
}
```

Figure 2-5. SlopeTree match-counting algorithm 2 (pseudocode).

Recursive SlopeTree function partitions blocks of matching amino acid sequences starting from the first column (left) and then moving right across the k-mer list.

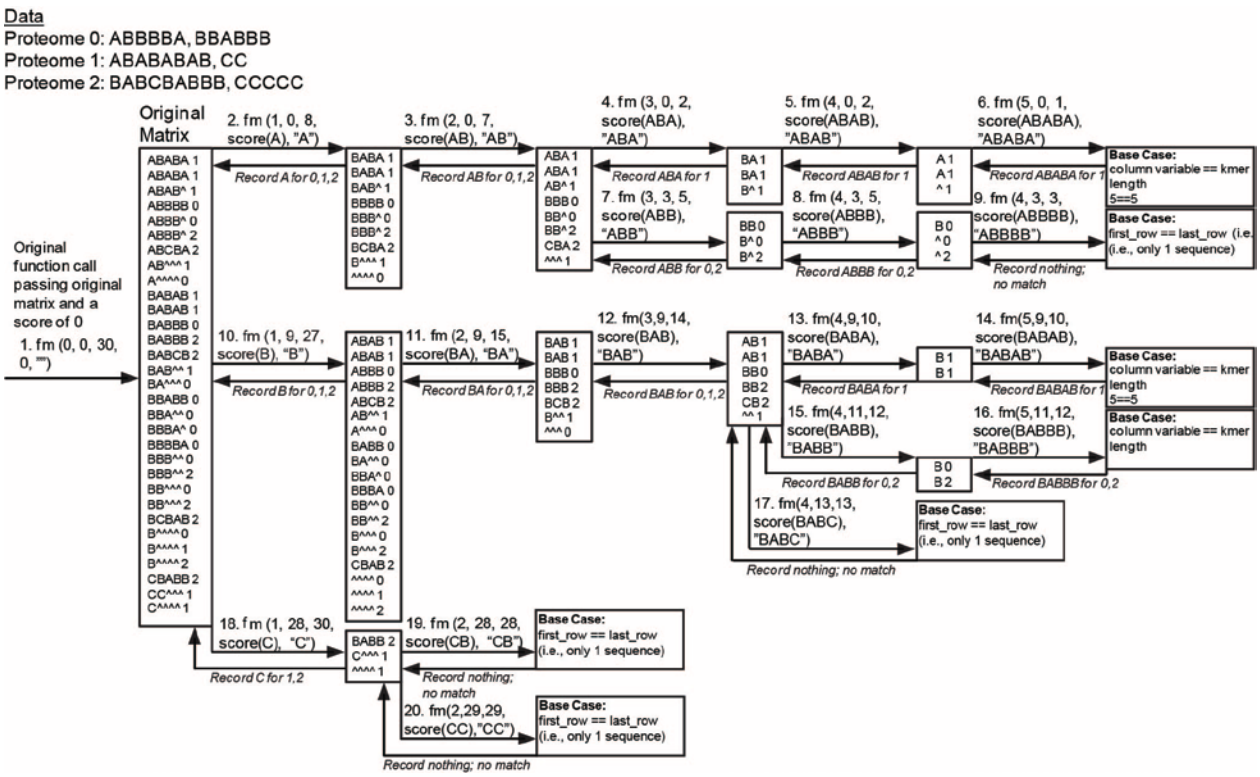


Figure 2-6. SlopeTree match-counting algorithm 2 (visual example).

5-mers and a reduced alphabet of 3 characters were used in this example. For the sake of clarity, this example does not include gene IDs. The matrix is partitioned into smaller and smaller matrices, with each partition consisting of adjacent sequences having the same amino acid; this lead amino acid is stripped off and the function called on the remaining sequences.

organism p

A	88091	0.1379	1.9815
C	3931	0.0062	5.091
D	34744	0.0544	2.9118
E	39115	0.0612	2.7933
F	17377	0.0272	3.6047
G	56887	0.089	2.4188
H	14109	0.0221	3.813
I	26203	0.041	3.194
K	8229	0.0129	4.3522
L	69397	0.1086	2.22
M	10003	0.0157	4.157
N	10097	0.0158	4.1476
P	34984	0.0548	2.905
Q	16329	0.0256	3.6669
R	55728	0.0872	2.4394
S	35198	0.0551	2.8989
T	34484	0.054	2.9194
V	63332	0.0991	2.3115
W	9145	0.0143	4.2466
Y	11584	0.0181	4.0102

organism q

A	134064	0.1129	2.181
C	10742	0.009	4.7052
D	56418	0.0475	3.0466
E	66360	0.0559	2.8843
F	44790	0.0377	3.2774
G	93780	0.079	2.5384
H	30788	0.0259	3.6522
I	57618	0.0485	3.0255
K	40301	0.0339	3.383
L	120815	0.1018	2.2851
M	28134	0.0237	3.7424
N	38228	0.0322	3.4358
P	64528	0.0544	2.9123
Q	50415	0.0425	3.1591
R	77765	0.0655	2.7257
S	72373	0.061	2.7975
T	66695	0.0562	2.8792
V	83279	0.0701	2.6572
W	16710	0.0141	4.2634
Y	33408	0.0281	3.5706

Nit scores for example matches between p and q:

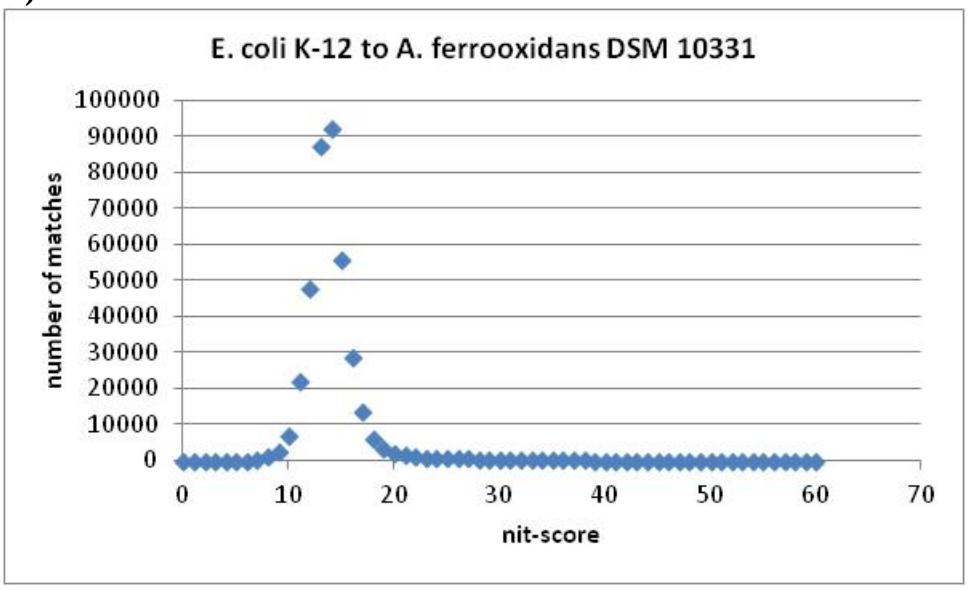
nit-score(ACA) =
(1.9815+2.181)/2+(5.091+4.7052)/2+(1.9815+2.181)/2 =
9.0606 ~= 9

nit-score(WYSH) =
(4.2466+4.2634)/2+(4.0102+3.5706)/2+(2.8989+2.7975)/2+(3.813+3.6522)/2 =
14.6262 ~= 15

Figure 2-7. Calculating nit-scores for a sequence match between two organisms.

For two organisms, p and q, an example of their amino acid total counts and frequencies, and nit-scores calculated for two sequence matches between p and q: ACA, and WYSH. Nit-score values are decimal numbers that are rounded to the nearest integer. Although WYSH is only longer than ACA by one amino acid, it has a much higher nit-score due to its amino acids.

A)



B)

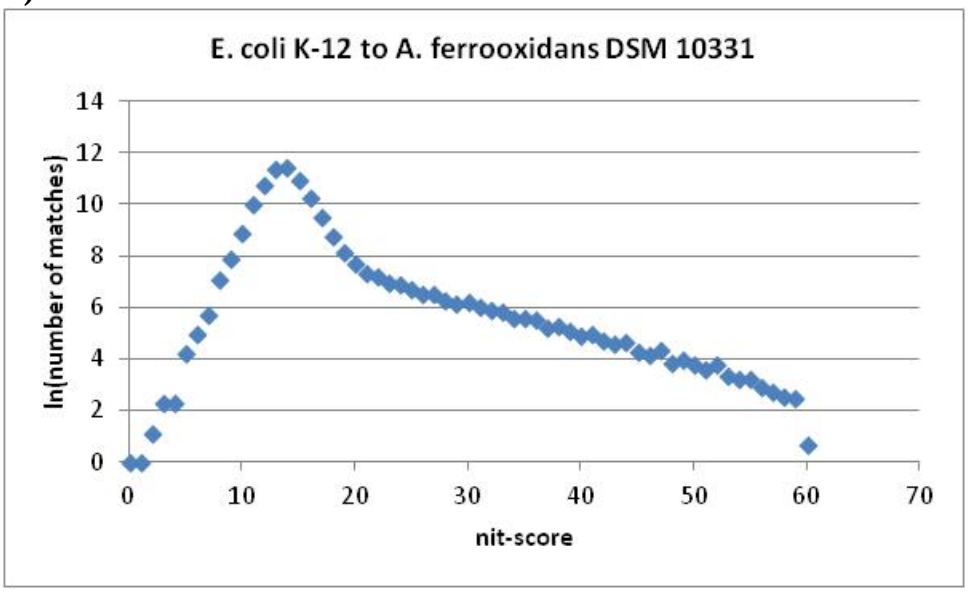


Figure 2-8. SlopeTree plot.

Plot for *Escherichia coli K-12* and *Acidimicrobium ferrooxidans DSM 10331*. A) The number of unique sequence matches between the pair, from length 1 to 20, scored using nit-scores rather than length. B) The same plot as in (A), in natural log.

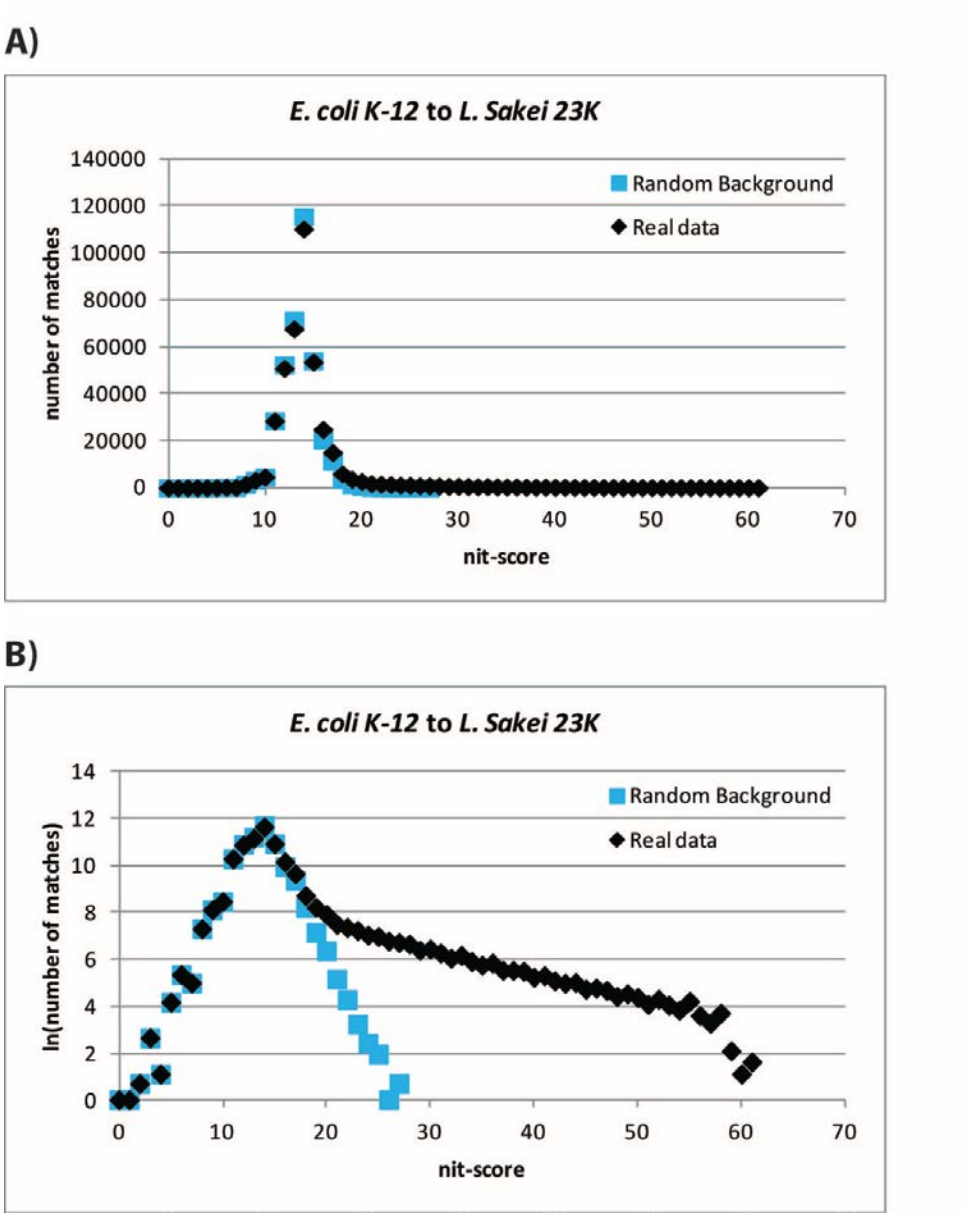


Figure 2-9. Subtracting the background.

Randomly selected fragments of the amino acid sequences of the above pair were scrambled and then passed through SlopeTree to generate plots in which the evolutionary signal had been eliminated. A) The SlopeTree plots for the randomly generated sequences (blue) and real protein sequences (black). B) The same as (A), in natural log; the plots diverge because, due to their evolutionary relatedness, the organisms share sequences more sequences than would be expected by pure chance.

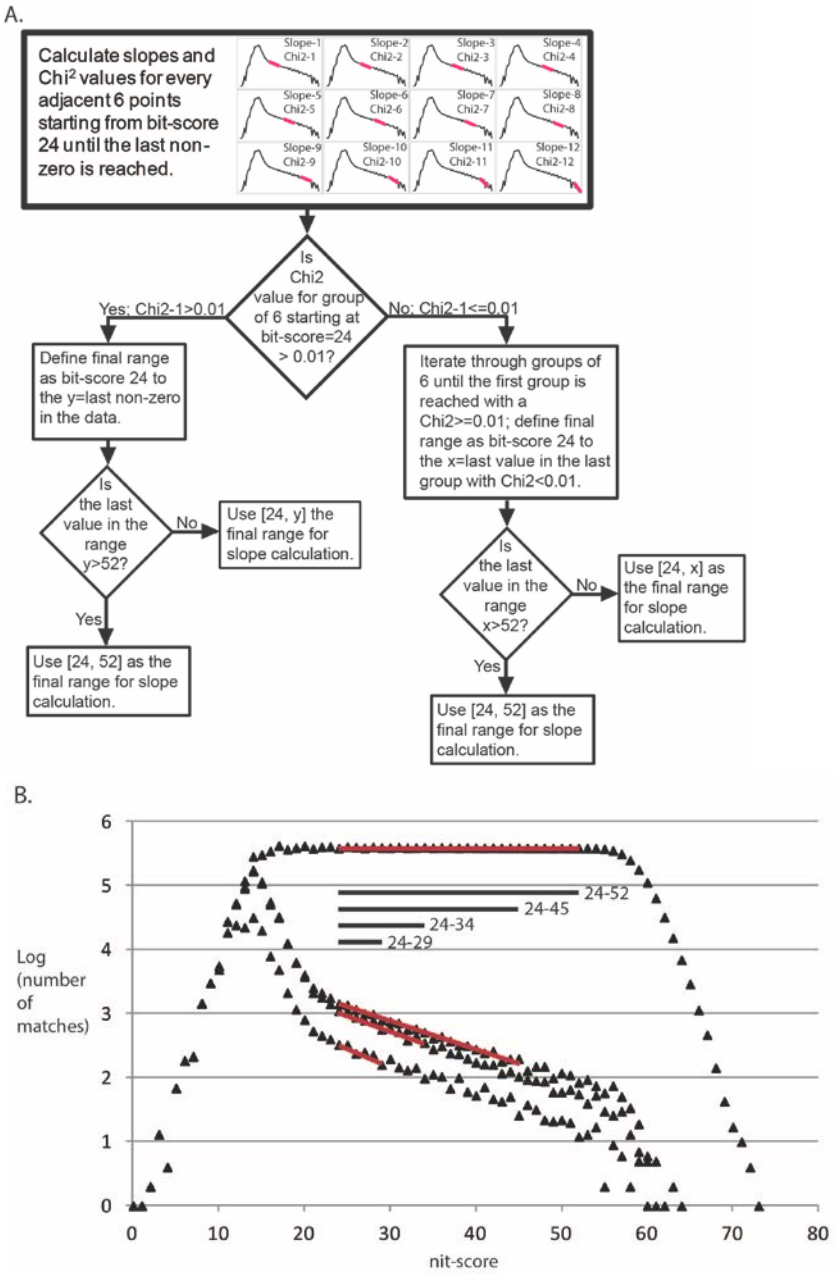
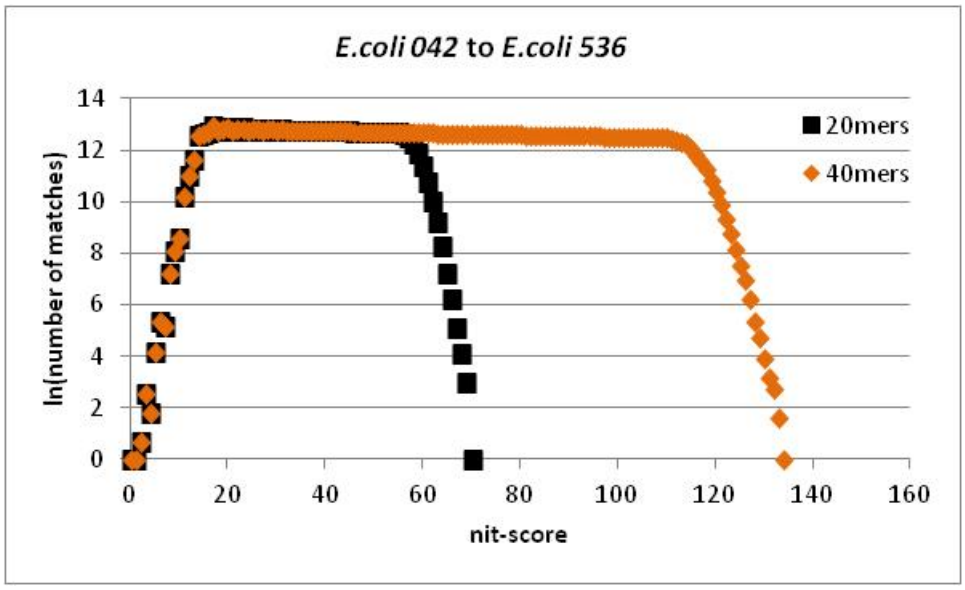


Figure 2-10. Original bounds selection.

A) Flowchart of right bound selection. B) Example bounds for different plots, with very smooth plots having longer ranges of points and very noisy plots using very narrow ranges of points.

A)



B)

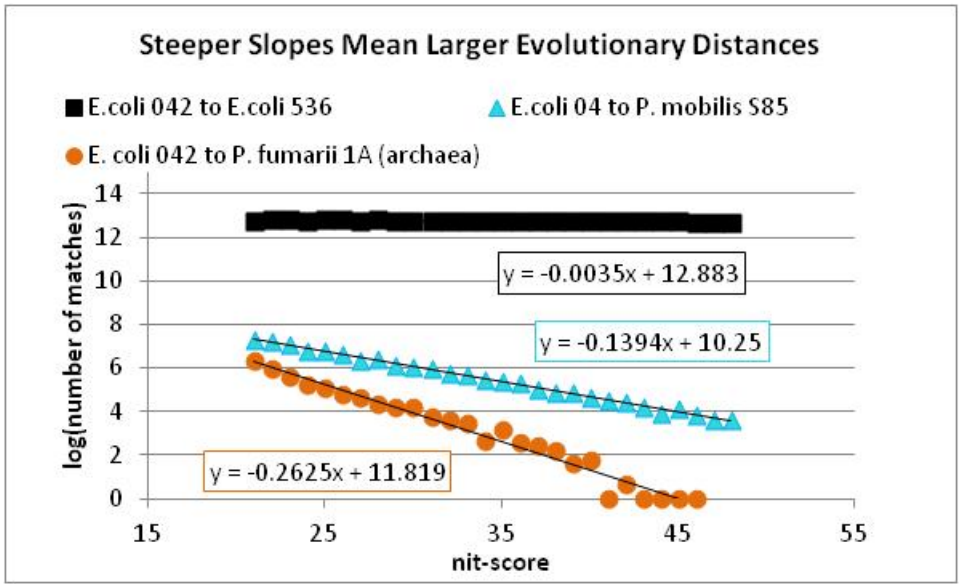


Figure 2-11. The meaning of SlopeTree slopes.

A) SlopeTree plot for two strains of *E.coli.* Slope is nearly zero. B) Slopes for organisms at different distances; steeper sloeps mean larger evolutionary distances.

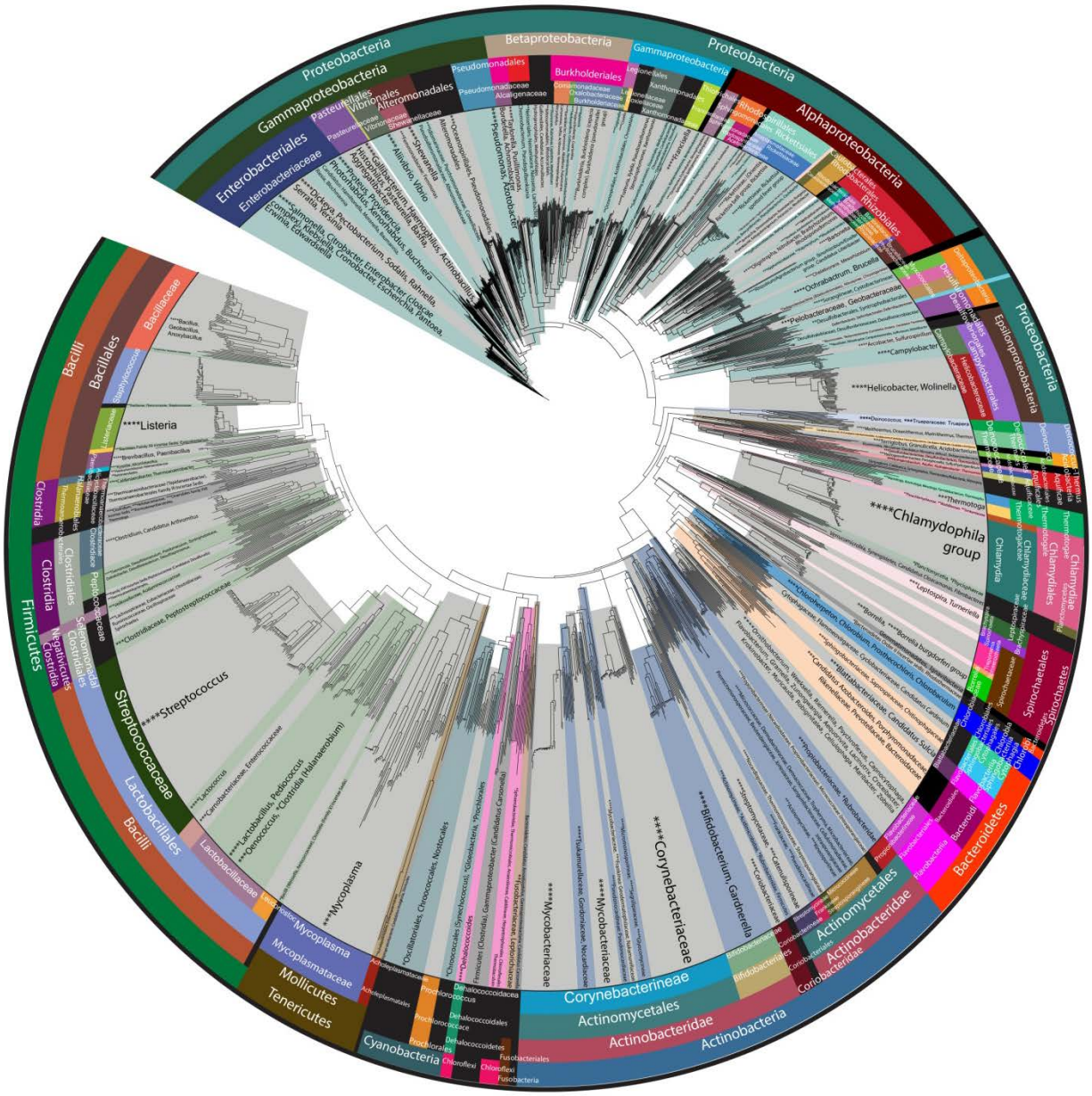


Figure 2-12. SlopeTree (v1) applied to 2001 bacteria.

The external ring of the figure consists of the phylum level. The second ring in consists mostly of bacterial classes, etc.

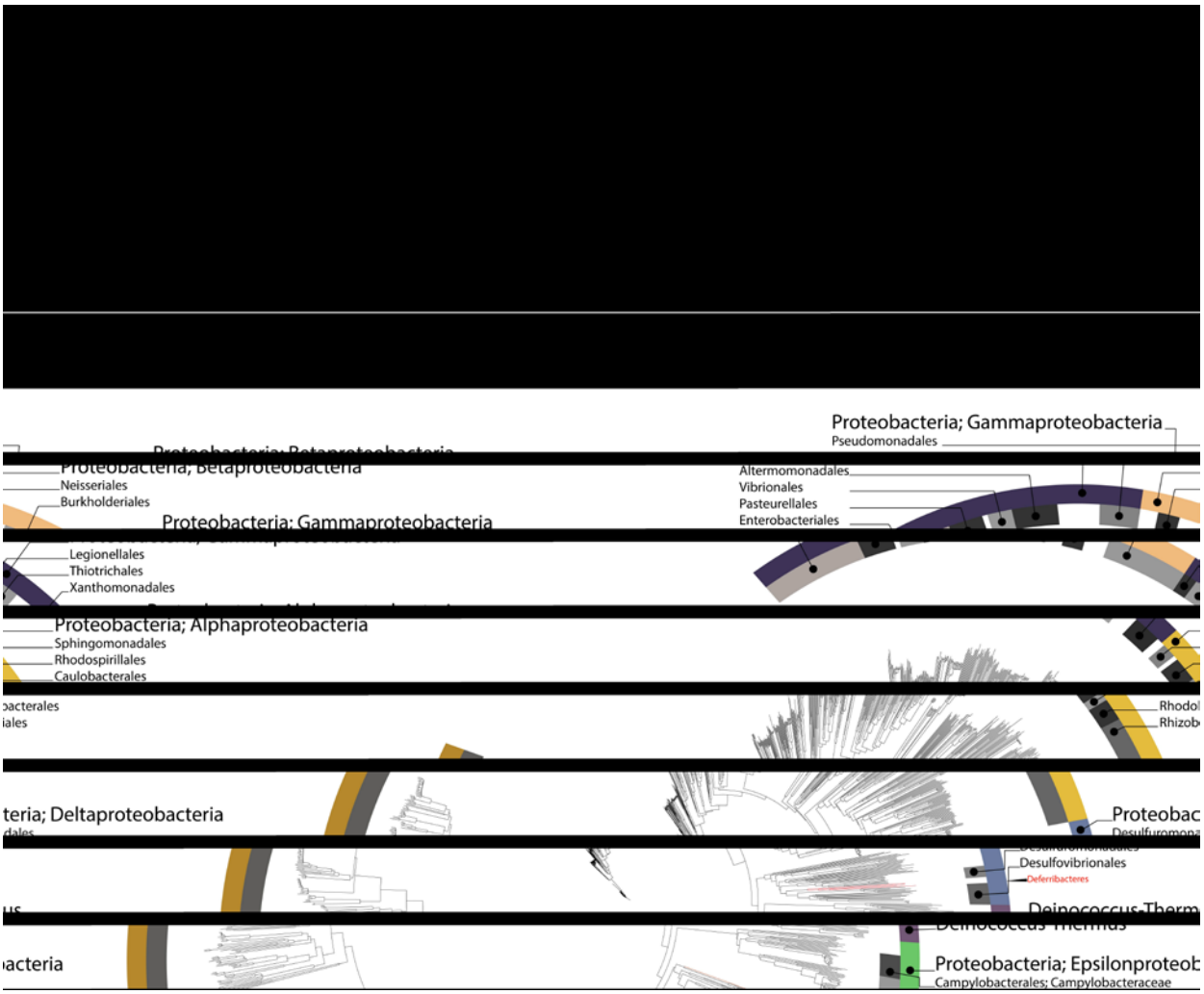


Figure 2-13. Phylogenetic tree constructed by SlopeTree (v1).

All the bacteria considered possibly problematic were removed from the set of 2001 in the construction of this tree, reducing it to 1718 leafs. I highlighted in red a group of 5 bacteria that had a very small number of representatives in the dataset.

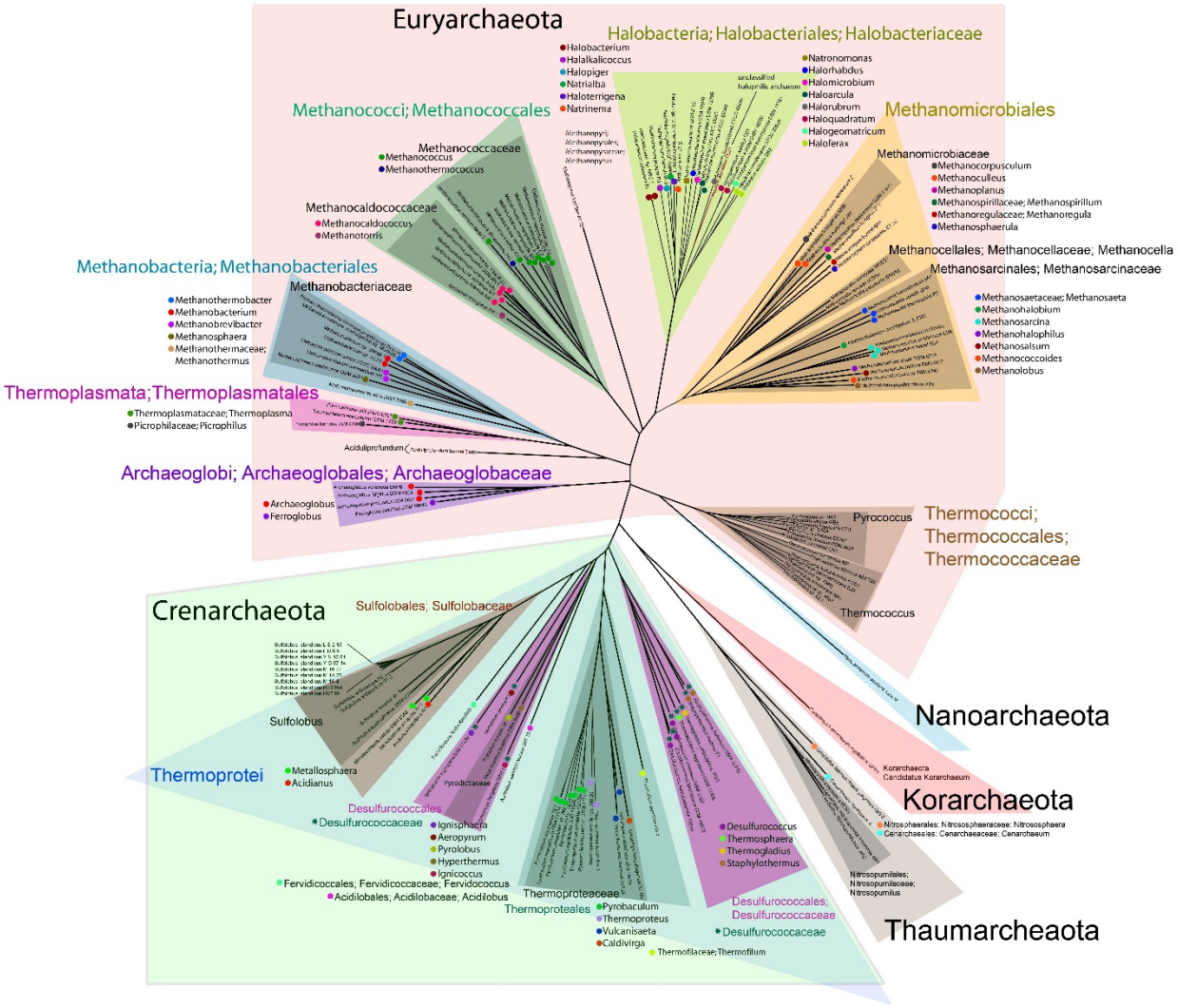


Figure 2-14. SlopeTree (v1) applied to 137 archaea.

Early version of SlopeTree using the linear fit applied to all archaea in the database.

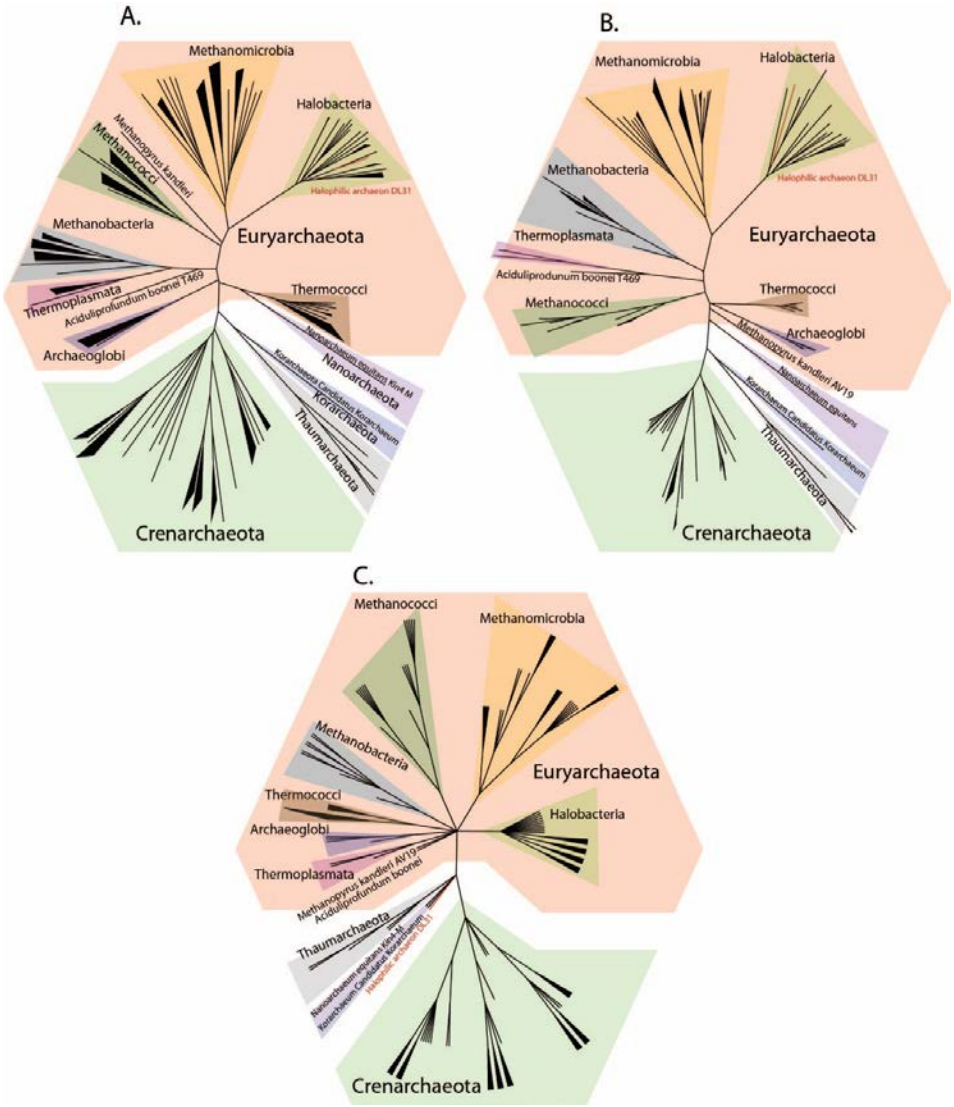


Figure 2-15. Phylogenetic Trees for SlopeTree (v1), 16S rRNA tree, and NCBI over 137 archaea.

A) SlopeTree tree. B) 16S rRNA tree. C) Tree using NCBI taxonomy. All phyla and classes are colored in groups, with colors consistent across all three trees. All the *Crenarchaeota* are of the class *Thermoprotei*, for which reason there are no colored groupings within the phylum.

	NCBI classification	SlopeTree classification
Acidothermus cellulolyticus 11B	Actinobacteria	Verrucomicrobia
Thermanaerovibrio acidaminovorans DSM 6589	Synergistetes	Verrucomicrobia
Cupriavidus metallidurans CH34	Betaproteobacteria	Alphaproteobacteria
Clostridium cellulovorans 743B	Firmicutes	Actinobacteria
Treponema succinifaciens DSM 2489	Spirochaetes	Firmicutes
Treponema brennaborense DSM 12168	Spirochaetes	Firmicutes
Hippea maritima DSM 10411	Deltaproteobacteria	Aquificae

Table 2-1. Seven misplaced bacteria for early version of SlopeTree.

Of the forty-two bacteria SlopeTree misplaced, only these seven lacked a clear explanation.

	1480 bacteria		500 bacteria	
	SD	BSD	SD	BSD
SlopeTree	1642	9.68E-01	538	9.44E-01
CVTree	1770	4.67E+00	666	3.69E+00
ACS	1604	4.58E+01	574	4.23E+01
FFP	2340	1.13E+00	936	1.11E+00

Table 2-2. Comparison to other methods (distance to the 16S rRNA trees).

Symmetric difference and branch score distance between the 16S rRNA tree and trees built by SlopeTree, CVTree, ACS, and FFP.

CHAPTER THREE

CONSIDERING HORIZONTAL GENE TRANSFER

3.1 MOTIVATION

In prokaryotes, evolution is not purely by descent. This being the case, it is necessary to find a way to define what evolves mostly by descent, and to clearly define what is meant by "mostly." Alignment-based methods, such the one the one in Lang et al. (28), often end up identifying the evolution of the largest conserved complexes and the proteins that interact with them. In evolution and archaea, one of these is the ribosome. For a group of molecules that interact with each other, horizontal gene transfer (HGT) is less likely, particularly between remote branches, because it may no longer be possible for the alien proteins to engage in the complementary interactions needed for the function of the given complex. This is an advantage of using proteins from these large, conserved complexes. However, the result is that the phylogenetic approach essentially reduces to assessing the history of the ribosome, or whatever other complex is being considered. Even if the history of the ribosome is relevant to the evolution of organisms, it is still necessary to look further for additional conserved features, to see for example if they represent a coevolving consensus consistent with that provided by the ribosome alone. The method presented here provides one such alternative.

Aquifex aeolicus is a member of the bacterial phylum Aquificae, with a history of extensive horizontal gene transfer that makes it extremely difficult to classify (120). It is an example

of the impact of HGT on phylogenetic analysis and shows that this is a situation that happens and that the corrections for HGT that I present in this chapter and the next are not addressing a merely hypothetical problem. It is not clear how often such transfers happen, but the current database, which is not that large at this time considering the rate at which it is growing, already contains several instances of such events.

3.2 ADDRESSING MISPLACED ORGANISMS IN SLOPETREE TOPOLOGIES

Misplacement of *Petrotoga mobilis*

One of the most troublesome cases of misplacement in the early SlopeTree topologies was that of *Petrotoga mobilis,* a member of the phylum Thermotoga which was consistently placed within the Clostridia, a class within the Firmicute phylum. Unlike the majority of the other misplacements, which could be explained either by *Candidatus* status, having a reduced genome, or being mis-assigned by the current classification, *P. mobilis* had none of these issues. Originally, lacking any other explanation, I ascribed its misplacement to the poor data quality of the proteome, but when I manually assessed a set of the organism's proteins with BLAST, I was unable to find proof of any data quality issues. There was also no indication of chimerism, another theory, or missing proteins, yet another theory. This misplacement was serious because the purpose of SlopeTree was to serve as a trustworthy first approximation of a prospectively very large input set's evolutionary history; the topologies were never intended to be perfect, but gross (i.e. at the phylum level)

misplacements of organisms, with no explanation as to what caused them, went against the core purpose of the method.

The source of the *P. mobilis* problem only became apparent when I looked more closely at the SlopeTree plot, which until this point I had always displayed over the entire range of nit scores. When I isolated the phylogenetic signal from the rest of the data, I was able to observe that there was curvature over that range of points, unlike the majority of SlopeTree plots which are straight lines (Figure 3-1 and 3-2). I had observed curvature in other plots during the initial development of SlopeTree, but when looking at the whole histogram, this curvature always looked relatively minor, possibly just created by the noisy end of the data. For this reason, for some time I remained convinced that the linear fit was adequate. However, when considering the curvature of the plot for *P. mobilis* and *C. clariflavum,* especially after having isolated the phylogenetic signal as in Figure 3-1, I saw that this curvature was much more severe than I had first thought and it was immediately apparent that its source was horizontal gene transfer (HGT). The pair most likely shared a typical SlopeTree plot that could be fitted by a linear function when plotted in log. However, in addition to this 'typical' SlopeTree plot, the pair also likely shared a second group of proteins that were much closer evolutionarily than the first group. With this case, I saw how such groups of horizontally transferred proteins could skew a SlopeTree histogram.

Longer length matches between any pair are expected to be from highly conserved proteins, except in cases where the organisms are very close evolutionarily, which was not the case for *P. mobilis* and *C. clariflavum,* members of separate phyla. When I manually assessed the proteins contributing k-mers to the longer length matches between the pair, I

found that several of these proteins were not highly conserved. Instead, they were associated with adaptation to a toxic environment. These proteins included arsenical resistance proteins, chromate transporters, and mercuric transport proteins.

The misplacement of *P. mobilis* and *C. clariflavum* demonstrated that even methods that use entire genomes or entire proteomes as their input are not robust to HGT merely by virtue of the fact that *most* of the proteins are not horizontally transferred. Because the goal was to develop an unsupervised method whose results were not necessarily perfect but trustworthy within some reasonable limit, it was necessary to implement an automatic correction for cases such as that of *P. mobilis and C. clariflavum.* Once I was aware of the case of *P. mobilis* misplacement and why, I investigated the plots for the small number of other organisms that were also grossly misplaced without explanation; the most notable of these was the group: *Dehalogenimonas lykanthroporepellens* (Chloroflexi), *Syntrophobacter fumaroxidans* (Deltaproteobacteria), and *Desulfarculus baarsii DSM 2075* (Deltaproteobacteria). These plots exhibited even more curvature (Figure 3-3) than the one between *P. mobilis* and *C. clariflavum.*

3.3 IMPLEMENTATION REFINEMENTS

I implemented a series of refinements to the SlopeTree code in an attempt to improve the distances and correct some of the gross misplacements mentioned above. I describe these refinements here. They include a correction for binning artifacts, improved bounds selection, the introduction of weighted fits, a conversion of slopes to evolutionary distances and a correction for the possibility of backwards mutations, and an application of a positive

restraint on the slopes. The most important change we made to SlopeTree at this point, and the only change that successfully fixed some of the misplacements discussed above, was the replacement of the linear fit with a quadratic fit. However, while the quadratic fit fixed the placement of *P. mobilis*, it did not address the problem with *D. lykanthroporepellens*. For this reason, I then developed the filters, which will be described after the implementation refinements.

Correcting for binning artifacts.

SlopeTree corrects for binning artifacts caused by amino acid frequencies and unusual patterns in amino acid composition. For every pair of organisms, an additional histogram is produced consisting of the nit scores from every single sequence in either proteome, from length 1 to the k-mer length. Sequences are counted regardless of whether or not they have matches (Figure 3-4). These sequences do not have to be unique, unlike the main SlopeTree algorithm. These sequences are scored using the nit scores derived for the particular pair, just as in the main match-counting code (b_i). For example, for some protein starting with the sequence MACLLKPSFTLSPWRTINCKA, the sliding window identifies the first 20-mer (assuming k=20), which is the sequence MACLLKPSFTLSPWRTINCK. For this sequence, every substring from the front is then scored and added to the data, e.g. M, MA, MAC, MACL, …, MACLLKPSFTLSPWRTINCK. Then the sliding window shifts over by one amino acid and the process is repeated, adding nit scores to the binning histogram for A, AC, ACLL,…, ACLLKPSFTLSPWRTINCKA.

To produce a histogram corrected for binning artifacts (y_i), where i corresponds to rounded nit scores, for each score in the natural log of the real data (t_i), the natural log of these bin-correction counts (b_i) is subtracted, and the average of the bin-correction ($\langle B \rangle$) added back:

$$y_i = \ln(t_i) - \ln(b_i) + \langle B \rangle, \quad (6)$$

This correction was particularly important for improving the accuracy of the slope-measurement because it mostly applied to the data in the lower nit scores to which SlopeTree gives the highest weights (weights described below).

Improved bounds selection

In the current version of SlopeTree, for the nit-scores in which the counts for the scrambled data (see Background subtraction in previous chapter) are more than 25% the counts for the real data, the real data values are set to 0, and the left bound set to the nit-score with the maximum count. To select the right bound, the binning correction also described above is used. This correction provides an estimate of the nit-score at which the cap on matching sequences, imposed by the maximum k-mer length, would cause the match counts to begin to decline. For each binning correction plot, a rolling average $\langle R \rangle$ across the counts is calculated; starting at nit-score 0, $\ln(\langle R \rangle)$ for each index is stored in a vector. This vector is then scanned for the largest nit-score at which the value of the natural log of the bin correction counts is within 0.1 of the natural log for the rolling average at that same index (i). The right bound is set to i-1, *assuming* the match counts are greater than 0 at this value. Otherwise, it is set to the lowest nit-score for which the pair had no matches.

Introducing a weighted fit

As the matches become sparser for higher nit scores, the data becomes increasingly noisy. Therefore, the slope is measured using a weighted fit, where the scores with higher counts are given more weight (*w(i)*) than those with lower counts:

$$w_i = \frac{t_i}{t_i + W}, \tag{7}$$

w is a constant set to 100 by default. As with so many refinements of SlopeTree, the weights were added to the package later. The slope (*d*) is invariant in the linear equation.

Converting slopes to evolutionary distances and correcting for revertants

I performed two operations to convert our slopes into evolutionary distances. During the initial compilation of the sorted k-mer list, the entropy for each pair of organisms was calculated. For organisms *p* and *q*, this entropy (H_{pq}) is calculated as:

$$H_{pq} = -\sum_{k=1}^{20} \frac{\left(\frac{c_i(k)}{T_p} + \frac{c_j(k)}{T_q}\right)}{2} \ln \frac{\left(\frac{c_i(k)}{T_p} + \frac{c_j(k)}{T_q}\right)}{2} \tag{8}$$

The final slopes are the slopes derived from the quadratic fit multiplied by their respective entropy.

The other operation was necessary due to backwards mutations (i.e. revertants). Alignment-based methods have very complex mathematics for the accumulation of multiple mutations. However, alignment-free methods only have to consider multiple mutations when they revert to their original position. In the absence of backwards mutations, the slope would be the evolutionary distance for the highly conserved subset of a proteome. This simplified

the evolutionary model, which essentially became a two-state model for each amino acid in the starting k-mer (either preserved or not). It was necessary to know, at least roughly, what number of amino acids a starting position could mutate to. In principle, this number would be 19, but in highly conserved positions that were still variable, selection restricted the effective number of possible states. If the total number of possible states was *n*, and *D* was the evolutionary distance, *d* the slope, and *x* the point at which the slope was taken for the quadratic, then our model was that:

$$D = -w\ln((w - dH)/w) \tag{9}$$

$$w = 1 - 1/n \tag{10}$$

$$d = -(2ax + b) \tag{11}$$

This formula was easy to invert to pass from slope to evolutionary distance, but there remained the problem of how to estimate the factor of *n*. I performed a somewhat simplified calculation in order to estimate this value by observing the number of alternative amino acids in k-mers longer than *n*. I found the possible range of *n* to be somewhere between 2.8 and 20 (Figure 3-4). This estimate was likely a lower bound for the actual number. Because of the finite length of the evolutionary distances, we did not observe all possible alternative states, so this presumably caused the estimate to be an underestimation of the actual *n*. This restrained the range of the nonlinearity correction in our model. I expect that the true number would be much closer to the bottom of the range than 20, and *n*=2.8 is the default setting. But even taking the smallest value corresponding to the largest correction for nonlinearity, within the groups of free-living bacteria or free-living archaea, this nonlinearity correction is

not large. As n becomes larger, the formula becomes more linear; a nonlinearity correction using n=20 would be minimal.

This is an incomplete description, because the number of alternative amino acids will be different at every position, and this will make the nonlinearity correction somewhat different from simply averaging the number of possible states. However, seeing as how there is already some uncertainty in our nonlinearity correction, this is a secondary consideration. Furthermore, distance-based methods are robust in terms of the nonlinearity of their measure with respect to evolutionary distance. This robustness depends on the type of the phylogenetic inference from the distance method. CVTree is the best example of limited sensitivity to nonlinearity correction; it has a highly nonlinear distance measure, but nevertheless produces meaningful trees. Considering that I faced a minimal range of nonlinearity uncertainty, in terms of tree construction, this could not have been a major factor.

Applying a Tikhonov positive restraint

I applied a positive restraint on the a-coefficients to both the linear fit and also to the quadratic fit. This restraint requires that the data be fitted twice: in the first pass, the average slope ($\langle A \rangle$) over all plots, the root mean square deviation for the fit (*RMSD*), and the uncertainty of the slope (σ) are calculated. These values are then included in the summation terms used to calculate the fit, resulting in a restrained version of the fit. When calculating the fit for the quadratic equation, I first multiplied out the square of the quadratic equation, which I divided into sums, where $S_{40} = \sum x_i^j y_i^k = \sum x_i^4 y_i^0 = \sum x_i^4$ and $S_{21} = \sum x_i^2 y_i^1$. These

two terms were then modified for the new fit, such that U_{40} and U_{21} were used in the subsequent fit calculations:

$$U_{40} = S_{40} + \left(\frac{RMSD}{\sigma}\right)^2$$

$$U_{21} = S_{21} + \langle A \rangle * \left(\frac{RMSD}{\sigma}\right)^2$$

Figure 3-6 shows the restraint's effect on the distance distribution.

Replacing SlopeTree's linear fit with a quadratic fit

Chapter 2 presented the linear fit and the equations for the regression. Here I present a quadratic fit.

Quadratic fit:

$$y = ax^2 + bx + c$$

$$d = -(2ax + b)$$

Least squares regression for a quadratic equation

$$\sum_{i=0}^{n-1} (ax_i^2 + bx_i + c - y_i)^2$$

When multiplied out and simplified, this is equal to

$$a^2x_i^4 + b^2x_i^2 + c^2 + y_i^2 + 2abx_i^3 + 2acx_i^2 + 2bcx_i - 2ax_i^2y_i - 2bx_iy_i - 2cy_i$$

I then plugged this back into the summation and split the sum:

$$\sum_{i=0}^{n-1}(ax_i^2 + bx_i + c - y_i)^2 = c^2 n + a^2 \sum_{i=0}^{n-1} x_i^4 + (b^2 + 2ac) \sum_{i=0}^{n-1} x_i^2$$

$$+ \sum_{i=0}^{n-1} y_i^2 + 2ab \sum_{i=0}^{n-1} x_i^3 + 2bc \sum_{i=0}^{n-1} x_i - 2a \sum_{i=0}^{n-1} x_i^2 y_i - 2b \sum_{i=0}^{n-1} x_i y_i - 2c \sum_{i=0}^{n-1} y_i$$

Notation:

$$S_{jk} = \sum x_i^j y_i^k$$

$$S_{00} = \sum x_i^j y_i^k = \sum x_i^0 y_i^0 = n$$

Using the new notation:

$$\sum_{i=0}^{n-1}(ax_i^2 + bx_i + c - y_i)^2$$

$$= a^2 S_{40} + (b^2 + 2ac) S_{20} + c^2 S_{00} + S_{02} + 2ab S_{30} + 2bc S_{10} - 2a S_{21}$$

$$- 2b S_{11} - 2c S_{01}$$

Derivatives in terms of a, b and c:

a: $2aS_{40} + 2cS_{20} + 2bS_{30} - 2S_{21}$

b: $2bS_{20} + 2aS_{30} + 2cS_{10} - 2S_{11}$

c: $2aS_{20} + 2cS_{00} + 2bS_{10} - 2S_{01}$

This is a system of linear equations:

$$[S_{40} \quad S_{30} \quad S_{20}][a] = [S_{21}]$$

$$[S_{30} \quad S_{20} \quad S_{10}][b] = [S_{11}]$$

$$[S_{20} \quad S_{10} \quad S_{00}][c] = [S_{01}]$$

Solve for *a*:

$$a = \begin{bmatrix} S_{21} & S_{30} & S_{20} \\ S_{11} & S_{20} & S_{10} \\ S_{20} & S_{10} & S_{00} \end{bmatrix} \Big/ \begin{bmatrix} S_{40} & S_{30} & S_{20} \\ S_{30} & S_{20} & S_{10} \\ S_{20} & S_{10} & S_{00} \end{bmatrix}$$

$$= \frac{[(S_{21}S_{20}S_{00}+S_{30}S_{10}S_{01}+S_{20}S_{11}S_{10})-(S_{21}S_{10}S_{10}+S_{30}S_{11}S_{00}+S_{20}S_{20}S_{01})]}{[(S_{40}S_{20}S_{00}+S_{30}S_{10}S_{20}+S_{20}S_{30}S_{10})-(S_{20}S_{20}S_{20}+S_{10}S_{10}S_{40}+S_{00}S_{30}S_{30})]}$$

The slope (*d*) was invariant in the linear equation. However, in the quadratic equation, the slope varies as a function of *x*, with the choice of *x* having an effect on the final trees. By default, *x* is set to 15, used in all trees presented in this chapter.

A third fit was introduced later (Chapter 4). However, for reasons explained in the next chapter, the current SlopeTree implementation uses the quadratic fit introduced here.

3.4 INTRODUCING SLOPETREE FILTERS FOR PRE-PROCESSING INPUT DATA

The quadratic fit brought the SlopeTree bacterial tree topology in closer agreement with the NCBI classification. For instance, the quadratic fit moved *P. mobilis*, which was misplaced by the linear fit with the Clostridia to the Thermotogae. However, the new fit did not fix the misplacement of *D. lykanthroporepellens*, a Chloroflexi that was placed with the Deltaproteobacteria. The curvature in the plots between *D. lykanthroporepellens* and *S. fumaroxidans*, and between *D. lykanthroporepellens* and *D. baarsii* was much more pronounced than the curvature for *P. mobilis* and *C. clariflavum* (Figure 3-3). Investigating the proteins that might have been transferred, I found several dozen phage proteins shared between *D. lykanthroporepellens* and the other two bacteria, indicating a transfer of a single copy phage.

Filtering mobile elements

With the failure of the quadratic fit to address all HGT-caused misplacements, I eventually decided to develop a method for automatically identifying proteins that might have been transferred, or at least for identifying proteins that are not conserved, and to automatically remove them from the analysis at an early stage. This led me to the list of sorted k-mers, where I already knew that the frequently long blocks of full-length sequence matches corresponded to very highly conserved proteins. While considering the statistics of these groups of k-mers, I together with my mentor eventually noticed that mobile elements exhibit completely different copy number patterns within proteomes and between them as compared to coevolving orthologous proteins. This rule underlying the presence and absence of mobile elements made it possible to identify them automatically and then remove them.

The formal algorithm is presented below. In prose, the process consists of generating alphabetically sorted k-mer lists for each proteome at the very beginning of a SlopeTree run. For each organism separately, these k-mers are clustered by comparing immediately neighboring sequences in the list. By default, k-mers that are identical in 19 out of 20 amino acids are put into the same cluster. The values for *a* and *b*, mentioned in Algorithm 1, are by default 1.0 and 3.0, respectively. In this way, I eliminated the proteins in each genome that appeared to be present in high copy number. These were almost always parasitic elements such as phage proteins. These are removed from the analysis prior to calling the main SlopeTree algorithm (Figure 1-1).

EF-Tu is the one consistent exception to this. EF-Tu is frequently present in multiple copies in a single genome. Therefore, at the stage of k-mer generation, k-mers are compared

to a small set of conserved, hardcoded sequences from EF-Tu. Proteins with k-mers that overlap with these sequences by 60% or more are considered matches and are marked so that the filters cannot remove them.

The mobile element filter was initially a small module embedded within the conservation/stability filter. This was because I preferred to refer to the conservation information before deleting a protein from my set due to its copy number. However, this was ultimately not an appropriate combination, as sometimes one might want to run the one without the other. For this reason, I removed the mobile element filter from the conservation filter code, and then had to introduce a reference set of conserved proteins on the side. How exactly I define this reference set is described in this chapter's Material's and Methods, but I found that the inclusion of this reference set made the mobile element significantly better both at identifying mobile elements and also at not throwing away conserved proteins. Appendix A presents a set of proteins marked for deletion by means of this filter.

Algorithm 1: Mobile Element Filter

Input: A set S of *n* proteomes $\langle S_1, S_2, \dots, S_n \rangle$ and a set T = $\langle T_1, T_2, \dots, T_l \rangle$, with T taken from *l* taxonomically diverse organisms where T_i consists solely of the highly conserved proteins of the organism *i*. In practice, *l* is generally much smaller than *n*, but this is not required.

Output: A set V = $\langle V_1, V_2, \dots, V_n \rangle$ where each V_i consists of all proteins in S_i, minus the mobile elements.

Algorithm: Let p_{ij} be the j^{th} protein in S_i, and let $p_k^{ij}[h]$ be a k-mer from p_{ij} of length *k*, starting at index *h*, where $0 \leq h < f$ given that p_{ij} has length *f*. For those k-mers at the end of

each protein where $h+k>f$, the suffix is expanded by the necessary number of empty characters to fill the remainder of the k-mer. Each k-mer is stored as a 2-tuple consisting of the k-mer and the gene ID (j). Let A_i be the alphabetically sorted list of all 2-tuples from S_i. For every protein p_{ij}, there is a pair of integers, r_{ij} and c_{ij}, both initialized to 0. Starting from the first k-mer in A_i, we pass down the list until a k-mer with more than u mismatches with this first k-mer is found. For all proteins with k-mers in this block, r_{ij} is incremented. This process is repeated until the end of A_i is reached, always starting from the first k-mer to not be a member of the current block of matches.

Separately, we repeat the k-mer compilation process described above on T to generate a single, alphabetically sorted list of 2-tuples across all proteomes in T. Duplicates are removed from this list to make a new list B consisting of each k-mer and the number of times it appears in T. Those k-mers appearing only once are given a count of 1. Then for every k-mer in A_j, we query B; the value of c_{ij} is increased by the count stored in B for every exact match between B and any k-mer in any protein p_{ij}.

Having set all r_{ij} and c_{ij} for all p_{ij} in S_i, we define a linear function such that all p_{ij} with $r_{ij} \geq ac_{ij}+b$ are removed from proteome P_i and the reduced proteome we call V_i.

Computational complexity: For n organisms and m amino acids in S, let $m=m_1+m_2+\ldots+m_n$. For l organisms and k amino acids in T, let $k=k_1+k_2+\ldots+k_l$. The compilation of A_i is done in $O(m)$ time, and the time required for sorting each A_i is $O(m_i \log m_i)$, which summed over all n organisms is $O(m \log m)$. Similarly, the time to compile all k-mers in T is $O(k)$ and to sort them requires $O(k \log k)$ time. The order of the algorithm is dominated by the sorting, and therefore the computational complexity of the filter is $O(m \log m + k \log k)$.

Filtering by conservation

The mobile element filter identified many proteins not evolving by descent, but had a surprisingly small effect on the final trees; it did not correct any of the serious misplacements and seemed to mostly remove proteins that were not contributing many sequence matches in the first place. For this reason, I implemented a second filter that filtered according to copy number *across* the entire input (i.e. not just within single proteomes but now between them) as well as according to conservation.

The k-mers in the final alphabetically sorted list *across all organisms* are compared to their immediate neighbors and clustered together if *x* amino acids (default=13 out of 20) are identical (i.e. same amino acid in the same position). The default value of 13 matches (for 20-mers) for clustering is adjustable, with a higher cutoff (e.g. 19 or 20) being suitable for strain-level phylogeny. At the end of the clustering and counting process, paralogy scores are calculated by dividing the protein count field by the genome count field. Orthologs generally have a value of 1 for this ratio, whereas paralogs and mobile elements have ratios that are often much higher. These values are summed for each protein across all clusters. A final value of 0 causes the protein to be marked for elimination. Proteins with a paralogy score greater than an orthology cutoff (default=1.3) are also eliminated. The default value of 1.3 was chosen in consideration for EF-Tu.

Paralogy scores can be calculated for a range of conservation levels. A parameter, which we refer to as *o* in the text, refers to the level of filtering that was applied. The two variables mentioned above, genome count and protein count, are both arrays (default

size=10) in the implementation (arrays G_{ij} and F_{ij} in Algorithm 2). Genome count and protein count for index 0 (i.e. o=0) of this table would be updated for every cluster regardless of cluster size. For index 2 (o=2) of the table, on the other hand, the value would only be updated only for clusters in which 20% or more of the reference set was represented. Paralogy scores calculated from higher indices of the table therefore produced smaller proteomes consisting of more conserved proteins (Figure 3-7).

Algorithm 2: Conservation and Stability Filter

Input: A set W of $n+k$ proteomes consisting of two sets of proteomes: a set V of n proteomes $\langle V_1, V_2, \dots, V_n \rangle$ and a set U of z proteomes $\langle U_1, U_2, \dots, U_z \rangle$, with U taken from taxonomically diverse organisms.

Output: A set H = $\langle H_1, H_2, \dots, H_{n+k} \rangle$ where H_i is the subset of W_i containing conserved proteins with stable copy number.

Algorithm: Let p_{ij} be the j^{th} protein in W_i, and let $p_k^{ij}[h]$ be a k-mer from p_{ij} of length k, starting at index h, where $0 \leq h < f$ given that p_{ij} has length f. For those k-mers at the end of each protein where $h+k>f$, the suffix is expanded by the necessary number of empty characters to fill the remainder of the k-mer. Each k-mer is stored as a 3-tuple consisting of the k-mer, the proteome ID (i), and the gene ID (j). Let D be the alphabetically sorted list of all 3-tuples from both V and U.

We define a k-mer cluster to be a block of adjacent k-mers in D in which no k-mer has more than u mismatches with the previous k-mer. Starting from the first k-mer in D, we compare adjacent k-mers to identify all clusters in D. At the end of this process, the k-mers

in adjacent clusters are checked against one another and merged by the same rule of no more than u mismatches, a step which circumvents the frequent problem of stray k-mers interrupting what would otherwise be a single block of matches. We call this final set of clusters C.

Every protein in p_{ij} from W is assigned a pair of integer arrays, G_{ij} and F_{ij} each initialized at every index to 0 (default size=10). For each cluster in *C*, let g be the number of organisms from U with *at least* one k-mer in the cluster, and let f be the number of total 3-tuples in the cluster with k-mers from U, including repeats. We use G_{ij} and F_{ij} to accumulate the sums of f and g, respectively, for each cluster; the index of the array for a given cluster is selected by a function of the fraction of the total proteomes in U with hits in the cluster. If y is the number of proteomes in U with hits in the cluster, $o = \lfloor 10y/z \rfloor$. For every protein p_{ij} with a k-mer in a given cluster from C, let g and f be added to the values of G_{ij} and F_{ij} at index o, respectively.

After passing through all clusters in C, we assign a paralogy score for every protein p_{ij}, for each possible value of o, where we define a paralogy score X_{ij}^{o} for each value of o as $X_{ij}^{o} = \sum_{k=o}^{k<10} G_{ij}[o] / \sum_{k=o}^{k<10} F_{ij}[o]$. H consists of all proteomes in V and U, where only proteins that have $0 < X_{ij}^{o} \leq$ orthology cutoff (default=1.3) retained. How conserved the final set H is depends on the user's selection of o.

The reference set U is not mandatory. When a reference set is absent, the whole set V is treated as the reference by the algorithm.

Computational complexity: As in Algorithm 1, the time to compile the sorted list of k-mers is $O(m \log m)$, where m is the total number of amino acids in W. The clustering is performed

in $O(m)$ time, and the calculation of final scores is performed in $O(n+k)$ time. Therefore, the computational complexity of the filter is $O(m \log m)$.

Selecting a reference

The SlopeTree package includes an option for including a reference set in a run. This reference is a set of user-chosen organisms. The reference set is used for two purposes: 1) filtering out proteins (often but not necessarily mobile elements and non-conserved proteins) and 2) checking for HGT (described in Chapter 4). If filtering is performed in the absence of a reference set, the set is filtered against itself. 30 bacteria were chosen out of the bacterial input of 495 to be the reference, and 10 were chosen for the archaeal input of 73. The sets of diverse bacteria and archaea for the reference sets are listed in the Appendix B**.** Figures 3-8 and 3-9 show raw (unpruned, unfiltered) trees generated by SlopeTree for 495 bacteria and 73 archaea, with the reference organisms highlighted to show their distribution on the tree. Organisms with short branches from the root were chosen, and each set was made as diverse as possible. Sets were chosen multiple times as the SlopeTree method improved and as inputs changed, and the archaea highlighted in Figure 3-8 were eventually replaced by a new, more evenly distributed set.

Flagging potentially problematic inputs

SlopeTree identifies potential problems in the input such as: reduced genomes (<140,000 amino acids), under-representation of conserved genes, over-representation of conserved genes, and candidate status. Reduced genomes are detected at the early k-mer-counting step.

Candidate division organisms are identified simply by scanning the name of the organism for 'Candidatus.'

SlopeTree identifies proteomes with an under- or over- representation of conserved genes by means of calculations performed during the k-mer clustering described above on filtering. When a k-mer cluster contained a large fraction of the reference set (default=0.9), SlopeTree calculates the average number of hits for the cluster per reference proteome. Generally, clusters with hits in 90% of the reference set come from conserved proteins, and this average number of hits is close to 1. For every cluster, for every organism represented in the cluster, the difference between the number of hits that the organism has in the cluster and the average number of hits per reference organism is stored as a running sum. Some organisms are left with much higher values for these sums than others; the IDs of these organisms are written to file. SlopeTree identifies proteomes with an under-representation of conserved genes in a similar manner, using the same set of clusters discussed above (i.e. 90% or more of the reference set present in the cluster). For every organism, SlopeTree counts the number of times the organism has a hit in one of these cluster. At the end of the process, some organisms which were frequently absent from these conserved clusters had significantly lower values for this count, and were also written to file. The list of flagged proteomes for archaea and bacteria is available in Appendix C.

These tests identified that genomic sequences based on WGS assembly of environmental reads can have particular characteristics, such as paralogy, rather different from complete genome assemblies. This is very likely due to the intrinsic difficulties in performing assembly based on a non-homogeneous source.

3.5 RESULTS

Series of SlopeTree (ST) trees were generated for 72 *Escherichia coli* and *Shigella*, 73 archaea, and 495 bacteria. Reference trees using maximum likelihood applied to a set of concatenated proteins were also built; I refer to these trees as the Eisen-trees, and they are described in the Materials and Methods section below.

SlopeTree provides two filters that remove proteins from the input prior to the distance calculations. The Mobile Element (ME) Filter (Algorithm 1) removes mobile elements by taking advantage of their unique copy number patterns within individual proteomes. The Conservation and Stability Filter (Algorithm 2) removes proteins exhibiting an unstable pattern of presence and absence in a taxonomically diverse reference set, with a parameter (o) corresponding to the fraction of reference organisms that have to have k-mer matches with a given protein the protein to be retained.

SlopeTree proved to be an effective tool for strain-level phylogeny, despite the number of matches between strains of the same species being enormous and most distances being very close to zero. SlopeTree was applied to archaea and bacteria separately because matches for organisms belonging to different domains can be very sparse, branch-length nonlinearity is magnified at very large genetic distances (e.g. between the domains of life), and there are cases of occasional but extensive HGT between domains (121-123).

A more detailed presentation and discussion of the series of bacterial and archaeal trees can be found in Chapter 4, where the final HGT correction is described; in this way, the full series of trees can be considered at once, without having to leave any final correction for

later. The strain-level analysis is discussed here because the final HGT correction was not applied to that set.

Filtering for mobile elements and by stability and conservation

I observed occasional curvature in the SlopeTree histograms. The linear fit was inadequate for plots exhibiting this curvature. Manual inspection of the proteins associated with long length matches between organisms with unexpectedly close distances identified several cases of horizontal gene transfer (HGT). I implemented a quadratic fit to address this, which produced better slopes for a number of cases. However, the quadratic fit also performed poorly when it came to large-scale HGT, e.g. cases involving single copy phages. For this reason, I developed the two filters and the final HGT correction.

Mobile elements are often present in multiple copies in a single genome, with their k-mers therefore also being present in multiple copies; I used this feature of mobile element k-mer copy number to identify and remove these proteins. This criteria removed an average of 118 proteins from each archaea (stdev=116) and 162 proteins from each bacteria (stdev=246). The archaea with the most mobile elements removed was *Methanosarcina acetivorans C2A*, which had 744 proteins removed out of a total 4540. The bacteria with the most mobile elements removed, and which did not show issues with data quality, was *Arthrospira platensis NIES-39*, which had 2143 proteins removed out of a total 6630.

The effect this filtering had on the distance to the Eisen-trees was variable; SlopeTree and CVTree show negligible difference before and after the application of the filter; ACS and

kmacs showed a small reduction in distance to the Eisen-trees; and D2 and Spaced Words showed a significant reduction in distance to the Eisen-trees (Table 4-1).

The conservation filter used a taxonomically diverse reference set of organisms to identify proteins with k-mers that had hits for a minimum fraction (~o) of the reference set, and calculated paralogy scores that provided an estimate of a protein's copy number profile across the entire reference set. This filter was applied to the majority of the ST-trees, in conjunction with the ME-filter. The purpose was to observe how the phylogenetic trees might change as the input was reduced to an increasingly conserved core, and to assess whether these automatic filters could help produce higher quality trees while keeping the methods completely unsupervised. As a validation, I generated histograms from the paralogy scores for proteins with specific keywords in their annotations, with for example 'ribosomal' as an instance of a core protein and 'chemotaxis' as an instance of an unstable, often horizontally transferred protein (Figure 3-10). The former has a sharp peak at the paralogy score of 1 which decreased but does not disappear for increasing o. The latter has two peaks at 0 and 5, with all paralogy scores of 1 disappearing by o=2, indicating that chemotaxis proteins are frequently absent or present in multiple copies. Proteins with paralogy scores less than 1 and greater than 1.3 are filtered out; therefore, as o is raised, chemotaxis and other similar proteins are gradually eliminated while the majority of ribosomal proteins and other stable, conserved proteins are retained. For every method, this filtering steadily reduced the distance to the Eisen-trees (Table 4-1) and organisms that were misplaced (according to the NCBI taxonomy) in the unfiltered trees were frequently placed correctly in the more filtered trees.

Strain-level analysis

A series of ST-trees was built for 62 *E.coli* and 10 *Shigella* (main tree shown in Figure 3-11), which were all the complete proteomes available for these species at the time of this writing. This was to test the range at which SlopeTree could still resolve sensible evolutionary distances. *Escherichia fergusonii* and *Escherichia blattae* were included in the run as outgroups to root the trees, but were removed from the final distance matrices prior to tree-building because their presence excessively compressed the other distances (Figure 3-12). To assess whether longer k-mers might produce more accurate distances at the strain-level, I built a tree using 20-mers and another using 40-mers (Figure 3-13). I did not observe an improvement; the 20-mer and 40-mer trees were in very close agreement, with topological differences arising from short branches mainly in the B2 phylogroup. I built additional trees using proteomes filtered for mobile elements, and also proteomes filtered for stability and conservation, in which the reference set for the conservation filter was simply the entire input. The average number of proteins per proteome for the 72 *E.coli* and *Shigella*, prior to filtering, was 4730 (stdev=485). When the set was filtered just for mobile elements, the average size was reduced to an average of 4282 proteins (stdev=402). This set, with mobile elements removed, was filtered against itself for the smallest possible filtering parameter (o=0), reduced the average proteome size to 4071 (stdev=362); for self-filtering on o=5, the average size was then 3465 (stdev=209); and for o=10, the average size was 1290 (stdev=9). For all trees, the trees were highly similar to the unfiltered trees. I performed more aggressive conservation filtering against a reference set of 30 diverse bacteria (o=3), leaving

an average of 343 (stdev=41) proteins per proteome. This was done to investigate whether the trees built from the most conserved genes across the entire domain of bacteria matched those built without filtering and those built with loose filtering. Again, I observed only minor changes in topology, mostly involving short branches. As an additional validation, I reduced the unfiltered 20-mer tree to the set considered in Touchon et al. (124) which was used as a reference for another alignment-free method in Sims et al. (61); these two topologies were also found to be in agreement.

The ST strain-level topology also agreed with current phylogroups of *E.coli* and *Shigella*. There are different means for determining phylogroups, with some assignments varying between approaches (125, 126); SlopeTree supports the grouping of *E. coli IAI39 uid59381*with phylogroup D and *E. coli APEC O78 uid187277* with phylogroup C. Pathotypes do not follow phylogeny (127) and when they were mapped the trees, their placement was scattered. The genes responsible for pathogenicity are frequently mobile elements (15, 128, 129), so I constructed an ST-tree from mobile elements and less conserved proteins removed during filtering on o=0, to investigate whether strains of the same pathotype would cluster. I did not see this effect; not surprisingly, this tree differed from the other trees in several placements, but nevertheless held many groupings in common, particularly between the more closely related strains.

When strains differ by very few mutations in DNA, most of these will not cause changes in coding sequence. For such cases, performing phylogenetic analyses by following the easily identifiable mutations at the DNA level is the more accurate and practical approach.

SlopeTree filtering benefits other methods

Filtering lessened the distance to the Eisen-trees for all methods (Figure 3-14B; Table 4-1). The filters developed for SlopeTree are equally applicable to any method that takes proteomes as its input. As the level of filtering increased, the distances between the ST-trees and the Eisen-73 or the Eisen-495 trees decreased. All other alignment-free methods that we tested also benefited from filtering the data prior to running, at least in terms of their distances becoming closer to the Eisen trees. An additional benefit was that filtering the data beforehand decreases the run-times.

I also observed a distinct difference in the nature of the branch lengths between different methods; D2, SlopeTree and Spaced Words fall into one group, having a wider range of branch lengths, while ACS, CVTree, kmacs, and ALFRED-G have branch lengths that are restricted to a more narrow range (Figure 3-14A, C, D). ACS appears to be the most restricted in this regard, and we found that by applying the conservation filter, the range for a given method's distances was somewhat widened.

3.6 MATERIALS AND METHODS

Downloading proteomes, selecting input sets, and building Eisen-trees

I downloaded the archive all.faa.tar from the NCBI ftp website (ftp://ftp.ncbi.nih.gov/genomes/Bacteria/) in May 2015, as discussed in Chapter 2. I downloaded the Maximum Likelihood trees, S1 and S4 files from Lang et al. (28), built from the concatenations of 24 conserved proteins, and compared their set of organisms to those

present in the FASTA archive. Allowing for some imperfect matches (e.g. *Haliangium ochraceum SMP 2 DSM 14365* in the ML tree, opposed to *Haliangium ochraceum DSM 14365* in the archive) and some differences in strains (e.g. *Eubacterium siraeum DSM 15702 uid54603* in the ML tree, opposed to *Eubacterium siraeum uid197160* in the archive), 73 archaea and 495 bacteria were found in common between the ML trees and the archive. Two lists were compiled of organisms to remove from the ML trees and these lists and trees were given as input to the program nw_prune, from the package newick-utils (version 1.6) (130):

```
./nw_prune Eisen_newick_ML_journal.pone.0062510.s008.txt $(cat pruning_bacteria.txt) > eisen_495_tree_bacteria_newick.txt
```

```
./nw_prune Eisen_ML_841_journal.pone.0062510.s011.txt $(cat pruning_archaea.txt) > eisen_73__tree_archaea_newick.txt
```

These two supermatrix-derived trees are referred to as the Eisen-73 tree and the Eisen-495 tree and were produced for comparison purposes (Figures 3-15 and 3-16).

Pruning trees

A raw tree consisting of the full sets bacteria and archaea is available for each method (SlopeTree and alternative alignment-free methods). The remaining trees were pruned of the organisms that SlopeTree automatically flagged as problematic, 2 for archaea and 50 for bacteria (Appendix C). The distance matrices were pruned of the flagged organisms before being passed to rapidnj. Pruned versions of the Eisen-trees were also created, using nw_prune as described above with the organisms flagged by SlopeTree added to the file of

organisms to prune. This was necessary for the pruned trees to be comparable to the Eisen-trees (Figures 3-17 and 3-18).

Building SlopeTree Trees

The scripts I refer to in this section are included in the SlopeTree package (https://git.biohpc.swmed.edu/biohpc/slopetree).

Commands for constructing the raw SlopeTree trees for the sets of bacteria, archaea and *E.coli*

All bacterial proteomes were moved to the directory FAA within the directory Bacteria. All archaeal proteomes were moved to the directory FAA within the directory Archaea. All proteomes for the strain-level analysis were moved to the directory FAA within the directory Ecoli. The distance matrices for these two sets were then generated with the following two scripts:

```
bash dSTm.sh Bacteria/ 20 B  ../Taxonomy/
bash dSTm.sh Archaea/ 20 A ../Taxonomy/
bash dSTm.sh Ecoli/ 20 B  ../Taxonomy/
```

The distance matrices were then passed to rapidnj. We refer to these trees as the “raw” trees.

Selecting the reference sets for bacteria and archaea

I manually selected thirty diverse bacteria from the raw ST-tree as our reference set for the bacterial runs. Similarly, I manually selected ten diverse archaea for the archaeal runs. The specific organisms selected are listed in S22 Text.

Building ST-trees with mobile elements removed

The reference sets for bacteria and archaea were moved to Bacteria_ref/FAA and Archaea_ref/FAA, respectively. I then filtered them for conservation, using our conservation filter, for the parameter of o=7:

For bacteria:

```
bash pFilt.sh Bacteria_ref/ 20
./fpwrite Bacteria_ref/ –f 10 –o 7
```

For archaea:

```
bash pFilt.sh Archaea_ref/ 20
./fpwrite Archaea_ref/ –f 10 –o 7
```

These commands generated proteomes that had been reduced to their core proteins. These reduced proteomes were moved to new directories Bacteria_ref_10_7/FAA and Archaea_ref_10_7/FAA and the list of merged and sorted 20-mers generated for each of them:

```
bash dMT.sh Bacteria_ref_10_7/ 20 B
bash dMT.sh Archaea_ref_10_7/ 20 A
```

This created a set of sorted 20-mers from conserved proteins from a diverse reference set for bacteria and for archaea. These sets were used as the reference for the mobile element filtering:

```
./mef Bacteria/ Bacteria_ref_10_7/MERGED_TAGS/
./mef Archaea/ Archaea_ref_10_7/MERGED_TAGS/
./mef Ecoli/ Bacteria_ref_10_7/MERGED_TAGS/
```

This produced, for bacteria, archaea and our set of *E.coli*, a set of proteomes in which the mobile elements were eliminated. These reduced proteomes were automatically written out to Bacteria/FAA_mobelim, Archaea/FAA_mobelim and Ecoli/FAA_mobelim. I moved these reduced proteomes to Bacteria_MEF/FAA, Archaea_MEF/FAA and Ecoli_MEF/FAA

and moved the organisms that had been chosen for the reference sets to FAA_ref directories within each main directory. I then ran the main SlopeTree script to produce the final distance matrices:

```
bash dSTm.sh Bacteria_MEF/ 20 B  ../Taxonomy/

bash dSTm.sh Archaea_MEF/ 20 A ../Taxonomy/

bash dSTm.sh Ecoli_MEF/ 20 B  ../Taxonomy/
```

Trees were then built using rapidnj.

Building Trees Filtered by Conservation

The FAA and FAA_ref directories from Bacteria_MEF/ and Archaea_MEF/, and the FAA directory for Ecoli_MEF, were copied to Bacteria_MEF_CF, Archaea_MEF_CF, and ECOLI_MEF_CF, respectively. I then ran the filtering code:

```
bash pFilt.sh Bacteria_MEF_CF/ 20 B
bash pFilt.sh Archaea_MEF_CF/ 20 A
bash pFilt.sh Ecoli_MEF_CF/ 20 B
```

For bacteria and archaea separately, I generated five sets of proteomes filtered on *o*=0, *o*=1, *o*=3, *o*=5 and *o*=7. The following two commands use o=3 as an example:

```
./fpwrite Bacteria_MEF_CF/ –f 10 –o 3
./fpwrite Archaea_MEF_CF/ –f 10 –o 3
```

This command generated filtered proteomes, still divided into main set and reference set, for both bacteria and archaea. These filtered proteomes were moved to their own directories, Bacteria_MEF_CF_10_3 and Archaea_MEF_CF_10_3 for the case of *o*=3 and so on for other values of *o*. Finally, each of these new directories, which contained an FAA and FAA_ref that had been reduced for both mobile elements and also less conserved proteins, was passed to the main SlopeTree script:

```
bash dSTm.sh Bacteria_MEF_CF_10_3/ 20 B  ../Taxonomy/
bash dSTm.sh Archaea_MEF_CF_10_3/ 20 A ../Taxonomy/
```

Similar steps were followed to generate the filtered proteomes for our set of E.coli, using the same set of 30 bacteria in FAA_ref for the more aggressive filtering. In addition, *E.coli* was filtered against itself, i.e. no reference set. All that was required for this self-filtering was to not provide an FAA_ref directory when pFilt.sh was run.

Building Alternative Trees

Trees were built using several other, alignment-free methods: ACS, CVTree, D2, kmacs, Spaced Words, and ALFRED-G. Each method was run on the 495 bacteria and 73 archaea for: a) raw proteomes, b) proteomes filtered of mobile elements, and c) proteomes filtered of mobile elements and also filtered for conservation on *o*=0, 1, 3, 5, and 7. The final pair-wise HGT-correction which was applied to the SlopeTree runs for o=3, 5, and 7 was not applied to these alternative methods because unlike the mobile element filter and conservation filter, the pair-wise HGT correction currently cannot be run independently of SlopeTree. For the matrices produced by these alternative methods, we built trees using rapidnj.

Average Common Substring

Version 1.2 of the ACS code was used to build the ACS trees with the following command:
./ACS –a <path to ACS directory>/ACS_input_file –o distance_matrix.txt –A ACS_matrix.txt

Trees were built using rapidnj on the file written out by the –o option.

Composition Vector Tree (CVTree)

Version 4.2 of CVTree was used. The commands to build the matrices were the following:

```
./cvtree –i cvtree_input_file.txt -d FAA/ -k 6 -t aa -c out/
./batch_dist.pl 1.5 cvtree_input_file.txt out/
out_matrix_k6.txt
```

D2 Method

Version 1.0 of D2 was used. The command to build the matrices was the following:

```
java -Xmx126g -jar jD2Stat_1.0.jar -a aa -i input.faa -o
matrix
```

kmacs

We ran kmacs with k=14:

```
./kmacs input.faa 14
```

Spaced Words

We ran Spaced Words with k=12 and Euclidean distances. Evolutionary distances were not available for amino acid sequences:

```
./spaced –k 12 –d EU input_file.faa
```

ALFRED-G

We ran ALFRED-G with k=6 and x=1.

```
build/alfred.x -f input.fas -o output.txt -k 6 -x 1
```

Comparing Trees

All trees were compared to the Eisen-trees using the treedist tool from PHYLIP (117) for the symmetric difference distance. Using a keys file generated for the purpose of finding matches between the original FASTA archive and the Eisen-trees, we renamed the nodes of

the Eisen-trees and alignment-free trees so that they were identical and renamed the two tree files intree and intree2 for treedist.

A)

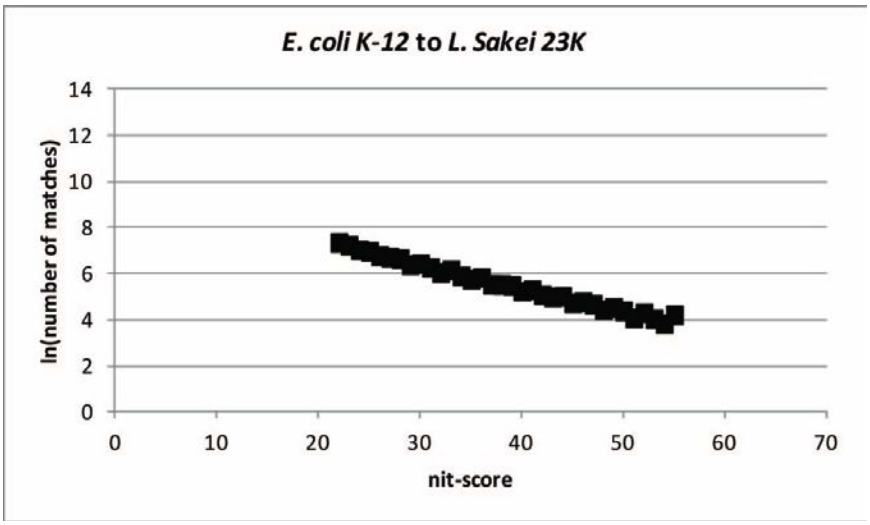
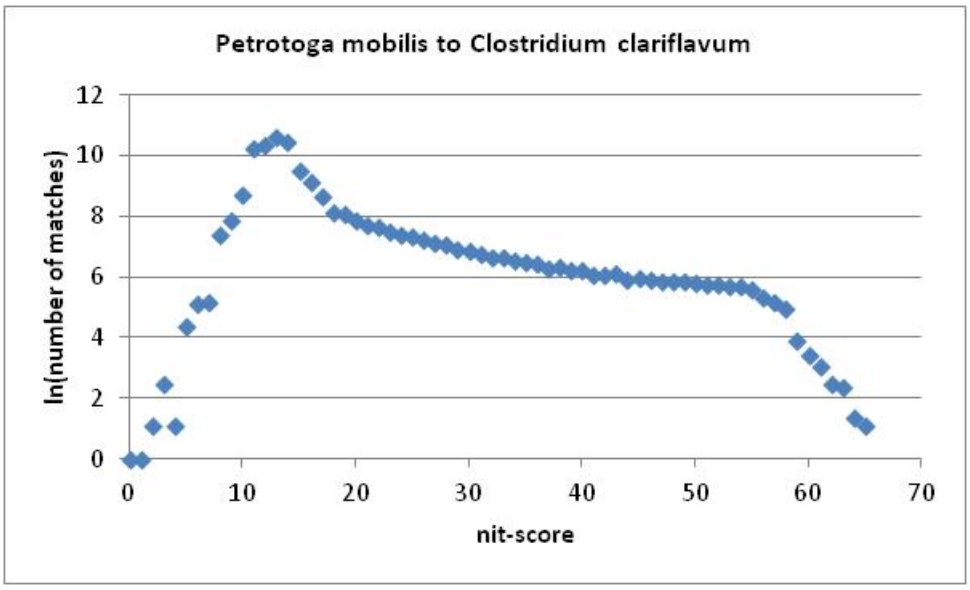


Figure 3-1. Extracted evolutionary signal from a SlopeTree plot.

A) Same data as in Figure 1-9, but with the background of coincidental matches subtracted and deleted, and the noisier right end sheared by the bounds selection criteria.

A)



B)

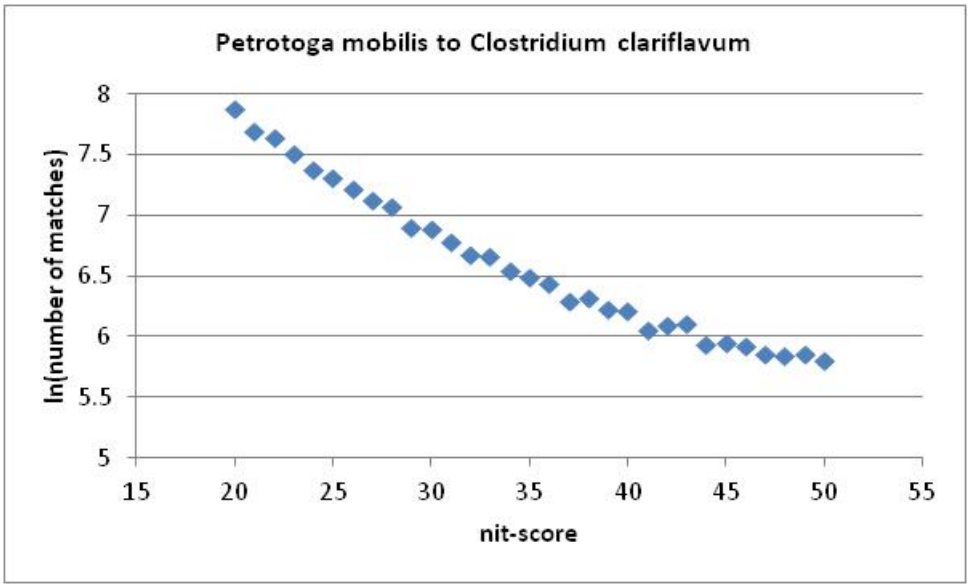


Figure 3-2. SlopeTree plot for HGT instance.

Petrotoga mobilis (Thermotogae) to *Clostridium clariflavum* (Firmicutes). A) SlopeTree plot over the whole range of nit scores. B) Phylogenetic signal extracted from the plot in (A).

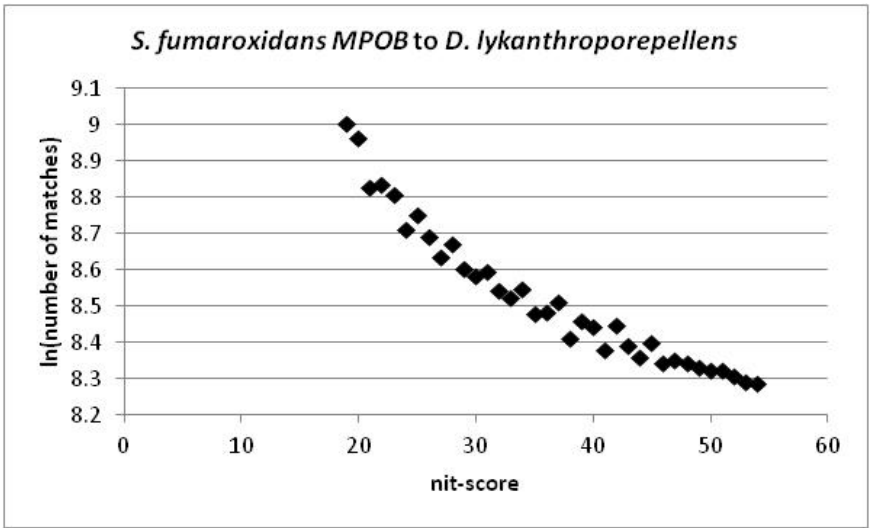
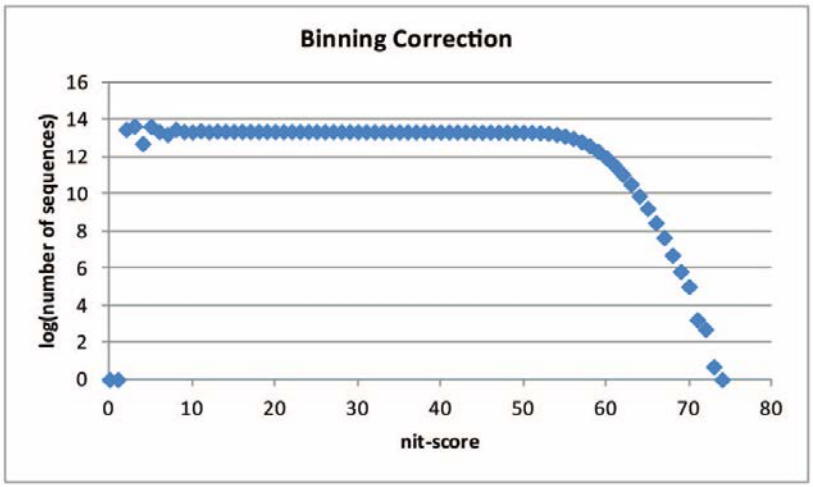


Figure 3-3. SlopeTree plot for pair sharing a transfer from a single copy phage.

SlopeTree plot for *Syntrophobacter fumaroxidans* and *Dehalogenimonas lykanthroporepellens*.

A.



B.

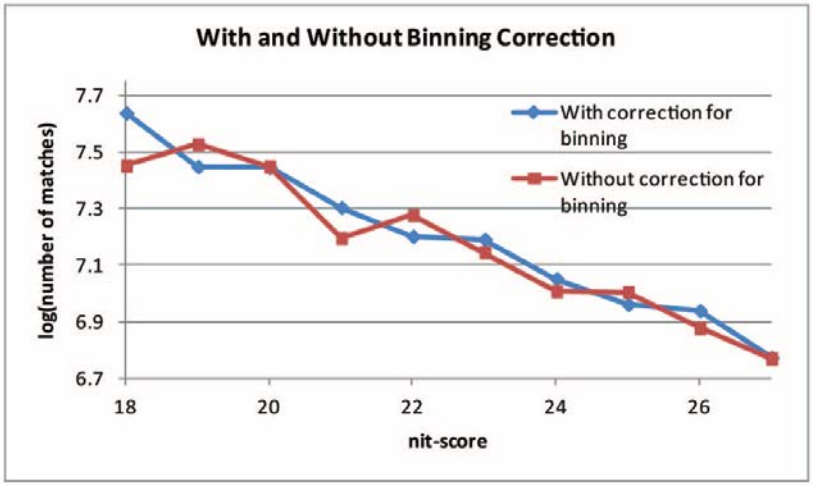


Figure 3-4. Binning artifacts

A) A SlopeTree-style plot for a pair of organisms, including *all* length 1 to 20 sequences in both proteomes, regardless of whether or not they were matches. B) Data with and without the binning correction, zoomed into the relevant range.

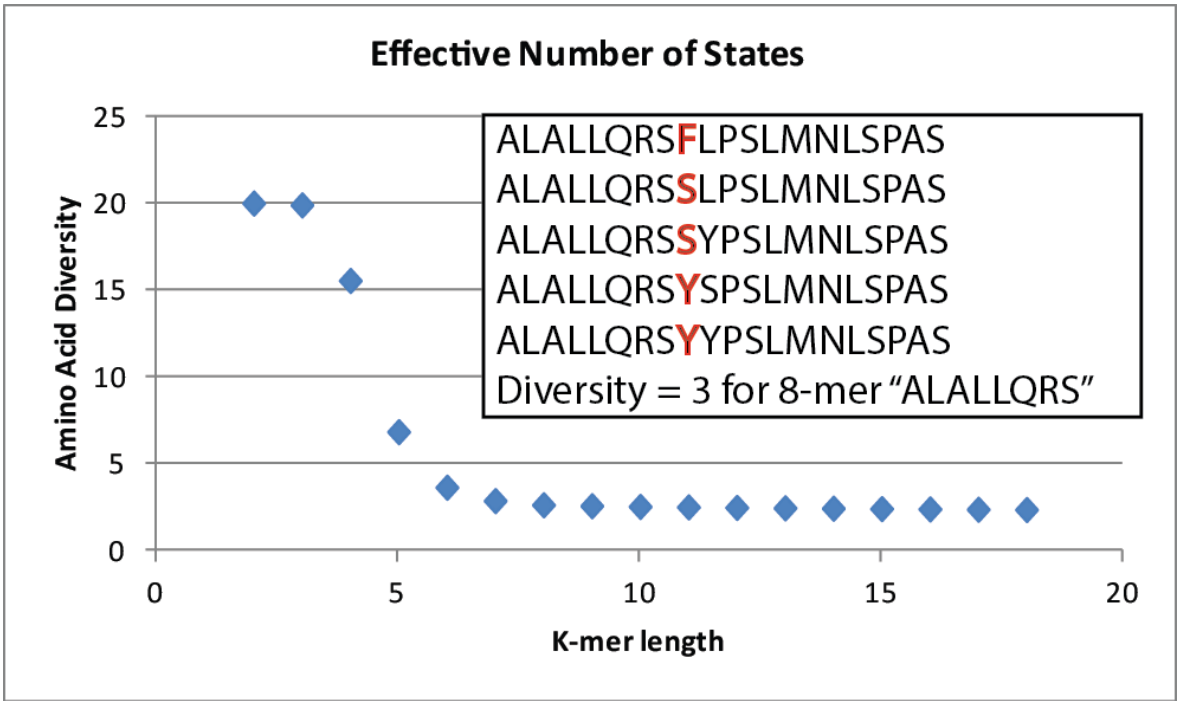


Figure 3-5. Calculating the effective amino acid population size.

For a large bacterial input, blocks of exact matches were identified for every possible length. A separate run was performed for each length. The above sequences are an example from a run on 8-mers; for the leading block 'ALALLQRS', the diversity in column 9, assuming would be 3 (repeating amino acids not counted).

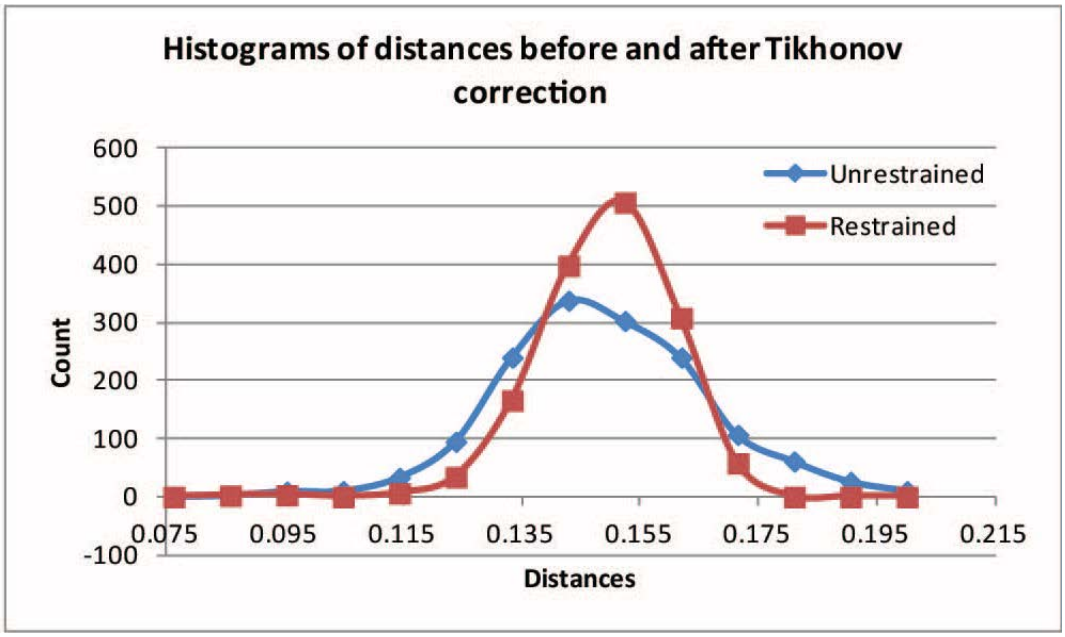


Figure 3-6. Positive restraint on SlopeTree distances.

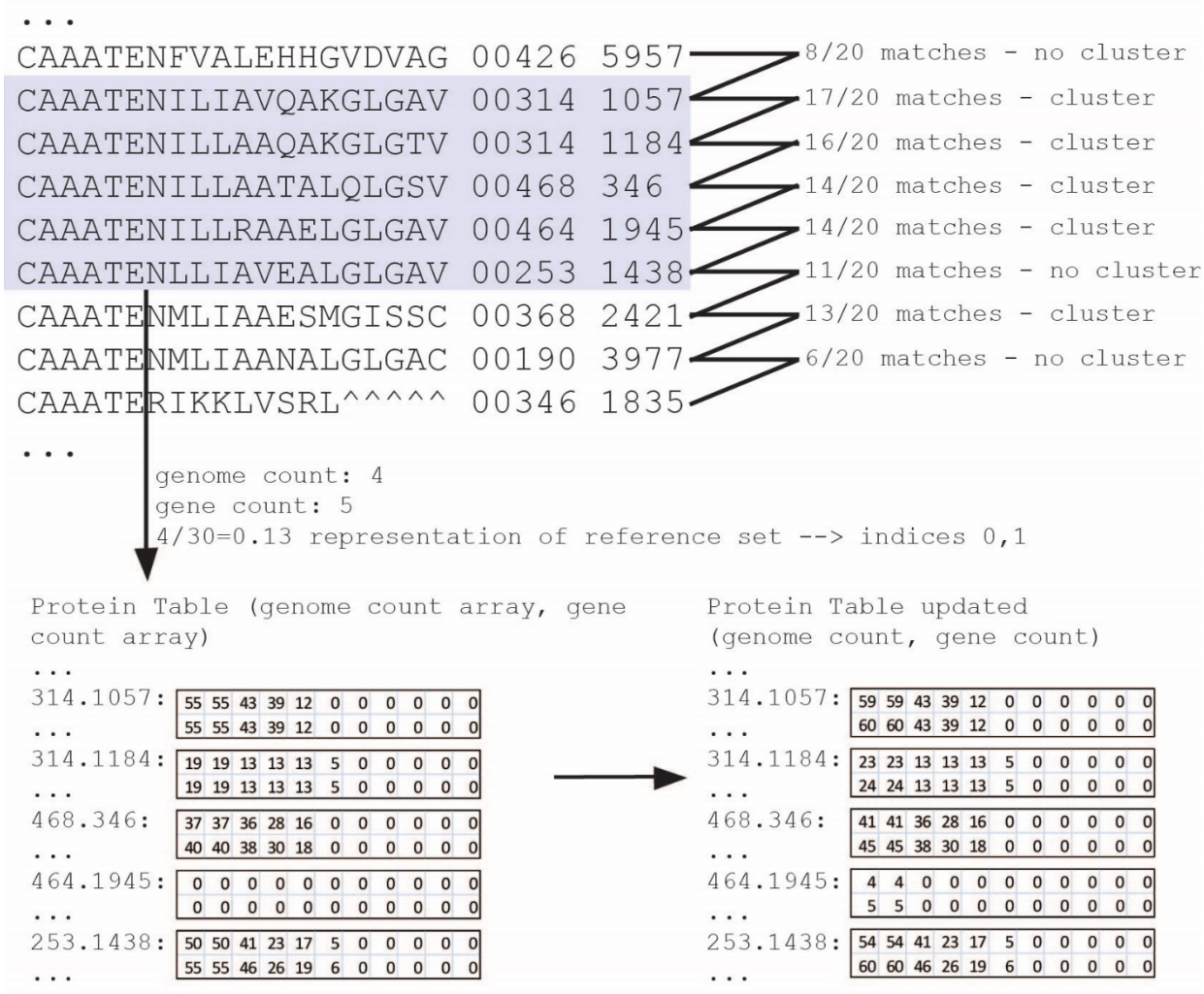


Figure 3-7. Conserved protein identification.

Highlighted block represents a cluster in the final, sorted k-mer list. Values on the right show what is meant by allowed number of matches to mismatches for sequences to cluster. Protein table on the left is the table before the update from the highlighted block, where the upper rows in the table correspond to G_{ij} and F_{ij} as described in the algorithm for stability and conservation filtering. Protein table on the right is equal to the table on the left after being updated by the cluster highlighted above.

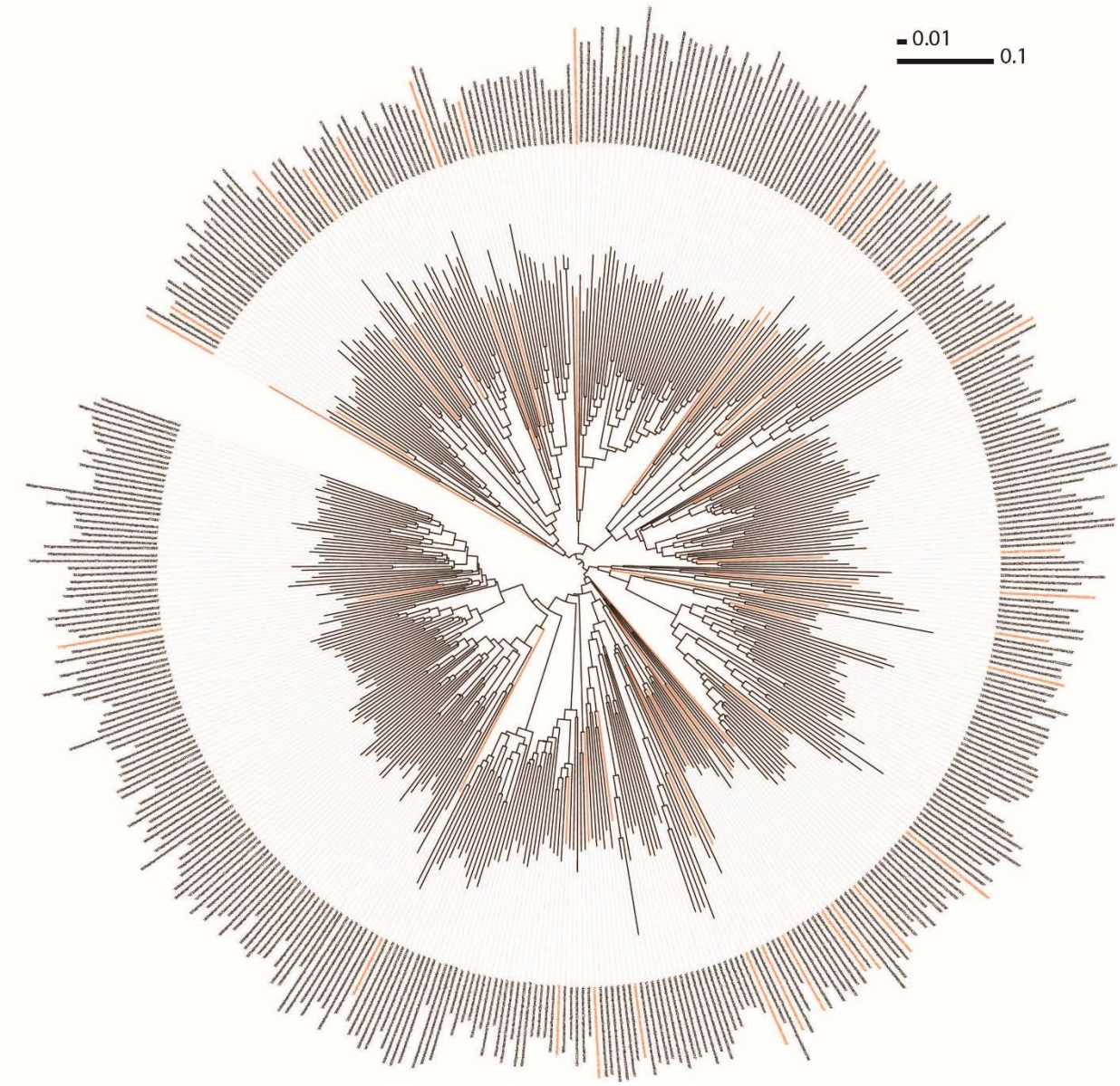


Figure 3-8. Bacterial reference set.

Reference set of 30 bacteria, mapped onto the raw tree of 495 bacteria.

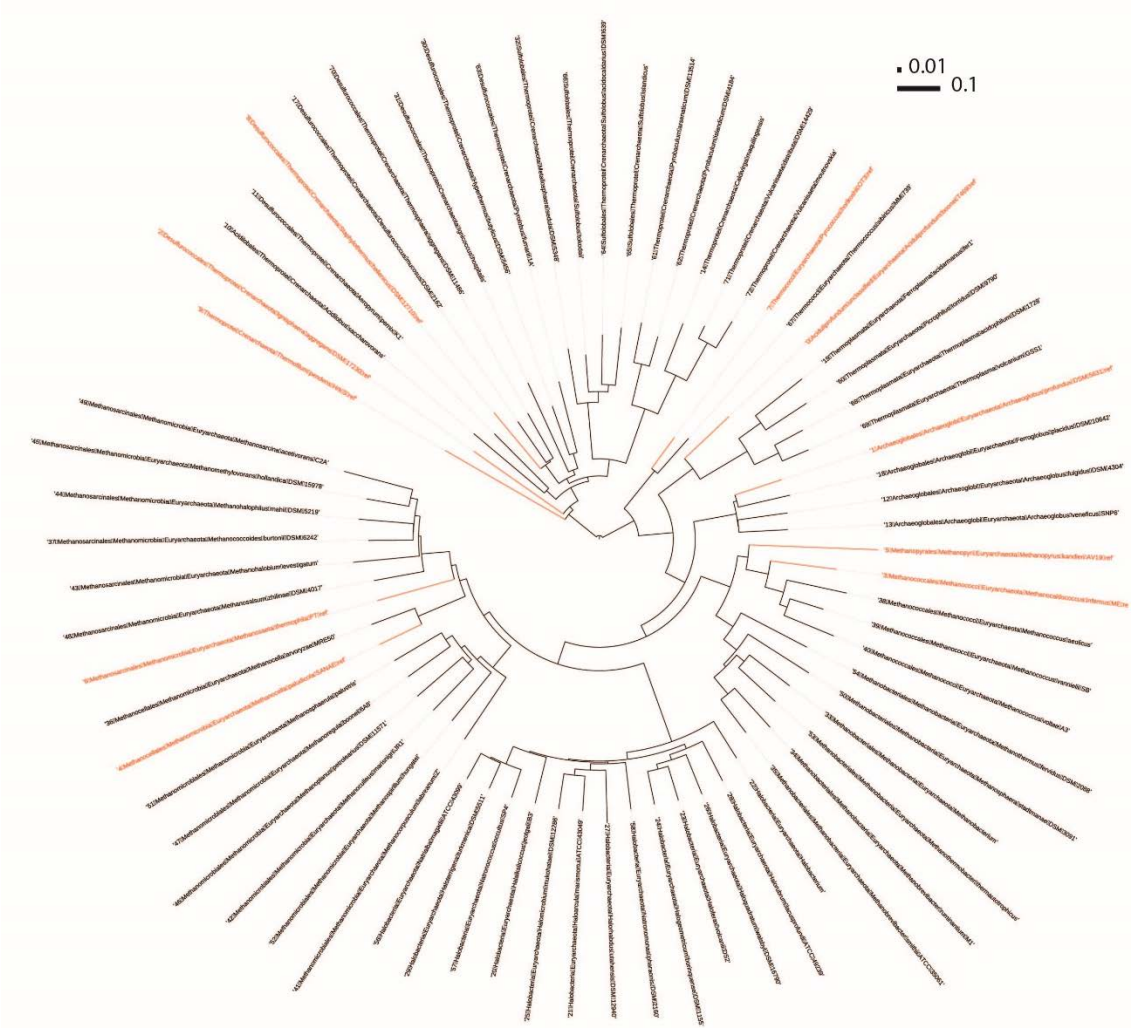
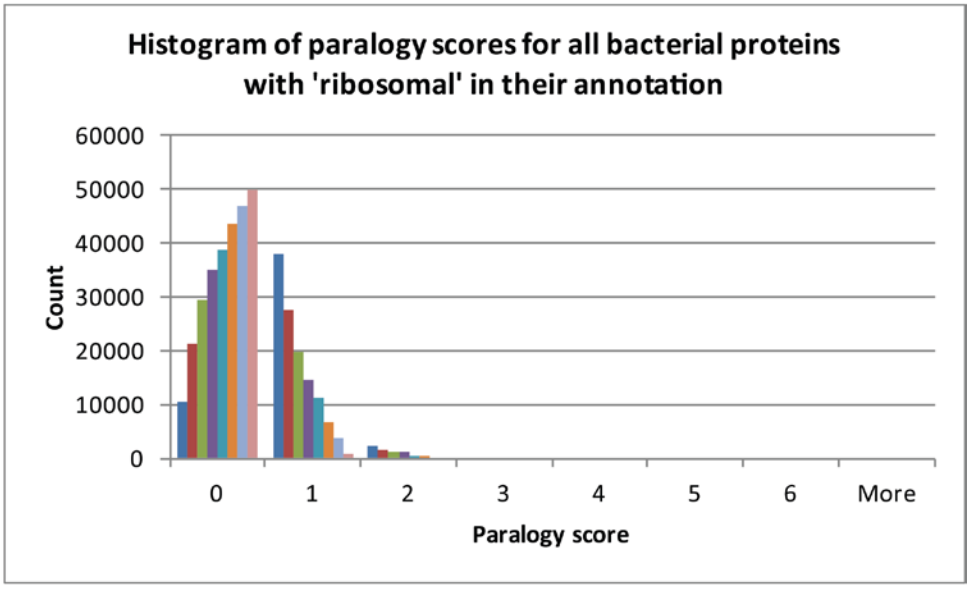


Figure 3-9. Archaeal reference set.

Reference set of 10 archaea, mapped onto the tree of 73 archaea.

A)



B)

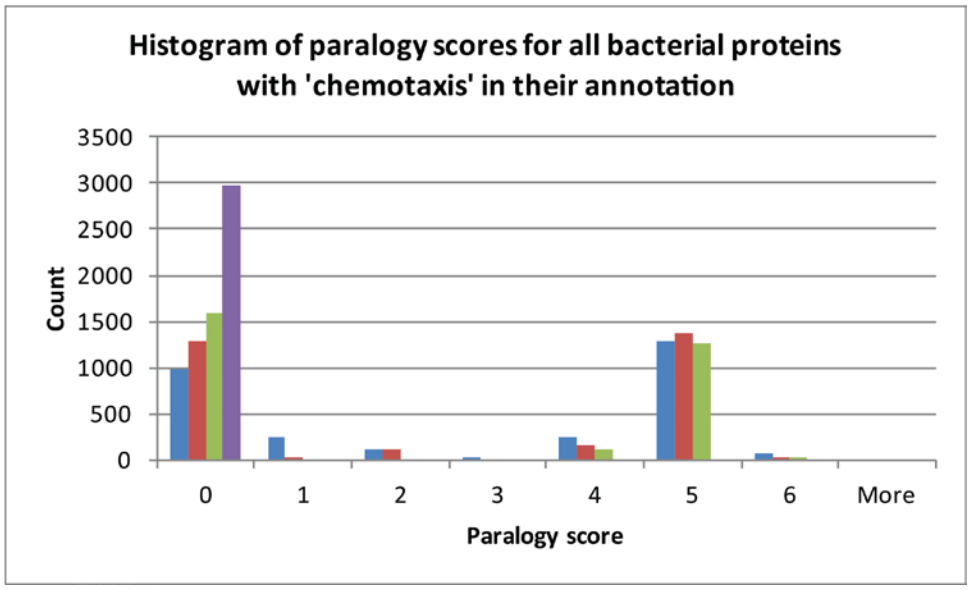


Figure 3-10. Paralogy score histograms over different values of conservation/stability filtering parameter *o*.

A) Histogram for all proteins with 'ribosomal' in their annotation, i.e. an example of paralogy scores for a highly conserved protein. B) Histogram for all proteins with 'chemotaxis' in their annotation, i.e. an example of paralogy scores for a non-conserved, frequently transferred protein.

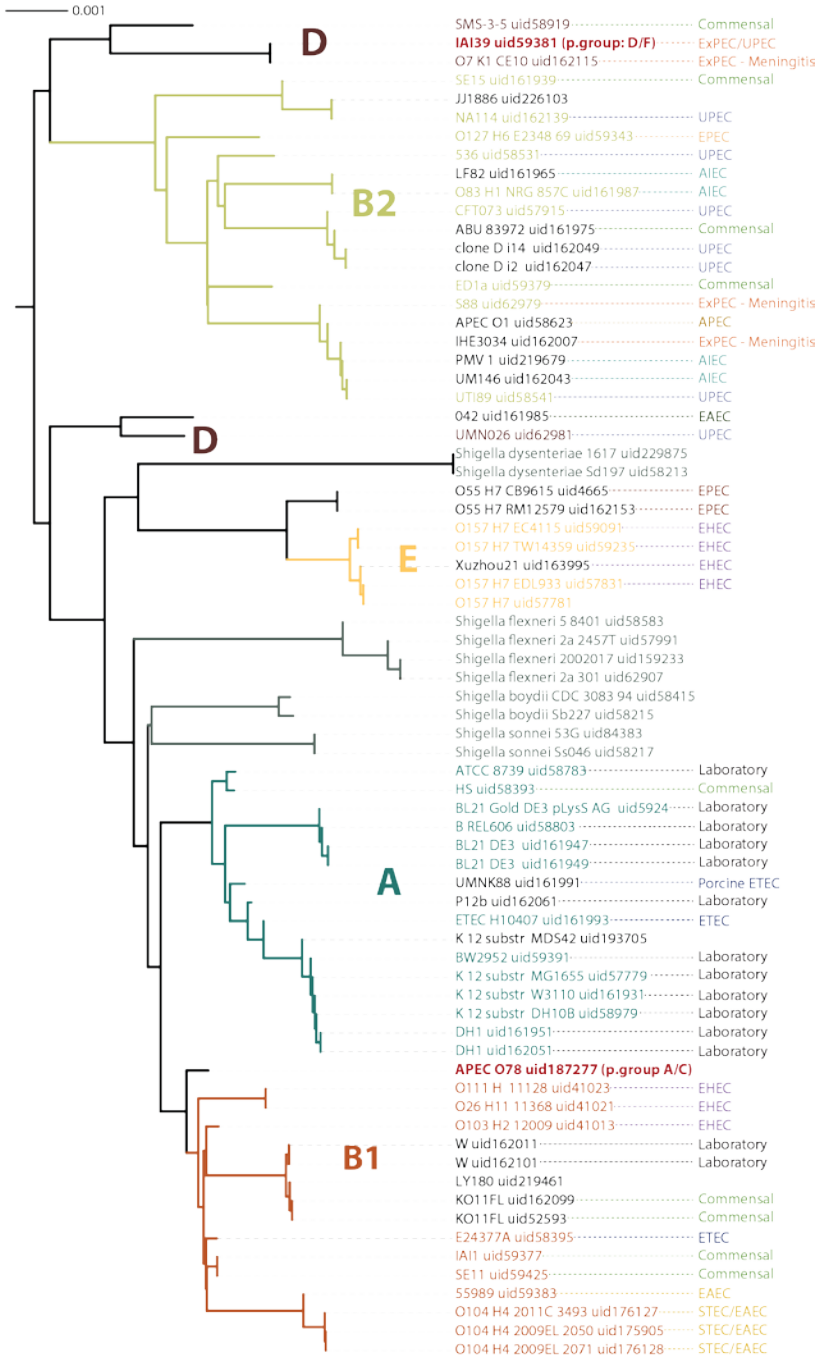


Figure 3-11. SlopeTree tree of 72 *Escherichia coli* and *Shigella* using 20-mers.

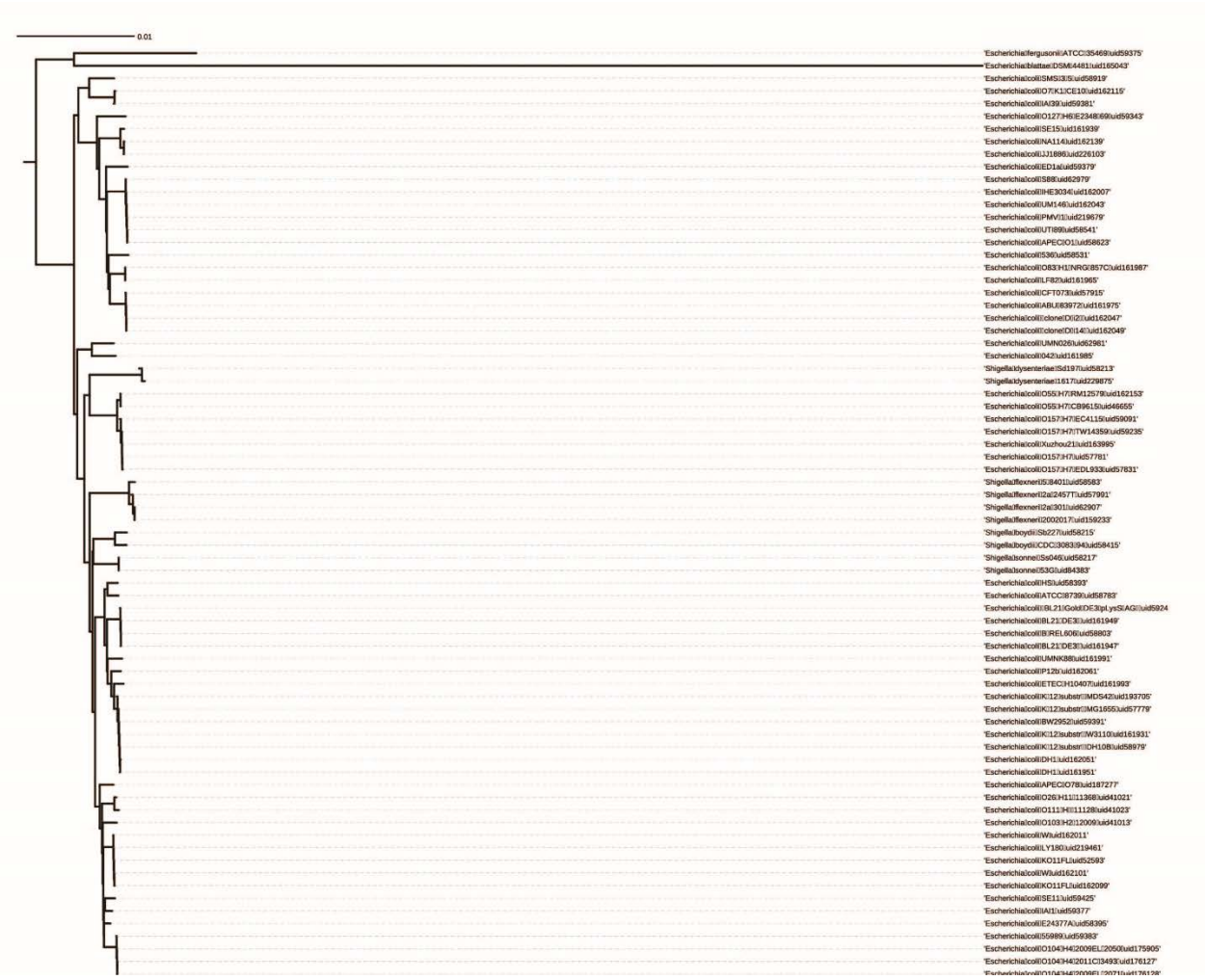


Figure 3-12. Outgroups used in strain level SlopeTree tree.

20-mer SlopeTree tree including 2 outgroups: *Escherichia fergusonii and Escherichia blattae.*

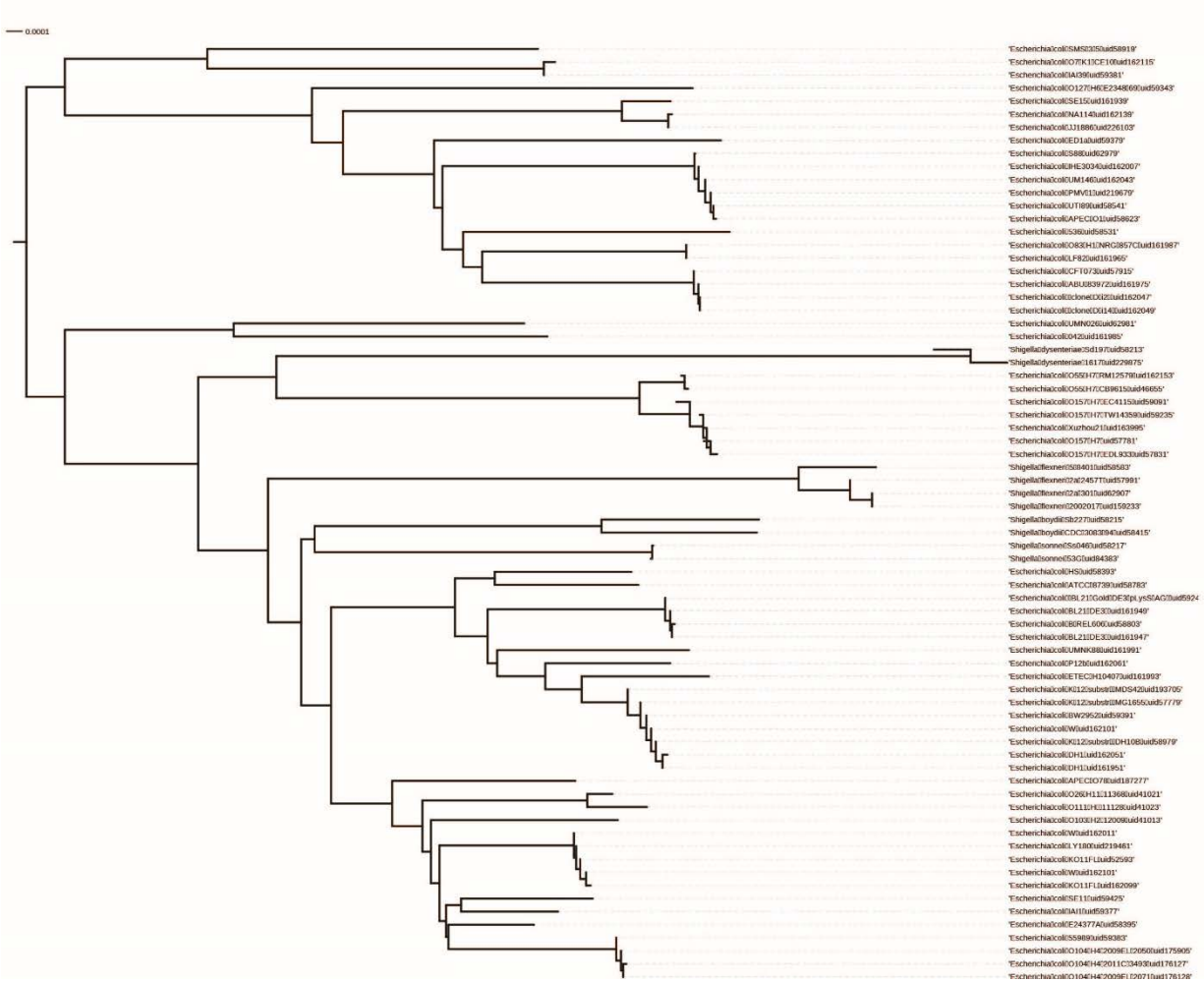


Figure 3-13. 40-mer tree of *Escherichia coli/Shigella*.

Topology almost identical to the 20-mer tree over the same set. (2 outgroups removed).

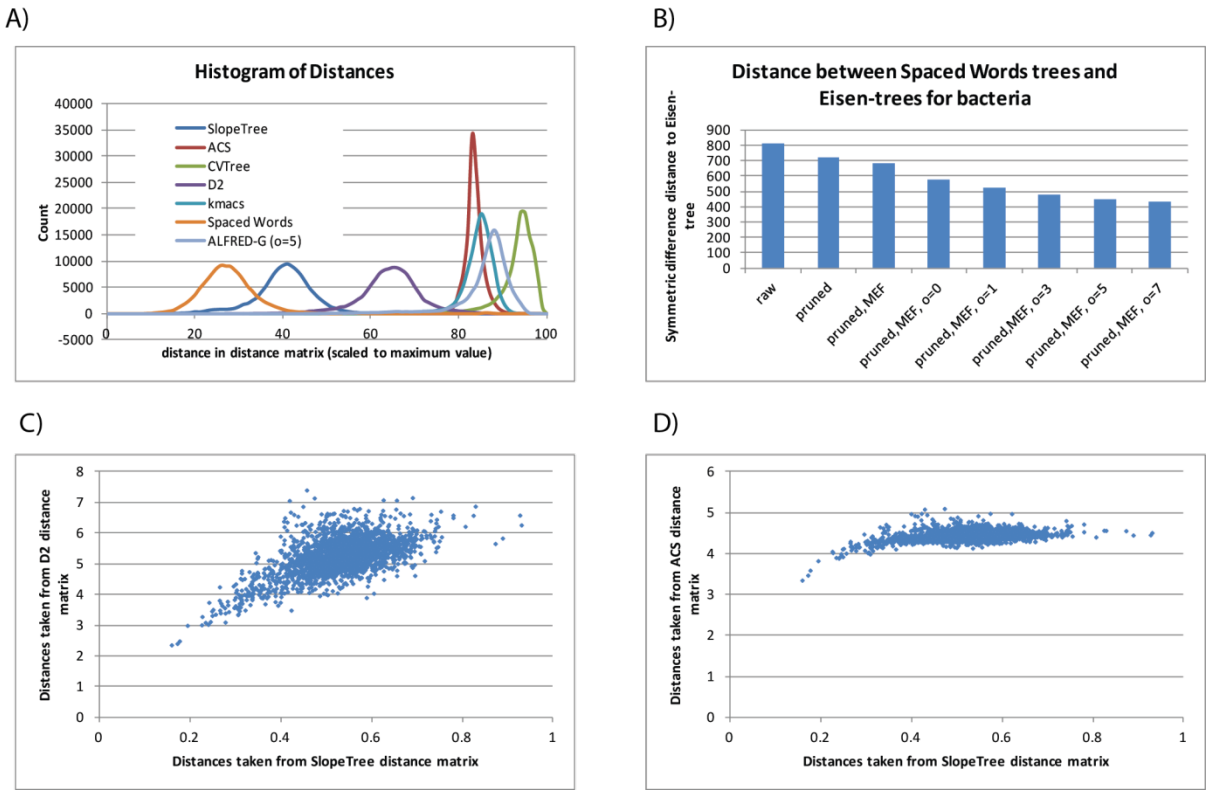


Figure 3-14. SlopeTree and other alignment-free methods.

A) Histogram of scaled distances produced by each method. B) Decrease in symmetric difference distance to the Eisen-495 tree for Spaced Words method. C) SlopeTree distances to D2 distances for a matching set of randomly selected organism pairs. D) SlopeTree distances to CVTree distances for a matching set of randomly selected organism pairs.

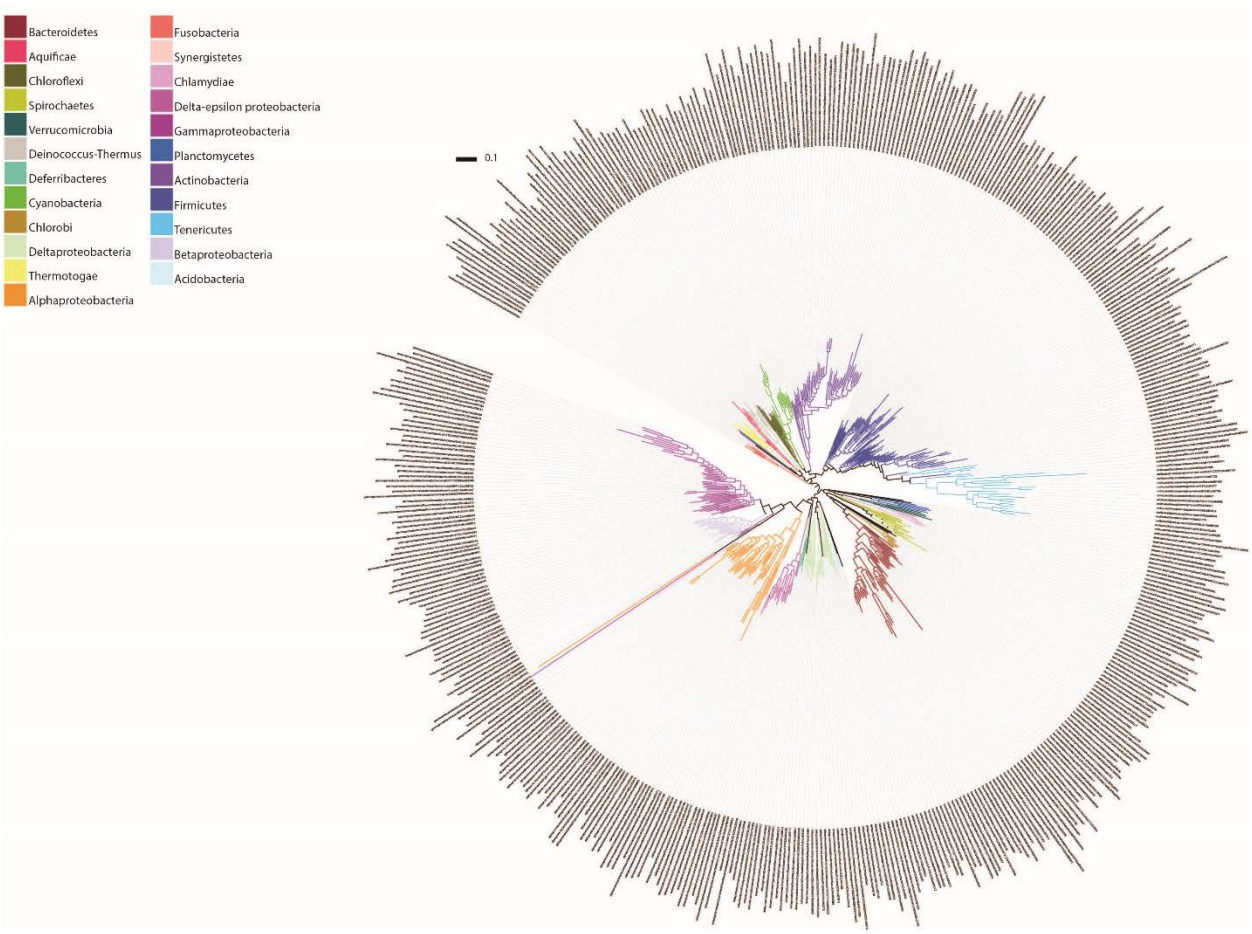


Figure 3-15. Eisen-495 trees for bacteria.

Maximum likelihood tree from Lang et al. (28), reduced to only bacteria and pruned to match the bacteria with whole proteomes in the NCBI database.

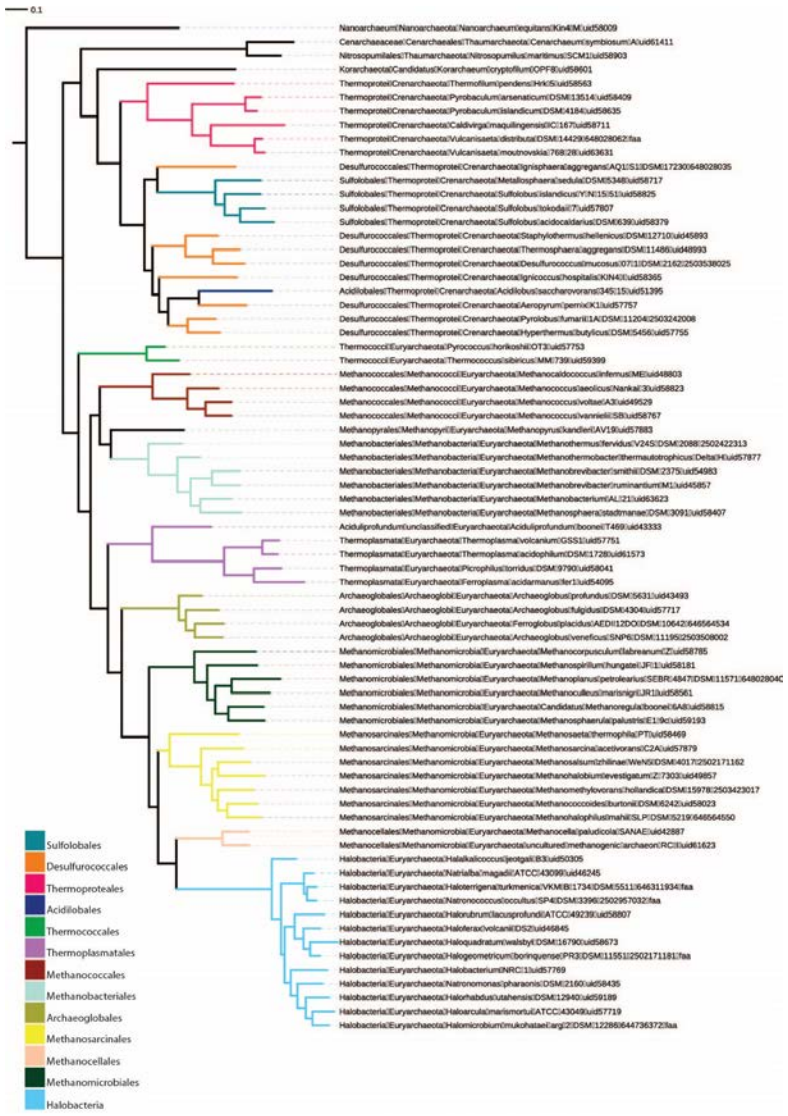


Figure 3-16. Eisen-73 trees for archaea.

Maximum likelihood tree from Lang et al. (28), reduced to only archaea and pruned to match the archaea with whole proteomes in the NCBI database.

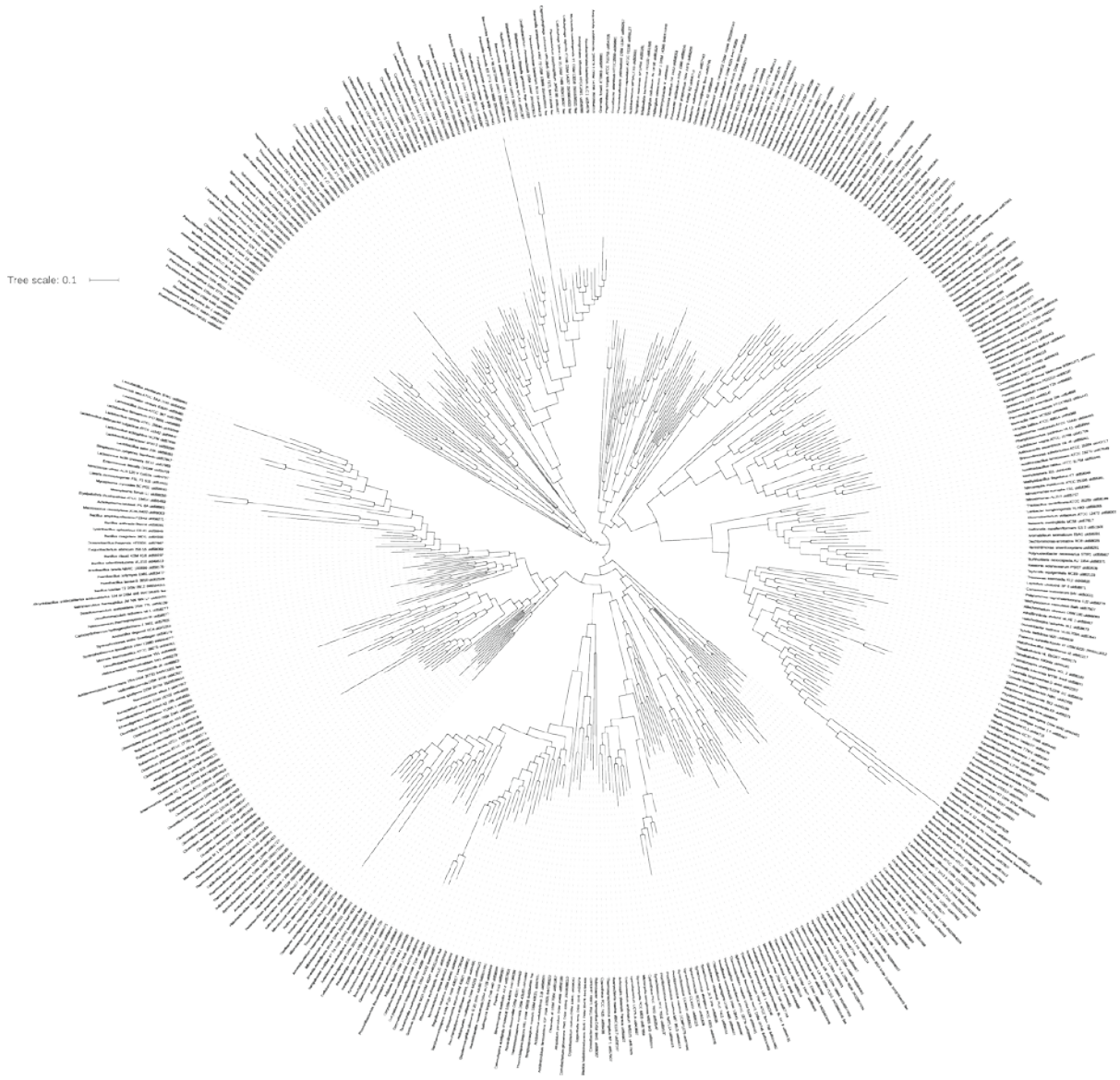


Figure 3-17. Eisen-445 trees for archaea.

Maximum likelihood tree from Lang et al. (28), reduced to only bacteria, pruned to match the bacteria with whole proteomes in the NCBI database, and then additionally pruned of organisms identified as SlopeTree as being problematic.

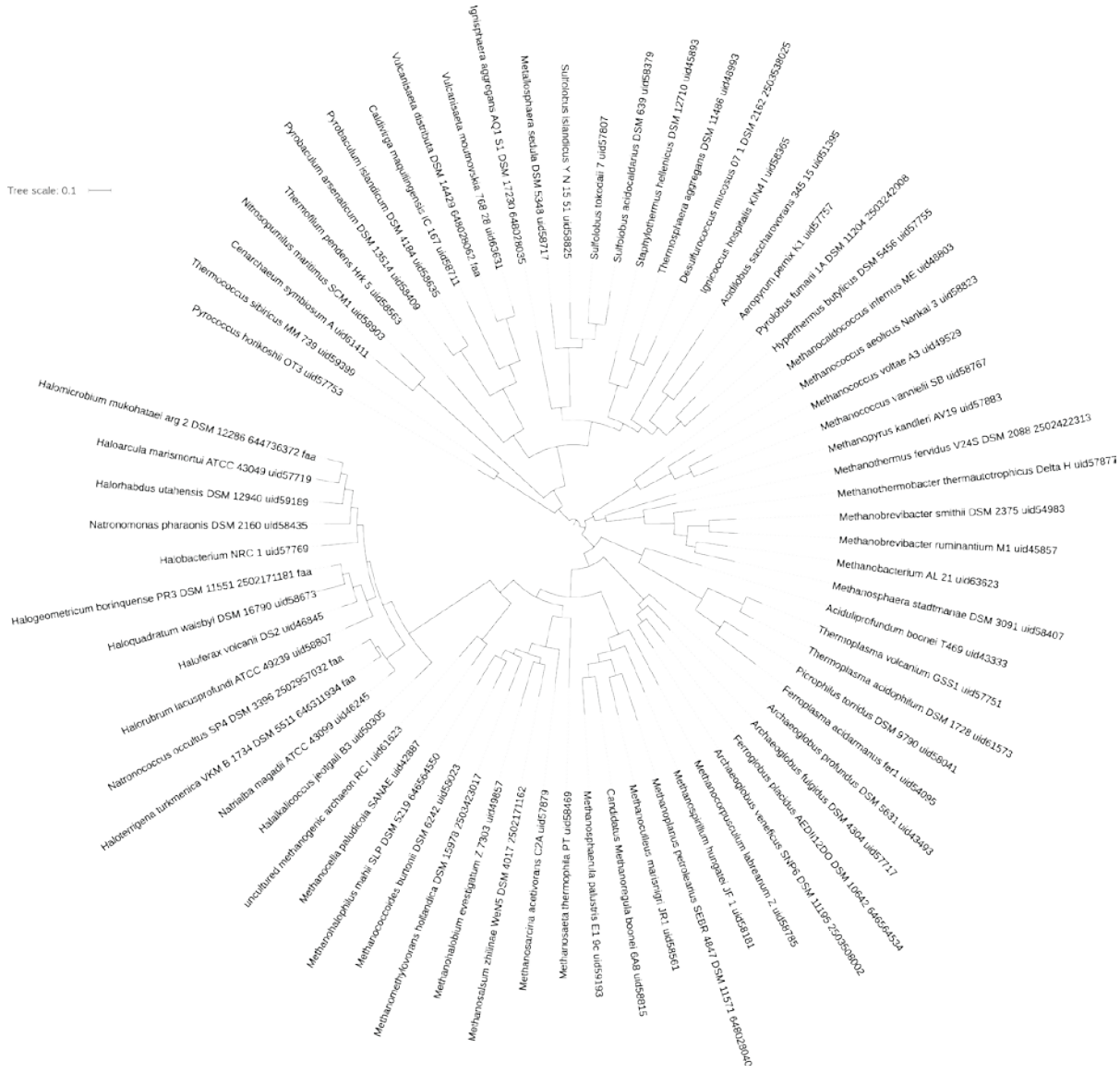


Figure 3-18. Eisen-71 trees for archaea.

Maximum likelihood tree from Lang et al. (28), reduced to only archaea, pruned to match the archaea with whole proteomes in the NCBI database, and then additionally pruned of organisms identified as SlopeTree as being problematic.

CHAPTER FOUR

CORRECTING FOR SINGLE COPY PHAGES

4.1 INTRODUCTION AND MOTIVATION

If the perturbations to the assumptions are small, then SlopeTree and other consensus methods perform acceptably. However, between some pairs of bacteria, I observed large HGT-coordinated contributions coming from phages. The heuristics from the previous chapter worked well for self-selecting elements that do not prevent the accumulations of additional copies. However, some phages have evolved the mechanisms to avoid the insertion of additional copies into the genome. I found instances of such phages that were capable of moving between different phyla. These phages can provide a substantial number of sequences that are quite similar and that have nothing to do with evolution by descent. Although such instances are infrequent, they can cause phylum-level misplacements of organisms in the final phylogenetic trees, and so need to be addressed.

In this chapter, I discuss an additional, separate correction for horizontal gene transfer (HGT) (Algorithm 4) SlopeTree provides, which identifies specific pairs of organisms that appear to have transferred genes and re-calculates the distance using the main SlopeTree routine, with the suspicious proteins removed from the data. This correction is not expected

to be effective for extremely ancient transfers, but is adequate for recent transfers such as those involving phage proteins.

4.2 AUTOMATIC IDENTIFICATION AND CORRECTION FOR SPECIFIC TYPES OF HORIZONTAL GENE TRANSFER

Between the quadratic fit, the mobile-element filter, and the conservation and stability filter, SlopeTree (ST) trees were brought much closer to the Eisen-trees and the number of organisms whose placement contracted the NCBI classification was greatly reduced. However, there remained a small set of organisms exhibiting such large-scale horizontal transfer (HGT) that only extremely aggressive conservation filtering could correct their placement on the tree. An instance of such a set of organisms was *Dehalogenimonas lykanthroporepellens*, a Chloroflexi, with two Gammaproteobacteria, *Desulfarculus baarsii* and *Syntrophobacter fumaroxidans*. The conservation filter, for a stringent enough setting, did separate this group. However, one strength of alignment-free, whole-genome methods is that they use all the data in the genome or proteome and so may be a closer approximation of organismal evolution than trees based on single genes or even groups of genes. Although not all of the information in a genome or proteome reflects vertical descent and eliminating proteins that contribute a spurious signal can benefit topologies (as demonstrated in Chapter 3), at the same time, I wanted to avoid any extreme reductions of my input data. Reducing proteomes to a tenth or less of their original size, throwing away thousands of proteins in the process (many of which reflect vertical descent), in order to remove a relatively small number of horizontally transferred proteins was not an acceptable solution. One example of

why such extreme reductions are problematic is that typically, only very specific proteins will be remain. Such a final set would be certainly enriched with proteins interacting with the ribosome, making the final trees once again less about organismal evolution and more about the evolution of the ribosome. One of the original goals of the SlopeTree project was to escape this exact problem.

4.3 IMPLEMENTATION

I tried several approaches before finding one that addressed the single copy phage transfers without requiring an extreme reduction in the input data. First I implemented a new fit, described below, which ultimately proved unstable and could not be used to generate distances, although the code still uses it to identify suspicious pairs that may have horizontally exchanged material. Then I implemented the pair-wise HGT correction, which is currently a part of the SlopeTree methodology.

Implementing a new fit: a sum of two exponentials

I first implemented a new fit for the data, this time a sum of two exponentials. The equations used to calculate this fit are available in Appendix D. The maximum possible slope for SlopeTree is 0.3. The minimum possible slope is 0. An array, or grid, was initialized for all possible combinations of slopes within and including these bounds, with 0.01 being the starting difference between adjacent elements (i.e. first elements of the first row would be [0,0] [0, 0.01] [0,0.2], etc. and the last element of the last row would be [0.3, 0.3]). The data was fitted using each element in the grid and the best fit was selected. Then a new grid was

initialized around this best fit point, with the parameter for the difference between elements parameter (0.01 as the starting value) being used to set the new bound. For instance, if the selected point was [0.23, 0.05] after the first iteration through the grid, then the new grid's bounds would be [0.22, 0.04] and [0.24, 0.06] using the initial value of 0.01 and the difference between adjacent elements was set to the original value divided by 10. This was repeated until the fit did not improve between the best value from the previous grid and the best value from the new grid.

Problems with the fit

From the beginning, the new fit did not produce better trees. This was almost immediately apparent. I generated a table called fits_vector for a large number of statistics using all three fits for each plot. These statistics included: from the linear fit, the slope and the weighted rmsd; from the quadratic fit, the value of *a, b,* and the weighted rmsd; and from the fit from the sum of the two exponentials, the value of *b1, b2, F, G,* the weighted rmsd, and the match score cutoff (described below). After some manipulations, I found that the ratio of the weighted rmsd from the quadratic fit to the weighted rmsd from the fit from the sum of the two exponentials initially seemed promising as a criterion to use for identifying suspicious pairs for HGT. In addition, I generated a value, called match score cutoff, which identified the nit-score at which the second exponential began to dominate the first by a factor of 10.

$$MSC = \frac{\log\left(\frac{0.1F}{G}\right)}{b1 - b2} + xvals[0]$$

where *xvals[0]* was the lowest nit-score in the range of data being fitted. Only matches scoring equal to or greater than the match score cutoff, which was typically in the range of 40-55, were assessed (in subsequent procedures, described below) as possibly coming from horizontally transferred proteins.

Eventually, I observed that the statistics in fits_vector for the same pairs appeared to change from one run to the next. I ran SlopeTree twice on the same data and plotted the match score cutoffs from the one run against the other. These values should have been identical, and plotting them against one another should have resulted in a perfectly straight diagonal. Instead, I observed a plot as the one shown in 4-1B. After several runs of SlopeTree, I found that the source of the inconsistency was in the randomized generation of the background. When I ran SlopeTree from the beginning on the same input set, it generated a plot as in Figure 4-1A. However, if I substituted the set of randomly generated k-mers from one run to another, but otherwise ran SlopeTree from the beginning, I got identical results (Figure 4-1B). The new fit was so unstable that even the very small differences in the plots due to subtracting slightly different backgrounds were enough for the fit to produce different values. Therefore, in the end I used the new fit only in part of the process of identifying organism pairs that appeared likely to have exchanged genetic material.

The current implementation of SlopeTree uses three fits for the data, for various calculations. However, because the linear fit was too sensitive to horizontal gene transfer and the sum of the two exponentials was unstable, the genomic distances used for the trees come from the quadratic fit.

4.4 CORRECTING FOR HGT EXPLICITLY

Because the new fit did not fix the problem of large-scale HGT, I decided to implement a SlopeTree module that (1) identified all organism pairs that exhibited signs of HGT and (2) explicitly identified and then removed the transferred proteins.

Flagging organism pairs exhibiting signs of HGT

First the pair-wise HGT correction identifies pairs with signs of HGT. Pairs in which the double exponential weighted RMSD (*x)* produces a better fit than the quadratic fit weighted RMSD (*y)* are flagged for the correction (default cutoff: $x/y < 0.9$). A shallow slope (i.e. indicating evolutionary closeness) but a high RMSD for the linear fit (default: RMSD>0.12; slope<0.06) also cause a pair to be flagged, because the RMSD is typically very low for slopes from truly close organisms.

These criteria for flagging pairs are highly imperfect and several other cutoffs and rules were applied before them with even less success. The biggest problem was in a very large number of false positives, which for even small inputs could range in the thousands if the criteria were too loose.

Two passes through the main SlopeTree match-counting algorithm

For each flagged pair, two iterations through the SlopeTree match-counting code are performed. First, k-mers from a flagged pair are passed through the match-counting code

alongside a diverse, pre-selected reference set. This typically was the same reference set selected for the conservation filtering described in chapter 3. During this match-counting run, two tables of integers, each with an entry corresponding to each protein in each of the two flagged organisms, are held in memory, one corresponding to matches between proteins from either member of the flagged pair, and the other corresponding to matches between proteins from either member of the pair and proteins in the reference set. A conserved protein is expected to have many matches with the reference set. Entries are incremented for all matching k-mers of a given length or longer (default=12 or more amino acids). Previously, the match score cutoff was used to identify k-mers of interest. At the end of the match-counting, these tables are compared; proteins shared by the pair that are not present in a certain number (default=3) of reference set organisms are flagged. The pair, without the reference set, is then passed through the match-counting code once more, with all flagged proteins excluded.

I now describe this process formally.

Algorithm 4: Pair-Wise Horizontal Gene Transfer (HGT) Correction

Input: A previously calculated SlopeTree distance matrix D (defined in Algorithm 3), a list Q of proteome pairs flagged as requiring additional correction, and a set R of proteomes, with R taken from taxonomically diverse organisms.

Output: A new distance matrix D` identical to D except for the distances between all pairs in Q, which have been recalculated.

Algorithm: Let p_{ij} be the j^{th} protein in R_i, and let $p_k^{ij}[h]$ be a k-mer from p_{ij} of length k, starting at index h, where $0 \leq h < f$ given that p_{ij} has length f. For those k-mers at the end of each protein where $h+k>f$, the suffix is expanded by the necessary number of empty characters to fill the remainder of the k-mer. Each k-mer is stored as a 3-tuple consisting of the k-mer, the proteome ID, and the gene ID. Let S be the alphabetically sorted list of all 3-tuples from R.

Let v and w be a pair in Q. Then for this pair, we compile an alphabetically sorted list of 3-tuples and call this list P. Let S and P be merged and this list passed to Algorithm 3, i.e. the SlopeTree Main Algorithm for counting matches. During the match-counting, let any protein p_{ij} contributing a match between v and w with a nit-score (proportional to the length of the match, described in Implementation) higher than some cutoff x, and with fewer than y hits among the reference set, be marked. Having reached the end of the merged list of S and P, and having marked all proteins from v and w, we rerun Algorithm 3 on P, but ignoring matches from the marked proteins, to produce a new distance, $D`_{vw}$.

Let the original distance D_{vw} be replaced by the new distance $D`_{vw}$, and the matrix $D`$ be the matrix in which every element has been updated in this way for all pairs in Q.

Computational complexity: Compiling the alphabetically sorted list S takes $O(r \log r)$ time, where r is the total number of amino acids in R. Similarly, compiling P takes $O(p \log p)$ time, where p is the total number of amino acids in v and w. Each first iteration of the SlopeTree main algorithm then requires $O(r \log r + p \log p)$ time, and running the pair requires $O(p \log p)$ time. This must be repeated for every pair in Q. For a total of n organisms, i.e. a distance matrix to recalculate that is n by n, the worst case scenario is that

every pair has been flagged, requiring that $n^2/2$ distances be recalculated, but in practice, and especially after having applied the filters described in Algorithms 1 and 2, the number of pairs in Q is much smaller.

Examples of HGT, identified by the SlopeTree HGT correction

I observed two main classes of HGT for the pair-wise HGT correction. The first was associated with single copy phages. *D. lykanthroporepellens* and both *Syntrophobacter fumaroxidans* and *Desulfarculus baarsi* serve as an example of this. The second was related to adaptation-associated proteins. *Petrotoga mobilis* and *Mahella australiensis,* which shared a transfer of proteins associated with resistance to a toxic environment, are an example (Figure 4-2). While the misplacement of the latter pair was addressed by the previous corrections, the HGT correction was also able to fix the misplacement, with the additional advantage that it explicitly identifies the proteins likely to be contributing to the spurious signal. For the two pairs listed above, the proteins identified by the HGT correction are available in Appendix E. The ME filter, conservation filter and pair-wise HGT correction are separate modules in SlopeTree that are applied at different times and address slightly different issues in the data. However, they partially overlap in the proteins that they remove (Figure 4-2); for instance, the conservation filter removes many proteins that the HGT filter would remove, were the conservation filter not applied, and vice versa.

4.5 FINAL RESULTS ACROSS ALL CORRECTIONS AND FILTERS

I built ST-trees on "raw" (i.e. no filtering) proteomes, proteomes filtered of mobile elements, proteomes filtered of mobile elements and also non-conserved, unstable proteins, and finally filtered proteomes passed through the additional HGT-correction. Most of these trees were pruned of organisms flagged by SlopeTree as problematic, e.g. reduced organisms. For comparison purposes, I calculated symmetric difference (SD) (131) distances between all ST-trees and the supermatrix trees (28), which we call Eisen-495 (bacteria) and Eisen-73 (archaea), and Eisen-445 and Eisen-71 for their pruned counterparts (Eisen-trees introduced in Chapter 3). I also calculated the distances to the Eisen-trees for trees built using other alignment-free methods, namely Average Common Substring (ACS), CVTree, D2, kmacs, and Spaced Words and ALFRED-G. These alternative methods were given both raw data and also a variety of filtered inputs.

SlopeTree applied to 73 archaea

A series of ST-trees was constructed for 73 archaea (Figure 4-3). These 73 were all the archaea in Lang et al. (28) that had available proteomes in NCBI. Two archaea were pruned from the distance matrix prior to building the trees: *Candidatus Korarchaeum cryptofilum OPF8 uid58601,* and *Nanoarchaeum equitans Kin4 M uid58009.* Both were automatically flagged by SlopeTree for having an unusually low number of conserved genes compared to the rest of the set. As with the strain-level analysis, we generated both unfiltered ST-trees and also filtered ST-trees, and also applied our pair-wise HGT correction. These trees were compared to the Eisen-73 and Eisen-71 trees. Differences in filtering parameters produced some changes in topology, with distances to the Eisen-73 tree generally decreasing as

filtering increased. For instance, without filtering (but with pruning), the symmetric difference distance was 52, compared to 38 for filtering on *o*=5. For the purpose of comparison, I also built trees on unfiltered and filtered data using five other alignment-free methods: ACS, CVTree, D2, kmacs, and Spaced Words. A smaller set of trees, due to the long run-time of the program, was calculated for ALFRED-G. The symmetric difference distances to the Eisen-73 and Eisen-71 trees are shown in Table 4-1.

SlopeTree applied to 495 bacteria

I built a series of ST-trees for 495 bacteria on unfiltered data and filtered data (varying the value of *o*). The closest of these to the reference is shown in Figure 4-4. As the root, I chose the division between the gram-negative and gram-positive bacteria. Organisms identified by SlopeTree as problematic (e.g. unusual number of conserved genes, reduced genomes, significantly fragmented assemblies, candidate division, etc.) were retained throughout the entire SlopeTree run, but pruned from the majority of the final trees (Appendix B). Mobile element and conservation filtering reduced the distance to the Eisen-495 tree for all methods, fixing several misplacements of individual organisms as well as shifting whole branches to locations more in keeping with the current NCBI classifications. By 'misplacement' I mean a disagreement with the current NCBI classification. For the purpose of comparison, I built trees on full and filtered data using ACS, CVTree, D2), kmacs, and Spaced Words. I also built trees using ALFRED-G, but could only test the *o*=5 and *o*=7 inputs due to the long run-time of the program. All distances to the Eisen-trees are included in Table 4-1.

There is no consensus regarding the positions of the deep branches of phylogenetic trees. Even the attempt to root the tree on the division between gram-positive and gram-negative bacteria could not be done cleanly, with the Chlamydiae, Cyanobacteria and Spirochaetes moving between these two groups for different levels of filtering. Not just SlopeTree, but all alignment-free methods have changes in their tree topologies as the inputs are filtered more aggressively. Nevertheless, I observed some stable features in the ST-trees that are stable for the other methods as well. These include a clade consisting of the Gammaproteobacteria, Betaproteobacteria, and Alphapoteobacteria. The Bacteroidetes, Chlorobi, and Gemmatimonadetes form another stable clade, typically neighboring a group consisting of the Spirochaetes and some subset of the Planctomycetes-Verrucomicrobia-Chlamydia (PVC) superphylum (132, 133). These features are consistent with the Eisen-495 tree. The Deltaproteobacteria however are almost always polyphyletic or paraphyletic. The position of the Acidobacteria is also variable, grouping with the Proteobacteria (mainly the Deltaproteobacteria) or the PVC group. The Epsilonproteobacteria are consistently monophyletic, but they group with the Proteobacteria for raw and less-filtered trees (up to o=3) and the Aquificae or PVC group for more filtered trees (o=5 or more).

SlopeTree usually places the Aquificae and a diverse, sulfur-reducing thermophilic group with the gram-negative bacteria, close to a group of Deltaproteobacteria. Filtering and the pair-wise HGT correction move this clade to an area that is separate from the majority of the gram-negative bacteria (Proteobacteria, Bacteroidetes, Chlorobi, Verrucomicrobia, Planctomycetes, etc.) and the gram-positive bacteria (Actinobacteria, Firmicutes) alike. The Cyanobacteria are also often found in this area; they are typically on a short, deep branch and

in the filtered trees, they neighbor the Deinococcus-Thermus. In the unfiltered ST-tree in which the pair-wise HGT correction was not performed, the Cyanobacteria are grouped with the Proteobacteria, which agrees with the Eisen-495 tree. However, a cursory investigation of the prospective HGT pairs for the members of Cyanobacteria present in the analysis revealed numerous possible transfers with the Proteobacteria, and the pair-wise HGT correction alone, even with no filtering, moved the Cyanobacteria away from the gram-negative bacteria and into the neutral area. This area also often includes a clade consisting of the Thermotogae and Synergistetes, another stable group whose placement in the trees varies between this area and a placement deep within the gram-positive bacteria.

The remainder of the tree consists predominantly of gram-positive bacteria. The Firmicutes and Actinobacteria typically share a common root, in agreement with the Eisen-495 tree. The Firmicutes are polyphyletic in all ST-trees, with the Tenericutes branching from within them. Whether the Tenericutes are their own phylum or belong within the Firmicutes is debated (134); SlopeTree consistently groups them within the Firmicutes, matching the Eisen-495 tree. The occasional presence of the Thermotogae within the Firmicutes is at least in part due to a clear instance of HGT discussed later, but it has been observed that the Thermotogae and Firmicutes, in particular Clostridia, show similarity at the whole-genome level (6, 135). The Fusobacteria are also in this clade, first nested within the Firmicutes but then more and more basal as filtering increases. The placement of the Fusobacteria with the gram-positive bacteria, despite their being gram negative, has support (6, 136). This generally gram-positive clade also often included the Chloroflexi. Like the Thermotogae, the Chloroflexi mostly stain Gram negative, but are monoderms (137)[77].

This placement is seen in the majority of trees produced by the other alignment-free methods and is also seen in the Eisen-495 tree.

Bacteria that diverge from the Eisen-495 tree or the NCBI classification

It is to be expected that different phylogenetic methods will produce different phylogenetic trees. However, the set of organisms that is misplaced in the trees according to the current NCBI taxonomy is remarkably consistent between all alignment-free methods and many of these misplacements were present in the supermatrix tree and specifically discussed in Lang et al. (28). I discuss some of them below.

***Coprothermobacter proteolyticus,* Dictyoglomi, Thermotogae and Synergistetes.** *C. proteolyticus*, currently classified as a member of Clostridia, is a thermophilic, gram-negative bacterium which was classified first as Thermobacteroides proteolyticus before being reclassified as a Firmicute, order Thermoanaerobacterales (138). Through the entire range of ST-trees without exception, it maintains a stable position alongside Dictyoglomus turgidum DSM 6724, a member of the Dictyoglomi. Together, C. proteolyticus and D. turgidum neighbor the Thermotogae, and this group in turn neighbors the Synergistetes. This placement is supported by the Eisen-495 tree and by other, independent observations from the literature (105, 134). Trees built by CVTree, D2, ACS, kmacs, Spaced Words, and ALFRED-G also support this classification.

A sulfur-reducing thermophilic cluster. There was tendency for sulfur-reducing thermophiles to cluster together in the tree, irrespective of their phylum. This cluster generally consisted of the Aquificae, a group of Deltaproteobacteria, and four additional

bacteria: *Thermodesulfobium narugense* (Clostridia), *Thermodesulfatator indicus* DSM 15286 (Thermodesulfobacteria), *Thermodesulfovibrio yellowstonii DSM 11347* (Nitrospira), and *Hippea maritima* (Deltaproteobacteria). All four were specifically described in Lang et al. (28) for their unusual phylogeny. *H. maritima* was placed in the Desulfurellaceae family of the Deltaproteobacteria by means of 16S rRNA (139); Lang et al. propose (28) to move it to the Epsilonproteobacteria. In the ST-phylogeny, *H. maritima* consistently appears closest to the Aquificae, forming a clade with this phylum in every ST-tree except for the most stringently filtered (o=7) ST-tree, in which it finally joins a clade consisting of Nitrospirae, Fibrobacteres, Verrucomicrobia, Planctomycetes, and the Epsilonproteobacteria. For *T. yellowstonii*, until filtering at o=5, it groups with the Aquificae, but then moves to the Epsilonproteobacteria. On the other hand, *T. narugense*, for o=3 groups with C. proteolyticus, D. turgidum, the Thermotogae and the Synergistetes. This is the placement supported in the Eisen-495 tree. However, for o=5 and o=7, it groups with the Deinoccocus-Thermus. *T. desulfatator*, for the totally raw tree, the pruned tree, and filtered for o=0, o=1, and o=5, it is found among the sulfur-reducing group; for o=3 and o=7, and also in the tree where no conservation filtering (only mobile element filtering) has been performed, it groups with the Deltaproteobacteria.

These four, together with the Aquificae, indicate that the less filtered trees, which provide a phylogenetic perspective that is inaccessible to alignment-based approaches, sometimes reflect phenetics over phylogeny. This grouping was present in all alignment-free methods, persisting to different extents as the data were filtered.

Acidithiobacillus ferrooxidans ATCC 23270* and *Acidithiobacillus caldus. NCBI currently classifies these two acidophiles as Gammaproteobacteria. In the unfiltered ST-tree, they form a basal group within the Gammaproteobacteria as well. However, their placement is unstable, and filtering can move them to within the Betaproteobacteria or make them a basal group for the two phyla. Compounding this ambiguity is the fact that under the heaviest conservation, they return to the Gammaproteobacteria. This ill-defined behavior was apparent in the other alignment-free phylogenies as well. It has been noted before that the Acidithiobacillales behave ambivalently (140, 141). Lang et al. (28) propose the creation of an "eta-proteobacteria" lineage for them. The alignment-free trees do not contradict this proposal.

***Dehalogenimonas lykanthroporepellens* and *Dehalococcoides mccartyi 195*.** *D. lykanthroporepellens* and *D. mccartyi* are members of the Chloroflexi. Both stain Gram negative, with the former being a mesophile—a somewhat unusual feature for a Chloroflexi. Both were classified by means of the 16S rRNA gene (142, 143). When no filtering was performed, SlopeTree misclassified this pair, grouping *D. lykanthroporepellens* with the Gammaproteobacteria and D. mccartyi with the Firmicutes. This pair was also misclassified by all other alternative methods (ACS, CVTree, D2 and Spaced Words) up to some level of filtering, although D2 showed the most robustness to this misplacement. The misplacement of *D. lykanthroporepellens* is due to a phage transfer shared with *Syntrophobacter fumaroxidans*, and *Desulfarculus baarsi* (Gammaproteobacteria). The pair-wise HGT correction also flagged the Firmicute *Natranaerobius thermophilus JW/NM-WN-LF* as being a possible partner of D. mccartyi, and removed several transporters prior to recalculating the

evolutionary distance. The lightest level of conservation filtering (o=0) was sufficient to fix the misplacement of these two Chloroflexi. We also found that the pair-wise HGT correction, even without filtering, also corrected their placement.

***Rhodothermus marinus* and *Salinibacter ruber*.** Every ST-tree contains the Bacteroidetes and Chlorobi clade. However, the family Rhodothermaceae, which consists of *R. marinus* and *S. ruber* and is classified as belonging to the Bacteroidetes, is frequently either grouped with the Chlorobi or placed on a branch basal to both phyla. The Eisen-495 tree places this pair of bacteria with the Bacteroidetes, but all alignment-free methods frequently set this pair apart from the Bacteroidetes. When no ME filtering or conservation filtering were performed, or for very low levels of conservation filtering, ACS, CVTree and kmacs can completely misplace these two bacteria. For instance when no mobile element and conservation filtering are performed, kmacs groups the pair with the three Actinobacteria discussed above, the Myxococcales, and Deinococcus-Thermus.

Distances to Eisen-trees and other whole-proteome or alignment-free methods

The symmetric difference distance (131) was calculated between all alignment-free trees and the Eisen-trees, using the treedist program in PHYLIP (117). However, the Eisen-trees are only approximations of the real evolutionary history, and that the methods should not be judged as "better" or "worse" purely according to their distances to these approximations. The kmacs method, with mobile element filtering and conservation filtering on *o=7*, achieved the closest tree to the Eisen-tree for both bacteria and archaea, with a symmetric

difference distance of 350 and 32. D2 also achieved a distance of 32 to the Eisen-71 tree. For bacteria and archaea, SlopeTree achieved 384 and 38, both at o=5.

SlopeTree trees using the HGT correction

The pair-wise HGT correction was applied to all SlopeTree inputs already described; this comprises for both archaea and bacteria: the unpruned, unfiltered set; the unfiltered, pruned set; and the sets that were filtered on different conservation parameters. The pair-wise HGT correction amended the placement of *D. lykothroporepellens*, *D. mccartyi*, and *P. mobilis*. In addition, it amended the placement of *Leptospira biflexa serovar Patoc* and *Leptospira interrogans serovar Lai*, two Spirochaetes which every alignment-free method misplaced unless using a very high level of conservation filtering. *Rhodothermus marinus* and *Salinibacter ruber M8*, classified as Bacteroidetes, were also moved from the Chlorobi back to the Bacteroidetes. The correction also caused some substantial reordering of the deeper branches. The Gammaproteobacteria, which are completely monophyletic in the uncorrected tree, are split into two groups in the corrected tree, in both cases forming a monophyletic clade with the Betaproteobacteria; this split is often seen in the other alignment-free methods and may be an indication of a missing "eta" class for the Proteobacteria (28, 144). The pair-wise HGT correction also removed the Cyanobacteria from the Proteobacteria, placing them close to the root alongside the Deinococcus-Thermus which were also shifted out of the Firmicutes. The Spirochaetes and Chlamydiae were also moved from the gram-positive bacteria to the gram-negative bacteria.

4.6 DISCUSSION AND CONCLUSIONS

I tested SlopeTree, a new, alignment-free method for phylogenetic reconstruction, on a set of strains and also on two domains of life. The method implements three types of gene-filtering: filtering for parasitic elements using copy number within a genome; filtering of genes by their overall conservation; and filtering of gene pairs indicating HGT. The method also includes a bulk correction for genome-specific HGT, it corrects for nonlinearity of the distance measure, and it corrects for compositional bias affecting the background. Some of these corrections work cleanly, for example the mobile element (ME) filter which removes parasitic elements. Others represent only minor corrections to the distance estimate. The biggest influence came from the filtering of gene pairs and filtering for overall conservation, which corrected for various artifacts and helped in the analysis of the global patterns of co-evolution. For sets of core genes and also for complete genomes, SlopeTree produced trees that were close but not identical to those produced by traditional MSA approaches (28). These results point to the general validity of species evolution by descent, but with various types of exceptions.

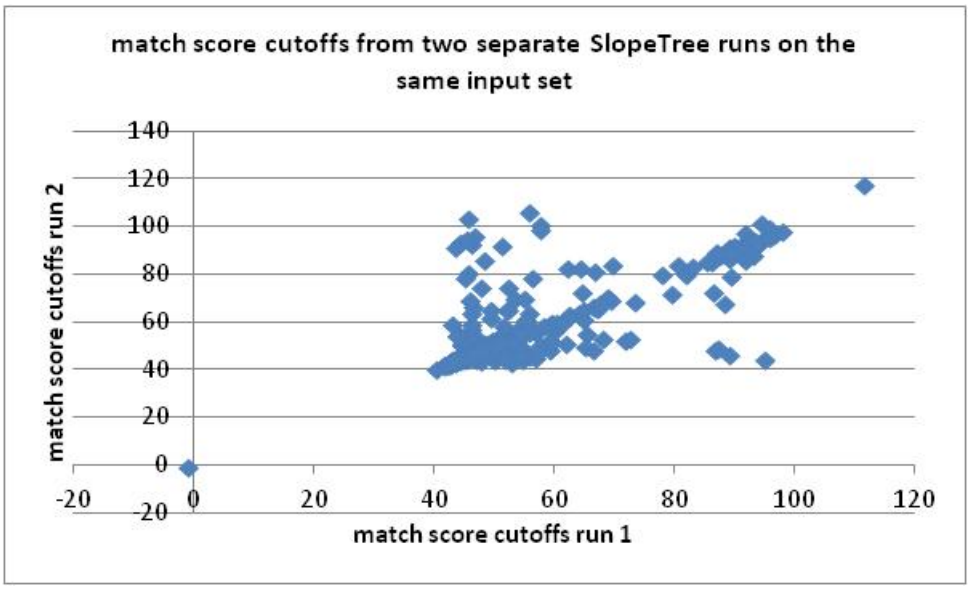
SlopeTree filtering benefits other methods

SlopeTree includes a filter for mobile elements and a conservation filter which is applied to all proteomes prior to the main run. A conservation filter follows, which is adjustable. As the level of filtering increased, the distances between the ST-trees and the Eisen-73 or the Eisen-495 trees decreased. All other alignment-free methods that we tested also benefited from filtering the data prior to running, at least in terms of their distances becoming closer to

the Eisen trees. An additional benefit to this is that filtering the data beforehand decreases the run-times.

The number of matches contributing to the assessment of evolutionary distances can be limited for longer distances or small genomes. Including mismatches adds a substantial number of informative, i.e. non-random, matches to the analysis. As can be seen with kmacs, the inclusion of mismatches can greatly improve phylogenetic distances. SlopeTree is essentially a type of survival analysis; therefore, it can apply to partial matches just as well as to those that are exact, and it is our expectation that such extension will produce even better results.

A



B

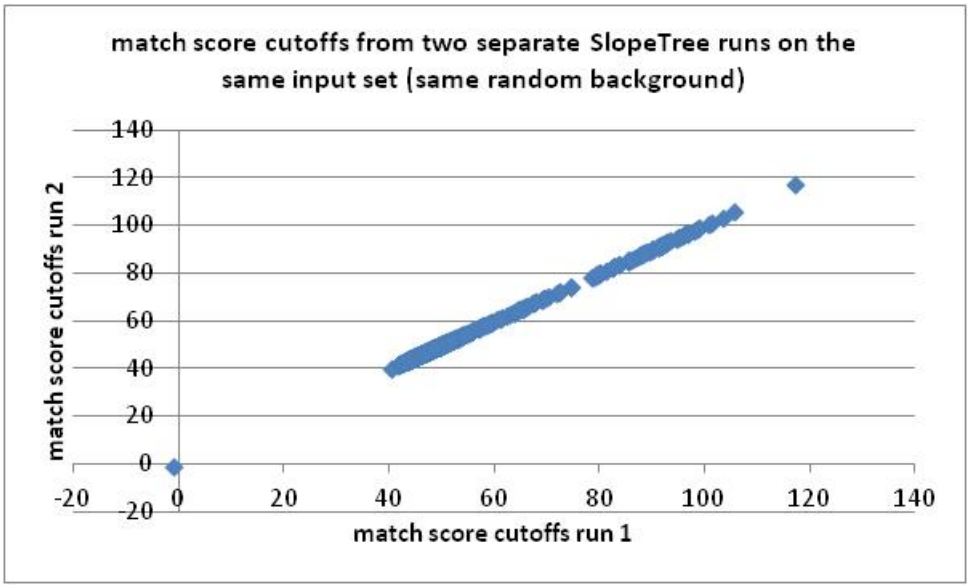


Figure 4-1. Instability of the fit from the sum of two exponentials.

A) For two runs on the same original input set, the match score cutoff values from run 1 plotted against those from run 2. B) For two runs on the same original input set *and* using the same randomly generated background, the match score cutoff values from run 1 plotted against those from run 2.

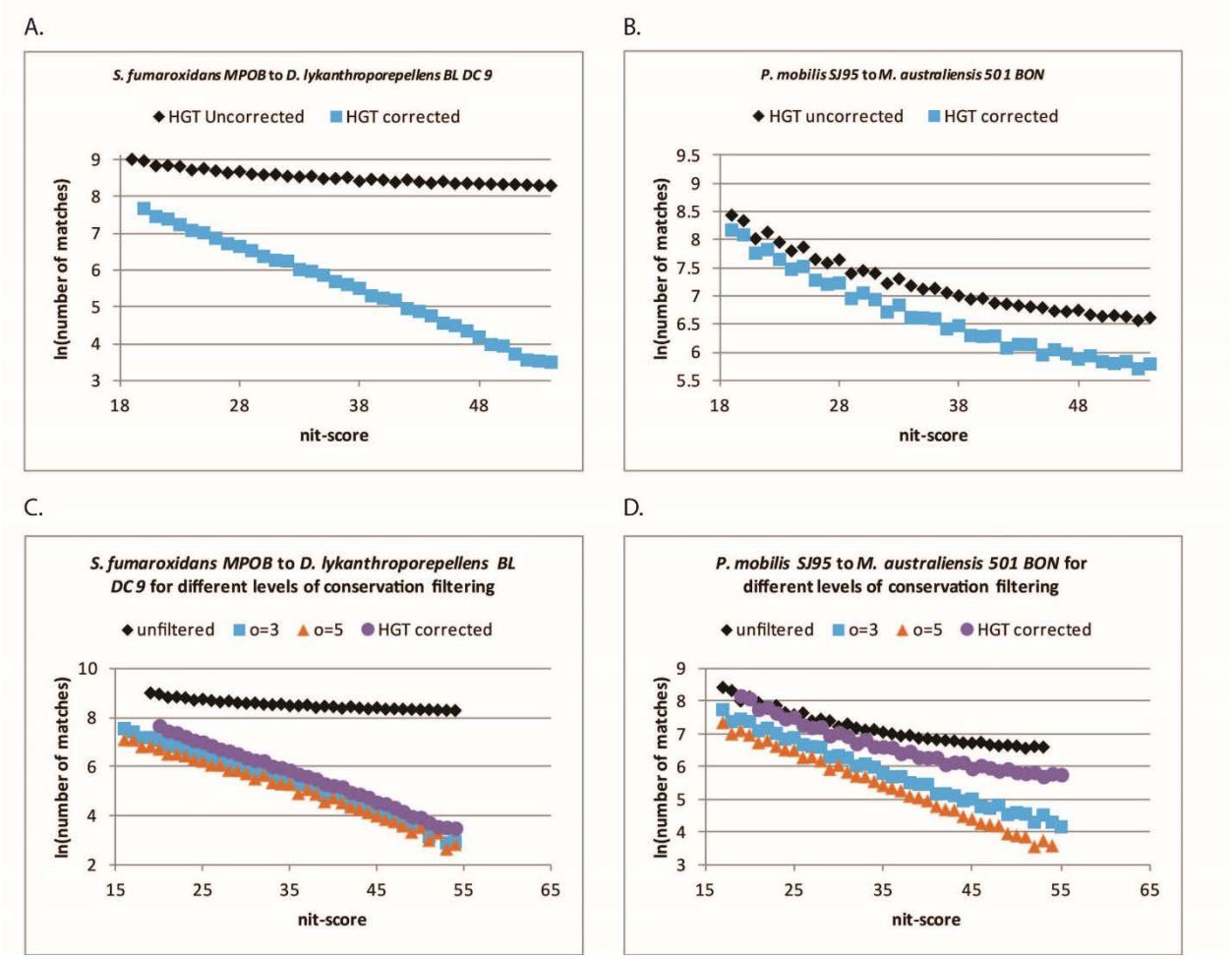


Figure 4-2. Correcting the 2 main classes of large-scale HGT.

A) A pair sharing a single copy phage. B) A pair sharing large-scale transfer of proteins associated with adaptation to environment. C) For pair sharing phage, effect on plots that mobile element filtering and conservation filtering have compared to the pair-wise HGT correction. D) For a pair sharing adaptive proteins, effect on plots that mobile element filtering and conservation filtering have compared to the pair-wise HGT correction.

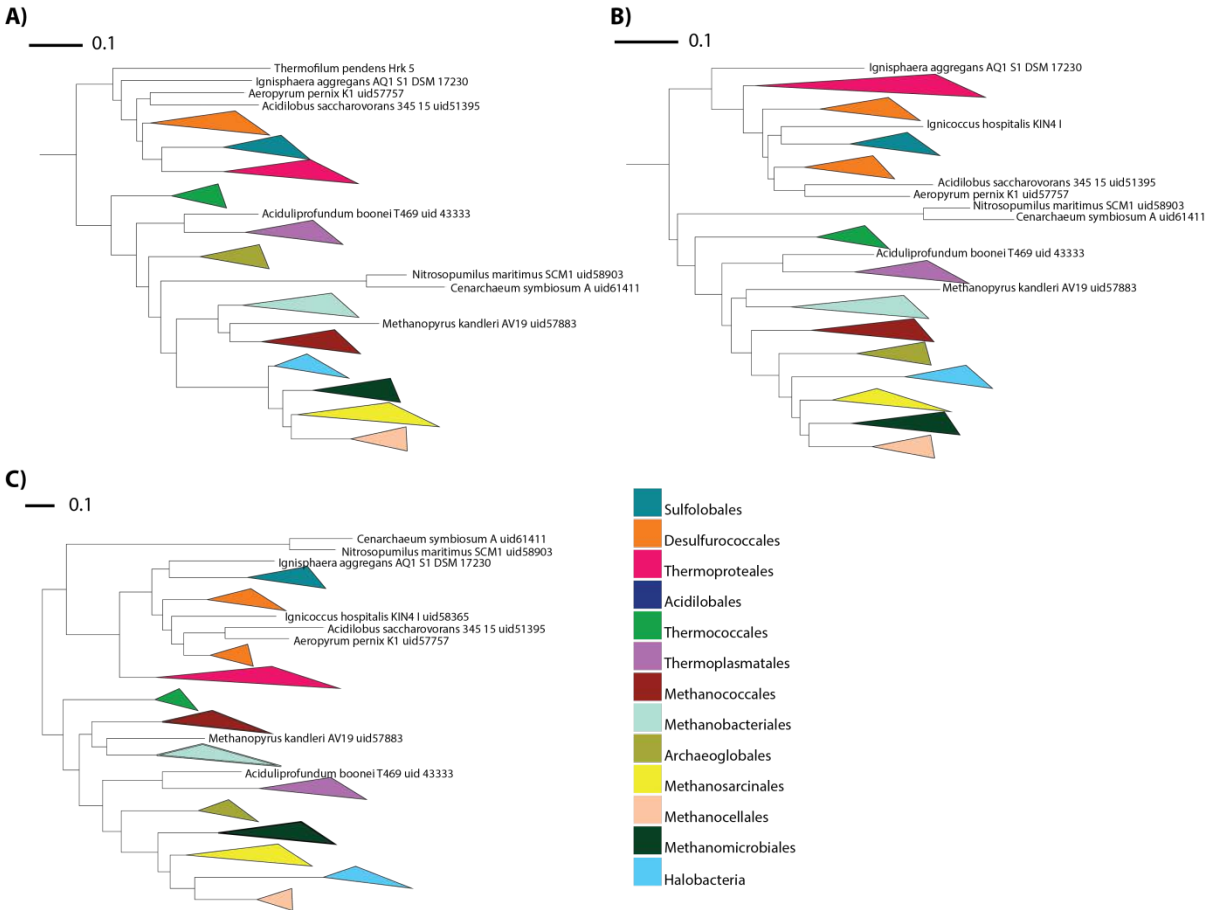


Figure 4-3. Phylogenetic trees for 73 Archaea.

A) ST-tree, raw and pruned. B) ST-tree, pruned, with mobile element filtering and conservation filtering (*o*=5). C) Eisen-71 tree.

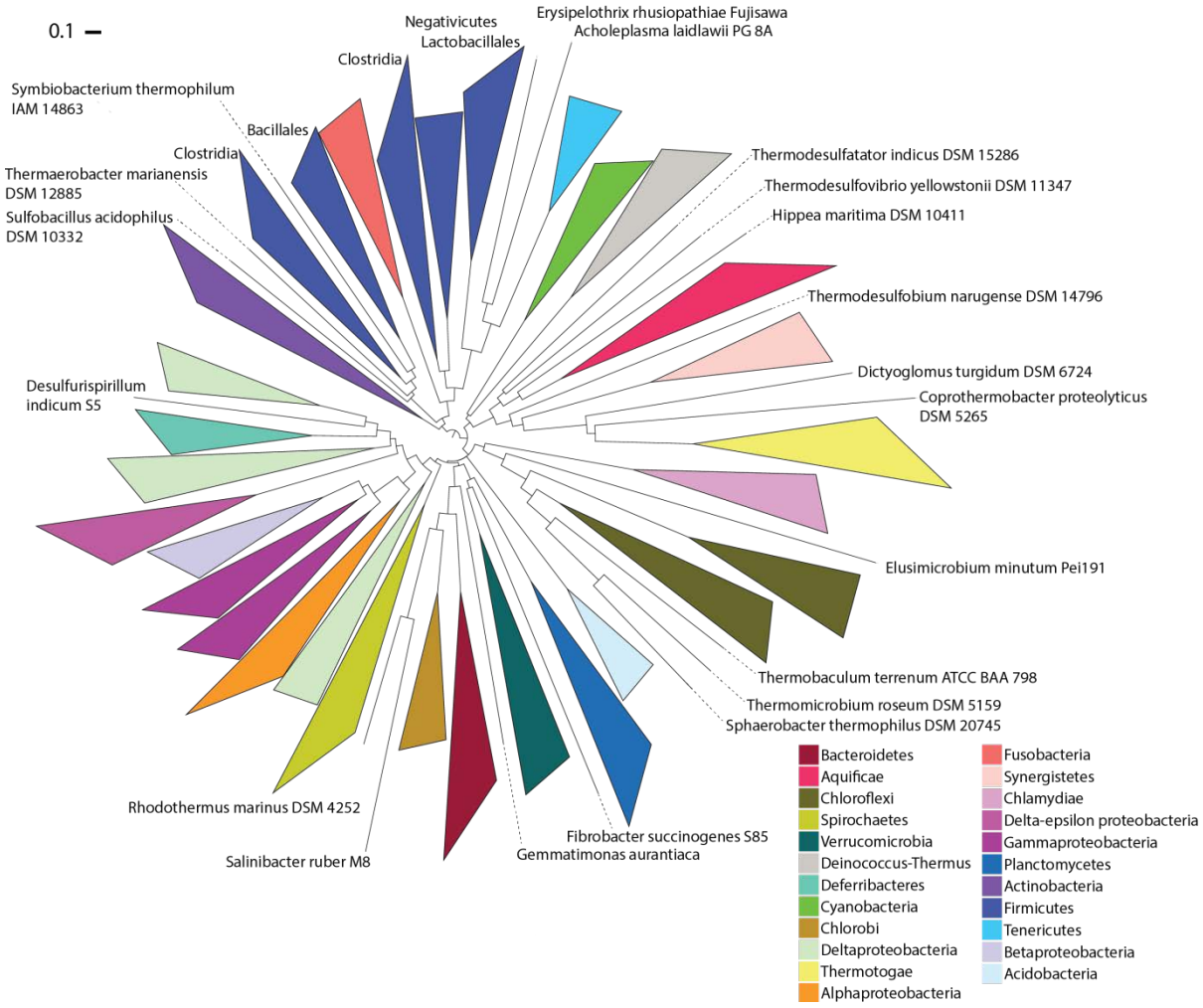


Figure 4-4. ST-tree of 495 Bacteria.

Tree was pruned to 445, with mobile element filtering and conservation filtering (*o*=3). Pair-wise HGT correction was performed.

Distance to the Eisen-495 tree and Eisen-445 tree	Symmetric difference	Distance to the Eisen-73 tree and Eisen 71-tree	Symmetric difference
ST - raw	506	ST - raw	56
ST - pruned	426	ST - pruned	52
ST - pruned, ME	432	ST - pruned, ME	54
ST - pruned, ME, o=0	402	ST - pruned, ME, o=0	52
ST - pruned, ME, o=1	388	ST - pruned, ME, o=1	56
ST - pruned, ME, o=3	388	ST - pruned, ME, o=3	42
ST - pruned, ME, o=3, HGT	384	ST - o=3, HGT	50
ST - pruned, ME, o=5	388	ST - ME, o=5	38
ST - pruned, ME, o=5, HGT	390	ST - ME, o=5, HGT	38
ST - pruned, ME, o=7	404	ST - ME, o=7	42
ST - pruned, ME, o=7, HGT	404	ST - ME, o=7, HGT	42
ACS - raw	554	ACS - raw	58
ACS - pruned	480	ACS - pruned	56
ACS - pruned, ME	474	ACS - pruned, ME	50
ACS - pruned, ME, o=0	448	ACS - pruned, ME, o=0	46
ACS - pruned, ME, o=1	442	ACS - pruned, ME, o=1	46
ACS - pruned, ME, o=3	412	ACS - pruned, ME, o=3	36
ACS - pruned, ME, o=5	420	ACS - pruned, ME, o=5	34
ACS - pruned, ME, o=7	410	ACS - pruned, ME, o=7	34
CVTree - raw	676	CVTree - raw	64
CVTree - pruned	868	CVTree - pruned	60
CVTree - pruned, ME	868	CVTree - pruned, ME	62
CVTree - pruned, ME, o=0	856	CVTree - pruned, ME, o=0	42
CVTree - pruned, ME, o=1	856	CVTree - pruned, ME, o=1	44
CVTree - pruned, ME, o=3	838	CVTree - pruned, ME, o=3	34
CVTree - pruned, ME, o=5	832	CVTree - pruned, ME, o=5	34
CVTree - pruned, ME, o=7	830	CVTree - pruned, ME, o=7	34
D2 - raw	528	D2 - raw	50
D2 - pruned	458	D2 - pruned	44
D2 - pruned, ME	416	D2 - pruned, ME	36
D2 - pruned, ME, o=0	410	D2 - pruned, ME, o=0	42
D2 - pruned, ME, o=1	398	D2 - pruned, ME, o=1	40
D2 - pruned, ME, o=3	386	D2 - pruned, ME, o=3	32
D2 - pruned, ME, o=5	366	D2 - pruned, ME, o=5	34

D2 - pruned, ME, o=7	366	D2 - pruned, ME, o=7	32
kmacs - raw	524	kmacs - raw	50
kmacs - pruned	448	kmacs - pruned	48
kmacs - pruned, ME	440	kmacs - pruned, ME	48
kmacs - pruned, ME, o=0	414	kmacs - pruned, ME, o=0	44
kmacs - pruned, ME, o=1	408	kmacs - pruned, ME, o=1	44
kmacs - pruned, ME, o=3	390	kmacs - pruned, ME, o=3	36
kmacs - pruned, ME, o=5	372	kmacs - pruned, ME, o=5	34
kmacs - pruned, ME, o=7	350	kmacs - pruned, ME, o=7	32
spaced - raw	810	spaced - raw	88
spaced - pruned	720	spaced - pruned	80
spaced - pruned, ME	684	spaced - pruned, ME	76
spaced - pruned, ME, o=0	578	spaced - pruned, ME, o=0	66
spaced - pruned, ME, o=1	526	spaced - pruned, ME, o=1	66
spaced - pruned, ME, o=3	478	spaced - pruned, ME, o=3	64
spaced - pruned, ME, o=5	452	spaced - pruned, ME, o=5	56
spaced - pruned, ME, o=7	430	spaced - pruned, ME, o=7	46
		ALFRED_G - raw	56
		ALFRED_G - pruned	48
		ALFRED_G - pruned, ME	48
		ALFRED_G - pruned, ME, o=0	44
ALFRED-G - pruned, ME, o=5	390		
ALFRED_G - pruned, ME, o=7	384	ALFRED_G - pruned, ME, o=7	34

Table 4-1. Symmetric difference distance to Eisen trees for SlopeTree and for six other whole-genome methods, over different levels of mobile-element and conservation filtering.

CHAPTER FIVE

CONCLUSIONS AND FUTURE DIRECTIONS

Non-sexual, clonal evolution with horizontal transfer creates a problem for defining the rules of species evolution. These rules would inform us on how to interpret genomic data, given the assumption of evolution by descent. The traditional approach to this problem is to define the genes that always evolve together (28, 145, 146). Such analyses are generally limited to the number of genes that are trustworthy, and these sets of genes in practice frequently correspond to ribosomal genes and proteins that interact with the ribosome (28, 35). However, if possible, we would like to have a concept of species evolution in prokaryotes that is not dominated by the evolution of the ribosome.

Alignment-free approaches using complete genomes are an alternative to MSA approaches. It is to be expected that alignment-free methods, which look for consensus phylogenetic signals at the level of individual k-mers rather than gene-long alignments, provide alternative insights into evolutionary history. For instance, alignment-free methods identified a cluster of sulfur-reducing thermophiles which was absent from the traditional MSA tree. To assess alignment-free methods, their trees can be compared to the ribosomal evolution tree, which is what we did here, but it may not always be clear to what extent disagreements are due to the method or to the lack of co-evolution.

Despite some objections to the validity of phylogenetic trees and the concept of evolution by descent, evolution by descent still appears to be the predominant mode of preserving organismal identity. However, this cannot be assumed to be universal in prokaryotes, and requires that the methods be implemented such that they are aware of exceptions to vertical evolution. When larger databases are available, it will be important to keep these exceptions in mind when making decisions about taxonomy. Compared to the current characterization of species using SSU rRNA, a sensible future direction for methods capable of characterizing species may be by means of full genome sequencing. Species characterization by full genome sequencing will require more data approaches that have multiple, automatic layers of analysis, as I developed for SlopeTree.

5.1 FUTURE DIRECTIONS

SlopeTree future development

Databases have now reached a size such that phylogenetic analyses must be fully automatic rather than relying on curation at intermediate stages. SlopeTree achieved this. The method is fast, unsupervised, and addresses the main challenges in phylogenetic analysis (compositional bias, heterotachy, horizontal transfer, etc.), producing trustworthy evolutionary distances. Unsupervised methods already exist in this category, but they include relatively few corrections for the problems we encounter in evolutionary analysis. However, the methods that I implemented still have considerable room for improvement and

expansion. The identification of HGT in particular requires refinement, with more sophisticated analyses for duplications, paralogs, and horizontal transfer.

SlopeTree refinements

A modification to SlopeTree, which would increase the method's utility, would be extending the algorithm to consider nucleotide sequences. This addition would not be useful for large evolutionary distances, but for strain-level analysis, it would improve the accuracy of the results. The optimal k-mer length for such analyses would also have to be determined, and is most likely longer than the default of k=20 used for amino acids.

Another modification would be to the amino acid frequencies used to calculate the nit scores. Originally, I used an average set of frequencies over all the organisms in the input. This I eventually replaced with calculating the nit score for each match using an average of only the two contributing organisms' frequencies. While SlopeTree takes whole proteomes as its input, the distances ultimately come not from the whole genome but only the pieces of proteins that find long length matches. And these fragments correspond to the conserved regions of proteins, which exhibit different amino acid frequencies than those observed across the whole proteome. The correction for composition would improve if we used this new set of frequencies, especially for distant organisms.

As mentioned above, permitting mismatches has greatly improved the results of other alignment-free methods. Incorporating this into SlopeTree could improve both the topology and the linearity of the branch lengths with evolutionary time. However, this would also

demand a significant re-implementation of the method, most likely finally requiring a transition to some form of suffix arrays.

The SlopeTree implementation contains a large number of hardcoded values that were determined mostly by trial and error. The point x=15, from which the slope is taken from the quadratic fit, is one such example. The criteria by which pairs are flagged as suspect for horizontal transfer is another particularly troublesome example; as this portion of the code stands, it is likely that users will have to adjust the cutoffs for these criteria to obtain an acceptable number of flagged pairs. More work needs to be put into the calculation of these values.

SlopeTree was intended for arbitrarily large inputs. However, the addition of some pre-processing steps would make its application to large inputs much more practical. For instance, the NCBI database of genomes has a large number of *Escherichia coli* in it. This type of oversampling should be addressed early on. One approach would be a preliminary SlopeTree run on all the input data, but generating only a fraction of the tags—e.g. only tags with a leading 'C'. This would be enough data to identify members of the same species and either eliminate all but one or else merge them into a pangenome. Alternatively, groups of strains could be passed to the SlopeTree module using nucleotides rather than amino acids.

Generating fast, high-quality, automatic alignments

A byproduct of the SlopeTree match-counting algorithm is a list consisting of every single matching sequence between every single pair. Every single time the match-counting algorithm hits the end of a block in the recursive algorithm and counts the sequence in the

correlation matrix, the sequence is also written to this file. I performed extensive preliminary work on the sequences in this file, first removing the short-length sequences, then grouping the remaining sequences into clusters. I first built the clusters simply by comparing the top sequence in the list to all sequences below it; any sequence it matched for a given number of amino acids was then added to the cluster, and then all sequences in the new cluster were compared to the remainder of the list, until the cluster stopped growing. This algorithm served as a proof-of-concept that the sequences could be clustered, but was unacceptably slow. I replaced it with a k-mer based method similar to what SlopeTree already does. The matching sequences were given IDs and then split into shorter k-mers which, along with their IDs, were sorted into a list. Adjacent k-mers were then compared to form clusters, and these clusters were then merged using the original IDs.

I ignored all clusters of size four or less. For the rest, I aligned all the sequences in a cluster and then built a profile from it and scanned through a set of proteomes. These profiles were narrow, sometimes only 8 amino acids wide, and yet they proved extremely effective in identifying homologs. Running these profiles left me with a list of short, aligned sequences corresponding to a conserved region of a group of homologs. I could generate a matrix from these short sequences and separate them into groups (e.g. separate orthologs from paralogs) by means of singular value decomposition.

This work was done on single clusters in a one-at-a-time fashion, but could eventually be extended to generate high quality, automatic alignments for large inputs.

Web-server for SlopeTree with selection for automatically generated, diverse taxa

Many alignment-free methods offer web-servers that generated trees on user-defined inputs, as discussed in the introduction. I am currently developing one for SlopeTree, but this server will be somewhat unique among the current alignemtn-free servers. Rather than calculating distance matrices for user-defined inputs, the server will contian large, precalculated matrices that are regulraly updated out of the current NCBI database. Users will then be able to select the taxa that they want in their tree, with tools for genrating diverse sets for any group, at any taxonomic level. Uneven sampling of taxa (e.g. there is a large number of *E. coli* sequences in the databases) can make phylogenetic analysis needlessly difficult. The SlopeTree web-server will make it possibel to BLAST a user-inputted protein on a truly diverse set of organisms.

APPENDIX A

Example list of proteins removed by mobile element filter

gene=12: 26 8 >gi|42560573|ref|NP_975024.1| prolipoprotein [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=13: 26 0 >gi|42560574|ref|NP_975025.1| immunodominant protein P72 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=14: 239 3 >gi|42560575|ref|NP_975026.1| IS1296SQ transposase protein B [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=15: 161 0 >gi|42560576|ref|NP_975027.1| IS1296SQ transposase protein A [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=32: 38 12 >gi|42560593|ref|NP_975044.1| permease [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=37: 522 0 >gi|42560598|ref|NP_975049.1| IS1634BT transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=54: 249 3 >gi|42560615|ref|NP_975066.1| IS1296IE transposase protein B [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=55: 154 0 >gi|42560616|ref|NP_975067.1| IS1296IE transposase protein A [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=59: 538 0 >gi|42560620|ref|NP_975071.1| IS1634BK transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=69: 488 1 >gi|42560630|ref|NP_975081.1| IS1634BA transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=72: 526 0 >gi|42560633|ref|NP_975084.1| IS1634BV transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=91: 8 0 >gi|42560652|ref|NP_975103.1| hypothetical protein MSC_0093 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=95: 538 0 >gi|42560656|ref|NP_975107.1| IS1634BO transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=96: 538 0 >gi|42560657|ref|NP_975108.1| IS1634AP transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=105: 52 0 >gi|42560666|ref|NP_975117.1| hypothetical protein MSC_0107 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=106: 388 0 >gi|42560667|ref|NP_975118.1| glycosyltransferase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=107: 276 0 >gi|42560668|ref|NP_975119.1| glycosyltransferase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=108: 263 0 >gi|42560669|ref|NP_975120.1| UTP-glucose-1-phosphate uridylyltransferase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=109: 54 3 >gi|42560670|ref|NP_975121.1| hypothetical protein MSC_0111 [Mycoplasma mycoides subsp. mycoides SC str. PG1]

GRAY ZONE: gene=116: 6 0 >gi|42560677|ref|NP_975128.1| hexosephosphate transport protein [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=118: 70 0 >gi|42560679|ref|NP_975130.1| prolipoprotein lppC [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=119: 438 0 >gi|42560680|ref|NP_975131.1| transposase ISMmy1E [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=121: 19 0 >gi|42560682|ref|NP_975133.1| hypothetical protein MSC_0126 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=128: 296 0 >gi|42560689|ref|NP_975140.1| hypothetical protein MSC_0135 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=129: 281 0 >gi|42560690|ref|NP_975141.1| hypothetical protein MSC_0136 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=130: 267 3 >gi|42560691|ref|NP_975142.1| IS1296MP transposase protein B [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=131: 141 0 >gi|42560692|ref|NP_975143.1| IS1296MP transposase protein A [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=133: 150 0 >gi|42560694|ref|NP_975145.1| IS1296EH transposase protein A [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=154: 109 0 >gi|42560715|ref|NP_975166.1| leucyl aminopeptidase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=159: 501 0 >gi|42560720|ref|NP_975171.1| IS1634BZ transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=163: 538 0 >gi|42560724|ref|NP_975175.1| IS1634BP transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=164: 144 0 >gi|42560725|ref|NP_975176.1| IS1296UK transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=165: 267 3 >gi|42560726|ref|NP_975177.1| IS1296UK transposase protein B [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=177: 96 12 >gi|42560738|ref|NP_975189.1| DNA methylase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=178: 225 0 >gi|42560739|ref|NP_975190.1| hypothetical protein MSC_0187 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=179: 165 0 >gi|42560740|ref|NP_975191.1| hypothetical protein MSC_0188 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=185: 48 10 >gi|42560746|ref|NP_975197.1| prophage protein (ps3) [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=198: 28 0 >gi|42560759|ref|NP_975210.1| hypothetical protein MSC_0207 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=202: 263 3 >gi|42560763|ref|NP_975214.1| IS1296JI transposase protein B [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=219: 446 0 >gi|42560780|ref|NP_975231.1| transposase ISMmy1B [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=220: 161 0 >gi|42560781|ref|NP_975232.1| IS1296PX transposase protein A [Mycoplasma mycoides subsp. mycoides SC str. PG1]

gene=221: 8 0 >gi|42560782|ref|NP_975233.1| hypothetical protein MSC_0233 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=223: 109 0 >gi|42560784|ref|NP_975235.1| leucyl aminopeptidase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=226: 137 3 >gi|42560787|ref|NP_975238.1| IS1296FJ transposase protein A [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=227: 267 3 >gi|42560788|ref|NP_975239.1| IS1296FJ transposase protein B [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=232: 469 1 >gi|42560793|ref|NP_975244.1| IS1634AC transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=233: 263 3 >gi|42560794|ref|NP_975245.1| IS1296OD transposase protein B [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=236: 533 0 >gi|42560797|ref|NP_975248.1| IS1634AG transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=254: 30 0 >gi|42560815|ref|NP_975266.1| branched-chain alpha-keto acid dehydrogenase subunit E2 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=255: 30 0 >gi|42560816|ref|NP_975267.1| dihydrolipoamide dehydrogenase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=282: 26 1 >gi|42560843|ref|NP_975294.1| oxidoreductase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=288: 26 0 >gi|42560849|ref|NP_975300.1| oxidoreductase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=301: 11 2 >gi|42560862|ref|NP_975313.1| hypothetical protein MSC_0314 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=302: 27 0 >gi|42560863|ref|NP_975314.1| permease [Mycoplasma mycoides subsp. mycoides SC str. PG1]
GRAY ZONE: gene=311: 5 0 >gi|42560872|ref|NP_975323.1| hypothetical protein MSC_0325 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=318: 538 0 >gi|42560879|ref|NP_975330.1| IS1634AW transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=325: 518 0 >gi|42560886|ref|NP_975337.1| IS1634BR transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=346: 528 0 >gi|42560907|ref|NP_975358.1| IS1634AE transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
GRAY ZONE: gene=358: 3 0 >gi|42560919|ref|NP_975370.1| prolipoprotein [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=370: 19 0 >gi|42560931|ref|NP_975382.1| hypothetical protein MSC_0391 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=407: 28 0 >gi|42560968|ref|NP_975419.1| hypothetical protein MSC_0433 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=408: 9 0 >gi|42560969|ref|NP_975420.1| ABC transporter ATP-binding protein and permease [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=452: 48 0 >gi|42561013|ref|NP_975464.1| hypothetical protein MSC_0478 [Mycoplasma mycoides subsp. mycoides SC str. PG1]

gene=478: 8 0 >gi|42561039|ref|NP_975490.1| peptidase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=494: 528 0 >gi|42561055|ref|NP_975506.1| IS1634AD transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=510: 538 0 >gi|42561071|ref|NP_975522.1| IS1634AB transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=523: 538 0 >gi|42561084|ref|NP_975535.1| IS1634AA transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=540: 451 0 >gi|42561101|ref|NP_975552.1| transposase ISMmy1F [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=571: 538 0 >gi|42561132|ref|NP_975583.1| IS1634CA transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=573: 538 0 >gi|42561134|ref|NP_975585.1| IS1634BH transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=593: 44 0 >gi|42561154|ref|NP_975605.1| prolipoprotein [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=594: 27 0 >gi|42561155|ref|NP_975606.1| hypothetical protein MSC_0626 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=595: 58 1 >gi|42561156|ref|NP_975607.1| prolipoprotein [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=596: 67 0 >gi|42561157|ref|NP_975608.1| hypothetical protein MSC_0628 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=597: 253 0 >gi|42561158|ref|NP_975609.1| IS1296LL transposase protein B [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=598: 161 0 >gi|42561159|ref|NP_975610.1| IS1296LL transposase protein A [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=599: 41 0 >gi|42561160|ref|NP_975611.1| hypothetical protein MSC_0631 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=601: 121 0 >gi|42561162|ref|NP_975613.1| hypothetical protein MSC_0633 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=602: 54 0 >gi|42561163|ref|NP_975614.1| prolipoprotein [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=604: 81 0 >gi|42561165|ref|NP_975616.1| hypothetical protein MSC_0637 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=629: 161 0 >gi|42561190|ref|NP_975641.1| IS1296AB_B transposase protein A [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=630: 249 0 >gi|42561191|ref|NP_975642.1| IS1296AB_B transposase protein B [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=632: 248 0 >gi|42561193|ref|NP_975644.1| hypothetical protein MSC_0665 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=633: 148 0 >gi|42561194|ref|NP_975645.1| hypothetical protein MSC_0667 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=634: 96 0 >gi|42561195|ref|NP_975646.1| hypothetical protein MSC_0668 [Mycoplasma mycoides subsp. mycoides SC str. PG1]

gene=638: 538 0 >gi|42561199|ref|NP_975650.1| IS1634AS transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=639: 161 0 >gi|42561200|ref|NP_975651.1| IS1296QT transposase protein A [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=650: 538 0 >gi|42561211|ref|NP_975662.1| IS1634BL transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=661: 8 0 >gi|42561222|ref|NP_975673.1| endopeptidase O [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=662: 480 1 >gi|42561223|ref|NP_975674.1| IS1634BM transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=665: 207 0 >gi|42561226|ref|NP_975677.1| carbamate kinase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=740: 30 0 >gi|42561300|ref|NP_975751.1| prolipoprotein [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=741: 30 0 >gi|42561301|ref|NP_975752.1| prolipoprotein [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=746: 337 0 >gi|42561306|ref|NP_975757.1| transposase ISMmy1D [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=747: 584 9 >gi|42561307|ref|NP_975758.1| prolipoprotein [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=748: 33 0 >gi|42561308|ref|NP_975759.1| hypothetical protein MSC_0783 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=749: 188 1 >gi|42561309|ref|NP_975760.1| hypothetical protein MSC_0784 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=750: 538 0 >gi|42561310|ref|NP_975761.1| IS1634AM transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=751: 199 1 >gi|42561311|ref|NP_975762.1| alkylphosphonate ABC transporter permease [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=752: 230 0 >gi|42561312|ref|NP_975763.1| alkylphosphonate ABC transporter ATP-binding protein [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=753: 444 0 >gi|42561313|ref|NP_975764.1| alkylphosphonate ABC transporter substrate-binding protein [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=755: 28 1 >gi|42561315|ref|NP_975766.1| hypothetical protein MSC_0792 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=756: 153 0 >gi|42561316|ref|NP_975767.1| hypothetical protein MSC_0793 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=757: 373 0 >gi|42561317|ref|NP_975768.1| aminotransferase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=758: 114 1 >gi|42561318|ref|NP_975769.1| translation initiation inhibitor [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=759: 438 0 >gi|42561319|ref|NP_975770.1| transposase ISMmy1G [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=760: 538 0 >gi|42561320|ref|NP_975771.1| IS1634AU transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]

gene=761: 584 9 >gi|42561321|ref|NP_975772.1| prolipoprotein [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=762: 33 0 >gi|42561322|ref|NP_975773.1| hypothetical protein MSC_0799 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=763: 188 1 >gi|42561323|ref|NP_975774.1| hypothetical protein MSC_0800 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=764: 199 9 >gi|42561324|ref|NP_975775.1| ABC transporter permease [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=765: 524 0 >gi|42561325|ref|NP_975776.1| IS1634AY transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=766: 230 0 >gi|42561326|ref|NP_975777.1| alkylphosphonate ABC transporter ATP-binding protein [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=767: 444 0 >gi|42561327|ref|NP_975778.1| ABC transporter substrate-binding protein [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=768: 181 1 >gi|42561328|ref|NP_975779.1| HAD superfamily hydrolase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=769: 373 0 >gi|42561329|ref|NP_975780.1| aminotransferase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=770: 114 1 >gi|42561330|ref|NP_975781.1| translation initiation inhibitor [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=771: 439 0 >gi|42561331|ref|NP_975782.1| transposase ISMmy1C [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=772: 9 0 >gi|42561332|ref|NP_975783.1| variable surface protein [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=773: 488 0 >gi|42561333|ref|NP_975784.1| variable surface protein [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=774: 460 0 >gi|42561334|ref|NP_975785.1| variable surface protein [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=775: 488 0 >gi|42561335|ref|NP_975786.1| variable surface protein [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=776: 533 0 >gi|42561336|ref|NP_975787.1| IS1634CHBZ transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=777: 448 0 >gi|42561337|ref|NP_975788.1| variable surface protein [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=778: 368 0 >gi|42561338|ref|NP_975789.1| variable surface protein [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=779: 49 0 >gi|42561339|ref|NP_975790.1| variable surface protein [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=780: 301 0 >gi|42561340|ref|NP_975791.1| variable surface protein [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=784: 530 0 >gi|42561344|ref|NP_975795.1| IS1634CI transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=804: 262 3 >gi|42561364|ref|NP_975815.1| IS1296HV transposase protein B [Mycoplasma mycoides subsp. mycoides SC str. PG1]

gene=811: 161 0 >gi|42561371|ref|NP_975822.1| IS1296GZ transposase protein A [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=818: 538 0 >gi|42561378|ref|NP_975829.1| IS1634AL transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=819: 652 0 >gi|42561379|ref|NP_975830.1| PTS system, glucose-specific IIBC component [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=820: 300 0 >gi|42561380|ref|NP_975831.1| hypothetical protein MSC_0861 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=821: 538 0 >gi|42561381|ref|NP_975832.1| IS1634AX transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=822: 215 0 >gi|42561382|ref|NP_975833.1| glucokinase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=823: 285 0 >gi|42561383|ref|NP_975834.1| carbamate kinase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=824: 83 0 >gi|42561384|ref|NP_975835.1| membrane arginine transporter [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=825: 30 1 >gi|42561385|ref|NP_975836.1| hypothetical protein MSC_0866 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=826: 62 0 >gi|42561386|ref|NP_975837.1| hypothetical protein MSC_0867 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=827: 596 0 >gi|42561387|ref|NP_975838.1| Mg(2+) transport ATPase, P-type [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=828: 231 0 >gi|42561388|ref|NP_975839.1| hypothetical protein MSC_0869 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=829: 112 0 >gi|42561389|ref|NP_975840.1| hypothetical protein MSC_0870 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=830: 538 0 >gi|42561390|ref|NP_975841.1| IS1634CE transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=832: 652 0 >gi|42561392|ref|NP_975843.1| PTS system, glucose-specific, IIBC component [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=833: 300 0 >gi|42561393|ref|NP_975844.1| hypothetical protein MSC_0874 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=834: 215 0 >gi|42561394|ref|NP_975845.1| glucokinase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=835: 534 0 >gi|42561395|ref|NP_975846.1| IS1634AZ transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=836: 260 0 >gi|42561396|ref|NP_975847.1| carbamate kinase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=837: 83 0 >gi|42561397|ref|NP_975848.1| membrane arginine transporter [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=838: 30 1 >gi|42561398|ref|NP_975849.1| hypothetical protein MSC_0879 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=839: 62 0 >gi|42561399|ref|NP_975850.1| hypothetical protein MSC_0880 [Mycoplasma mycoides subsp. mycoides SC str. PG1]

gene=840: 596 0 >gi|42561400|ref|NP_975851.1| magnesium ABC transporter ATPase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=841: 343 0 >gi|42561401|ref|NP_975852.1| hypothetical protein MSC_0882 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=856: 161 0 >gi|42561416|ref|NP_975867.1| IS1296AC_R transposase protein A [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=857: 267 3 >gi|42561417|ref|NP_975868.1| IS1296AC_R transposase protein B [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=858: 281 0 >gi|42561418|ref|NP_975869.1| hypothetical protein MSC_0900 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=859: 296 0 >gi|42561419|ref|NP_975870.1| hypothetical protein MSC_0901 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=860: 170 112 >gi|42561420|ref|NP_975871.1| CTP synthetase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
GRAY ZONE: gene=861: 6 0 >gi|42561421|ref|NP_975872.1| hexose phosphate transport protein [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=862: 263 3 >gi|42561422|ref|NP_975873.1| IS1296DS transposase protein B [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=863: 138 0 >gi|42561423|ref|NP_975874.1| IS1296DS transposase protein A [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=874: 72 0 >gi|42561434|ref|NP_975885.1| hypothetical protein MSC_0916 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=875: 72 0 >gi|42561435|ref|NP_975886.1| hypothetical protein MSC_0918 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
GRAY ZONE: gene=877: 6 0 >gi|42561437|ref|NP_975888.1| hypothetical protein MSC_0920 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=878: 492 0 >gi|42561438|ref|NP_975889.1| IS1634CD transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=890: 538 0 >gi|42561450|ref|NP_975901.1| IS1634BY transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=924: 301 0 >gi|42561484|ref|NP_975935.1| oligopeptide ABC transporter ATP-binding protein [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=925: 533 0 >gi|42561485|ref|NP_975936.1| IS1634BQ transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=926: 285 0 >gi|42561486|ref|NP_975937.1| UDP-galactopyranose mutase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=927: 310 20 >gi|42561487|ref|NP_975938.1| UDP-glucose 4-epimerase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=928: 29 1 >gi|42561488|ref|NP_975939.1| hypothetical protein MSC_0972 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=929: 363 0 >gi|42561489|ref|NP_975940.1| glycosyltransferase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=930: 241 0 >gi|42561490|ref|NP_975941.1| glycosyltransferase [Mycoplasma mycoides subsp. mycoides SC str. PG1]

gene=931: 301 0 >gi|42561491|ref|NP_975942.1| oligopeptide ABC transporter permease [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=932: 534 0 >gi|42561492|ref|NP_975943.1| IS1634BN transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=933: 284 0 >gi|42561493|ref|NP_975944.1| UDP-galactopuranose mutase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=934: 310 20 >gi|42561494|ref|NP_975945.1| UDP-glucose 4-epimerase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=935: 53 1 >gi|42561495|ref|NP_975946.1| hypothetical protein MSC_0979 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=936: 363 0 >gi|42561496|ref|NP_975947.1| glycosyltransferase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=937: 538 0 >gi|42561497|ref|NP_975948.1| IS1634AV transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=938: 247 0 >gi|42561498|ref|NP_975949.1| glycosyltransferase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=939: 268 0 >gi|42561499|ref|NP_975950.1| oligopeptide ABC transporter ATP-binding protein [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=940: 283 0 >gi|42561500|ref|NP_975951.1| UDP-galactopyranose mutase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=941: 187 3 >gi|42561501|ref|NP_975952.1| UDP-glucose 4-epimerase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=942: 53 1 >gi|42561502|ref|NP_975953.1| hypothetical protein MSC_0986 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=943: 363 0 >gi|42561503|ref|NP_975954.1| glycosyltransferase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=944: 285 0 >gi|42561504|ref|NP_975955.1| glycosyltransferase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=945: 530 0 >gi|42561505|ref|NP_975956.1| IS1634CB transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=946: 263 0 >gi|42561506|ref|NP_975957.1| UTP-glucose-1-phosphate uridylyltransferase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=947: 31 3 >gi|42561507|ref|NP_975958.1| hypothetical protein MSC_0991 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=948: 75 3 >gi|42561508|ref|NP_975959.1| hypothetical protein MSC_0992 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=949: 429 0 >gi|42561509|ref|NP_975960.1| glycosyltransferase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=955: 25 0 >gi|42561515|ref|NP_975966.1| prolipoprotein [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=957: 529 0 >gi|42561517|ref|NP_975968.1| IS1634BG transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=959: 70 0 >gi|42561519|ref|NP_975970.1| variable surface prolipoprotein [Mycoplasma mycoides subsp. mycoides SC str. PG1]

gene=964: 538 0 >gi|42561524|ref|NP_975975.1| IS1634BX transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=965: 43 0 >gi|42561525|ref|NP_975976.1| hypothetical protein MSC_1012 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=966: 66 0 >gi|42561526|ref|NP_975977.1| hypothetical protein MSC_1013 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=967: 30 0 >gi|42561527|ref|NP_975978.1| hypothetical protein MSC_1014 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=968: 307 0 >gi|42561528|ref|NP_975979.1| asparagine synthetase AsnA [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=969: 234 0 >gi|42561529|ref|NP_975980.1| hypothetical protein MSC_1016 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=970: 610 348 >gi|42561530|ref|NP_975981.1| tRNA uridine 5-carboxymethylaminomethyl modification protein GidA [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=971: 501 0 >gi|42561531|ref|NP_975982.1| proton/glutamate symporter [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=972: 447 0 >gi|42561532|ref|NP_975983.1| NADH oxidase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=973: 147 3 >gi|42561533|ref|NP_975984.1| pyrazinamidase/nicotinamidase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=974: 426 0 >gi|42561534|ref|NP_975985.1| prolipoprotein Q [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=975: 131 1 >gi|42561535|ref|NP_975986.1| hypothetical protein MSC_1022 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=976: 53 4 >gi|42561536|ref|NP_975987.1| hypothetical protein MSC_1023 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=977: 78 0 >gi|42561537|ref|NP_975988.1| hypothetical protein MSC_1024 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=978: 239 0 >gi|42561538|ref|NP_975989.1| hypothetical protein MSC_1025 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=979: 453 19 >gi|42561539|ref|NP_975990.1| hypothetical protein MSC_1026 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=980: 133 0 >gi|42561540|ref|NP_975991.1| hypothetical protein MSC_1027 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=981: 50 0 >gi|42561541|ref|NP_975992.1| hypothetical protein MSC_1028 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=982: 23 0 >gi|42561542|ref|NP_975993.1| hypothetical protein MSC_1029 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=983: 17 0 >gi|42561543|ref|NP_975994.1| hypothetical protein MSC_1030 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=984: 451 0 >gi|42561544|ref|NP_975995.1| transposase ISMmy1A [Mycoplasma mycoides subsp. mycoides SC str. PG1]

gene=985: 44 0 >gi|42561545|ref|NP_975996.1| variable prolipoprotein [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=986: 62 0 >gi|42561546|ref|NP_975997.1| hypothetical protein MSC_1034 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=987: 538 0 >gi|42561547|ref|NP_975998.1| IS1634CM transposase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=988: 43 0 >gi|42561548|ref|NP_975999.1| hypothetical protein MSC_1037 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=989: 66 0 >gi|42561549|ref|NP_976000.1| hypothetical protein MSC_1038 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=990: 30 0 >gi|42561550|ref|NP_976001.1| hypothetical protein MSC_1039 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=991: 307 0 >gi|42561551|ref|NP_976002.1| asparagine synthetase AsnA [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=992: 234 0 >gi|42561552|ref|NP_976003.1| hypothetical protein MSC_1041 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=993: 610 348 >gi|42561553|ref|NP_976004.1| tRNA uridine 5-carboxymethylaminomethyl modification protein GidA [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=994: 501 0 >gi|42561554|ref|NP_976005.1| proton/glutamate symporter [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=995: 447 0 >gi|42561555|ref|NP_976006.1| NADH oxidase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=996: 147 3 >gi|42561556|ref|NP_976007.1| pyrazinamidase/nicotinamidase [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=997: 426 0 >gi|42561557|ref|NP_976008.1| prolipoprotein Q [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=998: 131 1 >gi|42561558|ref|NP_976009.1| hypothetical protein MSC_1047 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=999: 53 4 >gi|42561559|ref|NP_976010.1| hypothetical protein MSC_1048 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=1000: 78 0 >gi|42561560|ref|NP_976011.1| hypothetical protein MSC_1049 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=1001: 239 0 >gi|42561561|ref|NP_976012.1| hypothetical protein MSC_1050 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=1002: 453 19 >gi|42561562|ref|NP_976013.1| hypothetical protein MSC_1051 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=1003: 133 0 >gi|42561563|ref|NP_976014.1| hypothetical protein MSC_1052 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=1004: 50 0 >gi|42561564|ref|NP_976015.1| hypothetical protein MSC_1053 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=1005: 23 0 >gi|42561565|ref|NP_976016.1| hypothetical protein MSC_1054 [Mycoplasma mycoides subsp. mycoides SC str. PG1]

gene=1006: 17 0 >gi|42561566|ref|NP_976017.1| hypothetical protein MSC_1055 [Mycoplasma mycoides subsp. mycoides SC str. PG1]
gene=1007: 451 0 >gi|42561567|ref|NP_976018.1| transposase ISMmy1I [Mycoplasma mycoides subsp. mycoides SC str. PG1]

gene=1008: 106 0 >gi|42561568|ref|NP_976019.1| variable prolipoprotein [Mycoplasma mycoides subsp. mycoides SC str. PG1]

APPENDIX B

Reference Set of Diverse Bacteria and Archaea

30 Diverse Bacteria

	Organism	Phylum
1	Acidimicrobium_ferrooxidans_DSM_10331_uid59215	Actinobacteria
2	Acidobacterium_capsulatum_ATCC_51196_uid59127	Acidobacteria
3	Aminobacterium_colombiense_DSM_12261_uid47083	Synergistetes
4	Bdellovibrio_bacteriovorus_HD100_uid61595	Deltaproteobacteria
5	Chlamydia_trachomatis_A_HAR_13_uid58333	Chlamydia
6	Chlorobium_phaeovibrioides_DSM_265_uid58129	Chlorobi
7	Coraliomargarita_akajimensis_DSM_45221_uid47079	Verrucomicrobia
8	Desulfurispirillum_indicum_S5_uid45897	Chrysiogenetes
9	Desulfurobacterium_thermolithotrophum_DSM_11699_uid63405	Aquificae
10	Elusimicrobium_minutum_Pei191_uid58949	Elusimicrobia
11	Enterococcus_faecalis_62_uid159663	Firmicutes: Lactobacillales
12	Gemmatimonas_aurantiaca_T_27_uid58813	Gemmatimonas
13	Geobacter_sulfurreducens_PCA_uid57743	Deltaprotoebacteria
14	Kyrpidia_tusciae_DSM_2912_uid48361	Firmicutes: Bacillales
15	Lawsonia_intracellularis_PHE_MN1_00_uid61575	Deltaproteobacteria
16	Leptospira_interrogans_serovar_Lai_56601_uid57881	Spirochaetes
17	Magnetococcus_MC_1_uid57833	Alphaproteobacteria
18	Mycoplasma_mycoides_SC_PG1_uid58031	Mollicutes
19	Prochlorococcus_marinus_CCMP1375_uid57995	Cyanobacteria
20	Riemerella_anatipestifer_ATCC_11845___DSM_15868_uid159857	Bacteroidetes
21	Roseiflexus_castenholzii_DSM_13941_uid58287	Chloroflexi
22	Rubrobacter_xylanophilus_DSM_9941_uid58057	Actinobacteria
23	Streptobacillus_moniliformis_DSM_12112_uid41863	Fusobacteria
24	Sulfobacillus_acidophilus_DSM_10332_uid88061	Firmicutes: Clostridia
25	Thermincola_potens_JR_uid48823	Firmicutes: Clostridia
26	Thermodesulfovibrio_yellowstonii_DSM_11347_uid59257	Nitrospirae
27	Thermus_thermophilus_HB8_uid58223	Deinococcus-Thermus
28	Thioalkalivibrio_sulfidophilus_HL_EbGr7_uid59179	Gammaproteobacteria
29	Veillonella_parvula_DSM_2008_uid41927	Firmicutes: Negativicutes
0	_Clostridium__sticklandii_uid59585	Firmicutes: Clostridia

10 Diverse Archaea

1	Aciduliprofundum_boonei_T469_uid43333	Euryarchaeota: unclassified
2	Archaeoglobus_profundus_DSM_5631_uid43493	Eurychaeota: Archaeoglobales

3 Ignisphaera_aggregans_DSM_17230_uid51875 Crenarchaeota: Thermoprotei: Desulfurococcales
4 Methanocaldococcus_infernus_ME_uid48803 Eurychaeota: Methanococcales
5 Methanocella_paludicola_SANAE_uid42887 Euryarchaeota: Methanomicrobia: Methanocellales
6 Methanopyrus_kandleri_AV19_uid57883 Euryarchaeota: Methanopyri
7 Methanosaeta_thermophila_PT_uid58469 Euryarchaeota: Methanomicrobia: Methanosarcinales
8 Pyrococcus_horikoshii_OT3_uid57753 Euryarchaeota: Thermococci
9 Staphylothermus_hellenicus_DSM_12710_uid45893 Crenarchaeota: Thermoprotei: Desulfurococcales
10 Thermofilum_pendens_Hrk_5_uid58563 Crenarchaeota: Thermoprotei

APPENDIX C

Pruned Organisms (flagged as problematic and removed from pruned trees)

Pruned Archaea

Candidatus_Korarchaeum_cryptofilum_OPF8_uid58601
Nanoarchaeum_equitans_Kin4_M_uid58009

Pruned Bacteria

Acidiphilium_cryptum_JF_5_uid58447
Acidobacterium_MP5ACTX9_uid50551
Aeromonas_salmonicida_A449_uid58631
Anaplasma_phagocytophilum_HZ_uid57951
Aster_yellows_witches_broom_phytoplasma_AYWB_uid58297
Azospirillum_B510_uid46085
Buchnera_aphidicola_Cc__Cinara_cedri__uid58579
Campylobacter_hominis_ATCC_BAA_381_uid58981
Candidatus_Accumulibacter_phosphatis_clade_IIA_UW_1_uid59207
Candidatus_Amoebophilus_asiaticus_5a2_uid58963
Candidatus_Azobacteroides_pseudotrichonymphae_genomovar__CFP2_uid59163
Candidatus_Blochmannia_floridanus_uid57999
Candidatus_Blochmannia_pennsylvanicus_BPEN_uid58329
Candidatus_Blochmannia_vafer_BVAF_uid62083
Candidatus_Carsonella_ruddii_uid58773
Candidatus_Cloacamonas_acidaminovorans_Evry_uid62959
Candidatus_Desulforudis_audaxviator_MP104C_uid59067
Candidatus_Hamiltonella_defensa_5AT__Acyrthosiphon_pisum__uid59289
Candidatus_Hodgkinia_cicadicola_Dsem_uid59311
Candidatus_Koribacter_versatilis_Ellin345_uid58479
Candidatus_Liberibacter_asiaticus_psy62_uid59227
Candidatus_Liberibacter_solanacearum_CLso_ZC1_uid61245
Candidatus_Nitrospira_defluvii_uid51175
Candidatus_Phytoplasma_australiense_uid61641
Candidatus_Phytoplasma_mali_uid59087
Candidatus_Protochlamydia_amoebophila_UWE25_uid58079
Candidatus_Puniceispirillum_marinum_IMCC1322_uid47081
Candidatus_Riesia_pediculicola_USDA_uid46841
Candidatus_Solibacter_usitatus_Ellin6076_uid58139
Candidatus_Sulcia_muelleri_CARI_uid52535
Candidatus_Sulcia_muelleri_SMDSEM_uid59393
Candidatus_Vesicomyosocius_okutanii_HA_uid59427
Candidatus_Zinderia_insecticola_CARI_uid52459

Cyanothece_PCC_7822_uid52547
Emticicia_oligotrophica_DSM_17448_uid177079
Gluconobacter_oxydans_621H_uid58239
Lactococcus_lactis_cremoris_SK11_uid57983
Macrococcus_caseolyticus_JCSC5402_uid59003
Mycoplasma_agalactiae_PG2_uid61619
Mycoplasma_arthritidis_158L3_1_uid58005
Mycoplasma_conjunctivae_uid59325
Mycoplasma_fermentans_JER_uid53543
Mycoplasma_gallisepticum_R_low__uid57993
Mycoplasma_genitalium_G37_uid57707
Mycoplasma_haemofelis_Langford_1_uid62461
Mycoplasma_hominis_ATCC_23114_uid41875
Mycoplasma_hyopneumoniae_232_uid58205
Mycoplasma_hyorhinis_HUB_1_uid51695
Mycoplasma_mobile_163K_uid58077
Mycoplasma_penetrans_HF_2_uid57729
Mycoplasma_pneumoniae_M129_uid57709
Mycoplasma_pulmonis_UAB_CTIP_uid61569
Mycoplasma_suis_KI3806_uid63665
Mycoplasma_synoviae_53_uid58061
Nitrosococcus_watsonii_C_113_uid50331
Orientia_tsutsugamushi_Boryong_uid61621
Polaromonas_naphthalenivorans_CJ2_uid58273
Porphyromonas_asaccharolytica_DSM_20707_uid66603
Runella_slithyformis_DSM_19594_uid68317
Spirosoma_linguale_DSM_74_uid43413
Synechococcus_PCC_7002_uid59137
Ureaplasma_parvum_serovar_3_ATCC_27815_uid58887
Zymomonas_mobilis_ATCC_10988_uid55403
uncultured_Termite_group_1_bacterium_phylotype_Rs_D17_uid59059

APPENDIX D

A fit using the sum of two exponentials

$$y = Fxe^{-b_1 x} + Gxe^{-b_2 x}$$

$$d = max(b1, b2)$$

Least squares regression of sum of two exponentials

$$\sum_{i=0}^{n-1} [Fe^{-b_1 x} + Ge^{-b_2 x} - y]^2$$

When multiplied out and simplified, this is equal to

$$\sum_{i=0}^{n-1} [Fe^{-b_1 x} + Ge^{-b_2 x} - y]^2$$

$$= \sum_{i=0}^{n-1} (F^2 e^{-2b_1 x} + 2FGe^{-b_1 x} e^{-b_2 x} - 2yF\, e^{-b_1 x} + G^2 e^{-2b_2 x} - 2yGe^{-b_2 x}$$

$$+ y^2)$$

$$\frac{dY}{dF} = \sum_{i=0}^{n-1} 2Fe^{-2b_1 x} + 2Ge^{-b_1 x} e^{-b_2 x} - 2ye^{-b_1 x}$$

$$\frac{dY}{dG} = \sum_{i=0}^{n-1} 2Ge^{-2b_2 x} + 2Fe^{-b_1 x} e^{-b_2 x} - 2ye^{-b_2 x}$$

$$0 = 2F \sum_{i=0}^{n-1} e^{-2b_1 x} + 2G \sum_{i=0}^{n-1} e^{-b_1 x - b_2 x} - 2 \sum_{i=0}^{n-1} ye^{-b_1 x}$$

$$0 = 2G \sum_{i=0}^{n-1} e^{-2b_2 x} + 2F \sum_{i=0}^{n-1} e^{-b_1 x - b_2 x} - 2 \sum_{i=0}^{n-1} ye^{-b_2 x}$$

$$0 = 2F\sum_{i=0}^{n-1} e^{-2b_1x} - \frac{[2F\sum_{i=0}^{n-1} e^{-b_1x-b_2x} - 2\sum_{i=0}^{n-1} ye^{-b_1x}]}{\sum\ \ e^{-2b_2x}}\sum_{i=0}^{n-1} e^{-b_qx-b_2x} - 2\sum_{i=0}^{n-1} ye^{-b_1x}$$

$$F = \frac{2\sum_{i=0}^{n-1} e^{-b_1x} - \frac{2\sum_{i=0}^{n-1} ye^{-b_2x}}{\sum_{i=0}^{n-1} e^{-2b_2x}}\sum_{i=0}^{n-1} e^{-b_1x-b_2x}}{2\sum_{i=0}^{n-1} e^{-2b_1x} - \frac{2\sum_{i=0}^{n-1} e^{-b_1x-b_2x}}{\sum_{i=0}^{n-1} e^{-2b_2x}}\sum_{i=0}^{n-1} e^{-b_1x-b_2x}}$$

For G:

$$0 = 2G\sum_{i=0}^{n-1} e^{-2b_2x} - \left(\frac{2G\sum_{i=0}^{n-1} e^{-b_1x-b_2x} - 2\sum_{i=0}^{n-1} ye^{-b_1x}}{\sum_{i=0}^{n-1} e^{-2b_1x}}\right)\sum_{i=0}^{n-1} e^{-b_1x-b_2x} - 2\sum_{i=0}^{n-1} ye^{-b_2x}$$

$$G = \frac{-\frac{\sum_{i=0}^{n-1} ye^{-b_1x}}{\sum_{i=0}^{n-1} e^{-2b_1x}}\sum_{i=0}^{n-1} e^{-b_1x-b_2x} + 2\sum_{i=0}^{n-1} ye^{-b_2x}}{2\sum_{i=0}^{n-1} e^{-2b_2x} + \frac{2\sum_{i=0}^{n-1} e^{-b_1x-b_2x}}{\sum_{i=0}^{n-1} e^{-2b_1x}}\sum_{i=0}^{n-1} e^{-b_1x-b_2x}}$$

$$G = \frac{2\sum_{i=0}^{n-1} ye^{-b_2x} - \frac{2\sum_{i=0}^{n-1} ye^{-b_1x}}{\sum_{i=0}^{n-1} e^{-2b_1x}}\sum_{i=0}^{n-1} e^{-b_1x-b_2x}}{2\sum_{i=0}^{n-1} e^{2b_2x} + \frac{\sum_{i=0}^{n-1} e^{-b_1x-b_2x}}{\sum_{i=0}^{n-1} e^{-2b_1x}}\sum_{i=0}^{n-1} e^{-b_1x-b_2x}}$$

$$F = \frac{2\sum_{i=0}^{n-1} ye^{-b_1x} - 2\frac{\sum_{i=0}^{n-1} ye^{-b_2x}}{\sum_{i=0}^{n-1} e^{-b_2x}}\sum_{i=0}^{n-1} e^{-b_1x-b_2x}}{2\sum_{i=0}^{n-1} e^{-b_1x} - 2\frac{\sum_{i=0}^{n-1} e^{-b_1x-b_2x}}{\sum_{i=0}^{n-1} e^{-2b_2x}}\sum_{i=0}^{i=1} e^{-b_1x-b_2x}}$$

For dY/dF:

$$\alpha = 2\sum_{i=0}^{n-1} e^{-2b_1x}$$

$$\beta = 2\sum_{i=0}^{n-1} e^{-b_1x-b_2x}$$

$$\gamma = 2\sum_{i=0}^{n-1} y_i e^{-b_1 x_i}$$

Original equation:

$$0 = \alpha F + \beta G - \gamma$$

For dY/dG:

$$\alpha 1 = 2\sum_{i=0}^{n-1} e^{-2b_2 x}$$

$$\beta 1 = 2\sum_{i=0}^{n-1} e^{-b_1 x - b_2 x}$$

$$\gamma 1 = 2\sum_{i=0}^{n-1} y e^{-b_2 x}$$

$$0 = \alpha 1 \times G + \beta 1 \times F - \gamma 1$$

$$G = \frac{\gamma 1 - \beta 1 \times F}{\alpha 1}$$

$$0 = \alpha F + \beta \frac{\gamma 1 - \beta 1 \times F}{\alpha 1} - \gamma$$

$$\gamma - \frac{\beta \times \gamma 1}{\alpha 1} = F\left(\alpha - \frac{\beta \beta 1}{\alpha 1}\right) \xrightarrow{yields} F = \frac{\gamma - \frac{\beta \times \gamma 1}{\alpha 1}}{\alpha - \frac{\beta \beta 1}{\alpha 1}}$$

Similarly:

$$F = \frac{\gamma - \beta G}{\alpha}$$

$$0 = \alpha 1 \times G + \beta 1 \times \frac{\gamma - \beta G}{\alpha} - \gamma 1$$

$$\gamma 1 - \frac{\beta 1 \times G}{\alpha} = G\left(\alpha 1 - \frac{\beta\beta 1}{\alpha}\right)$$

Thus:

$$G = \frac{\gamma 1 - \frac{\beta 1 \times \gamma}{\alpha}}{\alpha 1 - \frac{\beta\beta 1}{\alpha}}$$

APPENDIX E

Proteins involved in environmental adaptation identified by pair-wise hgt correction

84 proteins removed from *P. mobilis* and *M. australiensis* by pair-wise HGT correction.

>gi|332980637|ref|YP_004462078.1| alpha-mannosidase [Mahella australiensis 50-1 BON]
>gi|332980646|ref|YP_004462087.1| iron-containing alcohol dehydrogenase [Mahella australiensis 50-1 BON]
>gi|332980839|ref|YP_004462280.1| hypothetical protein Mahau_0240 [Mahella australiensis 50-1 BON]
>gi|332980911|ref|YP_004462352.1| hypothetical protein Mahau_0314 [Mahella australiensis 50-1 BON]
>gi|332981342|ref|YP_004462783.1| 3-dehydroquinate dehydratase [Mahella australiensis 50-1 BON]
>gi|332981397|ref|YP_004462838.1| hypothetical protein Mahau_0818 [Mahella australiensis 50-1 BON]
>gi|332981411|ref|YP_004462852.1| electron transfer flavoprotein alpha/beta-subunit [Mahella australiensis 50-1 BON]
>gi|332981481|ref|YP_004462922.1| chromate transporter [Mahella australiensis 50-1 BON]
>gi|332981482|ref|YP_004462923.1| chromate transporter [Mahella australiensis 50-1 BON]
>gi|332981484|ref|YP_004462925.1| winged helix family two component transcriptional regulator [Mahella australiensis 50-1 BON]
>gi|332981528|ref|YP_004462969.1| 5'-nucleotidase; exopolyphosphatase; 3'-nucleotidase [Mahella australiensis 50-1 BON]
>gi|332981564|ref|YP_004463005.1| peptidase membrane zinc metallopeptidase [Mahella australiensis 50-1 BON]
>gi|332981581|ref|YP_004463022.1| hypothetical protein Mahau_1002 [Mahella australiensis 50-1 BON]
>gi|332981642|ref|YP_004463083.1| ribosome biogenesis GTP-binding protein YlqF [Mahella australiensis 50-1 BON]
>gi|332981923|ref|YP_004463364.1| PHP domain-containing protein [Mahella australiensis 50-1 BON]
>gi|332982036|ref|YP_004463477.1| hypothetical protein Mahau_1463 [Mahella australiensis 50-1 BON]
>gi|332982392|ref|YP_004463833.1| ROK family glucokinase [Mahella australiensis 50-1 BON]
>gi|332982519|ref|YP_004463960.1| LacI family transcriptional regulator [Mahella australiensis 50-1 BON]

>gi|332982572|ref|YP_004464013.1| binding-protein-dependent transport system inner membrane protein [Mahella australiensis 50-1 BON]
>gi|332982591|ref|YP_004464032.1| ketose-bisphosphate aldolase [Mahella australiensis 50-1 BON]
>gi|332982597|ref|YP_004464038.1| methylglyoxal reductase [Mahella australiensis 50-1 BON]
>gi|332982610|ref|YP_004464051.1| NAD/NADP octopine/nopaline dehydrogenase [Mahella australiensis 50-1 BON]
>gi|332982963|ref|YP_004464404.1| HNH endonuclease [Mahella australiensis 50-1 BON]
>gi|332983053|ref|YP_004464494.1| HNH endonuclease [Mahella australiensis 50-1 BON]
>gi|332983062|ref|YP_004464503.1| hypothetical protein Mahau_2532 [Mahella australiensis 50-1 BON]
>gi|332983120|ref|YP_004464561.1| glycerol kinase [Mahella australiensis 50-1 BON]
>gi|332983147|ref|YP_004464588.1| single-strand binding protein [Mahella australiensis 50-1 BON]
>gi|332983178|ref|YP_004464619.1| G-D-S-L family lipolytic protein [Mahella australiensis 50-1 BON]
>gi|332983298|ref|YP_004464739.1| ArsR family transcriptional regulator [Mahella australiensis 50-1 BON]
>gi|332983301|ref|YP_004464742.1| redox-active disulfide protein 2 [Mahella australiensis 50-1 BON]
>gi|332983302|ref|YP_004464743.1| BFD (2Fe-2S)-binding domain-containing protein [Mahella australiensis 50-1 BON]
>gi|332983303|ref|YP_004464744.1| signal peptidase II [Mahella australiensis 50-1 BON]
>gi|332983306|ref|YP_004464747.1| resolvase domain-containing protein [Mahella australiensis 50-1 BON]
>gi|332983308|ref|YP_004464749.1| resolvase domain-containing protein [Mahella australiensis 50-1 BON]
>gi|332983309|ref|YP_004464750.1| hypothetical protein Mahau_2803 [Mahella australiensis 50-1 BON]
>gi|332983310|ref|YP_004464751.1| peptidoglycan-binding lysin domain-containing protein [Mahella australiensis 50-1 BON]
>gi|332983311|ref|YP_004464752.1| toxin secretion/phage lysis holin [Mahella australiensis 50-1 BON]
>gi|332983312|ref|YP_004464753.1| glycosyl hydrolase-like protein [Mahella australiensis 50-1 BON]
>gi|160901538|ref|YP_001567119.1| hypothetical protein Pmob_0047 [Petrotoga mobilis SJ95]
>gi|160901619|ref|YP_001567200.1| hypothetical protein Pmob_0129 [Petrotoga mobilis SJ95]
>gi|160901620|ref|YP_001567201.1| electron transfer flavoprotein alpha/beta-subunit [Petrotoga mobilis SJ95]
>gi|160901655|ref|YP_001567236.1| chromate transporter [Petrotoga mobilis SJ95]
>gi|160901656|ref|YP_001567237.1| chromate transporter [Petrotoga mobilis SJ95]

>gi|160901657|ref|YP_001567238.1| peptidase S58 DmpA [Petrotoga mobilis SJ95]
>gi|160901731|ref|YP_001567312.1| hypothetical protein Pmob_0245 [Petrotoga mobilis SJ95]
>gi|160901766|ref|YP_001567347.1| FAD-dependent pyridine nucleotide-disulfide oxidoreductase [Petrotoga mobilis SJ95]
>gi|160901880|ref|YP_001567461.1| alkyl hydroperoxide reductase/ Thiol specific antioxidant/ Mal allergen [Petrotoga mobilis SJ95]
>gi|160901900|ref|YP_001567481.1| ArsR family transcriptional regulator [Petrotoga mobilis SJ95]
>gi|160901905|ref|YP_001567486.1| glycosyl hydrolase-like protein [Petrotoga mobilis SJ95]
>gi|160901906|ref|YP_001567487.1| toxin secretion/phage lysis holin [Petrotoga mobilis SJ95]
>gi|160901907|ref|YP_001567488.1| peptidoglycan-binding LysM [Petrotoga mobilis SJ95]
>gi|160901909|ref|YP_001567490.1| hypothetical protein Pmob_0427 [Petrotoga mobilis SJ95]
>gi|160901910|ref|YP_001567491.1| resolvase domain-containing protein [Petrotoga mobilis SJ95]
>gi|160901911|ref|YP_001567492.1| resolvase domain-containing protein [Petrotoga mobilis SJ95]
>gi|160901914|ref|YP_001567495.1| lipoprotein signal peptidase [Petrotoga mobilis SJ95]
>gi|160901915|ref|YP_001567496.1| mercuric transport protein periplasmic component [Petrotoga mobilis SJ95]
>gi|160901916|ref|YP_001567497.1| GDSL family lipase [Petrotoga mobilis SJ95]
>gi|160901918|ref|YP_001567499.1| ArsR family transcriptional regulator [Petrotoga mobilis SJ95]
>gi|160901975|ref|YP_001567556.1| LacI family transcription regulator [Petrotoga mobilis SJ95]
>gi|160901996|ref|YP_001567577.1| putative N-acetylmannosamine-6-phosphate epimerase [Petrotoga mobilis SJ95]
>gi|160902000|ref|YP_001567581.1| binding-protein-dependent transport systems inner membrane component [Petrotoga mobilis SJ95]
>gi|160902035|ref|YP_001567616.1| single-strand binding protein [Petrotoga mobilis SJ95]
>gi|160902086|ref|YP_001567667.1| RnfABCDGE type electron transport complex subunit C [Petrotoga mobilis SJ95]
>gi|160902119|ref|YP_001567700.1| fructose-1,6-bisphosphate aldolase, class II [Petrotoga mobilis SJ95]
>gi|160902147|ref|YP_001567728.1| ABC transporter-like protein [Petrotoga mobilis SJ95]
>gi|160902196|ref|YP_001567777.1| redox-active disulfide protein 2 [Petrotoga mobilis SJ95]
>gi|160902264|ref|YP_001567845.1| HNH endonuclease [Petrotoga mobilis SJ95]
>gi|160902309|ref|YP_001567890.1| HSR1-like GTP-binding protein [Petrotoga mobilis SJ95]

>gi|160902374|ref|YP_001567955.1| glucose-1-phosphate adenylyltransferase [Petrotoga mobilis SJ95]
>gi|160902438|ref|YP_001568019.1| UDP-N-acetylenolpyruvoylglucosamine reductase [Petrotoga mobilis SJ95]
>gi|160902444|ref|YP_001568025.1| 5-carboxymethyl-2-hydroxymuconate Delta-isomerase [Petrotoga mobilis SJ95]
>gi|160902448|ref|YP_001568029.1| NAD/NADP octopine/nopaline dehydrogenase [Petrotoga mobilis SJ95]
>gi|160902613|ref|YP_001568194.1| homoserine dehydrogenase [Petrotoga mobilis SJ95]
>gi|160902657|ref|YP_001568238.1| 3-dehydroquinate dehydratase [Petrotoga mobilis SJ95]
>gi|160902664|ref|YP_001568245.1| peptidase membrane zinc metallopeptidase putative [Petrotoga mobilis SJ95]
>gi|160902758|ref|YP_001568339.1| hypothetical protein Pmob_1311 [Petrotoga mobilis SJ95]
>gi|160902863|ref|YP_001568444.1| S-adenosylmethionine decarboxylase proenzyme [Petrotoga mobilis SJ95]
>gi|160902880|ref|YP_001568461.1| phosphoribulokinase/uridine kinase [Petrotoga mobilis SJ95]
>gi|160902946|ref|YP_001568527.1| 2,5-didehydrogluconate reductase [Petrotoga mobilis SJ95]
>gi|160903322|ref|YP_001568903.1| L-lactate dehydrogenase [Petrotoga mobilis SJ95]
>gi|160903345|ref|YP_001568926.1| 6-phosphofructokinase [Petrotoga mobilis SJ95]

178 Proteins removed from D. lykanthroporepellens and S. fumaroxidans.

>gi|300087203|ref|YP_003757725.1| ABC transporter-like protein [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300087240|ref|YP_003757762.1| ribosome-associated GTPase EngA [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300087269|ref|YP_003757791.1| ferredoxin-dependent glutamate synthase [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300087351|ref|YP_003757873.1| phospho-2-dehydro-3-deoxyheptonate aldolase [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300087389|ref|YP_003757911.1| cupin 2 conserved barrel domain-containing protein [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300087390|ref|YP_003757912.1| integral membrane sensor signal transduction histidine kinase [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300087629|ref|YP_003758151.1| 4Fe-4S ferredoxin [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300087674|ref|YP_003758196.1| flavodoxin/nitric oxide synthase [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300087704|ref|YP_003758226.1| response regulator receiver modulated metal dependent phosphohydrolase [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300087734|ref|YP_003758256.1| thiamine pyrophosphate domain-containing TPP-binding protein [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300087735|ref|YP_003758257.1| pyruvate flavodoxin/ferredoxin oxidoreductase domain-containing protein [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300087737|ref|YP_003758259.1| pyruvate/ketoisovalerate oxidoreductase subunit gamma [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300087787|ref|YP_003758309.1| ABC transporter-like protein [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300087813|ref|YP_003758335.1| FAD-dependent pyridine nucleotide-disulfide oxidoreductase [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300087831|ref|YP_003758353.1| nickel-dependent hydrogenase large subunit [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300087834|ref|YP_003758356.1| methyl-viologen-reducing hydrogenase subunit delta [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300087835|ref|YP_003758357.1| FAD-dependent pyridine nucleotide-disulfide oxidoreductase [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300087914|ref|YP_003758436.1| ferredoxin [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300087919|ref|YP_003758441.1| CO dehydrogenase/acetyl-CoA synthase subunit delta [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088098|ref|YP_003758620.1| PAS/PAC sensor signal transduction histidine kinase [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088143|ref|YP_003758665.1| CoA-substrate-specific enzyme activase [Dehalogenimonas lykanthroporepellens BL-DC-9]

>gi|300088160|ref|YP_003758682.1| aminoglycoside phosphotransferase [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088235|ref|YP_003758757.1| histidine kinase [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088343|ref|YP_003758865.1| helicase domain-containing protein [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088706|ref|YP_003759228.1| phage transcriptional regulator AlpA [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088707|ref|YP_003759229.1| AIG2 family protein [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088708|ref|YP_003759230.1| glucosamine 6-phosphate synthetase-like protein [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088709|ref|YP_003759231.1| hypothetical protein Dehly_1632 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088710|ref|YP_003759232.1| hypothetical protein Dehly_1633 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088711|ref|YP_003759233.1| hypothetical protein Dehly_1634 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088712|ref|YP_003759234.1| BNR repeat-containing glycosyl hydrolase [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088713|ref|YP_003759235.1| hypothetical protein Dehly_1636 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088714|ref|YP_003759236.1| hypothetical protein Dehly_1637 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088715|ref|YP_003759237.1| hypothetical protein Dehly_1638 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088716|ref|YP_003759238.1| phage tail protein [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088717|ref|YP_003759239.1| baseplate J family protein [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088718|ref|YP_003759240.1| GPW/gp25 family protein [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088719|ref|YP_003759241.1| hypothetical protein Dehly_1642 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088720|ref|YP_003759242.1| hypothetical protein Dehly_1643 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088721|ref|YP_003759243.1| hypothetical protein Dehly_1644 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088722|ref|YP_003759244.1| PaaR repeat-containing protein [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088723|ref|YP_003759245.1| hypothetical protein Dehly_1646 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088724|ref|YP_003759246.1| hypothetical protein Dehly_1647 [Dehalogenimonas lykanthroporepellens BL-DC-9]

>gi|300088725|ref|YP_003759247.1| hypothetical protein Dehly_1648 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088726|ref|YP_003759248.1| peptidase M15A [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088727|ref|YP_003759249.1| phage protein D-like protein [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088728|ref|YP_003759250.1| hypothetical protein Dehly_1651 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088729|ref|YP_003759251.1| hypothetical protein Dehly_1652 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088730|ref|YP_003759252.1| hypothetical protein Dehly_1653 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088731|ref|YP_003759253.1| phage tail tape measure protein [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088732|ref|YP_003759254.1| hypothetical protein Dehly_1655 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088733|ref|YP_003759255.1| hypothetical protein Dehly_1656 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088734|ref|YP_003759256.1| tail sheath protein [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088735|ref|YP_003759257.1| hypothetical protein Dehly_1658 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088736|ref|YP_003759258.1| hypothetical protein Dehly_1659 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088739|ref|YP_003759261.1| hypothetical protein Dehly_1662 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088748|ref|YP_003759270.1| hypothetical protein Dehly_1671 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088749|ref|YP_003759271.1| hypothetical protein Dehly_1672 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088750|ref|YP_003759272.1| hypothetical protein Dehly_1673 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088751|ref|YP_003759273.1| hypothetical protein Dehly_1674 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088752|ref|YP_003759274.1| adenine-specific DNA-methyltransferase [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088755|ref|YP_003759277.1| phage head morphogenesis protein [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088756|ref|YP_003759278.1| hypothetical protein Dehly_1679 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088757|ref|YP_003759279.1| hypothetical protein Dehly_1680 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088758|ref|YP_003759280.1| hypothetical protein Dehly_1681 [Dehalogenimonas lykanthroporepellens BL-DC-9]

>gi|300088759|ref|YP_003759281.1| hypothetical protein Dehly_1682 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088760|ref|YP_003759282.1| hypothetical protein Dehly_1683 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088761|ref|YP_003759283.1| hypothetical protein Dehly_1684 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088762|ref|YP_003759284.1| integrase family protein [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088763|ref|YP_003759285.1| hypothetical protein Dehly_1686 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088764|ref|YP_003759286.1| hypothetical protein Dehly_1687 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088765|ref|YP_003759287.1| hypothetical protein Dehly_1688 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088766|ref|YP_003759288.1| DNA primase catalytic core domain-containing protein [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088767|ref|YP_003759289.1| UvrD/REP helicase [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088768|ref|YP_003759290.1| ERCC4 domain-containing protein [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088769|ref|YP_003759291.1| hypothetical protein Dehly_1692 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088770|ref|YP_003759292.1| hypothetical protein Dehly_1693 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088771|ref|YP_003759293.1| hypothetical protein Dehly_1694 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088772|ref|YP_003759294.1| hypothetical protein Dehly_1695 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088773|ref|YP_003759295.1| ECF subfamily RNA polymerase, sigma-24 subunit [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088774|ref|YP_003759296.1| hypothetical protein Dehly_1697 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088775|ref|YP_003759297.1| hypothetical protein Dehly_1698 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088776|ref|YP_003759298.1| hypothetical protein Dehly_1699 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088777|ref|YP_003759299.1| LexA family transcriptional regulator [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088778|ref|YP_003759300.1| hypothetical protein Dehly_1701 [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088779|ref|YP_003759301.1| putative bacteriophage-like protein [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|300088780|ref|YP_003759302.1| resolvase domain-containing protein [Dehalogenimonas lykanthroporepellens BL-DC-9]

>gi|300088781|ref|YP_003759303.1| putative phage-like protein [Dehalogenimonas lykanthroporepellens BL-DC-9]
>gi|116747614|ref|YP_844301.1| ABC transporter-like protein [Syntrophobacter fumaroxidans MPOB]
>gi|116748066|ref|YP_844753.1| histidine kinase [Syntrophobacter fumaroxidans MPOB]
>gi|116748363|ref|YP_845050.1| 4Fe-4S ferredoxin [Syntrophobacter fumaroxidans MPOB]
>gi|116748442|ref|YP_845129.1| ferredoxin [Syntrophobacter fumaroxidans MPOB]
>gi|116748615|ref|YP_845302.1| methyl-viologen-reducing hydrogenase subunit delta [Syntrophobacter fumaroxidans MPOB]
>gi|116749128|ref|YP_845815.1| thiamine-monophosphate kinase [Syntrophobacter fumaroxidans MPOB]
>gi|116749237|ref|YP_845924.1| 4Fe-4S ferredoxin [Syntrophobacter fumaroxidans MPOB]
>gi|116749422|ref|YP_846109.1| ferredoxin [Syntrophobacter fumaroxidans MPOB]
>gi|116749651|ref|YP_846338.1| nickel-dependent hydrogenase large subunit [Syntrophobacter fumaroxidans MPOB]
>gi|116749990|ref|YP_846677.1| acetyl-CoA decarbonylase/synthase complex subunit gamma [Syntrophobacter fumaroxidans MPOB]
>gi|116750218|ref|YP_846905.1| pyruvate/ketoisovalerate oxidoreductase subunit gamma [Syntrophobacter fumaroxidans MPOB]
>gi|116750220|ref|YP_846907.1| pyruvate flavodoxin/ferredoxin oxidoreductase domain-containing protein [Syntrophobacter fumaroxidans MPOB]
>gi|116750221|ref|YP_846908.1| thiamine pyrophosphate binding domain-containing protein [Syntrophobacter fumaroxidans MPOB]
>gi|116750357|ref|YP_847044.1| cupin [Syntrophobacter fumaroxidans MPOB]
>gi|116750393|ref|YP_847080.1| 4Fe-4S ferredoxin iron-sulfur binding domain-containing protein [Syntrophobacter fumaroxidans MPOB]
>gi|116750435|ref|YP_847122.1| 4Fe-4S ferredoxin iron-sulfur binding domain-containing protein [Syntrophobacter fumaroxidans MPOB]
>gi|116750581|ref|YP_847268.1| putative CoA-substrate-specific enzyme activase [Syntrophobacter fumaroxidans MPOB]
>gi|116750582|ref|YP_847269.1| 2-hydroxyglutaryl-CoA dehydratase subunit D [Syntrophobacter fumaroxidans MPOB]
>gi|116750830|ref|YP_847517.1| cobyrinic acid a,c-diamide synthase [Syntrophobacter fumaroxidans MPOB]
>gi|116750867|ref|YP_847554.1| 4Fe-4S ferredoxin [Syntrophobacter fumaroxidans MPOB]
>gi|116750982|ref|YP_847669.1| response regulator receiver modulated metal dependent phosphohydrolase [Syntrophobacter fumaroxidans MPOB]
>gi|116751086|ref|YP_847773.1| PAS/PAC sensor signal transduction histidine kinase [Syntrophobacter fumaroxidans MPOB]
>gi|116751192|ref|YP_847879.1| putative phage-like protein [Syntrophobacter fumaroxidans MPOB]
>gi|116751193|ref|YP_847880.1| recombinase [Syntrophobacter fumaroxidans MPOB]
>gi|116751194|ref|YP_847881.1| putative bacteriophage-like protein [Syntrophobacter fumaroxidans MPOB]

>gi|116751195|ref|YP_847882.1| hypothetical protein Sfum_3778 [Syntrophobacter fumaroxidans MPOB]
>gi|116751196|ref|YP_847883.1| SOS-response transcriptional repressor LexA [Syntrophobacter fumaroxidans MPOB]
>gi|116751197|ref|YP_847884.1| hypothetical protein Sfum_3780 [Syntrophobacter fumaroxidans MPOB]
>gi|116751198|ref|YP_847885.1| hypothetical protein Sfum_3781 [Syntrophobacter fumaroxidans MPOB]
>gi|116751199|ref|YP_847886.1| hypothetical protein Sfum_3782 [Syntrophobacter fumaroxidans MPOB]
>gi|116751200|ref|YP_847887.1| ECF subfamily RNA polymerase sigma-24 factor [Syntrophobacter fumaroxidans MPOB]
>gi|116751201|ref|YP_847888.1| hypothetical protein Sfum_3784 [Syntrophobacter fumaroxidans MPOB]
>gi|116751202|ref|YP_847889.1| hypothetical protein Sfum_3785 [Syntrophobacter fumaroxidans MPOB]
>gi|116751203|ref|YP_847890.1| hypothetical protein Sfum_3786 [Syntrophobacter fumaroxidans MPOB]
>gi|116751204|ref|YP_847891.1| hypothetical protein Sfum_3787 [Syntrophobacter fumaroxidans MPOB]
>gi|116751205|ref|YP_847892.1| ERCC4 domain-containing protein [Syntrophobacter fumaroxidans MPOB]
>gi|116751206|ref|YP_847893.1| hypothetical protein Sfum_3789 [Syntrophobacter fumaroxidans MPOB]
>gi|116751207|ref|YP_847894.1| DNA primase catalytic core [Syntrophobacter fumaroxidans MPOB]
>gi|116751210|ref|YP_847897.1| hypothetical protein Sfum_3793 [Syntrophobacter fumaroxidans MPOB]
>gi|116751211|ref|YP_847898.1| hypothetical protein Sfum_3794 [Syntrophobacter fumaroxidans MPOB]
>gi|116751212|ref|YP_847899.1| hypothetical protein Sfum_3795 [Syntrophobacter fumaroxidans MPOB]
>gi|116751213|ref|YP_847900.1| phage integrase family protein [Syntrophobacter fumaroxidans MPOB]
>gi|116751214|ref|YP_847901.1| hypothetical protein Sfum_3797 [Syntrophobacter fumaroxidans MPOB]
>gi|116751215|ref|YP_847902.1| hypothetical protein Sfum_3798 [Syntrophobacter fumaroxidans MPOB]
>gi|116751216|ref|YP_847903.1| hypothetical protein Sfum_3799 [Syntrophobacter fumaroxidans MPOB]
>gi|116751217|ref|YP_847904.1| hypothetical protein Sfum_3800 [Syntrophobacter fumaroxidans MPOB]
>gi|116751218|ref|YP_847905.1| hypothetical protein Sfum_3801 [Syntrophobacter fumaroxidans MPOB]

>gi|116751219|ref|YP_847906.1| hypothetical protein Sfum_3802 [Syntrophobacter fumaroxidans MPOB]
>gi|116751220|ref|YP_847907.1| SPP1 family phage head morphogenesis protein [Syntrophobacter fumaroxidans MPOB]
>gi|116751223|ref|YP_847910.1| D12 class N6 adenine-specific DNA methyltransferase [Syntrophobacter fumaroxidans MPOB]
>gi|116751224|ref|YP_847911.1| hypothetical protein Sfum_3807 [Syntrophobacter fumaroxidans MPOB]
>gi|116751225|ref|YP_847912.1| hypothetical protein Sfum_3808 [Syntrophobacter fumaroxidans MPOB]
>gi|116751226|ref|YP_847913.1| hypothetical protein Sfum_3809 [Syntrophobacter fumaroxidans MPOB]
>gi|116751227|ref|YP_847914.1| hypothetical protein Sfum_3810 [Syntrophobacter fumaroxidans MPOB]
>gi|116751229|ref|YP_847916.1| hypothetical protein Sfum_3812 [Syntrophobacter fumaroxidans MPOB]
>gi|116751230|ref|YP_847917.1| hypothetical protein Sfum_3813 [Syntrophobacter fumaroxidans MPOB]
>gi|116751231|ref|YP_847918.1| hypothetical protein Sfum_3814 [Syntrophobacter fumaroxidans MPOB]
>gi|116751232|ref|YP_847919.1| phage tail sheath protein [Syntrophobacter fumaroxidans MPOB]
>gi|116751233|ref|YP_847920.1| hypothetical protein Sfum_3816 [Syntrophobacter fumaroxidans MPOB]
>gi|116751234|ref|YP_847921.1| hypothetical protein Sfum_3817 [Syntrophobacter fumaroxidans MPOB]
>gi|116751235|ref|YP_847922.1| TP901 family phage tail tape measure protein [Syntrophobacter fumaroxidans MPOB]
>gi|116751236|ref|YP_847923.1| hypothetical protein Sfum_3819 [Syntrophobacter fumaroxidans MPOB]
>gi|116751237|ref|YP_847924.1| hypothetical protein Sfum_3820 [Syntrophobacter fumaroxidans MPOB]
>gi|116751238|ref|YP_847925.1| hypothetical protein Sfum_3821 [Syntrophobacter fumaroxidans MPOB]
>gi|116751239|ref|YP_847926.1| phage protein D-like [Syntrophobacter fumaroxidans MPOB]
>gi|116751240|ref|YP_847927.1| peptidase M15A [Syntrophobacter fumaroxidans MPOB]
>gi|116751241|ref|YP_847928.1| hypothetical protein Sfum_3824 [Syntrophobacter fumaroxidans MPOB]
>gi|116751242|ref|YP_847929.1| hypothetical protein Sfum_3825 [Syntrophobacter fumaroxidans MPOB]
>gi|116751243|ref|YP_847930.1| hypothetical protein Sfum_3826 [Syntrophobacter fumaroxidans MPOB]

>gi|116751244|ref|YP_847931.1| PAAR repeat-containing protein [Syntrophobacter fumaroxidans MPOB]
>gi|116751245|ref|YP_847932.1| hypothetical protein Sfum_3828 [Syntrophobacter fumaroxidans MPOB]
>gi|116751246|ref|YP_847933.1| hypothetical protein Sfum_3829 [Syntrophobacter fumaroxidans MPOB]
>gi|116751247|ref|YP_847934.1| hypothetical protein Sfum_3830 [Syntrophobacter fumaroxidans MPOB]
>gi|116751248|ref|YP_847935.1| GPW/gp25 family protein [Syntrophobacter fumaroxidans MPOB]
>gi|116751249|ref|YP_847936.1| hypothetical protein Sfum_3832 [Syntrophobacter fumaroxidans MPOB]
>gi|116751250|ref|YP_847937.1| phage tail protein [Syntrophobacter fumaroxidans MPOB]
>gi|116751251|ref|YP_847938.1| hypothetical protein Sfum_3834 [Syntrophobacter fumaroxidans MPOB]
>gi|116751253|ref|YP_847940.1| hypothetical protein Sfum_3836 [Syntrophobacter fumaroxidans MPOB]
>gi|116751254|ref|YP_847941.1| hypothetical protein Sfum_3837 [Syntrophobacter fumaroxidans MPOB]
>gi|116751255|ref|YP_847942.1| BNR repeat-containing glycosyl hydrolase [Syntrophobacter fumaroxidans MPOB]
>gi|116751256|ref|YP_847943.1| hypothetical protein Sfum_3839 [Syntrophobacter fumaroxidans MPOB]
>gi|116751257|ref|YP_847944.1| hypothetical protein Sfum_3840 [Syntrophobacter fumaroxidans MPOB]
>gi|116751258|ref|YP_847945.1| hypothetical protein Sfum_3841 [Syntrophobacter fumaroxidans MPOB]
>gi|116751259|ref|YP_847946.1| glutamine amidotransferase, class-II [Syntrophobacter fumaroxidans MPOB]
>gi|116751260|ref|YP_847947.1| hypothetical protein Sfum_3843 [Syntrophobacter fumaroxidans MPOB]
>gi|116751261|ref|YP_847948.1| phage transcriptional regulator AlpA [Syntrophobacter fumaroxidans MPOB]
>gi|116751274|ref|YP_847961.1| hypothetical protein Sfum_3857 [Syntrophobacter fumaroxidans MPOB]
>gi|116751371|ref|YP_848058.1| nickel-dependent hydrogenase large subunit [Syntrophobacter fumaroxidans MPOB]
>gi|116751480|ref|YP_848167.1| ferredoxin-dependent glutamate synthase [Syntrophobacter fumaroxidans MPOB]

BIBLIOGRAPHY

1. Woese CR, Fox GE. Phylogenetic structure of the prokaryotic domain: the primary kingdoms. Proceedings of the National Academy of Sciences of the United States of America. 1977 Nov;74(11):5088-90. PubMed PMID: 270744. Pubmed Central PMCID: 432104.
2. Lombard J, Lopez-Garcia P, Moreira D. The early evolution of lipid membranes and the three domains of life. Nat Rev Microbiol. 2012 Jul;10(7):507-15. PubMed PMID: WOS:000305471800015. English.
3. Bapteste E, O'Malley MA, Beiko RG, Ereshefsky M, Gogarten JP, Franklin-Hall L, et al. Prokaryotic evolution and the tree of life are two different things. Biology direct. 2009;4:34. PubMed PMID: 19788731. Pubmed Central PMCID: 2761302.
4. Koonin EV, Wolf YI. The fundamental units, processes and patterns of evolution, and the Tree of Life conundrum. Biology direct. 2009 Sep 29;4. PubMed PMID: WOS:000271061400001. English.
5. Sapp J. The Bacterium's Place in Nature. In: Sapp J, editor. Microbial Phylogeny and Evolution. New York: Oxford University Press; 2005.
6. Wolf M, Muller T, Dandekar T, Pollack JD. Phylogeny of Firmicutes with special reference to Mycoplasma (Mollicutes) as inferred from phosphoglycerate kinase amino acid sequence data. International journal of systematic and evolutionary microbiology. 2004 May;54(Pt 3):871-5. PubMed PMID: 15143038.
7. Pauling L, Zuckerkandl E. Chemical Paleogenetics Molecular Restoration Studies of Extinct Forms of Life. Acta Chem Scand. 1963;17:9-&. PubMed PMID: WOS:A19631198A00037. English.
8. Margoliash E, Needleman SB, Stewart JW. A Comparison of Amino Acid Sequences of Cytochromes C of Several Vertebrates. Acta Chem Scand. 1963;17:250-&. PubMed PMID: WOS:A19631198A00029. English.
9. Zuckerka.E, Pauling L. Molecules as Documents of Evolutionary History. Journal of theoretical biology. 1965;8(2):357-&. PubMed PMID: WOS:A19656369100011. English.
10. Pace NR, Olsen GJ, Woese CR. Ribosomal RNA phylogeny and the primary lines of evolutionary descent. Cell. 1986 May 9;45(3):325-6. PubMed PMID: 3084106.
11. Olsen GJ, Overbeek R, Larsen N, Marsh TL, McCaughey MJ, Maciukenas MA, et al. The Ribosomal Database Project. Nucleic Acids Res. 1992 May 11;20 Suppl:2199-200. PubMed PMID: 1598241. Pubmed Central PMCID: 333993.
12. Olsen GJ, Woese CR. Ribosomal RNA: a key to phylogeny. Faseb J. 1993 Jan;7(1):113-23. PubMed PMID: 8422957.
13. Cole JR, Chai B, Farris RJ, Wang Q, Kulam-Syed-Mohideen AS, McGarrell DM, et al. The ribosomal database project (RDP-II): introducing myRDP space and quality controlled public data. Nucleic Acids Res. 2007 Jan;35:D169-D72. PubMed PMID: WOS:000243494600035. English.
14. Quast C, Pruesse E, Yilmaz P, Gerken J, Schweer T, Yarza P, et al. The SILVA ribosomal RNA gene database project: improved data processing and web-based tools.

Nucleic Acids Res. 2013 Jan;41(D1):D590-D6. PubMed PMID: WOS:000312893300084. English.
15. Ochman H, Lawrence JG, Groisman EA. Lateral gene transfer and the nature of bacterial innovation. Nature. 2000 May 18;405(6784):299-304. PubMed PMID: 10830951.
16. Rajendhran J, Gunasekaran P. Microbial phylogeny and diversity: small subunit ribosomal RNA sequence analysis and beyond. Microbiological research. 2011 Feb 20;166(2):99-110. PubMed PMID: 20223646.
17. Doolittle WF. Phylogenetic classification and the universal tree. Science. 1999 Jun 25;284(5423):2124-8. PubMed PMID: WOS:000081099300040. English.
18. Iwabe N, Kuma K, Hasegawa M, Osawa S, Miyata T. Evolutionary relationship of archaebacteria, eubacteria, and eukaryotes inferred from phylogenetic trees of duplicated genes. Proceedings of the National Academy of Sciences of the United States of America. 1989 Dec;86(23):9355-9. PubMed PMID: 2531898. Pubmed Central PMCID: 298494.
19. Hashimoto T, Hasegawa M. Origin and early evolution of eukaryotes inferred from the amino acid sequences of translation elongation factors 1alpha/Tu and 2/G. Advances in biophysics. 1996;32:73-120. PubMed PMID: 8781286.
20. Kamla V, Henrich B, Hadding U. Phylogeny based on elongation factor Tu reflects the phenotypic features of mycoplasmas better than that based on 16S rRNA. Gene. 1996 May 24;171(1):83-7. PubMed PMID: 8675036.
21. Bui ET, Bradley PJ, Johnson PJ. A common evolutionary origin for mitochondria and hydrogenosomes. Proceedings of the National Academy of Sciences of the United States of America. 1996 Sep 3;93(18):9651-6. PubMed PMID: 8790385. Pubmed Central PMCID: 38483.
22. Kwok AY, Su SC, Reynolds RP, Bay SJ, Av-Gay Y, Dovichi NJ, et al. Species identification and phylogenetic relationships based on partial HSP60 gene sequences within the genus Staphylococcus. Int J Syst Bacteriol. 1999 Jul;49 Pt 3:1181-92. PubMed PMID: 10425778.
23. Hirt RP, Logsdon JM, Jr., Healy B, Dorey MW, Doolittle WF, Embley TM. Microsporidia are related to Fungi: evidence from the largest subunit of RNA polymerase II and other proteins. Proceedings of the National Academy of Sciences of the United States of America. 1999 Jan 19;96(2):580-5. PubMed PMID: 9892676. Pubmed Central PMCID: 15179.
24. Kim BJ, Lee SH, Lyu MA, Kim SJ, Bai GH, Chae GT, et al. Identification of mycobacterial species by comparative sequence analysis of the RNA polymerase gene (rpoB). J Clin Microbiol. 1999 Jun;37(6):1714-20. PubMed PMID: 10325313. Pubmed Central PMCID: 84932.
25. Lloyd AT, Sharp PM. Evolution of the recA gene and the molecular phylogeny of bacteria. Journal of molecular evolution. 1993 Oct;37(4):399-407. PubMed PMID: 8308907.
26. Woese CR, Olsen GJ, Ibba M, Soll D. Aminoacyl-tRNA synthetases, the genetic code, and the evolutionary process. Microbiology and molecular biology reviews :

MMBR. 2000 Mar;64(1):202-36. PubMed PMID: 10704480. Pubmed Central PMCID: 98992.
27. Ludwig W, Strunk O, Klugbauer S, Klugbauer N, Weizenegger M, Neumaier J, et al. Bacterial phylogeny based on comparative sequence analysis. Electrophoresis. 1998 Apr;19(4):554-68. PubMed PMID: WOS:000073099500015. English.
28. Lang JM, Darling AE, Eisen JA. Phylogeny of Bacterial and Archaeal Genomes Using Conserved Genes: Supertrees and Supermatrices. PloS one. 2013 Apr 25;8(4). PubMed PMID: WOS:000318341400055. English.
29. Erdos PL, Steel MA, Szekely LA, Warnow TJ. A few logs suffice to build (almost) all trees (I). Random Struct Algor. 1999 Mar;14(2):153-84. PubMed PMID: WOS:000078618000003. English.
30. Erdos PL, Steel MA, Szekely LA, Warnow TJ. A few logs suffice to build (almost) all trees: Part II. Theor Comput Sci. 1999 Jun 28;221(1-2):77-118. PubMed PMID: WOS:000081328400006. English.
31. Brown JR, Douady CJ, Italia MJ, Marshall WE, Stanhope MJ. Universal trees based on large combined protein sequence data sets. Nat Genet. 2001 Jul;28(3):281-5. PubMed PMID: WOS:000169656400023. English.
32. Bapteste E, Brinkmann H, Lee JA, Moore DV, Sensen CW, Gordon P, et al. The analysis of 100 genes supports the grouping of three highly divergent amoebae: Dictyostelium, Entamoeba, and Mastigamoeba. Proceedings of the National Academy of Sciences of the United States of America. 2002 Feb 5;99(3):1414-9. PubMed PMID: WOS:000173752500059. English.
33. Rokas A, Williams BL, King N, Carroll SB. Genome-scale approaches to resolving incongruence in molecular phylogenies. Nature. 2003 Oct 23;425(6960):798-804. PubMed PMID: WOS:000186118500036. English.
34. Wang Z, Wu M. A phylum-level bacterial phylogenetic marker database. Molecular biology and evolution. 2013 Jun;30(6):1258-62. PubMed PMID: 23519313.
35. Ciccarelli FD, Doerks T, von Mering C, Creevey CJ, Snel B, Bork P. Toward automatic reconstruction of a highly resolved tree of life. Science. 2006 Mar 3;311(5765):1283-7. PubMed PMID: WOS:000235870400041. English.
36. Sanger F, Air GM, Barrell BG, Brown NL, Coulson AR, Fiddes CA, et al. Nucleotide sequence of bacteriophage phi X174 DNA. Nature. 1977 Feb 24;265(5596):687-95. PubMed PMID: 870828.
37. Fleischmann RD, Adams MD, White O, Clayton RA, Kirkness EF, Kerlavage AR, et al. Whole-Genome Random Sequencing and Assembly of Haemophilus-Influenzae Rd. Science. 1995 Jul 28;269(5223):496-512. PubMed PMID: WOS:A1995RL49500017. English.
38. Chan EY. Advances in sequencing technology. Mutat Res-Fund Mol M. 2005 Jun 3;573(1-2):13-40. PubMed PMID: WOS:000228887700003. English.
39. Metzker ML. Sequencing technologies - the next generation. Nat Rev Genet. 2010 Jan;11(1):31-46. PubMed PMID: 19997069.
40. van Dijk EL, Auger H, Jaszczyszyn Y, Thermes C. Ten years of next-generation sequencing technology. Trends Genet. 2014 Sep;30(9):418-26. PubMed PMID: WOS:000342036000005. English.

41. Kemena C, Notredame C. Upcoming challenges for multiple sequence alignment methods in the high-throughput era. Bioinformatics. 2009 Oct 1;25(19):2455-65. PubMed PMID: 19648142. Pubmed Central PMCID: 2752613.
42. Linder CR, Suri R, Liu K, Warnow T. Benchmark datasets and software for developing and testing methods for large-scale multiple sequence alignment and phylogenetic inference. PLoS currents. 2010;2:RRN1195. PubMed PMID: 21113335. Pubmed Central PMCID: 2989560.
43. Blair C, Murphy RW. Recent trends in molecular phylogenetic analysis: where to next? The Journal of heredity. 2011 Jan-Feb;102(1):130-8. PubMed PMID: 20696667.
44. Stamatakis A. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. Bioinformatics. 2014 May 1;30(9):1312-3. PubMed PMID: WOS:000336095100024. English.
45. Edgar RC, Batzoglou S. Multiple sequence alignment. Current opinion in structural biology. 2006 Jun;16(3):368-73. PubMed PMID: 16679011.
46. Land M, Hauser L, Jun SR, Nookaew I, Leuze MR, Ahn TH, et al. Insights from 20 years of bacterial genome sequencing. Functional & integrative genomics. 2015 Mar;15(2):141-61. PubMed PMID: 25722247. Pubmed Central PMCID: 4361730.
47. Schneiker S, Perlova O, Kaiser O, Gerth K, Alici A, Altmeyer MO, et al. Complete genome sequence of the myxobacterium Sorangium cellulosum. Nature biotechnology. 2007 Nov;25(11):1281-9. PubMed PMID: 17965706.
48. Gans J, Wolinsky M, Dunbar J. Computational improvements reveal great bacterial diversity and high metal toxicity in soil. Science. 2005 Aug 26;309(5739):1387-90. PubMed PMID: WOS:000231543300048. English.
49. Sogin ML, Morrison HG, Huber JA, Mark Welch D, Huse SM, Neal PR, et al. Microbial diversity in the deep sea and the underexplored "rare biosphere". Proceedings of the National Academy of Sciences of the United States of America. 2006 Aug 8;103(32):12115-20. PubMed PMID: WOS:000239701900053. English.
50. Vinga S, Almeida J. Alignment-free sequence comparison-a review. Bioinformatics. 2003 Mar 1;19(4):513-23. PubMed PMID: 12611807.
51. Bonham-Carter O, Steele J, Bastola D. Alignment-free genetic sequence comparisons: a review of recent approaches by word analysis. Briefings in bioinformatics. 2014 Nov;15(6):890-905. PubMed PMID: 23904502. Pubmed Central PMCID: 4296134.
52. Haubold B. Alignment-free phylogenetics and population genetics. Briefings in bioinformatics. 2014 May;15(3):407-18. PubMed PMID: 24291823.
53. Song K, Ren J, Reinert G, Deng M, Waterman MS, Sun F. New developments of alignment-free sequence comparison: measures, statistics and next-generation sequencing. Briefings in bioinformatics. 2014 May;15(3):343-53. PubMed PMID: 24064230. Pubmed Central PMCID: 4017329.
54. Ragan MA, Bernard G, Chan CX. Molecular phylogenetics before sequences: oligonucleotide catalogs as k-mer spectra. RNA biology. 2014;11(3):176-85. PubMed PMID: 24572375. Pubmed Central PMCID: 4008546.

55. Konstantinidis KT, Tiedje JM. Prokaryotic taxonomy and phylogeny in the genomic era: advancements and challenges ahead. Current opinion in microbiology. 2007 Oct;10(5):504-9. PubMed PMID: 17923431.
56. Qi J, Luo H, Hao BL. CVTree: a phylogenetic tree reconstruction tool based on whole genomes. Nucleic Acids Res. 2004 Jul 1;32:W45-W7. PubMed PMID: WOS:000222273100009. English.
57. Zuo G, Xu Z, Hao B. Phylogeny and Taxonomy of Archaea: A Comparison of the Whole-Genome-Based CVTree Approach with 16S rRNA Sequence Analysis. Life. 2015;5(1):949-68. PubMed PMID: 25789552. Pubmed Central PMCID: 4390887.
58. Hao BL, Qi J, Wang B. Prokaryotic phylogeny based on complete genomes without sequence alignment. Proceedings of the International Symposium on Frontiers of Science. 2003:441-4. PubMed PMID: WOS:000186708100042. English.
59. Sims GE, Jun SR, Wua GA, Kim SH. Alignment-free genome comparison with feature frequency profiles (FFP) and optimal resolutions. Proceedings of the National Academy of Sciences of the United States of America. 2009 Feb 24;106(8):2677-82. PubMed PMID: WOS:000263652900039. English.
60. Sims GE, Jun SR, Wu GA, Kim SH. Whole-genome phylogeny of mammals: Evolutionary information in genic and nongenic regions. Proceedings of the National Academy of Sciences of the United States of America. 2009 Oct 6;106(40):17077-82. PubMed PMID: WOS:000270537500040. English.
61. Sims GE, Kim SH. Whole-genome phylogeny of Escherichia coli/Shigella group by feature frequency profiles (FFPs). Proceedings of the National Academy of Sciences of the United States of America. 2011 May 17;108(20):8329-34. PubMed PMID: WOS:000290719600052. English.
62. Blaisdell BE. A measure of the similarity of sets of sequences not requiring sequence alignment. Proceedings of the National Academy of Sciences of the United States of America. 1986 Jul;83(14):5155-9. PubMed PMID: 3460087. Pubmed Central PMCID: 323909.
63. Torney DC, Burks C, Davison D, Sirotkin KM. Computation of D2 - a Measure of Sequence Dissimilarity. Sfi S Sci C. 1990;7:109-25. PubMed PMID: WOS:A1990BQ92N00011. English.
64. Chan CX, Bernard G, Poirion O, Hogan JM, Ragan MA. Inferring phylogenies of evolving sequences without multiple sequence alignment. Scientific reports. 2014;4:6504. PubMed PMID: 25266120. Pubmed Central PMCID: 4179140.
65. Yi H, Jin L. Co-phylog: an assembly-free phylogenomic approach for closely related organisms. Nucleic Acids Res. 2013 Apr;41(7):e75. PubMed PMID: 23335788. Pubmed Central PMCID: 3627563.
66. Horwege S, Lindner S, Boden M, Hatje K, Kollmar M, Leimeister CA, et al. Spaced words and kmacs: fast alignment-free sequence comparison based on inexact word matches. Nucleic Acids Res. 2014 Jul 1;42(W1):W7-W11. PubMed PMID: WOS:000339715000003. English.
67. Ulitsky I, Burstein D, Tuller T, Chor B. The average common substring approach to phylogenomic reconstruction. J Comput Biol. 2006 Mar;13(2):336-50. PubMed PMID: WOS:000236954700015. English.

68. Leimeister CA, Morgenstern B. kmacs: the k-mismatch average common substring approach to alignment-free sequence comparison. Bioinformatics. 2014 Jul 15;30(14):2000-8. PubMed PMID: WOS:000339814300008. English.
69. Haubold B, Pfaffelhuber P, Domazet-Loso M, Wiehe T. Estimating mutation distances from unaligned genomes. J Comput Biol. 2009 Oct;16(10):1487-500. PubMed PMID: 19803738.
70. Sharma V. Thankachan SPC, Yongchao Liu, Ambujam Krishnan, Srinivas Aluru. A greedy alignment-free distance estimator for phylogenetic inference. Conference: 5th IEEE International Conference on Computational Advances in Bio and Medical Sciences. 2016.
71. Aluru S, Apostolico A, Thankachan SV. Efficient Alignment Free Sequence Comparison with Bounded Mismatches. Lect N Bioinformat. 2015;9029:1-12. PubMed PMID: WOS:000361983900001. English.
72. Comin M, Verzotto D. Alignment-free phylogeny of whole genomes using underlying subwords. Algorithm Mol Biol. 2012 Dec 6;7. PubMed PMID: WOS:000313792900001. English.
73. Qi J, Wang B, Hao BI. Whole proteome prokaryote phylogeny without sequence alignment: A K-string composition approach. Journal of molecular evolution. 2004 Jan;58(1):1-11. PubMed PMID: WOS:000188112200001. English.
74. Zhao Xu BH. CVTree update: a newly designed phylogenetic study platform using composition vectors and whole genomes. Nucleic Acids Res. 2009;37 (Web Server issue):W174-W8.
75. Guanghong Zuo BH. CVTree3 web server for whole-genome-based and alignment-free prokaryotic phyogeny and taxonomy. Genomics Proteomes & Bioinformatics. 2015;13:321-31.
76. Zuo G, Xu Z, Hao B. Shigella strains are not clones of Escherichia coli but sister species in the genus Escherichia. Genomics, proteomics & bioinformatics. 2013 Feb;11(1):61-5. PubMed PMID: 23395177. Pubmed Central PMCID: 4357666.
77. Fu M, Deng R, Wang J, Wang X. Whole-genome phylogenetic analysis of herpesviruses. Acta virologica. 2008;52(1):31-40. PubMed PMID: 18459833.
78. Wang H, Xu Z, Gao L, Hao B. A fungal phylogeny based on 82 complete genomes using the composition vector method. Bmc Evol Biol. 2009;9:195. PubMed PMID: 19664262. Pubmed Central PMCID: 3087519.
79. Gao L, Qi J, Sun J, Hao B. Prokaryote phylogeny meets taxonomy: an exhaustive comparison of composition vector trees with systematic bacteriology. Science in China Series C, Life sciences / Chinese Academy of Sciences. 2007 Oct;50(5):587-99. PubMed PMID: 17879055.
80. Chu KH, Qi J, Yu ZG, Anh V. Origin and phylogeny of chloroplasts revealed by a simple correlation analysis of complete genomes. Molecular biology and evolution. 2004 Jan;21(1):200-6. PubMed PMID: WOS:000189149300021. English.
81. Kullback S, Leibler RA. On Information and Sufficiency. Ann Math Stat. 1951;22(1):79-86. PubMed PMID: WOS:A1951UM01800005. English.
82. Kullback S. The Kullback-Leibler Distance. Am Stat. 1987 Nov;41(4):340-. PubMed PMID: WOS:A1987L300600025. English.

83. Lin JH. Divergence Measures Based on the Shannon Entropy. Ieee T Inform Theory. 1991 Jan;37(1):145-51. PubMed PMID: WOS:A1991EM05900015. English.
84. Ma B, Tromp J, Li M. PatternHunter: faster and more sensitive homology search. Bioinformatics. 2002 Mar;18(3):440-5. PubMed PMID: 11934743.
85. Li M, Ma B, Kisman D, Tromp J. PatternHunter II: highly sensitive and fast homology search. Genome informatics International Conference on Genome Informatics. 2003;14:164-75. PubMed PMID: 15706531.
86. Marcus Boden MS, Sebastian Horwege, Sebastian Lindner, Chris Leimeister, Burkhard Morgenstern. Alignment-free sequence comparison with spaced k-mers. German Conference on Bioinformatics. 2013:21–31.
87. Morgenstern B, Zhu BY, Horwege S, Leimeister CA. Estimating Evolutionary Distances from Spaced-Word Matches. Algorithms in Bioinformatics. 2014;8701:161-73. PubMed PMID: WOS:000343880000013. English.
88. Leimeister CA, Boden M, Horwege S, Lindner S, Morgenstern B. Fast alignment-free sequence comparison using spaced-word frequencies. Bioinformatics. 2014 Jul 15;30(14):1991-9. PubMed PMID: WOS:000339814300007. English.
89. Mohamed Ibrahim Abouelhoda SK, Enno Ohlebusch. Replacing suffix trees with enhanced suffix arrays. Journal of Discrete Algorithms. 2004:53-86.
90. Haubold B, Pierstorff N, Moller F, Wiehe T. Genome comparison without alignment using shortest unique substrings. BMC bioinformatics. 2005;6:123. PubMed PMID: 15910684. Pubmed Central PMCID: 1166540.
91. Haubold B, Wiehe T. How repetitive are genomes? BMC bioinformatics. 2006;7:541. PubMed PMID: 17187668. Pubmed Central PMCID: 1769404.
92. Ames SK, Hysom DA, Gardner SN, Lloyd GS, Gokhale MB, Allen JE. Scalable metagenomic taxonomy classification using a reference genome database. Bioinformatics. 2013 Sep 15;29(18):2253-60. PubMed PMID: WOS:000323943200005. English.
93. Lawrence JG, Ochman H. Reconciling the many faces of lateral gene transfer. Trends Microbiol. 2002 Jan;10(1):1-4. PubMed PMID: 11755071.
94. Syvanen M. Evolutionary implications of horizontal gene transfer. Annual review of genetics. 2012;46:341-58. PubMed PMID: 22934638.
95. Medigue C, Rouxel T, Vigier P, Henaut A, Danchin A. Evidence for horizontal gene transfer in Escherichia coli speciation. Journal of molecular biology. 1991 Dec 20;222(4):851-6. PubMed PMID: 1762151.
96. Koonin EV, Makarova KS, Aravind L. Horizontal gene transfer in prokaryotes: quantification and classification. Annual review of microbiology. 2001;55:709-42. PubMed PMID: 11544372.
97. Li X, Xing J, Li B, Yu F, Lan X, Liu J. Phylogenetic analysis reveals the coexistence of interfamily and interspecies horizontal gene transfer in Streptococcus thermophilus strains isolated from the same yoghurt. Molecular phylogenetics and evolution. 2013 Oct;69(1):286-92. PubMed PMID: 23769954.
98. Moura A, Savageau MA, Alves R. Relative amino acid composition signatures of organisms and environments. PloS one. 2013;8(10):e77319. PubMed PMID: 24204807. Pubmed Central PMCID: 3808408.

99. Simonsen M, Mailund T, Pedersen CNS. Inference of Large Phylogenies Using Neighbour-Joining. Biomedical Engineering Systems and Technologies. 2011;127:334-44. PubMed PMID: WOS:000289177200026. English.
100. Kendall MM, Liu Y, Sieprawska-Lupa M, Stetter KO, Whitman WB, Boone DR. Methanococcus aeolicus sp. nov., a mesophilic, methanogenic archaeon from shallow and deep marine sediments. International journal of systematic and evolutionary microbiology. 2006 Jul;56(Pt 7):1525-9. PubMed PMID: 16825624.
101. Brochier C, Gribaldo S, Zivanovic Y, Confalonieri F, Forterre P. Nanoarchaea: representatives of a novel archaeal phylum or a fast-evolving euryarchaeal lineage related to Thermococcales? Genome biology. 2005;6(5):R42. PubMed PMID: 15892870. Pubmed Central PMCID: 1175954.
102. Nakabachi A, Yamashita A, Toh H, Ishikawa H, Dunbar HE, Moran NA, et al. The 160-kilobase genome of the bacterial endosymbiont Carsonella. Science. 2006 Oct 13;314(5797):267. PubMed PMID: 17038615.
103. Kunisawa T. The phylogenetic placement of the non-phototrophic, Gram-positive thermophile 'Thermobaculum terrenum' and branching orders within the phylum 'Chloroflexi' inferred from gene order comparisons. International journal of systematic and evolutionary microbiology. 2011 Aug;61(Pt 8):1944-53. PubMed PMID: 20833875.
104. Lefevre CT, Menguy N, Abreu F, Lins U, Posfai M, Prozorov T, et al. A cultured greigite-producing magnetotactic bacterium in a novel group of sulfate-reducing bacteria. Science. 2011 Dec 23;334(6063):1720-3. PubMed PMID: 22194580.
105. Nishida H, Beppu T, Ueda K. Whole-genome comparison clarifies close phylogenetic relationships between the phyla Dictyoglomi and Thermotogae. Genomics. 2011 Nov;98(5):370-5. PubMed PMID: 21851855.
106. Gupta RS, Lorenzini E. Phylogeny and molecular signatures (conserved proteins and indels) that are specific for the Bacteroidetes and Chlorobi species. Bmc Evol Biol. 2007 May 8;7. PubMed PMID: WOS:000247144100001. English.
107. Iino T, Mori K, Uchino Y, Nakagawa T, Harayama S, Suzuki K. Ignavibacterium album gen. nov., sp. nov., a moderately thermophilic anaerobic bacterium isolated from microbial mats at a terrestrial hot spring and proposal of Ignavibacteria classis nov., for a novel lineage at the periphery of green sulfur bacteria. International journal of systematic and evolutionary microbiology. 2010 Jun;60(Pt 6):1376-82. PubMed PMID: 19671715.
108. Wagner M, Horn M. The Planctomycetes, Verrucomicrobia, Chlamydiae and sister phyla comprise a superphylum with biotechnological and medical relevance. Curr Opin Biotech. 2006 Jun;17(3):241-9. PubMed PMID: WOS:000238846300004. English.
109. Lee KC, Webb RI, Janssen PH, Sangwan P, Romeo T, Staley JT, et al. Phylum Verrucomicrobia representatives share a compartmentalized cell plan with members of bacterial phylum Planctomycetes. Bmc Microbiol. 2009 Jan 8;9. PubMed PMID: WOS:000264157600001. English.
110. Botero LM, Brown KB, Brumefield S, Burr M, Castenholz RW, Young M, et al. Thermobaculum terrenum gen. nov., sp. nov.: a non-phototrophic gram-positive thermophile representing an environmental clone group related to the Chloroflexi (green non-sulfur bacteria) and Thermomicrobia. Archives of microbiology. 2004 Apr;181(4):269-77. PubMed PMID: 14745485.

111. Yarza P, Munoz R. The All-Species Living Tree Project. Method Microbiol. 2014;41:45-59. PubMed PMID: WOS:000349345800004. English.
112. Yarza P, Richter M, Peplies J, Euzeby J, Amann R, Schleifer KH, et al. The All-Species Living Tree project: A 16S rRNA-based phylogenetic tree of all sequenced type strains. Systematic and applied microbiology. 2008 Sep;31(4):241-50. PubMed PMID: WOS:000260357000001. English.
113. Takaki Y, Shimamura S, Nakagawa S, Fukuhara Y, Horikawa H, Ankai A, et al. Bacterial Lifestyle in a Deep-sea Hydrothermal Vent Chimney Revealed by the Genome Sequence of the Thermophilic Bacterium Deferribacter desulfuricans SSM1. DNA Res. 2010 Jun;17(3):123-37. PubMed PMID: WOS:000279411300001. English.
114. Maidak BL, Cole JR, Parker CT, Garrity GM, Larsen N, Li B, et al. A new version of the RDP (Ribosomal Database Project). Nucleic Acids Res. 1999 Jan 1;27(1):171-3. PubMed PMID: WOS:000077983000042. English.
115. Maidak BL, Cole JR, Lilburn TG, Parker CT, Saxman PR, Stredwick JM, et al. The RDP (Ribosomal Database Project) continues. Nucleic Acids Res. 2000 Jan 1;28(1):173-4. PubMed PMID: WOS:000084896300050. English.
116. Maidak BL, Cole JR, Lilburn TG, Parker CT, Saxman PR, Farris RJ, et al. The RDP-II (Ribosomal Database Project). Nucleic Acids Res. 2001 Jan 1;29(1):173-4. PubMed PMID: WOS:000166360300046. English.
117. Felsenstein J. PHYLIP (Phylogeny Inference Package). 3.6 ed. Department of Genome Sciences, University of Washington, Seattle2005.
118. Letunic I, Bork P. Interactive Tree Of Life (iTOL): an online tool for phylogenetic tree display and annotation. Bioinformatics. 2007 Jan 1;23(1):127-8. PubMed PMID: 17050570.
119. Letunic I, Bork P. Interactive Tree Of Life v2: online annotation and display of phylogenetic trees made easy. Nucleic Acids Res. 2011 Jul;39(Web Server issue):W475-8. PubMed PMID: 21470960. Pubmed Central PMCID: 3125724.
120. Eveleigh RJM, Meehan CJ, Archibald JM, Beiko RG. Being Aquifex aeolicus: Untangling a Hyperthermophile's Checkered Past. Genome biology and evolution. 2013;5(12):2478-97. PubMed PMID: WOS:000329250400020. English.
121. Nelson KE, Clayton RA, Gill SR, Gwinn ML, Dodson RJ, Haft DH, et al. Evidence for lateral gene transfer between Archaea and bacteria from genome sequence of Thermotoga maritima. Nature. 1999 May 27;399(6734):323-9. PubMed PMID: 10360571.
122. Garcia-Vallve S, Romeu A, Palau J. Horizontal gene transfer in bacterial and archaeal complete genomes. Genome Res. 2000 Nov;10(11):1719-25. PubMed PMID: 11076857. Pubmed Central PMCID: 310969.
123. Metcalf JA, Funkhouser-Jones LJ, Brileya K, Reysenbach AL, Bordenstein SR. Antibacterial gene transfer across the tree of life. eLife. 2014;3. PubMed PMID: 25422936. Pubmed Central PMCID: 4241558.
124. Touchon M, Hoede C, Tenaillon O, Barbe V, Baeriswyl S, Bidet P, et al. Organised Genome Dynamics in the Escherichia coli Species Results in Highly Diverse Adaptive Paths. Plos Genet. 2009 Jan;5(1). PubMed PMID: WOS:000266221100026. English.

125. Clermont O, Gordon D, Denamur E. Guide to the various phylogenetic classification schemes for Escherichia coli and the correspondence among schemes. Microbiology. 2015 May;161(Pt 5):980-8. PubMed PMID: 25714816.
126. Turrientes MC, Gonzalez-Alba JM, del Campo R, Baquero MR, Canton R, Baquero F, et al. Recombination blurs phylogenetic groups routine assignment in Escherichia coli: setting the record straight. PloS one. 2014;9(8):e105395. PubMed PMID: 25137251. Pubmed Central PMCID: 4138120.
127. Sahl JW, Morris CR, Emberger J, Fraser CM, Ochieng JB, Juma J, et al. Defining the Phylogenomics of Shigella Species: a Pathway to Diagnostics. J Clin Microbiol. 2015 Mar;53(3):951-60. PubMed PMID: WOS:000350204600029. English.
128. Paulsen IT, Banerjei L, Myers GS, Nelson KE, Seshadri R, Read TD, et al. Role of mobile DNA in the evolution of vancomycin-resistant Enterococcus faecalis. Science. 2003 Mar 28;299(5615):2071-4. PubMed PMID: 12663927.
129. Li M, Du X, Villaruz AE, Diep BA, Wang D, Song Y, et al. MRSA epidemic linked to a quickly spreading colonization and virulence determinant. Nature medicine. 2012 May;18(5):816-9. PubMed PMID: 22522561. Pubmed Central PMCID: 3378817.
130. Junier T, Zdobnov EM. The Newick utilities: high-throughput phylogenetic tree processing in the UNIX shell. Bioinformatics. 2010 Jul 1;26(13):1669-70. PubMed PMID: 20472542. Pubmed Central PMCID: 2887050.
131. Robinson DF, Foulds LR. Comparison of Phylogenetic Trees. Mathematical biosciences. 1981;53(1-2):131-47. PubMed PMID: WOS:A1981LB66300008. English.
132. Gupta RS, Bhandari V, Naushad HS. Molecular Signatures for the PVC Clade (Planctomycetes, Verrucomicrobia, Chlamydiae, and Lentisphaerae) of Bacteria Provide Insights into Their Evolutionary Relationships. Frontiers in microbiology. 2012;3:327. PubMed PMID: 23060863. Pubmed Central PMCID: 3444138.
133. Fuerst JA. The PVC superphylum: exceptions to the bacterial definition? Antonie van Leeuwenhoek. 2013 Oct;104(4):451-66. PubMed PMID: 23912444.
134. Zhang WW, Lu ZT. Phylogenomic evaluation of members above the species level within the phylum Firmicutes based on conserved proteins. Env Microbiol Rep. 2015 Apr;7(2):273-81. PubMed PMID: WOS:000351407300014. English.
135. Zhaxybayeva O, Swithers KS, Lapierre P, Fournier GP, Bickhart DM, DeBoy RT, et al. On the chimeric nature, thermophilic origin, and phylogenetic placement of the Thermotogales. Proceedings of the National Academy of Sciences of the United States of America. 2009 Apr 7;106(14):5865-70. PubMed PMID: 19307556. Pubmed Central PMCID: 2667022.
136. Mira A, Pushker R, Legault BA, Moreira D, Rodriguez-Valera F. Evolutionary relationships of Fusobacterium nucleatum based on phylogenetic analysis and comparative genomics. Bmc Evol Biol. 2004 Nov 26;4. PubMed PMID: WOS:000226141200001. English.
137. Sutcliffe IC. A phylum level perspective on bacterial cell envelope architecture. Trends Microbiol. 2010 Oct;18(10):464-70. PubMed PMID: WOS:000283399900005. English.
138. Rainey FA, Stackebrandt E. Transfer of the Type Species of the Genus Thermobacteroides to the Genus Thermoanaerobacter as Thermoanaerobacter-

Acetoethylicus (Ben-Bassat and Zeikus 1981) Comb-Nov, Description of Coprothermobacter Gen-Nov, and Reclassification of Thermobacteroides-Proteolyticus as Coprothermobacter-Proteolyticus (Ollivier Et-Al 1985) Comb-Nov. Int J Syst Bacteriol. 1993 Oct;43(4):857-9. PubMed PMID: WOS:A1993MC21000035. English.
139. Huntemann M, Lu M, Nolan M, Lapidus A, Lucas S, Hammon N, et al. Complete genome sequence of the thermophilic sulfur-reducer Hippea maritima type strain (MH(2)). Standards in genomic sciences. 2011 Jul 1;4(3):303-11. PubMed PMID: 21886857. Pubmed Central PMCID: 3156395.
140. Williams KP, Kelly DP. Proposal for a new class within the phylum Proteobacteria, Acidithiobacillia classis nov., with the type order Acidithiobacillales, and emended description of the class Gammaproteobacteria (vol 63, pg 2901, 2013). International journal of systematic and evolutionary microbiology. 2013 Sep;63:3547-8. PubMed PMID: WOS:000326426100064. English.
141. Williams KP, Gillespie JJ, Sobral BWS, Nordberg EK, Snyder EE, Shallom JM, et al. Phylogeny of Gammaproteobacteria. J Bacteriol. 2010 May;192(9):2305-14. PubMed PMID: WOS:000276685800003. English.
142. Moe WM, Yan J, Nobre MF, da Costa MS, Rainey FA. Dehalogenimonas lykanthroporepellens gen. nov., sp. nov., a reductively dehalogenating bacterium isolated from chlorinated solvent-contaminated groundwater. International journal of systematic and evolutionary microbiology. 2009 Nov;59(Pt 11):2692-7. PubMed PMID: 19625421.
143. Loffler FE, Yan J, Ritalahti KM, Adrian L, Edwards EA, Konstantinidis KT, et al. Dehalococcoides mccartyi gen. nov., sp. nov., obligately organohalide-respiring anaerobic bacteria relevant to halogen cycling and bioremediation, belong to a novel bacterial class, Dehalococcoidia classis nov., order Dehalococcoidales ord. nov. and family Dehalococcoidaceae fam. nov., within the phylum Chloroflexi. International journal of systematic and evolutionary microbiology. 2013 Feb;63(Pt 2):625-35. PubMed PMID: 22544797.
144. Williams KP, Gillespie JJ, Sobral BW, Nordberg EK, Snyder EE, Shallom JM, et al. Phylogeny of gammaproteobacteria. J Bacteriol. 2010 May;192(9):2305-14. PubMed PMID: 20207755. Pubmed Central PMCID: 2863478.
145. Matte-Tailliez O, Brochier C, Forterre P, Philippe H. Archaeal phylogeny based on ribosomal proteins. Molecular biology and evolution. 2002 May;19(5):631-9. PubMed PMID: 11961097.
146. Bachvaroff TR, Handy SM, Place AR, Delwiche CF. Alveolate phylogeny inferred using concatenated ribosomal proteins. The Journal of eukaryotic microbiology. 2011 May-Jun;58(3):223-33. PubMed PMID: 21518081.