12-2018

Establishing Benchmark Criteria for Single Chromosome Bacterial Genome Assembly

Timothy Krause
tkrause1@blc.edu

ESTABLISHING BENCHMARK CRITERIA FOR SINGLE CHROMOSOME BACTERIAL GENOME ASSEMBLY

by

Timothy J. Krause

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Master of Science

Major: Biological Sciences

Under the Supervision of Joshua R. Herr

Lincoln, Nebraska

December, 2018

ESTABLISHING BENCHMARK CRITERIA FOR SINGLE CHROMOSOME BACTERIAL GENOME ASSEMBLY

Timothy J. Krause, M.S.
University of Nebraska, 2018

Adviser: Joshua R. Herr

Adequate recommendations for the amount and types of sequencing data necessary to optimize the recovery of single chromosomes from bacterial sequencing projects do not exist. Broad estimates for coverage depths needed to recover complete bacterial genomes are present in the literature, but required sequencing depths across bacterial and archaeal phylogenies needed for high-quality assembly are not known. Additionally, correlations between genomic complexity and expected quality of assembly have not been properly defined. Furthermore, the capabilities of multiplexing (sequencing more than one sample simultaneously on one flow cell) with long-read sequencing platforms in order to recover complete bacterial chromosomes are poorly documented. We first preface our research by discussing the benefits and challenges surrounding assembly of single chromosome bacterial genomes. Then, in order to address the role of genomic variability on genome assembly quality, we selected a clade of closely related *Escherichia coli* strains and assessed how strain-level genomic variation leads to differences in genome assembly quality. While variation in assembly quality among highly similar strains does occur, we show that the depth at which increased coverage does not improve assembly contiguity can be ascertained for strains of highly similar bacteria. We also show that there are significant correlations between genomic traits – such as genome size, repeat content, and number of coding sequences – and the resulting genome assembly quality. Furthermore, we simulated long-read data based on standard multiplexed read profiles of a phylogenetically diverse array of bacteria and archaea and found that although limitations due to genome size and repeat complexity exist, long-read x8 multiplexed data are able to complete many bacterial genomes without the need for additional short-read sequencing. This research provides a series of criteria for why short-read sequencing and assembly often does not result in the generation of complete genome assemblies, and how multiplexed, long-read data can greatly reduce time and financial resources for many bacterial and archaeal sequencing projects.

Keywords: Genome Assembly, Sequencing, Multiplexing, Comparative Genomics

ACKNOWLEDGEMENTS

We thank Hasan Otu, Etsuko Moriyama, James Krause, and all the members of the Herr Laboratory for constructive feedback and comments during the formulation of this research.

This work was completed utilizing the Holland Computing Center of the University of Nebraska, which receives support from the Nebraska Research Initiative.

GRANT INFORMATION

TJK and JRH both acknowledge support from US National Science Foundation Grant #1560163 and US National Science Foundation-EPSCoR Grant #1557417 that contributed to this research. JRH also recognizes research support from National Air & Space Administration Grant 80NSSC17K0737 and National Institute of Justice Grant 2017-IJ-CX-0025, as well as start-up funding from the University of Nebraska Agricultural Research Division and the Office of Research and Economic Development, which supported research and provided resources in his laboratory group.

Table of Contents

Chapter 1

Complete Genomes of Bacteria and Archaea Advance our Knowledge

1.1 Background

Improvements in sequencing technologies have transformed microbial whole-genome sequencing from the lengthy, labor-intensive effort of ”primer-walking” via Sanger sequencing [1], into a much cheaper, quicker, and streamlined research process. The knowledge gained through sequencing projects has revolutionized our view of the microbial world, most predominantly in the sub-fields of microbial diversity and taxonomy [2], evolution and phylogenetics [3], and pathogenicity [4].

Currently, high-throughput nucleotide sequencing technologies can generally be categorized by read length outputs. Short-read technologies generate relatively short (25-300 bp) low error rate reads (generally less than 0.01, although this depends on position of the read [5]) and comes with the advantage of having the lowest available cost per gigabase (Gb) on the market ($30 to $150, depending of the level of output, for Illumina’s HiSeq 2500 platform [6])). Short-read technologies generally come with the disadvantage of coverage bias in regions of high or low GC content [7], and they are not well suited to resolve repeated genomic regions due to the limits of read length [8, 9]. Characteristics of long reads – the two current market leaders in long-read sequencing technologies being PacBio and Oxford Nanopore – depend on sequencing platform and chemistry.

PacBio's newest platform, the Sequel system, can generate reads with a maximum read length of ≈100,000, a mean read length of ≈20,000, and a mean error rate of 11 to 14% [10]. Adequate estimations of cost per Gb do not yet exist for the Sequel system due to its recent release, but PacBio's second to most recent platform, the RS II, comes with a cost of approximately $1,000 per Gb, and the Sequel system's output ranges between x3 to x10 times greater than the RS II [6]. Estimating read characteristics generated by Oxford Nanopore's MinION portable sequencer is difficult, as output per flow cell using the R9.0 chemistry ranges widely from 0.2-1.2 Gb [11], but has been found to be 2.3 Gb per flow cell using the newest R9.4 chemistry [12] – cost per flow cell is $500 if bought in bulk of 48 or $900 if bought individually [13, 14]. Furthermore, depending on the sequencing protocol, maximum lengths of reads that are mappable to the corresponding reference genome range from ≈150,000 to ≈900,000 with mean read lengths of ≈7,000 and ≈16,000, respectively [11, 12]. Error rates range from 8% to 15% [11]. High error rates in long-read data can be greatly reduced by aligning long reads and generating consensus sequences, reaching accuracy of >99.999% with PacBio reads [15] and >99.5% with Nanopore reads [16].

1.2 Limitations of Draft Genomes

The number of draft genomes still exceed the number of completed genomes currently housed on and being deposited into published databases [17]. Of the vast number of microbial genomes sequenced over the last decade most are still in draft status (Figure 1.1). Completed microbial genomes can be characterized as assemblies that possess continuous chromosomal representation, low assembly error rates, and lack known misassembly issues such as assembly chimeras and false rearrangements [18].

While some specific questions can be answered without the completion of genome

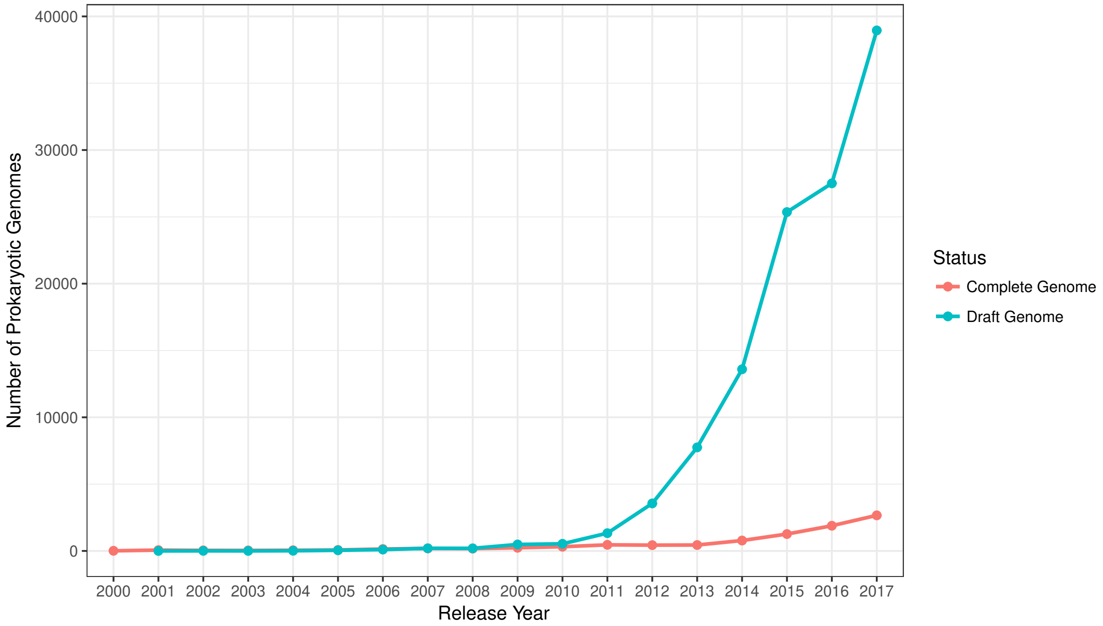

Figure 1.1: The number of complete vs. draft genomes deposited annually since 2000 on NCBI's Genome Assembly database [19] according to NCBI's assembly status classification.

sequencing and assembly, the generation of draft and fragmented genomes limit and hamper the research of others. On the most basic level, draft genomes may be missing functional genes, which can mislead others because it is not known whether a specific isolate lacks genes of interest or that the genes were not sequenced or assembled by the researchers. This can be detrimental to research of genes of interest, which includes their evolution and phylogenetic diversity, and their presence/absence across microbial lineages, which includes potential horizontal gene transfer events. However, even draft genomes that contain gaps in only non-coding regions [20] would most likely still be missing information on regions with important functions, as a significant amount of the DNA previously categorized as ”junk” has been shown to play important roles in biochemical processes and may serve as regulatory regions [21]. Gaps in genome assembly, even when they are in regions of unknown function, reduce the likelihood that future discoveries can be made. This is perhaps most notable in regards to repeat regions, such as CRISPR regions in microbes, which can be left unassembled in draft genome sequencing projects [22]. Given the limitations of fragmented genomes, the scientific community would benefit from a structured research effort or funding requirements to bring genome sequencing projects to completion.

1.3 Benefits of Completed Genomes

For fragmented genomes, the absence of a gene does not prove that a gene of interest is not present in the genome. If the presence of genes is unknown in two or more organisms, it is not possible to capture an accurate, complete picture of species variation through comparative genomic methods. As a result, the ability to carry out accurate comparative genomic analyses on both a small and large scale is greatly improved by complete genomes. The comparison of a suite of strategically chosen species can provide information about critical phenotypes, such as virulence [23]. Continued publication of complete genomes will increase the accuracy and feasibility of comparative genomics on a larger scale.

1.3.1 Illuminated Microbial Pathogenicity

Draft genomes limit our understanding of microbial pathogenicity. Incomplete genomes are often found to be fragmented at regions of mobile elements and genomic island sites, as assembly algorithms typically fail to resolve repeated elements often associated with these genomic regions [24]. As a result, this limits our understanding of key driving factors in microbial evolution as these sites are free to change because they are not selective. Having a clearer picture of genomic plasticity areas can give the ability to better track microbial pathogenicity [4], as virulence factors are often encoded in such regions [25]. Attempting to track outbreaks of closely related pathogenic strains may be much more difficult or impossible without comparison of complete genomes, such as the case for a group of certain *Salmonella enterica* serovars from a recent outbreak in Denmark [23] and an extensive analysis of carbapenem resistant bacteria isolated from hospital plumbing systems [4].

1.3.2 Increased Understanding of Genome Organization

Recent studies have found structural variants (SVs) to be common in microbial genomes [26]. A wide variety of genomic rearrangements ranging from insertions to duplications can result in SVs. These rearrangements are initiated by a variety of processes which are easier to trace with more complete genome information [27]. Regions that characterize SVs can be difficult for assembly algorithms to resolve and are generally not properly represented in fragmented assemblies [28]. Genomic rearrangements can cause a wide variety of phenotypic shifts, including virulence [29], and can act as a driving factor of microbial evolution [30]. Missing knowledge of SVs may translate into a lack of information about the basis of critically important microbial phenotypes and pathovars.

1.3.3 Improved Understanding of Orphan Genes

Complete genomes further our understanding of abundance and function of orphan genes, which some researchers have termed ”Taxonomically Restricted Genes” [31]. Orphan genes lack homologs in other taxonomic groups and can represent up to 30% of the genes in bacterial genomes [32]. In agreement with the concept that a limited amount of protein families and potential protein folds exist [33], there has been the expectation that genome sequencing would cause the discovery of new proteins to plateau over time [34]. This plateau has not been observed as non-redundant proteins are still being discovered at high rates [35].

The identification and annotation of orphan genes is often difficult, since the very definition of orphan genes involves the lack of a homolog. However, using a mixture of laboratory techniques and computational analysis, the functions of numerous orphan genes have been elucidated [3, 36]. Orphan genes have been found to have essential roles [36, 37] and to be associated with species evolution [31]. Presenting the complete pool of orphan genes with an unfragmented complete assembly is a vital step in fully identifying, annotating, and understanding these genes.

1.3.4 More Insights into Evolutionary Rates

Complete genomes may also provide more insights into evolution rates. Rates of molecular evolution are not universal over the Tree of Life [38], and calibrating the "molecular clock" for a species is a difficult task when basing estimates on specific clusters of genes, as the selective pressures that drive mutation rates differ in various regions [39]. Comparison of whole genomes of various taxa over long periods of times will give better species specific evolution rates than broad estimates based on selected genes from a limited amount of taxa [40].

1.3.5 Enables Others to Work with a Wider Variety of Organisms

Ultimately, bringing whole-genome sequencing projects to completion enables others to work with a wider range of microbes. Designing primers or gRNAs for CRISPR-Cas systems can be greatly facilitated by a complete genome. The difficult task of designing primers for oligonucleotides where only the partial sequence composition is known [41] can be avoided by simply knowing the target sequence in the first place. Complete genomes can also reduce unwanted off-target binding, as some primer design tools scan genome-wide in order to predict primers that are most likely to have the lowest levels of off-target binding [42].

1.4 Why are we Still not Completing Reference Genomes?

Given all the benefits of a non-fragmented assembly, why are bacterial and archaeal genomes still not being completed? With regard to microbial genomes, we are not limited by the scope of nucleotide sequencing technologies, the want of computational tools, or the lack of computer processing power and memory.

Utilizing the potential of long reads has shown to generate complete genomes through a variety of assembly methods [43]. All companies in recent years have made significant improvements on affordability and throughput of sequencing platforms – with current examples being Illumina's HiSeq Platform, PacBio's Sequel System, and Oxford Nanopore's portable MinION sequencer. However, relying only on short reads for microbial whole-genome assembly without supplementing them with long reads generally results in a fragmented assembly [8, 20].

As the ability to generate increasingly longer reads over the last ten years has occurred, many inventive algorithms have been developed specifically for assembling these types of data [44]. Despite many computational strategies able to produce complete genomes using long reads, mean contig length in genome assemblies still remains low due to heavy reliance on short-read sequencing (Figure 1.2).

One factor limiting complete genome assembly is its increased cost compared to producing a draft genome. While the cost per Gb of long-read sequencing has dropped significantly since first entering the market, long reads that are needed to resolve microbial repeat regions [9] come at a higher cost than short reads [43]. However, it is important to put the cost of long reads into the perspective of microbial genome sequencing. Higher cost per Gb may discourage researchers from using long-read sequencing, but in light of the typical size of a bacterial or archaeal genome, the cost is not very prohibitive. For example, multiplexing 8 *E. coli* K12 strains (approx. 4.5 Mb in size) on a single PacBio

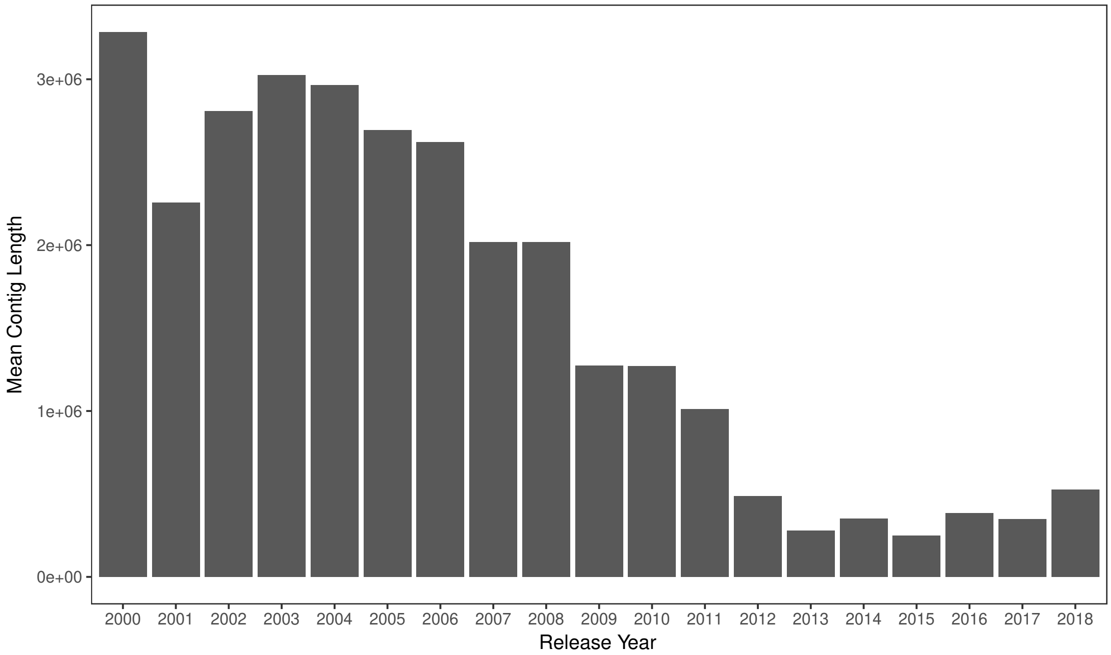

Figure 1.2: Mean contig length of all bacterial and archaeal genomes deposited on NCBI's Genome Assembly database [19] on a per-year basis as of May 2018. The advent of higher-throughput short-read sequencing in the mid 2000s caused a significant drop in mean contig length, and long-read sequencing has not caused much of an increase in average contig length since. The drop in contig length is not due to differences in average genome assembly size over time, as the mean genome size per year has remained relatively consistent since 2006 at 3.8 Mb.

Sequel SMRT Cell generated complete or nearly complete genomes for each sample [45]. Depending on the genome size, one sequencing cell can result in a maximum of 16 non-fragmented or close to complete bacterial assemblies [46, 47]. Additionally, 12 *Klebsiella pneumoniae* (5.3 Mb) high-quality, gapless genomes were assembled with a mixture of multiplexed MinION sequence data and short-read sequencing, despite the drastic coverage depth variation amongst the multiplexed MinION samples [48]. The researchers estimated generation of an individual *K. pneumoniae* genome to cost $150. However, ignoring the options of multiplexing may cause researchers to shy away from higher cost per Gb of long-read sequencing and rely on relatively cheaper short-read sequencing instead.

Another factor that may discourage researchers from generating complete genomes is an overabundance of outdated assembly strategies in the literature. As technologies improve, researchers respond with developing software that are optimized for new types of sequencing data, which in turn leads to a plethora of assembly methods, some of which quickly become outdated. For example, the strategy outlined for optimized microbial genome assembly by Nagarajan & Pop [49] may lead a researcher to prepare three DNA libraries – one Illumina mate-paired, one Illumina paired-end, and one PacBio single-read – at x50 coverage each. However, this estimate is based on the average read length distribution from an older version of the PacBio chemistry and sequencing platform, and newer versions of the chemistry and platform updates have shown that high-quality microbial genomes can be assembled with either one long-read and one short-read library or simply one long-read library [43]. The challenge of keeping up to date with assembly strategies can contribute to the preponderance of fragmented assemblies.

1.5 How can we Complete more Microbial Genomes?

Overall, the power of long reads to resolve complex, repeated regions must be utilized more often. Long-read sequencing can be optimal in the production of complete genomes for the majority of bacterial and archaeal microorganisms [8], and the ability to generate even longer, usable reads using various protocols and technologies will enable the completion of more microbial genomes [12]. In our estimation, there are three specific hypotheses that could be explored to enable the completion of more microbial genomes.

First, the question of ”to what depth of coverage should I sequence?” in a lineage-specific manner must be answered better. Answering this question does not come without challenges, as necessary depth of coverage depends on factors such as genome size and repeat complexity [50] and the choice of algorithm used for genome assembly [51]. Uneven sequencing depths at different regions in a genome adds additional complexity to this question [52]. However, a benchmarking study – which quantifies the correlations between genome size and genomic complexity across a phylogenetically diverse breadth of microbial taxa and the estimated optimal sequencing depths – would be very useful [9]. A researcher could then estimate optimal sequencing depths by looking at the genomic complexity of a closely related strain. If the taxonomy of the microbe being sequenced is not known, then Sanger-based sequencing of key marker genes, such as the V3-V4 region of the 16S ribosomal subunit, would be a logical choice for identifying closely related taxa. Even if a researcher chooses to skip marker based identification of unknown strains, the genome size and some aspects of repeat profile can be estimated by counting k-mers in short-read sequencing data [53, 54] and then optimal depth of long-read coverage may be calculated. Answering these questions can prevent researchers from not generating enough sequencing reads to effectively close the genome or the alternative of generating data past the point where increased coverage of short-read data does not improve assembly quality [9], which in turn increases financial expenditures.

Second, more robust evaluations of genome assembly are necessary to understand the nuances of factors contributing to the completion of bacterial and archaeal genomes. While many bacterial genome assemblies have been benchmarked, most of these benchmarks only report quality statistics that provide limited information, such as number of contigs and N50 values [55, 56]. The problem with such statistics is that a complete picture of actual assembly quality is not given, but only the number and size distribution of fragments, as the percentage of similarity between the *de novo* assembly and the target sequence is unknown. Comparing assemblies to closely related reference genomes can give an estimate of the percentage of correctly assembled sequence, but significant differences between assemblies can be due to naturally occurring SVs [26]. We suggest that future assembly algorithm benchmarks incorporate simulated sequencing data sets which could be used to compare against a known target assembly, as this allows for detection of mismapped regions that could be flagged as potential misassemblies. This would give a clearer picture of the strengths and weaknesses of various genome assemblers, allowing researchers to better select which algorithm software to utilize to generate the assembly of complete, high-quality genomes.

Lastly, providing estimates of both the capability and limitations of multiplexing samples with the goal of generating a complete genome is important. Multiplexing on both PacBio and Oxford Nanopore platforms has been successful, which significantly reduces sequencing cost per sample. This has resulted in high-quality assemblies of both microbial genomes [46, 48] and multidrug-resistance plasmids [57]. However, to the best of our knowledge, the question of how many microbial genomes, of a given genome size, can be effectively completed by sequence data from multiplexing on a single flow cell of long-read platforms has not been appropriately answered. Answering these questions will aid in the rapid completion of more microbial genomes and will positively impact the field of microbial genomics [58].

1.6 Conclusion

Despite the ability to rapidly and cost-effectively generate complete genome assemblies, the ratio of published draft to complete prokaryotic genomes is still high. More publicly available, unfragmented genomes will give key insights into pathogenicity, structural variance, orphan genes, and ultimately, microbial taxonomy and evolution. Financial burdens, an excessive amount of assembly algorithms, and no appropriate estimate for needed sequencing coverage may discourage bringing sequencing projects to completion.

We believe that providing scientists with a concept of how much sequencing should be done in order to enable assembly of a complete genome, carrying out benchmarks of assembly algorithms on various sequencing data profiles, and gaining more information regarding multiplexing options will aid in the completion of more bacterial and archaeal genomes.

Chapter 2

Genomic Variation Influences Genome Assembly Quality Metrics

2.1 Background

Genome assembly benchmark studies [55, 59, 60, 61] have demonstrated variation in standard bacterial and archaeal *de novo* genome assembly quality statistics. Variation in these statistics – most notably the number of contigs, N50 values, and percentage of a reference genome covered by *de novo* assemblies – may still be predominant in highly similar genomes when given equal sized datasets of similar read quality. This raises the question: if we hold genome size approximately the same and stardardize sequencing quality and depth of coverage, what causes the resulting variation in genome assembly quality and varying amounts of assembly fragmentation?

Variation in genome assemblies may stem from simply using different assembly algorithms [62, 63]. Natural differences in composition inherent in microbial genomes impact the assembly quality downstream [24]. Most notably, assembly quality metrics may vary among assemblies generated using the exact same algorithm independent of insufficient overall coverage differences [64]. Genome assembly quality is most notably negatively impacted by sequencing errors [65]. Short-read sequencing technologies can fail to produce sufficient coverage in genomic regions that are difficult for assembly programs to resolve [66], such as those with high GC content or regions affected by sequencing library preparation amplification biases [7, 67, 68, 69]. Additionally, the bias of certain motifs such as homopolymer runs [70] negatively impacts genome assembly quality. However, independent of biases that are introduced during sequencing, the complexity and frequency of repeat regions is considered the main factor that causes assembly quality variation [8, 71] and this lies at the crux of why genome assembly is classified as an NP-hard problem [72].

Short-read high throughput sequencing is insufficient to close gaps in many repeated regions when read length does not exceed the length of genomic repeats [20, 43, 51, 73]. Resolution of these repeated regions is difficult or impossible without manually intensive processes. Typically these techniques consist of cloning genomic fragments of small sizes into plasmids, then utilizing Sanger-type sequencing, and the subsequent assembly and ordering of the fragments into a complete sequence [1]. These methods are generally lengthy and consist of labor-intensive tasks that may prohibit the completion of single chromosome contiguous bacterial or archaeal genome assemblies. Genome assembly using long-read technologies is another solution to generating higher-quality genomes [74], but long-read sequencing comes at a much higher cost than short-read sequencing [75] and typically is more error prone than short-read technologies. Due to either labor-intensive processes, prohibitive costs of long reads, or the desire to use low error rate of short-read data, automated genome assembly using short reads (or ideally a hybrid of short and long-read data) will probably remain the common method for genome assembly in the immediate future.

The fact that genomic repeats of various lengths make assembling a complete genome using short reads nearly impossible is known [8]. It is not known how the overall abundance of genomic repeats affects genome assembly. Additionally, it is still unclear how other factors such as GC content, k-mer count variation and complexity, homopolymeric regions of sequencing reads, genome size, number of genes, fraction of the genome subsisting of coding regions, percentage of non-coding vs. coding sequences, and repeat density correlate to genome assembly quality [53, 76, 77]. Here, we address the role of various genomic characteristics on genome assembly quality metrics. In order to eliminate confounding factors of high levels of genome complexity variation, we chose to focus on datasets from genome sequencing projects that were both high in depth and from highly similar strains. Providing insights into what genome characteristics convolute genome assembly can aid in future algorithm development and help understand issues as to why genome assembly projects do not often provide complete genome assemblies.

2.2 Experiments and Results

2.2.1 Data Curation

To address the role of genomic variation on assembly quality, we began by selecting 96 publicly available short-read sequencing datasets of highly similar *E. coli* strains from NCBI's Sequence Read Archive (SRA). The list of SRA accession numbers can be found in Appendix A. The same DNA library preparation kit (Nextera XT shotgun) and sequencing platform (Illumina NextSeq 500) were used as a part of the same sequencing project, the U.S. Department of Agriculture's Genome Trakr Project [78]. The FDA Center for Food Safety and Applied Nutrition (College Park, Maryland) submitted all of the datasets, which consisted of 75 bp paired-end reads.

Overall read quality in the datasets was found to be high (mean Phred score no lower than 30, even at the most error prone read positions) as measured by assessment using FastQC [79] analysis. In order to focus solely on strain-level variation, data were subsampled to minimize differences in sequencing run statistics, such as quality score and depth, between pools of sequencing reads. Total coverage of the datasets ranged from x135 to x400 across the complete calculated and actual genome size. In order to reduce the variance among strains in GC bias due to Illumina sequencing technology [66], all selected strains had highly similar levels of GC content (50.5) with less than one percent of variance.

2.2.2 Assembly Quality Definition

While not always possible, we acknowledge that simply observing internal assembly statistics of a draft genome without comparing assemblies to a known reference genome results in a lack of information about assembly accuracy [80, 81], genome length [82], and coding sequences available to annotation [83]. In order to understand completeness, we mapped the assemblies (described in detail in following sections) using MUMmer4 [84] to more than 600 *E. coli* gapless genomes categorized as ″Complete″ by NCBI's GenBank and discovered that the percentage of various reference genomes represented by an assembly varied between 70 and 90 percent. In an attempt to find the most appropriate reference genome for our various strains, we looked at the highest percentages of a reference genome covered by a given strain. This turned out to be of limited value, as several of the reference genomes were all covered by around 90% of a *de novo* assembly, which indicated no clear appropriate reference genome for either an individual strain or the entire collection of *de novo* assemblies. This complicated our desire to compare the *de novo* assembled genomes to a standard reference. As no adequate reference genome existed for all of the selected strains, no way to measure assembly accuracy existed. In the second chapter of this thesis, we focused on assembly contiguity as our measure of assembly completeness and use the terms ″contiguity″ and ″quality″ interchangeably at times.

2.2.3 Assembly Quality over Coverage Increment

In order to assess genome assembly contiguity under differing sequencing coverage, we selected 34 out of the original 96 datasets (ranging from SRR3989774 to SRR3989808 but excluding SRR3989775, which was a *Mus musculus* sample) to subset over a coverage increment, assemble, and assess contiguity for each strain. We used our custom analysis pipeline (code available in Appendix C) to perform these analyses, which provided us with a mean of 41 independent datasets for each individual strain. We then assembled these normalized datasets using the SPAdes assembler [85] and assessed the quality of the subsequent assemblies using QUAST [86].

We found that for almost all of the *E. coli* sequencing data, great improvement in the number of contigs (in this case, fewer contigs of longer length) occurred at low coverage levels up until x50 coverage, but increased assembly contiguity leveled off around x100 coverage for most strains (Figure 2.1). At higher depths of coverage, the number of contigs increased for some of the selected strains, but this finding was not consistent or widespread. This is most likely due to the difficulty genome assembly programs have in dealing with large accumulations of sequencing errors found in larger datasets [87].

Using the entire pool of the 96 *E. coli* strains we initially selected, we subsampled each dataset down to 3.25 million reads. This translated into approximately 95-fold coverage when dividing the total number of sequenced base pairs by the median total length of these *E. coli* strains' genome size. We then assembled each subsampled dataset using two de Bruijn graph assemblers, IDBA [88] and SPAdes [85]. The resulting assemblies were quality assessed using QUAST [86]. We then counted the number of k-mers of length 21 using Jellyfish [53] and used the software Genome Scope [77] to estimate genome size and repeat region sequence lengths based on k-mer profiles. Analyzing k-mer profiles is a standard method of estimating genomic repeat complexity and genome size [53, 76].

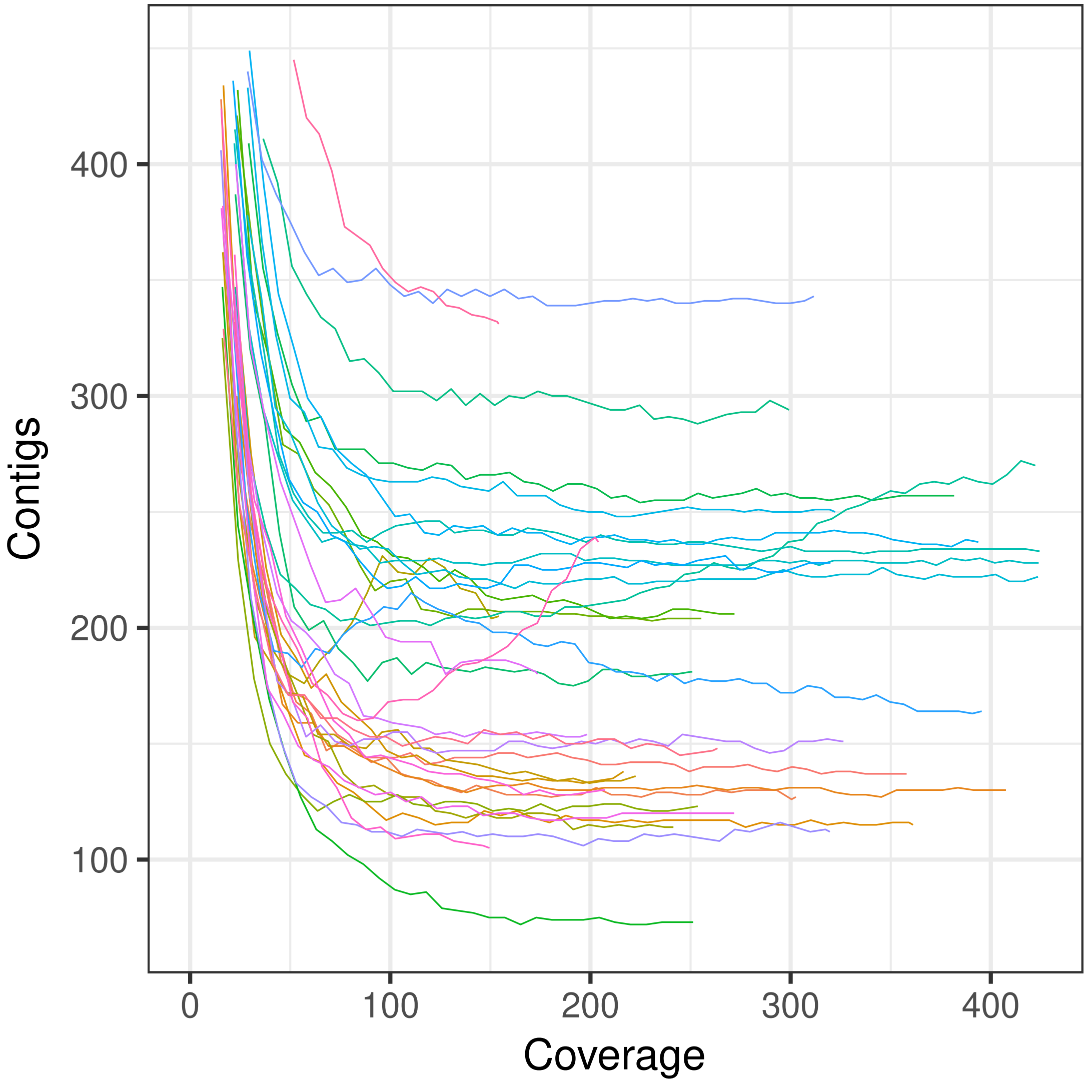

Figure 2.1: Depth of coverage vs. number of contigs from assemblies of 34 various *E. coli* strains. Each colored line represents a different strain. Assembly contiguity varies among the closely-related bacterial strains and increased coverage provides little increase in genome assembly contiguity past 100-fold coverage for most strains. Coverage was calculated by dividing number of sequenced base pairs by the total length of contigs $\geq$ 500 bp. The corresponding coverage vs. N50 graph based on analysis of the same strains can be found in Appendix B.

Using Prokka [89], we annotated all of the assemblies and calculated the number of coding sequences and the average coding sequence length. Assembly statistics of the

SPAdes and IDBA assemblies for each strain were very similar, supporting the observation that assembly quality statistics and gaps in genome assemblies do not simply stem from the failings of a specific genome assembly algorithm. All of the correlations reported in the second chapter of this thesis are based on information derived from SPAdes assemblies, except for Figure 2.2, where genome size was estimated by analyzing k-mer profiles. Correlations and graphs based on IDBA assemblies can be found in Appendix B.

2.2.4 Assembly Quality vs. Genome Size and Number of Coding Regions

Genome size in bacteria and archaea is positively correlated with the number of coding sequences, as well as the total length of the coding sequences [90, 91]. While the quality of annotation can be negatively impacted by gene fragmentation [83], Prokka can be modified to annotate partial or fragmented genes [89]. Keeping in mind that a typical gene length averages around 1,000 bp in *E. coli* [92] and that it is estimated that the average bacterial genome consists of between 5-15% of non-coding sequences [93, 94], we observed the expected number of coding sequences compared to genome size for the strains we assessed (Figure 2.2). After verifying the relationship between genome size and number of coding sequences, our expectations that larger genomes (and those consisting of a larger number of coding sequences) would have more fragmented genome assemblies were confirmed (Figure 2.3).

Not surprisingly, the number of contigs in the assemblies of closely related *E. coli* strains generally increased with an increase in estimated genome size and an increase in the number of coding sequences identified. In line with large scale differences in genome complexity and genome assembly contiguity observed across disparate lineages of organisms [95], the number of contigs generated in our analysis increased from 47 to 369 with only a 1.2 Mb increase in genome size when maintaining both read quality and

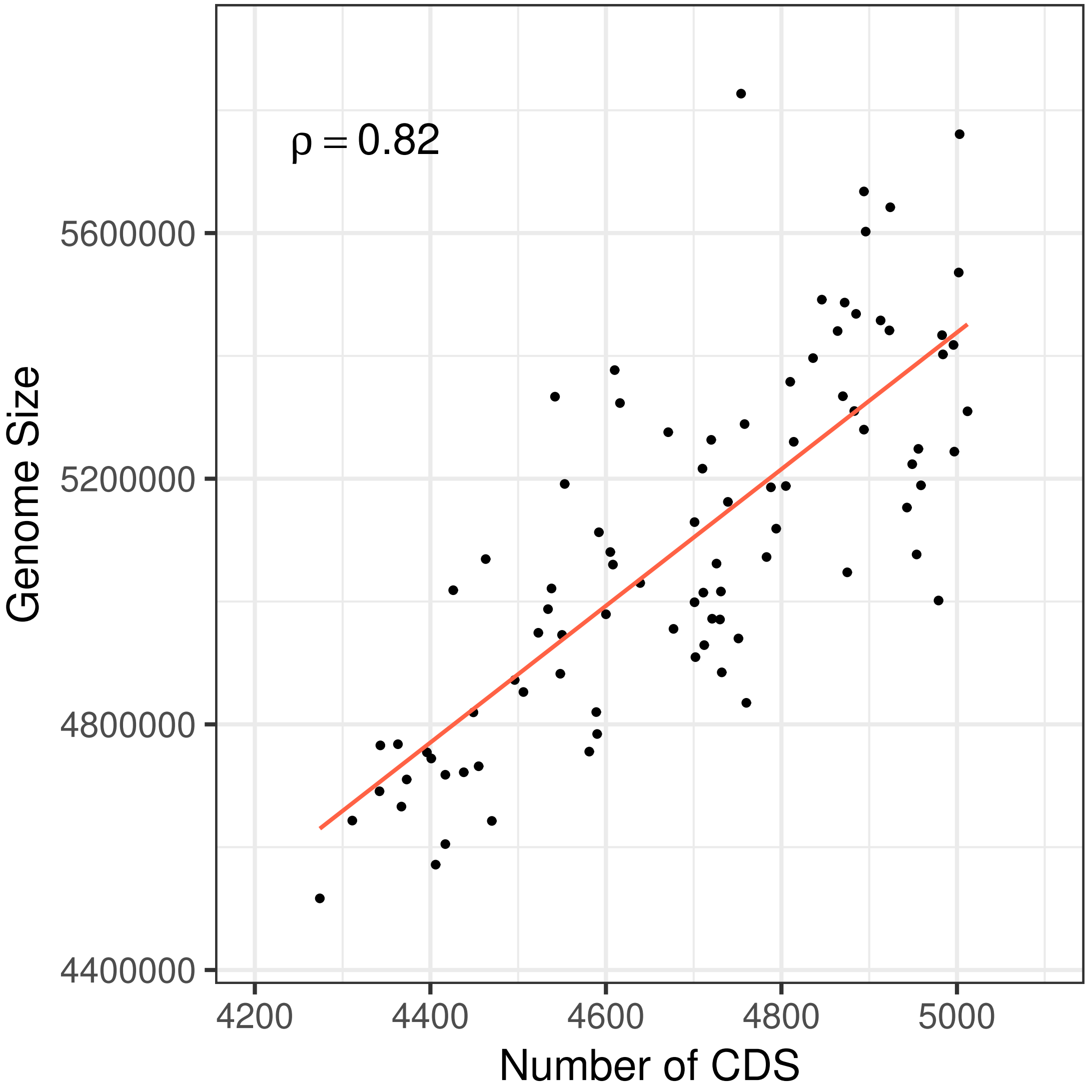

Figure 2.2: Number of coding sequences vs. genome length for 96 *E. coli* strains. The number of coding regions at various genome sizes lies within an expected range when factoring in the positive correlation between genome size and number of genes, mean coding region length in bacteria, and typical percentages of coding vs. non-coding sequences. The p value and 95% confidence interval of ρ are 2.2e-16 and 0.74–0.87.

read depth. It is important to note that other researchers have also observed similar trends when comparing genome assemblies of various organisms at different coverage levels [96].

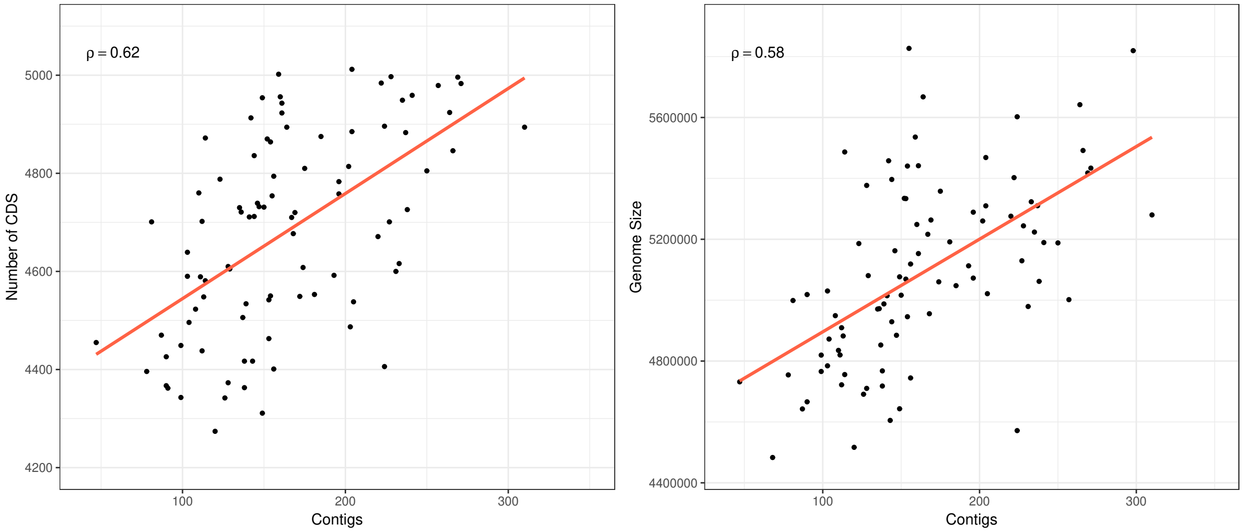

Figure 2.3: Contig number vs. the number of coding regions and genome size for 96 *E. coli* strains. Variation in genome assembly quality, assessed here as genome assembly contiguity, can be attributabed to the correlation between genome size and the number of coding regions. The p values and 95% confidence intervals of ρ from left to right are 1.29e-11 and 7.57e-10 and 0.48–0.73 and 0.43–0.70.

2.2.5 Assembly Quality vs. Repeat Density

Some complex genomic repeat regions are difficult or mathematically impossible for assembly algorithms to resolve when using short-read datasets [51, 73]. In terms of understanding the causes of variation in genome assembly quality, an obvious impact of assembly quality is the number of repeat regions present in a genome. Quantifying repeats is difficult, as many different classes and lengths of repeats exist [97]. Therefore, we did not exhaustively search for certain categories or types of repeats, but instead estimated total repeated length based on k-mer profiles. We chose to compare overall repeat density, which is defined as total repeat sequence length divided by genome length [98], to assembly quality (recognized here by NG50 values) instead of simply observing total repeated length vs. fragmentation. The NG50 statistic is a version of the N50 statistic that is weighted for genome size [99], which makes comparison of NG50 values across different genome size possible. By observing repeat density and NG50 (Figure 2.4), we effectively eliminate biases introduced by variations in genome size.

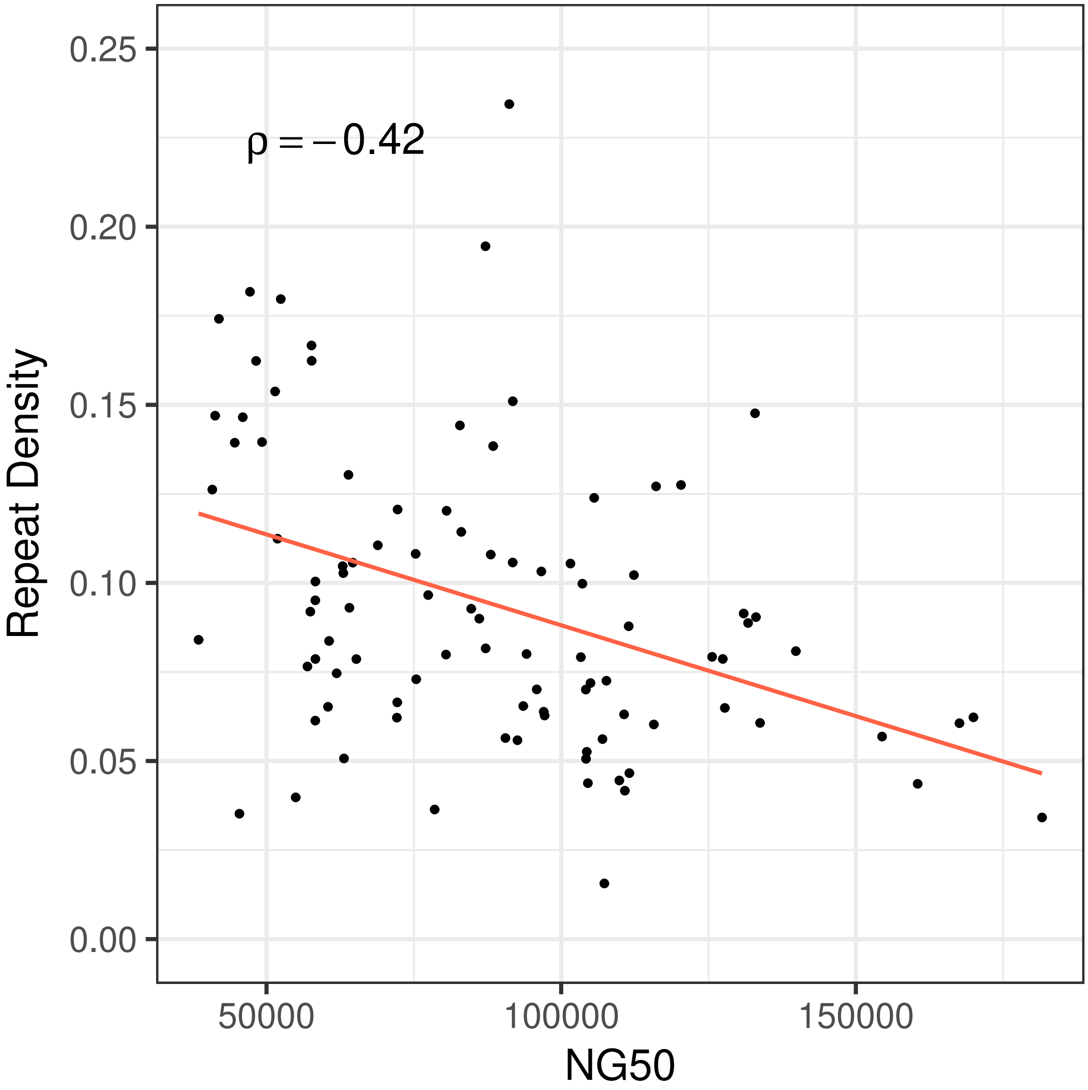

Figure 2.4: NG50 vs. repeat density of the assembled genomes. This correlation indicates that assembly quality does not only decrease while genome size increases due to a natural growth in repeats along with genome size, but rather that an increase in repeats in a genome negatively affects assembly quality independently of genome size. The p value and 95% confidence interval of ρ are 1.99e-05 and -0.57—-0.24.

2.2.6 Coding Sequence Density vs. Assembly Quality

Previous studies have shown the number of coding sequences to increase in a linear fashion along with genome length in microbes [91]. Therefore, simply comparing the number of open reading frames to a contiguity statistic would result in similar correlation between the number of coding regions and genome size in most cases. Since we previously reported the correlation between genome size and assembled contigs, here we compared coding sequence density (recognized here as the percentage of a genome that is coding sequence) to NG50 (Figure 2.5). We found no significant correlation, which indicates the density of a genome represented by coding regions by itself is not a significant factor that complicates genome assembly.

2.3 Discussion

Many studies have used reference genomes, typically assembled from incredibly high depth reads from multiple platforms or from Sanger-based ”primer-walking” sequencing, to provide a standard to infer the genome size and number of genes for particular *de novo* assemblies [55, 59, 60]. We realize that most standard sequencing projects for bacteria or archaea do not have the luxury of starting with much knowledge about the genomic structure of their organism of interest prior to sequencing. Some of the most common questions from researchers wanting to complete bacterial or archaeal genome sequencing projects are focused on how many reads to sequence and what the optimal length of the sequencing reads should be.

To assess the amount of reads needed to complete genome assemblies for a suite of single chromosome genomes of similar phylogenetic distance, we subsampled data from 96 *E. coli* strains and assembled the reads at a range of depths using different assembly algorithms. We found the point at which increased coverage did not improve assembly

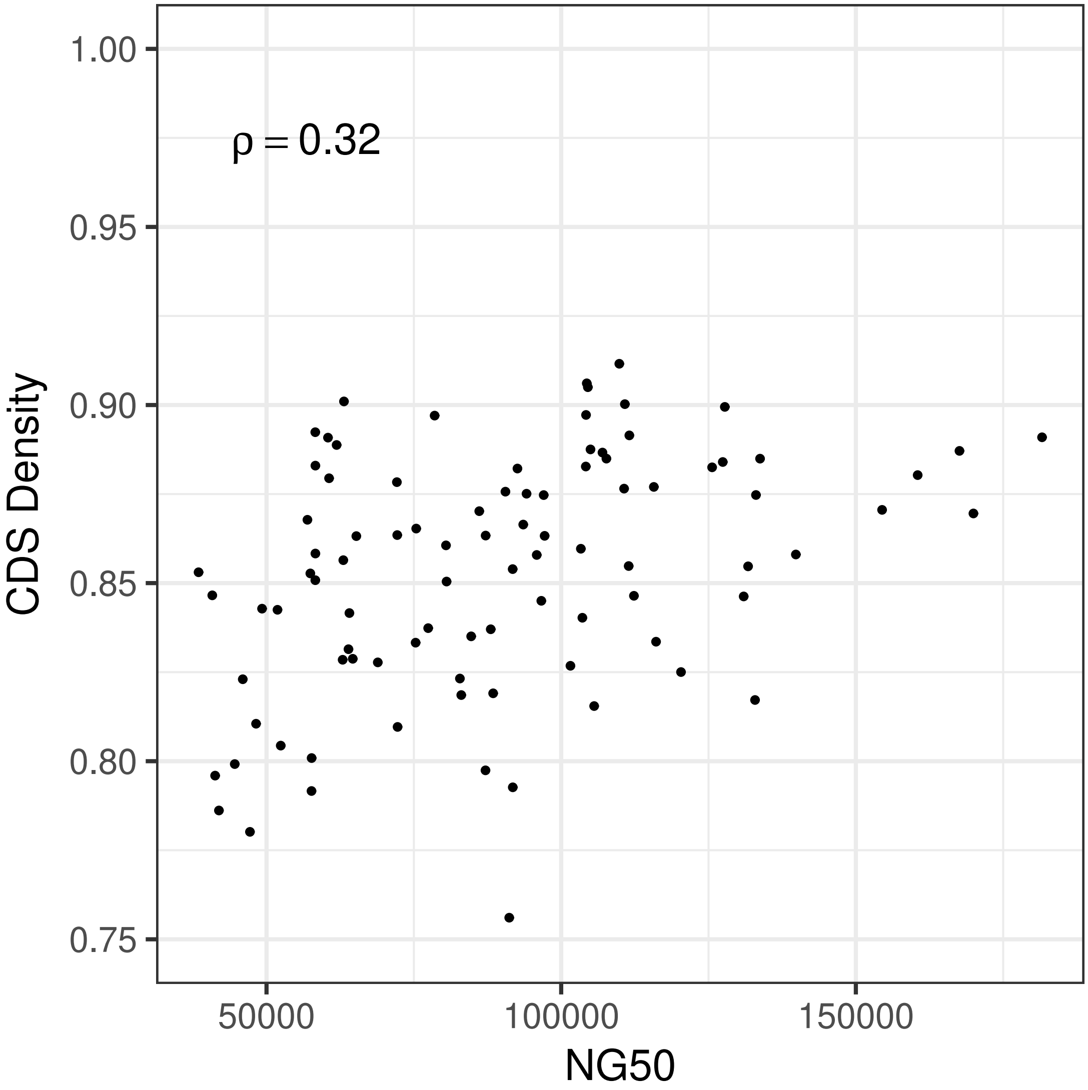

Figure 2.5: NG50 shows little correlation with the abundance of coding regions in a genome. The p value and 95% confidence interval of ρ are 1.69e-3 and 0.12–0.49.

contiguity for most of the *E. coli* strains to be around x100, but the question of how much coverage is enough should be answered in a more phylogenetically diverse manner.

The number of annotated genes compared to genome size fell within an expected range (Figure 2.2) in the assemblies. However, the correlation between coding sequence density and assembly quality (summarized in the NG50 value) may still be somewhat impacted by poor assembly affecting the ability to annotate a genome properly and partially explain why no significant correlation between coding region density and NG50 was found. The statement of the authors from the GAGE-B paper [55] that a single, short fragment sequencing library generally contains the majority of genes most likely does not apply to our assemblies, as read length was 75 bp. This lies in a length range where even a 25 bp length increase of reads can significantly improve assembly quality [100].

Expectations of assembly quality must be lowered as genome size and repeat density increases, because as microbial genomes grow in size, coding sequence, and repeat complexity, genomes become poorer in quality (Figures 2.3 and 2.4). Even gene-level questions for difficult to assemble genomes may not be fully answered if relying solely on short-reads. Sequencing short reads of longer length may increase quality [100], but one may have to turn to long-read technologies to be able to produce more accurately assembled genomes as closely representing the real genome structure as possible. As Richards [101] emphasizes, researchers must fully inform their audience regarding how the accuracy and quality of *de novo* genomes (as well as their subsequent annotations) especially when assembling with short reads. This may be extremely pronounced in genome structure variation. Instead of making unverifiable claims based on genomes whose exact sequence is not known, researchers must always keep in mind the uncertainties surrounding *de novo* assemblies.

Short-read sequencing alone simply does not have the ability to answer many biological questions requiring high quality genomes. Instead of the generally futile attempt to reduce fragmentation by sequencing short reads at unnecessarily high depths, providing the scientific community with better ways to lower the cost of long-read sequencing in terms of microbial sequencing is needed. Recently, both Heiner et al. [46] and Wick et al. [48] took advantage of new higher-throughput sequencing platforms (both PacBio's Sequel System and Oxford Nanopore's MinION, respectively) to sequence and assemble nearly gapless genomes for more than 8 microbes simultaneously on one flow cell, greatly reducing overall price per sample. This strategy of multiplexing is able to greatly reduce long-read sequencing costs, which could cause a shift away from short-read sequencing in the future. However, research about best practices and optimal number of microbes per multiplexed flow cell must be carried out.

Overall, we show here that repeat variation, often compounded over longer genomes, can be a main factor that contributes to the inability to assemble quality genomes when utilizing short-read data sets. We recommend that researchers utilize a combination of high-quality short-reads and more error prone (or high depth) longer-read technologies to complete bacterial and archaeal genomes.

Chapter 3

Coverage Benchmarks for Bacterial Genome Assembly

3.1 Background

In the last two decades, whole-genome sequencing has become common place. As a result, benchmarks of sequencing technologies and optimal assembly algorithms have been established [50, 55, 56, 59, 61, 102]. While our knowledge of the effects of assembly algorithm and sequencing error rate on genome assembly quality is well established, recommendations for optimal sequencing depth of coverage and read length are not well understood. This fact is a contributor to why so many bacterial and archaeal genome sequencing projects remain in draft stages (Chapter 1). Additionally, the capabilities of various levels of multiplexing with long-read platforms in order to generate highly accurate, single-contig assemblies for various taxonomic groups and complexity classes are lacking in the literature. For the few cases where this information exists, these estimates are only based on a few select species in a small subset of the microbial phylogenetic breadth. Factors such as genome size and repeat complexity vary dramatically among the bacteria and archaea [97, 103], which in turn adds uncertainty to optimal sequencing parameters and coverage depths [50].

3.1.1 Genome Assembly Algorithms

The choice of which genome assembler to use largely depends on the type of corresponding read data. Assemblers typically fall into one or more of the following broad algorithm categories – greedy, overlap-based, and de Bruijn graph. Overlap-based and de Brjuin graph approaches utilize graph theory [104], a mathematical theory in which nodes represent objects and edges that connect the vertices represent relationships between objects. In the case of genome assembly, nodes represent nucleotide strings and edges represent overlap between strings. Generally speaking, overlap-based methods are best suited for low coverage, long-read data and de Bruijn graph approaches for high coverage, short-read data [105]. The usefulness of most greedy approaches is currently limited, as they are not well suited for short-read data assembly or able to assemble strings larger than 100 kb [106]. Some software use a combination of two or more of these approaches during different steps of assembly [107].

One of the earliest completed bacterial genomes, *Mycoplasma genitalium* [108], was assembled using the TIGR assembler [109], which utilizes the greedy approach – a heuristic method that combines reads in the most optimal, local manner. The workflow for this project utilized a laboratory intensive strategy which fragmented the bacterial genome through bacterial cloning and subsequently Sanger sequenced [110] the resulting fragments. The researchers then used the TIGR assembler to assemble the reads into contigs, and used a smaller set of paired-end reads sequenced from λ-clones with an insert size of 10 kb to order the contigs. Using this approach, the TIGR assembler was able to produce a completely contiguous and accurate genome assembly. Such early assembly algorithms used heuristics to combine reads into the shortest possible string, an NP-complete problem when attempting to find the true answer [111]. These strings were generated through overlapping the strings by finding the best Hamiltonian paths through the connecting reads [106]. TIGR, for example, combines reads with the highest scoring overlaps in a decreasing manner. While this assembly strategy can be sufficient for small assembly targets in the 10's of kb in length with simple repeat structures, greedy algorithms often fail to produce accurate assemblies for larger, more complex strings [106, 112]. Indeed, reads from non-contiguous repeated regions can often score higher than the actual adjacent region, resulting in chimeric reads or collapsed repeats [113].

The use of greedy algorithms was soon replaced with overlap approaches, a method that was used in software as early as 1984 [114]. The Celera Assembler [115] was one of the most successful overlap assemblers and is still used today in different software packages [116]. Overlap methods look for statistically significant overlaps between reads and then use this overlap information to combine reads into contigs in a global manner, unlike the greedy approach. What makes overlap between reads statistically significant depends on a selected minimum length of overlap, which is determined by the mean read error rates [106]. In the terms of graph theory, nodes represent reads and edges the overlap between reads. Once the overlap graph is constructed, the optimal path traversed through the nodes represents the unpolished, assembled genome.

The advent of high throughput, short-read sequencing [117] caused a necessary shift in the foundation of assembly algorithms. Assembling high-coverage, short reads using an overlap method is a computationally unrealistic problem, as the subsequent overlap graph would be too large to deal with computationally [118]. This caused researchers to begin utilizing a de Bruijn graph approach [85, 119], which is based on early assemblers and theoretical work of Idury & Waterman [120] and Pevzner [121, 122]. Using this method, reads are first split into substrings, termed k-mers, with the selected length of *k*. A de Bruijn graph is then constructed with unique k-mers representing nodes and overlap between k-mers representing edges. The optimal Eulerian paths are then found in the graph, which in turn represent contigs. De Bruijn assembly is much more time efficient when compared to overlap approaches because large numbers of read pairwise alignments do not need not to be calculated to identify overlap between reads, as read overlap is implicit in graph topology [118]. While the de Bruijn graph approach is computationally efficient, assembly of short reads almost always results in fragmented assemblies with limited quality due to limited read length [20].

Methods for assembly shifted once again following the coming of high-throughput, long reads [123, 124]. While variations on de Bruijn graph methods have been proposed to assemble long-reads [125, 126], most long-read assemblers rely on some overlap based approach [16, 127]. De Bruijn graphs can become very tangled due to the high error rates of long reads and lose the most important characteristic of long reads – positioning information. Several methods have been developed to deal with long reads, such as filling gaps and scaffolding with long reads [128, 129, 130], but a certain class of overlap-based graph approaches, termed the hierarchical approach, has been shown to generate the least fragmented assemblies for long-read only assembly [116, 127, 131].

Hierarchical methods are well-suited to deal with the high error rates of long reads. One of the key steps of this method is error correcting long reads before the merging of reads, which takes advantage of sufficient read depth to reach a consensus sequence. This can at times increase the base calling accuracy rate of long reads from ≈85% to ≈99.9% for the consensus reads [132]. Assemblers that do not initially error correct reads have been developed [133, 134], but are outperformed by hierarchical approaches [127, 131]. Hierarchical methods can be further split into two subcategories: hybrid methods, in which accurate consensus reads can be generated by aligning high-quality, short-read sequencing data onto long reads [132, 135], and non-hybrid methods, which usually involve mapping long reads onto themselves [15], although errors can be corrected using a de Bruijn graph approach [136, 137]. Hybrid methods are generally thought to be best suited when less than x30 long-read coverage is available, and non-hybrid assembly when at least more than x30 is available [138]. Once reads are error corrected, an overlap graph is constructed through detecting read-to-read overlap [139] and then traversed, resulting in an assembly.

3.1.2 Sequencing for Complete Recovery of Prokaryotic Chromosomes

When put in the context of generating complete bacterial and archaeal genomes, a clear answer regarding which sequencing technologies to use exists. The presence of long-read data is clearly required in order to produce a complete genome [8], as read length must exceed repeat length in order for complete assembly to be likely [51, 73]. Short-read sequencing alone is insufficient to close the gaps in many repeated regions [20, 43]. Whether long-read only assembly quality can match that of hybrid remains at this point somewhat convoluted, as some comparative analyses give a positive answer [15, 116] and some a negative [44, 48].

3.1.3 Current Recommendations for Optimal Coverage Depths

Determining necessary coverage for sequencing projects depends on several factors that include the choice of sequencing technology, sequencing biases inherent in different technologies, sequence read length, read error rates, choice of assembly algorithm, and repeat complexity of the target genome [51, 52, 140]. We define optimal coverage as the point where increased coverage does not improve assembly quality in terms of contiguity or accuracy. The question of how much coverage is required to generate a complete *de novo* assembly is the equivalent of asking at what point will almost all of sequencing errors be corrected and even the most complex repeats in a genome be resolved.

In 1988, Lander and Waterman presented the equation C=LN/G, where C equals expected genome-wide coverage, L equals read length, N equals the number of reads, and G equals the haploid genome size [141]. Key assumptions when calculating expected percentage of genome covered by assembled contigs is that reads are sampled in a completely randomly fashion [142]. However, theoretical coverage depths ignore sequencing biases, such as the biases introduced by amplification in regions of high GC content [68], k-mer specific biases introduced by stochastic disturbances at the pores in Nanopore sequencing [143], sequence biases of homopolymer regions [70], and that not all reads are assembled into contigs [144]. Therefore, estimates of the percentage of a genome covered by contigs using the Lander-Waterman approach often do not always match up with the true percentage of the genome covered by assembled contigs [144]. Even if every base pair in a genome was represented at least once, higher depth of coverage would be necessary in order to correct sequencing errors.

Non-theoretical approaches to estimating optimal depths of coverage also exist [96, 145]. These approaches assemble sequencing data at incremental depths and then attempt to identify the point where increased coverage does not increase assembly quality. However, these approaches are limited for two main reasons. First, these coverage benchmarks are derived from sequencing and assembling only a small number of bacteria and ignore the fact that optimal depths of coverage vary among lineages [51, 52, 140]. Second, the accuracy of assemblies is determined by comparing the *de novo* assemblies to an already known reference genome. This method of determining accuracy can be inaccurate, as regions that are characterized as assembly errors can in reality be due to naturally occurring strain-level variation [9, 26, 55].

3.1.4 Capabilities of Multiplexing Remain Underexplored

Multiplexing on a single long-read sequencing flow cell has produced several high-quality, complete genomes using either a non-hybrid assembly approach with PacBio sequencing [45, 46] or a hybrid assembly approach with MinION sequencing [48]. The question of how many microbial genomes can be effectively completed by sequence data from multiplexing has not been accurately answered.

3.2 Experiments and Results

3.2.1 Analysis of Short-read Only Assemblies

3.2.1.1 Data Curation

We selected 30 publicly available short-read sequencing datasets from NCBI's Short Read Archive (SRA database numbers located in Appendix A) in order to determine if high levels of coverage increased contiguity of short-read only assemblies. The data represented sequencing of various *Salmonella* strains, for a total of 28 strains, and the other 2 from *E. coli* strains that came from a single-cell sequencing project. The average read length from the *Salmonella* strains was 150 bp and coverage ranged from x30 to x100. The reads from the two *E. coli* sequencing runs had average read lengths of ≈250 bp and coverages of ≈x2,000. All of the selected reads were paired-end. Collectively, our selection represented sequencing data with a diverse range of read lengths and coverage depths for two common model organisms.

3.2.1.2 Assembly Contiguity over Coverage Increment

We subset each dataset at an increment of 7-fold coverage, assembled the subsequent downsized reads using the SPAdes assembler [85], and assessed the contiguity of the subsequent assemblies using QUAST [86]. We previously found that for almost all of the *E. coli* sequencing data with read lengths of 75, great improvement in number of contigs occurs at low coverage levels up until ≈x50 coverage, but increased assembly contiguity levels off between x50-x100 coverage (Fig. 2.1). Here, even with longer read lengths of 150 bp, contiguity still levels off at a certain point, which was ≈x30 in the case of the *Salmonella* strains (Figure 3.1). Contiguity leveled off at around x150 and x200 coverage even when sequencing to an ultra-high depth of x2000 in the case of the two *E. coli* strains (Figure 3.2).

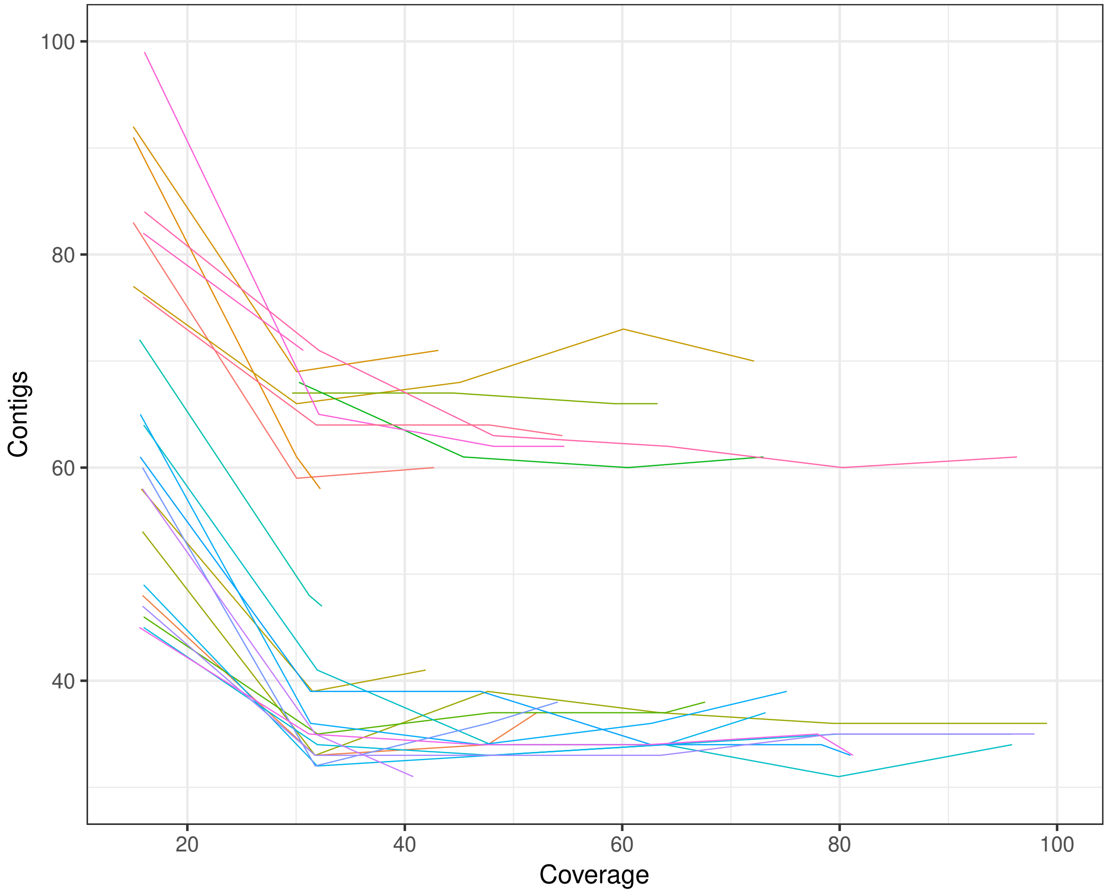

Figure 3.1: Contiguity of genome assemblies of 28 *Salmonella* strains at various depths of coverage. These datasets were 150 bp Illumina NextSeq 500 paired-end reads. All contigs are greater than 500 bp.

Indeed, at higher depths of coverage, the number of contigs increases in some of the selected strains. This is likely due to genome assemblers having difficulty dealing with large accumulations of sequencing errors [87]. In order to show that the leveling off of contiguity was not simply due to failings of the SPAdes algorithm, but simply due to the inherent biases of short-read sequencing and limits of short read length, we also ran our pipeline on an *E. coli* strain (SRA # SRR3989809) with the IDBA [88] and MEGAHIT

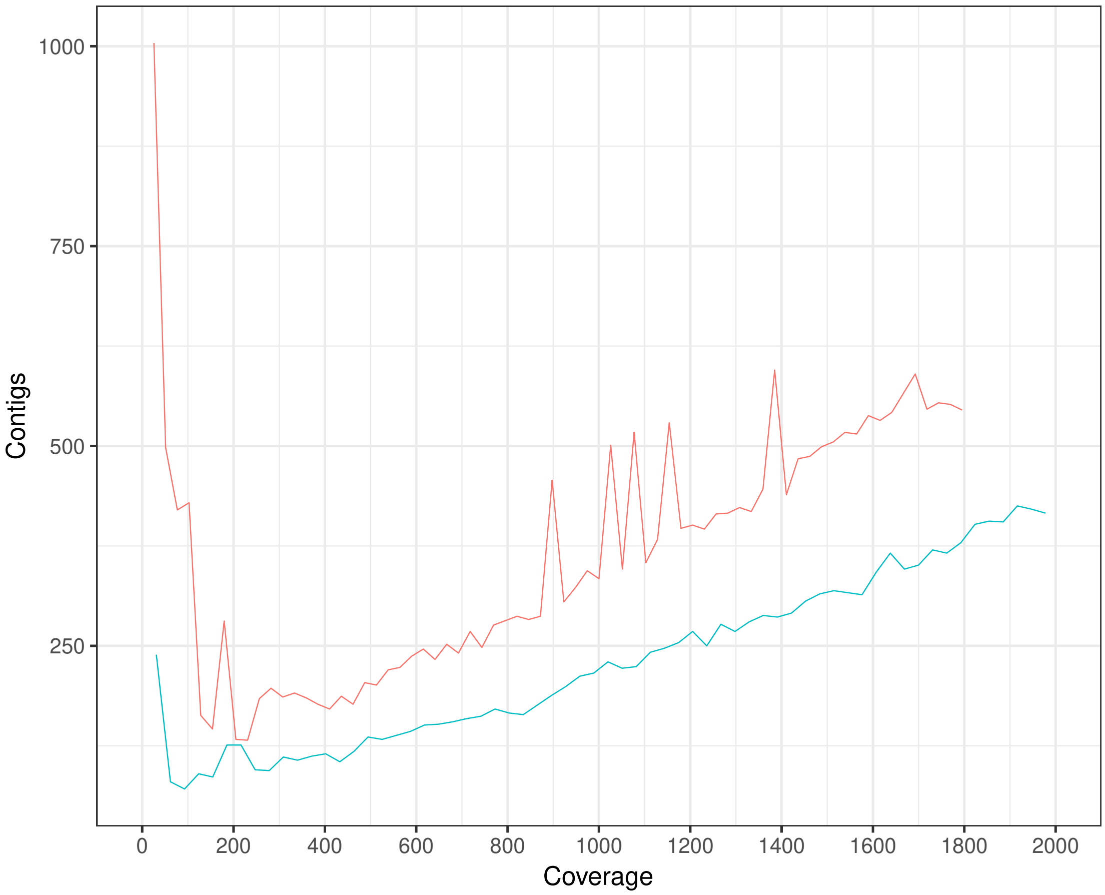

Figure 3.2: Contiguity of genome assemblies of 2 *E. coli* strains sequenced to ultra-high depths at various depths of coverage. All datasets were 250 bp Illumina MiSeq paired-end reads and were single-cell sequencing projects. All contigs are greater than 1,000 bp.

[146] assemblers, and the trend of the number of contigs decreasing none or very little at certain coverage depths remained (Appendix B).

3.2.2 Analysis of Non-hybrid Assemblies and Multiplexing

Due to the inherent limits of short reads being able to generate complete assemblies that we demonstrated in Chapter 2 of this thesis and that have been also demonstrated theoretically [20], we began to ask to what extent long-read only assemblies can produce high-quality assemblies. Long-read only assemblies have shown to be able to produce gapless genomes [8, 15]. However, simply gaging quality based on the number of contigs per chromosome or N50 value is not a full measure of completeness [80], as assembly accuracy is ignored. In order to get an exact measurement of the quality of a genome assembly, we simulated long-read sequencing data based on publicly available genomes classified as ”Complete” according to NCBI curation. Mapping the assemblies built with the simulated data unto their corresponding reference genomes gave us an exact measurement of assembly accuracy.

We chose to focus our analyses on PacBio data rather than Nanopore data. Non-hybrid assemblies of Nanopore data have shown to be limited in contiguity [48] most likely due to biases in base calling [16], while PacBio currently generates the least biased long reads available [70]. Furthermore, multiplexing on the Nanopore MinION produces highly variable distribution of reads per sample. Read output per sample has ranged from 0.092 Gb to 1.2 Gb per sample (a 13-fold difference) when multiplexing 12 *K. pneumoniae* (5.7 Mb) [48]. Multiplexing on PacBio's Sequel System has shown much less variation in output – output has ranged from 0.74 Gb to 1.7 Gb per sample [46] when multiplexing 8 various bacteria (≈4.0 Mb), 0.16 Gb to 0.4 Gb with 12 multiplexed Bacillus subtilis (4.0 Mb) [45], and 0.029 Gb to 0.056 Gb with 12 multiplexed *Helicobacter pylori* (1.6 Mb) [45]. In these studies, PacBio multiplexing exhibited at most only about a 2.5-fold coverage difference.

PacBio provides recommendations regarding the upper bound of how many total Mb of microbial genomes can be multiplexed (30-40 Mb depending on target genome complexity and quality of gDNA available) and has a cap of 16 on the level of multiplexing possible on their platforms [47]. To the best of our knowledge, there are no publicly available guidelines or recommendations to the upper-bound regarding the number of microbes that can be multiplexed on Nanopore's MinION. Without an idea of the upper bound of multiplexing level for a given sequencing platform, our recommending which types and sizes of microbial genomes that could be completed by multiplexing would be only marginally useful.

3.2.2.1 Data Curation

We filtered the complete list of single-chromosome prokaryotic, plasmid-free genomes from NCBI's GenBank that had an assembly status classified as "Complete Genomes" down to a count of 7,779. From this list, we selected 311 genomes over various size ranges to analyze (Table 3.1).

Genome Size (Mb)	# Selected for Analysis
1.9-2.1	64
2.9-3.1	48
3.9-4.1	42
4.9-5.1	32
5.1-6.1	89
6.1-7.1	36

Table 3.1: Size ranges of all selected genomes for analysis

For every selected reference genome (complete list of GenBank accession numbers located in Appendix A), we simulated long-read data with read profiles identical to previously sequenced x8 multiplexed *E. coli* genomes (available at https://github.com/PacificBiosciences/DevNet/wiki/8-plex-Ecoli-Multiplexed-Microbial-Assembly) using the software SimLoRD [147]. Error rate model was set to the standard for PacBio data (1, 2, and 12% error rate for substitutions, deletions, and insertions, respectively), read length was log-normal distributed, and mean read length was ≈4,410.

We randomly subsampled the simulated PacBio reads at coverage increments of x5 ranging from 5-fold to 150-fold and then assembled the subset read data using the Canu assembler [116]. We used the software QUAST [86] to assess the quality of each assembly across the range of coverage. We also used QUAST to map the assembled contigs back onto the corresponding reference genome off of which a read set was simulated, which provided the percentage of the true genome represented and amount of genes compared to the known amount. All assemblies at a coverage depth of 5 were filtered out of all downstream analyses, as every assembly at this depth was highly fragmented and possessed extremely low accuracy.

3.2.2.2 Gapless Assemblies and Genome Size

Genomes without gaps were generated for slightly over 90% of the 311 selected organisms at some coverage depth between x5 and x150. Assemblies for 96% of the organisms covered more than 99.5% of their corresponding reference genome. We then reported the percentage of genomes that were completed in terms of genome size (Table 3.2).

Genome Size (Mb)	% of Gapless Assemblies
1.9-7.1	91.0
1.9-4.1	94.2
4.9-7.1	87.9
1.9-2.1	92.2
2.9-3.1	97.9
3.9-4.1	92.8
4.9-5.1	87.5
5.1-6.1	89.9
6.1-7.1	83.3

Table 3.2: Percentage of gapless assemblies across genome size ranges

3.2.2.3 Optimal Coverage and Genome Size

We then found the minimum coverage that yielded the highest amount of contiguity or accuracy for each organism (Tables 3.3 and 3.4). Out of the entire pool, the mean and median of the optimal coverage were 36 and 30, respectively, and excluding the

28 fragmented assemblies had little effect on the mean and median of the depth of coverage. The higher means as compared to medians indicate that the distribution of required coverage is skewed towards genomes that require higher coverage depth.

Genome Size (Mb)	Mean	Median
1.9-7.1	x36	x30
1.9-2.1	x27	x25
2.9-3.1	x33	x26
3.9-4.1	x36	x30
4.9-5.1	x34	x30
5.1-6.1	x43	x30
6.1-7.1	x42	x35

Table 3.3: Optimal coverage in terms of contiguity across genome size ranges

Genome Size (Mb)	Mean	Median
1.9-7.1	x70	x61
1.9-2.1	x73	x63
2.9-3.1	x73	x66
3.9-4.1	x72	x61
4.9-5.1	x71	x61
5.1-6.1	x66	x66
6.1-7.1	x63	x56

Table 3.4: Optimal coverage in terms of accuracy across genome size ranges

As genome size increased, mean required coverage in terms of contiguity to generate a gapless genome increased by x15 and mean required coverage in terms of accuracy decreased by x10 as genome size increased. Due to assembling these genomes at a coverage increment of x5, the corresponding coverage estimates could be affected by probable variations. Therefore, the trends between coverage estimates and genome size could be impacted by assembling across a coverage increment and/or random variation of selected strains.

3.2.2.4 Assembly Accuracy over Coverage Increment

The authors of Canu [116] show that as little as x20-x30 coverage can produce complete genomes. They cite contiguity statistics which show that this coverage depth can produce a single contig per chromosome assemblies in bacteria. However, this definition of completeness is somewhat narrow, as it does not properly address assembly accuracy. By analyzing the mean fraction of the corresponding reference genome that is represented over a coverage increment for each of our 311 selected strains, we found that increasing coverage from x5 to x30 initially greatly improved assembly accuracy and increasing coverage beyond x30 generally returned minimal benefits (Figure 3.3). However, by plotting out a local regression line using the LOESS method [148], we observed that local variation in accuracy along the coverage increment exists and that 30-fold coverage is not adequate for all strains (Figure 3.3). Such local variations explain why the ≈x70 mean optimal coverage in terms of accuracy for various genome sizes (Table 3.4) and repeat profiles (Table 3.7) is higher than x30 coverage.

Even the seemingly small increase of accuracy from 99.7% to 99.9% translates into a decrease from 12,000 errors to 4,000 for a 4 Mb genome. An optional and computationally intensive post-assembly polishing step may be able to further the mean percentage of reference genome represented [149]. Only expending the resources needed to sequence to only x20-x30 coverage for bacteria is impossible on PacBio's Sequel System without multiplexing due to high sequencing data output.

Here we report coverage vs. percentage of reference genome represented. Percentage of assembly that represents the true reference assembly provides better measure of assembly quality than only contiguity statistics. Some of the assemblies only contained one contig at low coverage levels (x20-x30), but then became more fragmented as coverage increased, possibly due to Canu having difficulties with accumulation of sequencing

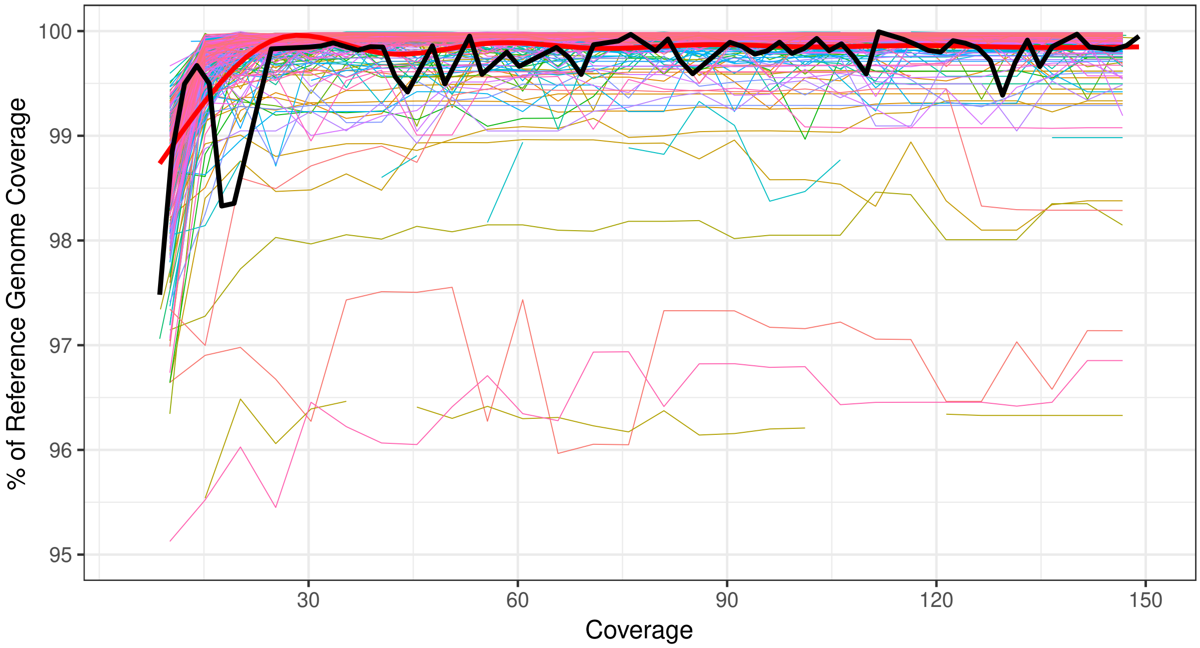

Figure 3.3: The mean fraction of an assembly's corresponding reference genome covered over a coverage increment. Each colored line represents a different strain. Gaps in a given colored line are simply due to lying out of the selected y-axis range. The smoothed regression line was set to red, and the local regression line calculated using the LOESS method (f = 0.09) was set to black. While generating a perfectly contiguous or almost contiguous assembly at depths as low as 20 in bacteria is possible [116], an increase of read data improves assembly accuracy for many strains. While the mean accuracy of the 311 assembled genomes was ≈99.9%, average percentage of reference genome covered varied among strains.

errors [87].

3.2.2.5 Gapless Assemblies and Complexity Class

Comparing total repeat length or percentage of the genome that contains repeated sequence with assembly quality as we did earlier with our short-read assemblies would not be useful. Long reads generally span most mid-sized repeats (repeats less than 5 kb) found in the majority of prokaryotic genomes [8]. Summing up total repeat length does not correctly measure genomic complexity in terms of long reads, as long reads typically resolve mid-sized repeats relatively easily. However, since PacBio read lengths are generally log-normally distributed [147], mid-sized repeats may not be covered by sufficiently long enough reads at all times.

Defining repeat complexity of an organism is difficult, as many different repeat classes exist [71, 150]. Koren et al. define three classes of repeat complexity in prokaryotes in terms of the rDNA operon [8], the largest repeat (5-7 kb) in most bacteria and archaea [97]. Class I prokaryotes contain few repeats other than the rDNA operon, Class II prokaryotes contain many mid-sized repeats while the rDNA operon remains the longest, and Class III prokaryotes contain repeats longer than the rDNA operon. The boundary between Classes I & II is set to the arbitrary count of 100. Out of all our 311 genomes analyzed, 86%, 5%, and 8% belonged to Class 1, 2, and 3, respectively.

We mapped each reference genome onto itself and found every repeat over 500 bp and 95% identity using MUMmer4 [84]. We found that out of the 26 Class III genomes (containing repeats longer than 7 kb) analyzed, 12 contained gaps. Class I & II genomes were on average much easier to resolve as compared to Class III genomes (Table 3.5).

Complexity Class	% of Gapless Assemblies
I, II, & III	91.0%
I & II	94.4%
III	53.8%

Table 3.5: Percentage of gapless assemblies in terms of genomic complexity classes

3.2.2.6 Optimal Coverage and Complexity Class

We then looked at the mean optimal coverage across our entire selected clade of organisms in terms of both contiguity and accuracy. We found that more complex repeat structure and higher desired assembly accuracy required higher levels of coverage (Tables 3.6 and 3.7).

Complexity Class	Mean	Median
I, II, & III	x36	x30
I & II	x34	x26
III	x60	x48

Table 3.6: Optimal coverage in terms of assembly contiguity across complexity classes

Complexity Class	Mean	Median
I, II, & III	x70	x61
I & II	x69	x61
III	x76	x71

Table 3.7: Optimal coverage in terms of assembly accuracy across complexity classes

Koren et al. [116] show that assembling with x20 PacBio coverage using Canu outperforms a x20 PacBio and x100 Illumina hybrid SPAdes assembly when simply looking at contiguity statistics and recommend using a hierarchical method when over x20 coverage of PacBio is attainable. We show that for the most complex class of genomes, x76 coverage produced the highest level of accuracy on average (Table 3.7). We note that since genomes were assembled across a coverage increment of x5, these coverage estimates lie within the corresponding range.

3.2.2.7 Gapless Assemblies and GC Content

Bacteria range in GC content from 17% to 75% and variation in base composition has been found to be corrlated with several factors, which includes genome size, coding sequence length, and and various environmental factors [151, 152]. The mean and median GC content of all 311 genomes were 53.5 and 54.4, respectively. We reported GC content ranges of selected genomes and percentage of completed genomes across each GC content range (Tables 3.8 and 3.9). Genomes at the high and low of the GC content spectrum were slightly more likely to contain gaps. We found no correlation between GC content and optimal coverage in terms of contiguity or accuracy.

% of GC Content	# Selected for Analysis
28.0-40.0	54
40.0-50.0	70
50.0-60.0	68
60.0-67.0	66
67.0-75.0	53

Table 3.8: GC content ranges of all selected genomes for analysis

% of GC Content	% of Gapless Assemblies
28.0-75.0	91.0
28.0-40.0	87.0
40.0-50.0	92.9
50.0-60.0	97.1
60.0-67.0	84.8
67.0-75.0	92.5

Table 3.9: Percentage of gapless assemblies across a GC content ranges

3.2.2.8 Phylogenetic Perspective

We then attempted to answer the following question: do genomes that are not able to be completely resolved cluster phylogenetically? Reporting which clades are especially problematic for assemblers would give researchers better guides to sequencing projects.

We then used the software RNAmmer [153] to annotate the 16S rDNA sequence of each reference genome and constructed a maximum-likelihood phylogenetic tree using IQTree [154]. Ideal substitution model was found using ModelFinder [155] and bootstrap values were calculated using the bootstrap method from Hoang et al. [156]. The phylogenetic tree was annotated using software from He et al. [157]. We found that unresolved genomes clustered together on some clades of the phylogenetic tree while many clades assembled without problem (Figure 3.4). Phylogenetic trees of the same organsisms as on Figure 3.4 that include taxonomic labels, bootstrap values, and branch lengths can be found in Appendix B.

3.2.2.9 Evolution of Repeats

The evolution of repeats, genome size, or other genomic characteristics in microbes has been well documented in the literature. Genome size has already been found to be correlated to bacterial and archaeal clades on the Tree of Life [103] and evolution of repeats is a complex research area that has previosuly been studied [97]. Our findings add that genomes that are unable to be resolved by a level of x8 multiplexing also cluster together to some extent. This level of clustering may be due to simple chance. Calculating if statistically significant clustering of traits across a phylogenetic tree occurs due to chance is possible using statistics such as Pascal's λ [158, 159]. However, such statistical methods have been scrutinized, as the resulting measures of significance can be based on the topology of only one tree [160] and varying evolutionary rates can cause drastic bias of phylogenetic signal [161].

Theoretically, an evolutionary event that introduces a repeat longer than 7 kb (as defined as the border between Class II & III genome complexity levels), would cause all subsequent lineages to be almost 40% more likely to be unresolved by our chosen level of multiplexing. This percentage increase is based on our previous findings of the percentages of genomes in each complexity class that were resolved. However, one may still be able to resolve genomes for species in complex clades at a level of multiplexing, even if the odds are significantly lower. Out of the pool of selected genomes containing the top 10 longest repeat regions ranging from 17 to 61 kb, two were still gapless. Two

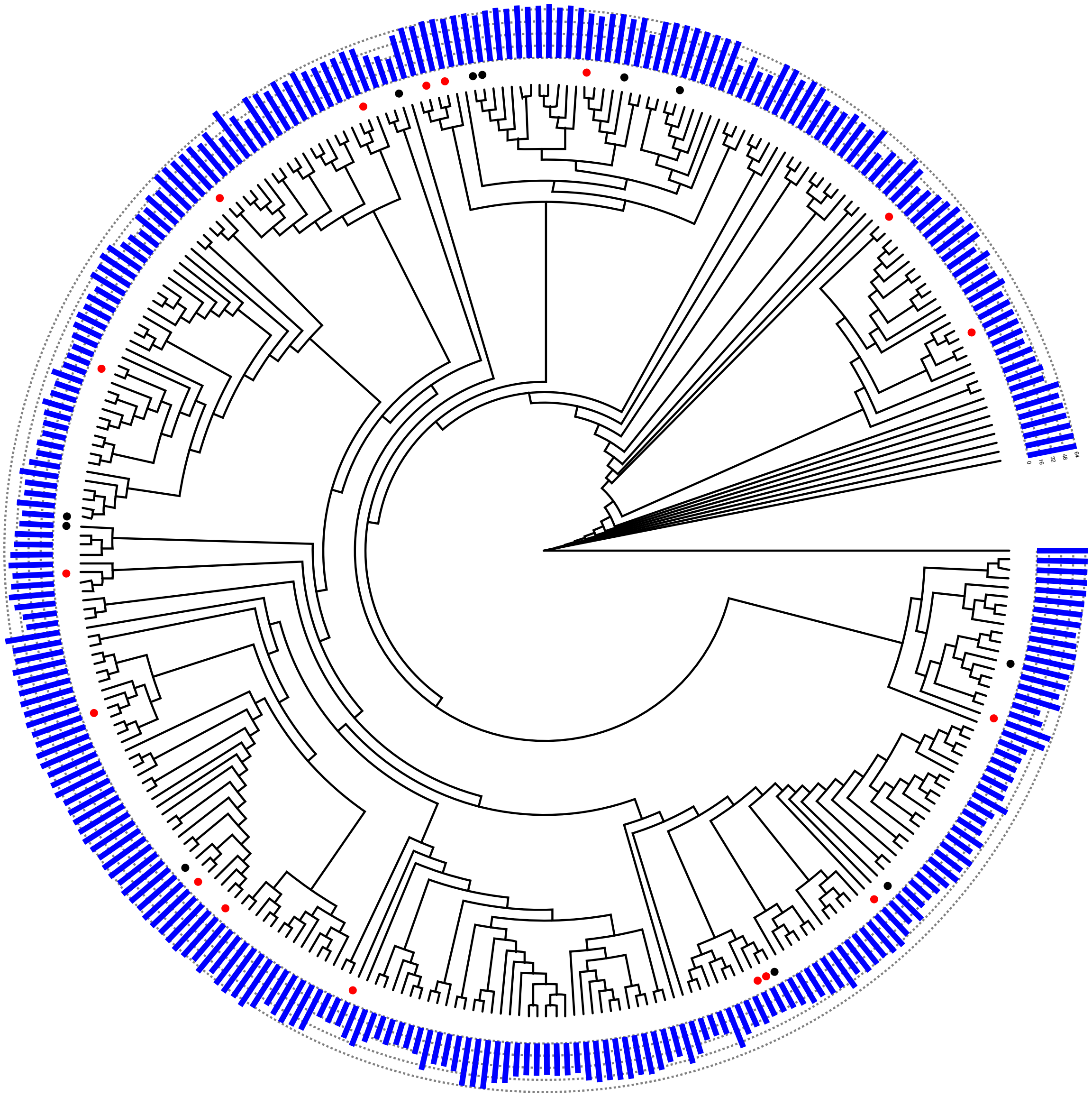

Figure 3.4: A maximum-likelihood unrooted tree of the 16S rRNA sequences of the 311 selected prokaryotic strains. Red dots represent Class I & II lineages that were unable to be fully resolved with x8 multiplexed PacBio simulated data, black dots represent Class III lineages that were unable to be fully resolved with x8 multiplexed PacBio simulated data, and blue bars signify GC content.

characteristics of long-read sequencing platforms explain this phenomenon – the ability of long-read sequencing platforms to generate sequence reads in a more uniform and random fashion across the genome as compared to short-read sequencers [15] and the fact that long-read length profiles possessing a log-normal distribution [147]. These characteristics can account for the longest repeats in these two genomes to be spanned by the longest reads simply due to chance.

3.3 Discussion

Improved developments of sequencing technologies are typically followed by optimization of genome assembly algorithms designed to deal with the types of new read data [162]. This can be seen in the shift of using greedy, de Bruijn, or overlap assembly approaches over time [106]. These trends are still seen, with recent assemblers specifically designed to deal with noisy long-reads [116, 133]. However, the practical use of such algorithms is still limited by researchers not knowing the ideal sequencing depth at the beginning of sequencing projects.

To the best of our knowledge, no usable model of how much coverage is optimal for genome assembly exists. The theoretical depth coverage from reads can vary greatly from the amount of coverage provided by assembled contigs [144]. Single, broad estimates for optimal coverage depth for bacterial genomes have been provided [96], but these estimates ignore the diverse levels of complexity in bacteria and archaea and the fact that different assembly algorithms require different depths of coverage [51]. Furthermore, simply assuming that longer total repeat sequence of a genome translates into higher optimal coverage is also a fallacy. Repeats of several kbs in length are harder than repeats of a few hundred bps to resolve, but the total length of repeats in the length of hundreds can add a significant amount to the total repeated sequence. Even if individual regions in prokaryotic repeat databases like the GenomeCRISPR database [163] or Microsatellite Database [164] had some sort of individual measure of complexity, no widely accepted genome-wide complexity classification system currently exists.

Excessive short-read coverage not only fails to produce high-quality genomes, but can also be an unnecessary expenditure of time and finances. Too much coverage translates into higher costs of sequencing, more data storage space, and increased runtime of assembly algorithms [165]. For example, one of the previously mentioned *E. coli* datasets (SRA # DRR079902) was sequenced to past the depth of x1,800. Past the depth of x600, there was inconsequential improvement of the number of contigs, largest contig, and N50 value. Therefore, the x1,200 additional coverage (≈5 Gb of sequencing data) represented unnecessary sequencing in terms of assembly contiguity. Additional required storage space may be multiplied when accounting for many funding sources now requiring multiple backups of raw sequencing data. The runtime of SPAdes at x600 coverage was only 1.5 hours, while at x1800 it was 8.25 hours, despite utilizing 16 cores, each with 30 GB of RAM, on our university's high-performance computing clusters. Ultra-deep sequencing may be needed to detect extremely rare variants [166], but for typical sequencing projects this represents unnecessary data.

We found that certain genomes were unable to be resolved with simulated reads possessing multiplexed length profiles. Naturally, criticisms that simulated data are not "real data" always will be raised, but without simulated reads, no efficient way to measure the true accuracy on a large scale exists due to the inherent nature of *de novo* assembly.

Different genomic characteristics can impact how much coverage is needed to assemble a complete genome. The largest change in mean optimal coverage in terms of either contiguity or accuracy occured as genomic complexity increased. The shift between Class I & II and III genomes caused a jump of 24-fold in mean optimal coverage in terms of contiguity and a 7-fold increase in terms of accuracy. Mean optimal coverage in terms of contiguity only increased by 15-fold for ≈2 Mb to ≈7 Mb genomes and decreased by 10-fold in terms of accuracy across the same size range. The fact that mean optimal coverage in terms of assembly accuracy slightly decreased over a genome size increment could be due to more larger genomes being unable to be completed than smaller genomes proportionally – the gaps more commonly found in larger genomes in our analyses that were unable to be resolved at any coverage between x5 and x150 limit higher assembly accuracy due to missing pieces that are unable to cover sections of the subsequent reference genome and increased coverage can do nothing to increase assembly accuracy in this case.

Our report on the percentages of Class I & II and Class III genomes that were able to be completed by multiplexed data (94% and 54%, respectively) using the Canu assembler and the estimates on optimal coverage depths in terms of contiguity and accuracy may be impacted by inherent bias in NCBI's GenBank. The genomes we analyzed were database dependent, and some difficult to assemble genomes may not have been deposited into the database as completed in the first place. Additionally, the quality of around 10% of publicly available, long-read assembled *de novo* genomes that were not closed with labor-intensive steps such as primer walking has been questioned [162]. For example, some *Pseudomonas koreensis* strains contain ultra long, highly similar repeats (around 70 kb) that cannot even be closed by PacBio reads in some cases [167]. Currently, these errors in such publicly available genomes cannot be corrected, outside of resequencing every *de novo* genome with ultra-long Nanopore reads. Due to current database biases, our estimates of the amount of genomes able to be resolved without gaps and major errors by a level of x8 multiplexing most likely lies on the upper bound.

We also reported that the inability to close genomes clustered somewhat in a GC content specific and phylogenetic manner. Genomes ranging in GC content from 50.0% to 60%, which can be described in terms of information theory as prokaryotic genomes with relatively high levels of entropy [168], were more likely to be completed (97.1%) than genomes at the highs or lows of GC content. It must be noted that bacterial genomes can abruptly change due to movement of mobile elements [169], so categorizing a certain clade as ”assembly friendly” or not can be contradicted by acquisition or loss of mobile elements.

We recognize that some microbial sequencing projects do not have the ability to definitively label the target genome into a certain complexity class prior to long-read sequencing and assembly. Yet there are methods to estimate genomic complexity prior to long-read sequencing. Repeat structure of a target organism can be inferred by the repeat profile of a closely related species if taxonomic information is known. Species belonging to some sort of taxonomic grouping, such as species belonging to the *Yersinia* genus, have similar repeat profiles [170]. When taxonomic information is unknown, total repeat sequence length can be estimated by analyzing k-mer profiles [77]. However, as repeats' lengths can not be estimated, this technique is limited in terms of placing genomes into the aforementioned complexity classes.

The coverage needed to generate the most accurate and contiguous assembly depends on the complexity class of the target species. We show here that genomes containing repeats longer than the rDNA operon are unsuccessfully assembled with x8 multiplexed reads almost 40% more than genomes with no such repeats. We recommend that when a bacterial or archaeal target genome contains repeats longer than the rDNA operon, researchers should use either non-multiplexed long-read data or a low level of multiplexing. We suggest that researchers utilize multiplexing on long-read sequencing platforms when working with lineages belonging to complexity Class I & II and that more research be carried out to explore appropriate levels of multiplexing for genomes in various complexity classes.

Appendix A

Sequencing Read Data

SRA #	Organism	SRA #	Organism	SRA #	Organism
SRR3989713	*E. coli*	SRR3989714	*E. coli*	SRR3989715	*E. coli*
SRR3989716	*E. coli*	SRR3989717	*E. coli*	SRR3989718	*E. coli*
SRR3989719	*E. coli*	SRR3989720	*E. coli*	SRR3989721	*E. coli*
SRR3989722	*E. coli*	SRR3989723	*E. coli*	SRR3989724	*E. coli*
SRR3989725	*E. coli*	SRR3989726	*E. coli*	SRR3989727	*E. coli*
SRR3989728	*E. coli*	SRR3989729	*E. coli*	SRR3989730	*E. coli*
SRR3989731	*E. coli*	SRR3989732	*E. coli*	SRR3989733	*E. coli*
SRR3989734	*E. coli*	SRR3989735	*E. coli*	SRR3989736	*E. coli*
SRR3989737	*E. coli*	SRR3989738	*E. coli*	SRR3989739	*E. coli*
SRR3989740	*E. coli*	SRR3989741	*E. coli*	SRR3989742	*E. coli*
SRR3989743	*E. coli*	SRR3989744	*E. coli*	SRR3989745	*E. coli*
SRR3989746	*E. coli*	SRR3989747	*E. coli*	SRR3989748	*E. coli*
SRR3989749	*E. coli*	SRR3989750	*E. coli*	SRR3989751	*E. coli*
SRR3989752	*E. coli*	SRR3989753	*E. coli*	SRR3989754	*E. coli*
SRR3989755	*E. coli*	SRR3989756	*E. coli*	SRR3989757	*E. coli*
SRR3989758	*E. coli*	SRR3989759	*E. coli*	SRR3989760	*E. coli*

SRR3989761	*E. coli*	SRR3989762	*E. coli*	SRR3989763	*E. coli*
SRR3989764	*E. coli*	SRR3989765	*E. coli*	SRR3989766	*E. coli*
SRR3989767	*E. coli*	SRR3989768	*E. coli*	SRR3989769	*E. coli*
SRR3989770	*E. coli*	SRR3989771	*E. coli*	SRR3989772	*E. coli*
SRR3989773	*E. coli*	SRR3989774	*E. coli*	SRR3989776	*E. coli*
SRR3989777	*E. coli*	SRR3989778	*E. coli*	SRR3989779	*E. coli*
SRR3989780	*E. coli*	SRR3989781	*E. coli*	SRR3989782	*E. coli*
SRR3989783	*E. coli*	SRR3989784	*E. coli*	SRR3989785	*E. coli*
SRR3989786	*E. coli*	SRR3989787	*E. coli*	SRR3989788	*E. coli*
SRR3989789	*E. coli*	SRR3989790	*E. coli*	SRR3989791	*E. coli*
SRR3989792	*E. coli*	SRR3989793	*E. coli*	SRR3989794	*E. coli*
SRR3989795	*E. coli*	SRR3989796	*E. coli*	SRR3989797	*E. coli*
SRR3989798	*E. coli*	SRR3989799	*E. coli*	SRR3989800	*E. coli*
SRR3989801	*E. coli*	SRR3989802	*E. coli*	SRR3989803	*E. coli*
SRR3989804	*E. coli*	SRR3989805	*E. coli*	SRR3989806	*E. coli*
SRR3989807	*E. coli*	SRR3989808	*E. coli*		

Table A.1: SRA accession numbers of all short reads analyzed in Chapter 2

SRA #	Organism	SRA #	Organism	SRA #	Organism
SRR3934217	*S. enterica*	SRR3934218	*S. enterica*	SRR3934230	*S. enterica*
SRR3934231	*S. enterica*	SRR3934240	*S. enterica*	SRR3934245	*S. enterica*
SRR3934246	*S. enterica*	SRR3934247	*S. enterica*	SRR3934248	*S. enterica*
SRR3934249	*S. enterica*	SRR3934250	*S. enterica*	SRR3934251	*S. enterica*
SRR3934251	*S. enterica*	SRR3934252	*S. enterica*	SRR3934253	*S. enterica*

SRR3934254	*S. enterica*	SRR3934255	*S. enterica*	SRR3934256	*S. enterica*
SRR3934257	*S. enterica*	SRR3934262	*S. enterica*	SRR3934263	*S. enterica*
SRR3934264	*S. enterica*	SRR3934265	*S. enterica*	SRR3934281	*S. enterica*
SRR3934282	*S. enterica*	SRR3934283	*S. enterica*	SRR3934284	*S. enterica*
SRR3934285	*S. enterica*	SRR3934286	*S. enterica*	DRR078802	*E. coli*
DRR078803	*E. coli*				

Table A.2: SRA accession numbers of all short reads analyzed in Chapter 3

GenBank #	Organism	GenBank #	Organism
GCA_000247605.1	*Acetobacterium*	GCA_001457475.1	*Achromobacter*
GCA_000021485.1	*Acidithiobacillus*	GCA_000176855.2	*Acidovorax*
GCA_001307195.1	*Acinetobacter*	GCA_002234535.1	*Actinoalloteichus*
GCA_001747425.1	*Actinoalloteichus*	GCA_001262055.1	*Actinomyces*
GCA_001553935.1	*Actinomyces*	GCA_001553565.1	*Actinomyces*
GCA_001543145.1	*Aerococcus*	GCA_001543175.1	*Aerococcus*
GCA_900097105.1	*Akkermansia*	GCA_000300005.1	*Alcanivorax*
GCA_001310225.1	*Algibacter*	GCA_000016985.1	*Alkaliphilus*
GCA_001698205.1	*Altererythrobacter*	GCA_000025885.1	*Aminobacterium*
GCA_900128415.1	*Anaerococcus*	GCA_000022145.1	*Anaeromyxobacter*
GCA_000092365.1	*Arcanobacterium*	GCA_000385565.1	*Archaeoglobus*
GCA_000194625.1	*Archaeoglobus*	GCA_001294625.1	*Arthrobacter*
GCA_000010525.1	*Azorhizobium*	GCA_000380335.1	*Azotobacter*
GCA_000196735.1	*Bacillus*	GCA_000007845.1	*Bacillus*
GCA_000008165.1	*Bacillus*	GCA_001318345.1	*Bacteroides*

GCA_001314995.1	*Bacteroides*	GCA_000012825.1	*Bacteroides*
GCA_000512915.1	*Barnesiella*	GCA_002007565.1	*Bartonella*
GCA_000046705.1	*Bartonella*	GCA_000743945.1	*Basilea*
GCA_000525675.1	*Bdellovibrio*	GCA_000265505.1	*Bernardetia*
GCA_000010425.1	*Bifidobacterium*	GCA_001025155.1	*Bifidobacterium*
GCA_000022965.1	*Bifidobacterium*	GCA_000800475.2	*Bifidobacterium*
GCA_001676705.1	*Bordetella*	GCA_000318015.1	*Bordetella*
GCA_002119665.1	*Bordetella*	GCA_001078275.1	*Bordetella*
GCA_000195715.1	*Bordetella*	GCA_000067205.1	*Bordetella*
GCA_000010165.1	*Brevibacillus*	GCA_000635915.2	*Brevundimonas*
GCA_000016545.1	*Caldicellulosiruptor*	GCA_000281175.1	*Caldilinea*
GCA_001886815.1	*Caldithrix*	GCA_000018305.1	*Caldivirga*
GCA_002024185.1	*Campylobacter*	GCA_000017465.2	*Campylobacter*
GCA_000612685.1	*Castellaniella*	GCA_000006905.1	*Caulobacter*
GCA_000022005.1	*Caulobacter*	GCA_001308265.1	*Celeribacter*
GCA_000016085.1	*Chlorobium*	GCA_000018865.1	*Chloroflexus*
GCA_002025665.1	*Chryseobacterium*	GCA_000833105.2	*Clostridium*
GCA_000022065.1	*Clostridium*	GCA_000145275.1	*Clostridium*
GCA_000473995.1	*Clostridium*	GCA_000331995.1	*Clostridium*
GCA_001584185.1	*Collimonas*	GCA_000012325.1	*Colwellia*
GCA_000739375.1	*Comamonas*	GCA_000025265.1	*Conexibacter*
GCA_000550805.1	*Corynebacterium*	GCA_000196315.1	*Croceibacter*
GCA_000222485.1	*Cyclobacterium*	GCA_000953715.1	*Defluviitoga*
GCA_000512895.1	*Dehalobacter*	GCA_001953175.1	*Dehalogenimonas*
GCA_001644565.1	*Deinococcus*	GCA_000018665.1	*Delftia*

GCA_000021925.1	*Desulfitobacterium*	GCA_000307105.1	*Desulfobacula*
GCA_000018405.1	*Desulfococcus*	GCA_000023225.1	*Desulfomicrobium*
GCA_000235605.1	*Desulfosporosinus*	GCA_000215085.1	*Desulfotomaculum*
GCA_000021385.1	*Desulfovibrio*	GCA_000177635.2	*Desulfurispirillum*
GCA_000092205.1	*Desulfurivibrio*	GCA_001278055.1	*Desulfuromonas*
GCA_900070355.1	*Devriesea*	GCA_002214645.1	*Diaphorobacter*
GCA_001644705.1	*Dickeya*	GCA_000020965.1	*Dictyoglomus*
GCA_000626635.1	*Draconibacterium*	GCA_000023125.1	*Dyadobacter*
GCA_000632805.1	*Dyella*	GCA_000013005.1	*Erythrobacter*
GCA_000026345.1	*Escherichia*	GCA_000178115.2	*Ethanoligenens*
GCA_000152265.2	*Ferroplasma*	GCA_000017545.1	*Fervidobacterium*
GCA_000163895.2	*Filifactor*	GCA_000724625.1	*Fimbriimonas*
GCA_001831475.1	*Flavobacterium*	GCA_000455605.1	*Flavobacterium*
GCA_000016645.1	*Flavobacterium*	GCA_000013345.1	*Frankia*
GCA_000016745.1	*Geobacter*	GCA_000025345.1	*Geodermatophilus*
GCA_001698225.1	*Gordonia*	GCA_900170005.1	*Gordonibacter*
GCA_001951155.1	*Gramella*	GCA_000178955.2	*Granulicella*
GCA_000940805.1	*Gynuella*	GCA_001886955.1	*Halodesulfurarchaeum*
GCA_002075285.2	*Halomicronema*	GCA_000696485.1	*Halomonas*
GCA_001545155.1	*Halomonas*	GCA_001460635.1	*Helicobacter*
GCA_000019165.1	*Heliobacterium*	GCA_002025725.1	*Herbaspirillum*
GCA_001267925.1	*Herbaspirillum*	GCA_001040945.1	*Herbaspirillum*
GCA_000184685.1	*Intrasporangium*	GCA_000723165.1	*Janthinobacterium*
GCA_001017655.1	*Kiritimatiella*	GCA_000215745.1	*Klebsiella*
GCA_000300455.4	*Kosakonia*	GCA_000011985.1	*Lactobacillus*

GCA_000785105.2	*Lactobacillus*	GCA_000010145.1	*Lactobacillus*
GCA_001050435.1	*Lactobacillus*	GCA_000016825.1	*Lactobacillus*
GCA_000026505.1	*Lactobacillus*	GCA_000224985.1	*Lactobacillus*
GCA_000269925.1	*Lactococcus*	GCA_000166395.1	*Leadbetterella*
GCA_000019785.1	*Leptothrix*	GCA_000196855.1	*Leuconostoc*
GCA_000196035.1	*Listeria*	GCA_001190945.1	*Luteipulveratus*
GCA_001543325.1	*Lutibacter*	GCA_001442535.1	*Lysobacter*
GCA_001442785.1	*Lysobacter*	GCA_000284615.1	*Marinobacter*
GCA_001043175.1	*Marinobacter*	GCA_000024425.1	*Meiothermus*
GCA_000024185.1	*Methanobrevibacter*	GCA_000006175.2	*Methanococcus*
GCA_001560915.1	*methanogenic*	GCA_001889405.1	*Methanohalophilus*
GCA_000025865.1	*Methanohalophilus*	GCA_000306725.1	*Methanolobus*
GCA_000007345.1	*Methanosarcina*	GCA_000970285.1	*Methanosarcina*
GCA_000970085.1	*Methanosarcina*	GCA_000021965.1	*Methanosphaerula*
GCA_000785705.2	*Methylomonas*	GCA_000214665.1	*Methylomonas*
GCA_000093025.1	*Methylotenera*	GCA_002209385.1	*Methylovulum*
GCA_000202635.1	*Microbacterium*	GCA_001617625.1	*Microbulbifer*
GCA_000010625.1	*Microcystis*	GCA_000270245.1	*Microlunatus*
GCA_000145235.1	*Micromonospora*	GCA_000306785.1	*Modestobacter*
GCA_900078775.1	*Mycobacterium*	GCA_001632805.1	*Mycobacterium*
GCA_002105755.1	*Mycobacterium*	GCA_001307545.1	*Mycobacterium*
GCA_000277125.1	*Mycobacterium*	GCA_002007745.1	*Mycobacterium*
GCA_001583415.1	*Mycobacterium*	GCA_000230895.3	*Mycobacterium*
GCA_000317305.3	*Mycobacterium*	GCA_000015005.1	*Mycobacterium*
GCA_001655245.1	*Mycobacterium*	GCA_000015305.1	*Mycobacterium*

GCA_000833025.1	*Myroides*	GCA_000024365.1	*Nakamurella*
GCA_002156705.1	*Natrialbaceae*	GCA_000591055.1	*Natronomonas*
GCA_001654455.1	*Niabella*	GCA_001007935.1	*Nitrosomonas*
GCA_900169565.1	*Nitrospira*	GCA_000284035.1	*Nocardia*
GCA_000294515.1	*Nocardiopsis*	GCA_000332115.1	*Nonlabens*
GCA_001586165.1	*Obesumbacterium*	GCA_002162375.1	*Oleiphilus*
GCA_000143845.1	*Olsenella*	GCA_000019965.1	*Opitutus*
GCA_000236705.1	*Owenweeksia*	GCA_000961095.1	*Paenibacillus*
GCA_001465255.1	*Paenibacillus*	GCA_000758725.1	*Paenibacillus*
GCA_000758685.1	*Paenibacillus*	GCA_001644605.1	*Paenibacillus*
GCA_001685395.1	*Paenibacillus*	GCA_000767615.3	*Pandoraea*
GCA_002079945.1	*Parasaccharibacter*	GCA_000017565.1	*Parvibaculum*
GCA_000152825.2	*Parvularcula*	GCA_001590605.1	*Pedobacter*
GCA_000023825.1	*Pedobacter*	GCA_001721645.1	*Pedobacter*
GCA_000020645.1	*Pelodictyon*	GCA_000271665.2	*Pelosinus*
GCA_001678945.1	*Phaeobacter*	GCA_001010285.1	*Photorhabdus*
GCA_000785495.1	*Pimelobacter*	GCA_000025185.1	*Pirellula*
GCA_000317025.1	*Pleurocapsa*	GCA_000757785.1	*Pluralibacter*
GCA_001017435.1	*Polyangium*	GCA_000973625.1	*Polynucleobacter*
GCA_000973725.1	*Pontibacter*	GCA_001663175.1	*Porphyrobacter*
GCA_001026985.1	*Pragia*	GCA_000193395.1	*Prevotella*
GCA_000014225.1	*Pseudoalteromonas*	GCA_001563225.1	*Pseudodesulfovibrio*
GCA_000006765.1	*Pseudomonas*	GCA_000237065.1	*Pseudomonas*
GCA_000213805.1	*Pseudomonas*	GCA_000016565.1	*Pseudomonas*
GCA_000397205.1	*Pseudomonas*	GCA_000007565.2	*Pseudomonas*

GCA_000012245.1	*Pseudomonas*	GCA_002119765.1	*Pseudorhodoplanes*
GCA_000217815.1	*Pseudothermotoga*	GCA_000007305.1	*Pyrococcus*
GCA_001577775.1	*Pyrococcus*	GCA_001412615.1	*Pyrodictium*
GCA_000215705.1	*Ramlibacter*	GCA_001580455.1	*Ramlibacter*
GCA_002116905.1	*Rhizobacter*	GCA_000982715.1	*Rhodococcus*
GCA_000196695.1	*Rhodococcus*	GCA_000166055.1	*Rhodomicrobium*
GCA_000013365.1	*Rhodopseudomonas*	GCA_000013745.1	*Rhodopseudomonas*
GCA_000014825.1	*Rhodopseudomonas*	GCA_001483865.1	*Roseateles*
GCA_000017805.1	*Roseiflexus*	GCA_000165715.3	*Rubinisphaera*
GCA_000284255.1	*Rubrivivax*	GCA_000013665.1	*Saccharophagus*
GCA_000018265.1	*Salinispora*	GCA_000016425.1	*Salinispora*
GCA_001006005.1	*Serratia*	GCA_000513215.1	*Serratia*
GCA_001572725.1	*Serratia*	GCA_002075795.1	*Shewanella*
GCA_000018285.1	*Shewanella*	GCA_000014885.1	*Shewanella*
GCA_002005305.1	*Shewanella*	GCA_000018025.1	*Shewanella*
GCA_000091325.1	*Shewanella*	GCA_000025705.1	*Sideroxydans*
GCA_001586195.1	*Solibacillus*	GCA_000242635.3	*Solitalea*
GCA_000485905.1	*Spiribacter*	GCA_001988955.1	*Spirosoma*
GCA_000974425.1	*Spirosoma*	GCA_002067135.1	*Spirosoma*
GCA_000024545.1	*Stackebrandtia*	GCA_000831485.1	*Streptococcus*
GCA_000463355.1	*Streptococcus*	GCA_000007465.2	*Streptococcus*
GCA_000007045.1	*Streptococcus*	GCA_000253395.1	*Streptococcus*
GCA_000993785.2	*Streptomyces*	GCA_900079115.1	*Sulfolobus*
GCA_000014965.1	*Syntrophobacter*	GCA_000014725.1	*Syntrophomonas*
GCA_001483385.1	*Tenacibaculum*	GCA_000023025.1	*Teredinibacter*

GCA_000265425.1	*Terriglobus*	GCA_000179915.2	*Terriglobus*
GCA_000355675.1	*Thalassolituus*	GCA_000305935.1	*Thermacetogenium*
GCA_002214465.1	*Thermococcus*	GCA_000265525.1	*Thermococcus*
GCA_000816105.1	*Thermococcus*	GCA_000009965.1	*Thermococcus*
GCA_000585495.1	*Thermococcus*	GCA_000517445.1	*Thermococcus*
GCA_001647085.1	*Thermococcus*	GCA_002214505.1	*Thermococcus*
GCA_002214545.1	*Thermococcus*	GCA_000020985.1	*Thermodesulfovibrio*
GCA_000024385.1	*Thermomonospora*	GCA_000021285.1	*Thermosipho*
GCA_000016905.1	*Thermosipho*	GCA_000828655.1	*Thermotoga*
GCA_000321415.2	*Thioalkalivibrio*	GCA_001020955.1	*Thioalkalivibrio*
GCA_000012745.1	*Thiobacillus*	GCA_000227745.3	*Thiocystis*
GCA_000214825.1	*Thiomicrospira*	GCA_000212415.1	*Treponema*
GCA_000214375.1	*Treponema*	GCA_000184745.1	*Variovorax*
GCA_001677435.1	*Woeseia*	GCA_000007145.1	*Xanthomonas*
GCA_000019585.2	*Xanthomonas*	GCA_000973105.1	*Zobellia*
GCA_000023465.1	*Zunongwangia*		

Table A.3: GenBank accession numbers of all complete genomes used to simulate long-read data in Chapter 3

Appendix B

Supplementary Graphs

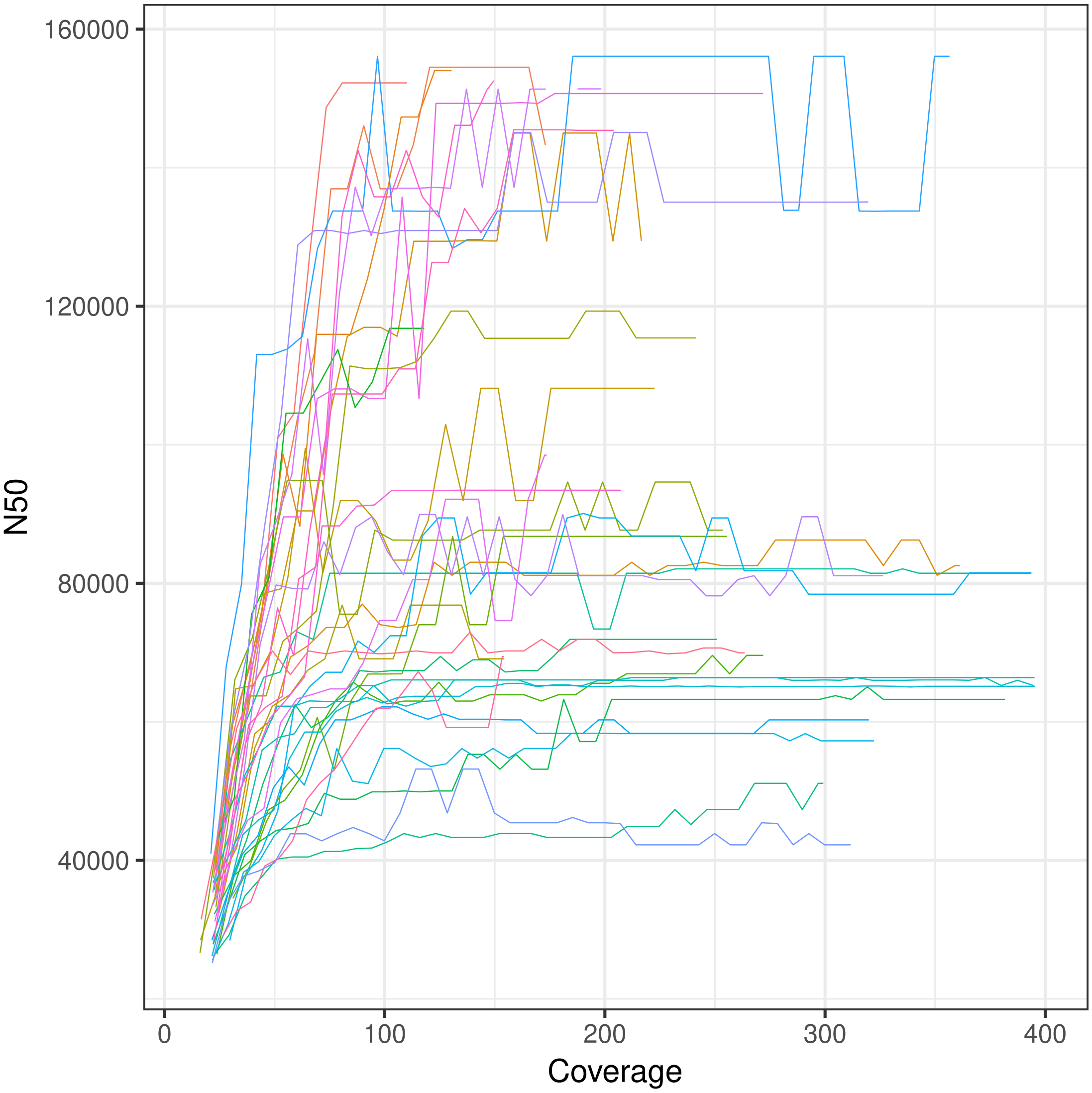

Figure B.1: Depth of coverage vs. N50 from assemblies of 34 various *E. coli* strains. Each colored line represents a different strain. Assembly contiguity varies among the closely-related bacterial strains and increased coverage provides little increase in genome assembly contiguity past 100-fold coverage for most strains. Coverage was calculated by dividing number of sequenced base pairs by the total length of contigs $\geq$ 500 bp.

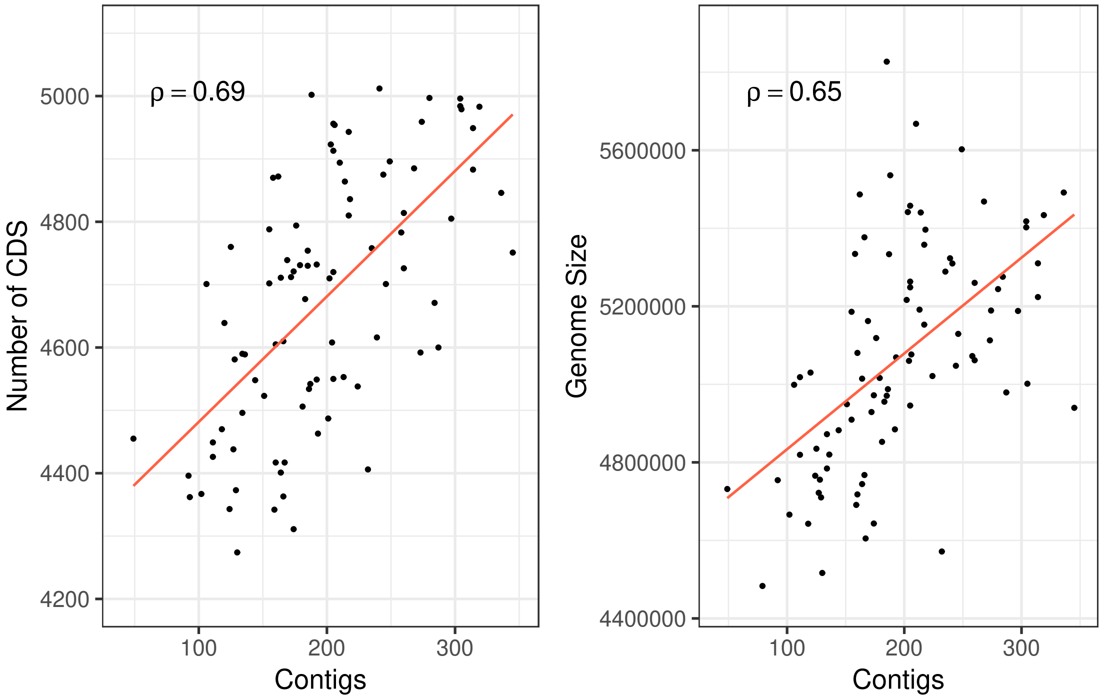

Figure B.2: Contig number vs. the number of coding regions and genome size – based on IDBA assemblies. Variation in genome assembly quality, assessed here as genome assembly contiguity, can be attributabed to the correlation between genome size and the number of coding regions. The p values and 95% confidence intervals of ρ from left to right are 4.77e-15 and 7.50e-13 and 0.57–0.78 and 0.52–0.75.

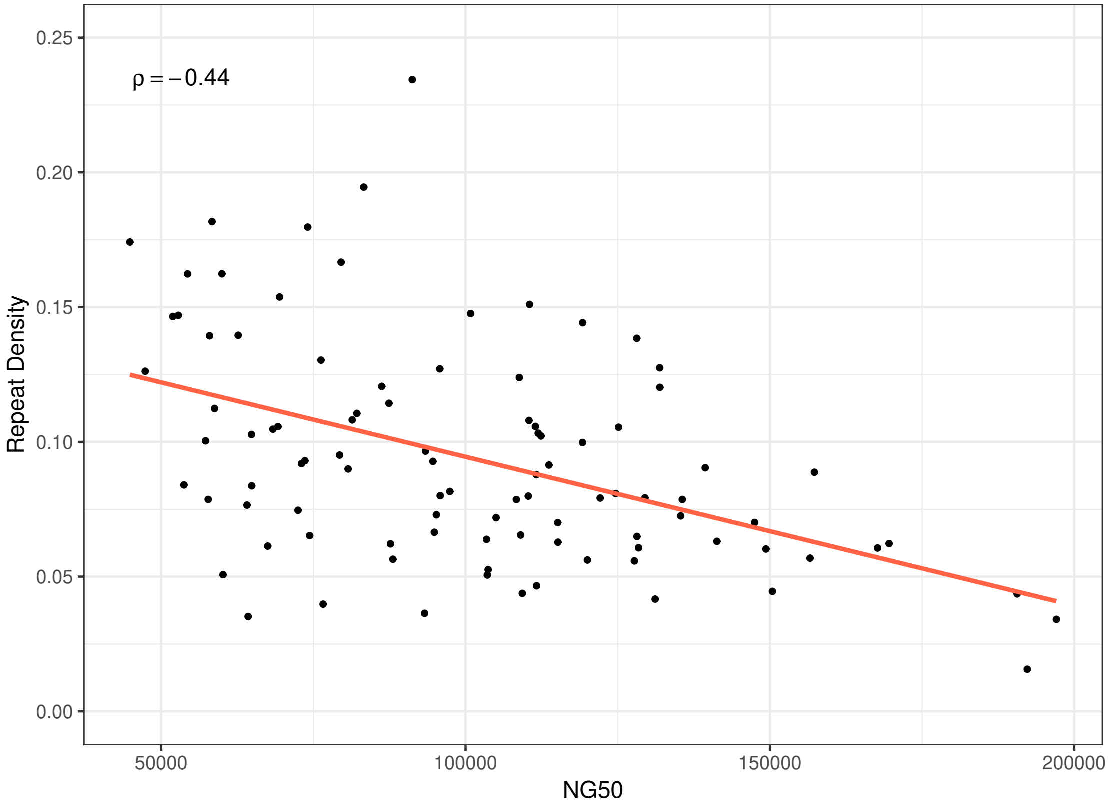

Figure B.3: NG50 vs. repeat density of the assembled genomes – based on IDBA assemblies. This correlation indicates that assembly quality does not only decrease while genome size increases due to a natural growth in repeats along with genome size, but rather that an increase in repeats in a genome negatively affects assembly quality independently of genome size. The p value and 95% confidence interval of ρ are 8.72e-06 and -0.59—-0.26.

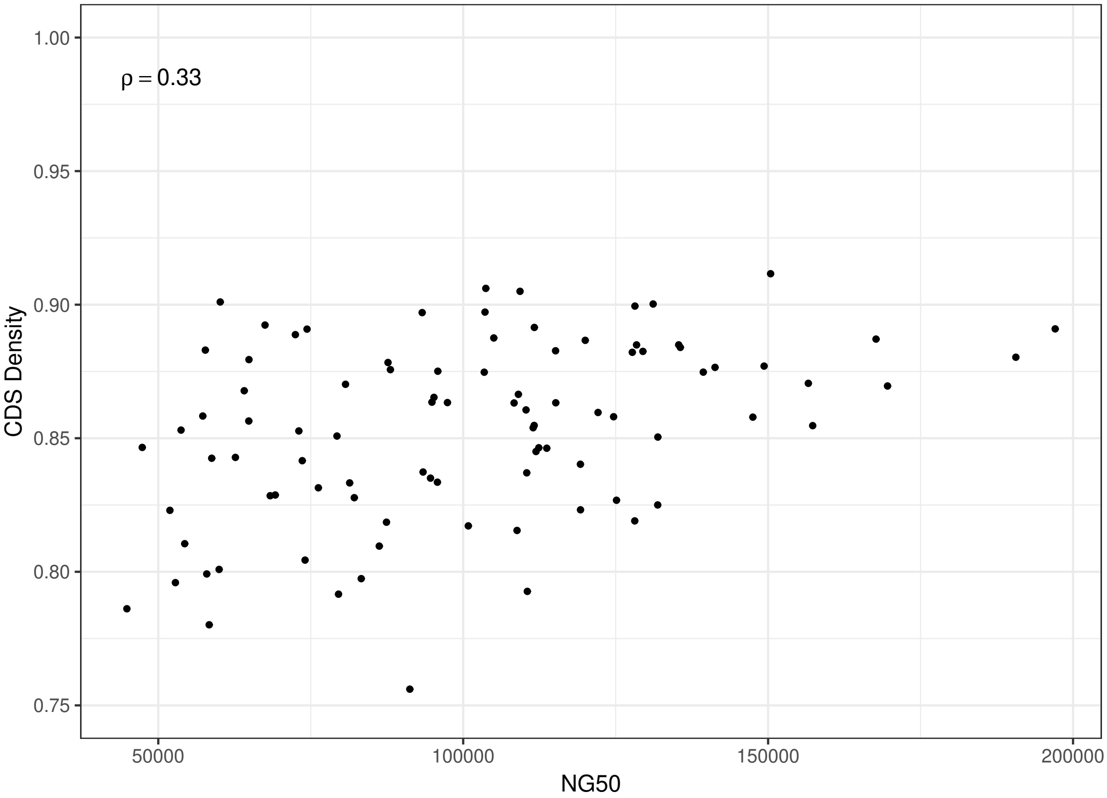

Figure B.4: NG50 shows little correlation with the abundance of coding regions in a genome – based on IDBA assemblies. The p value and 95% confidence interval of ρ are 1.25e-3 and 0.13–0.49.

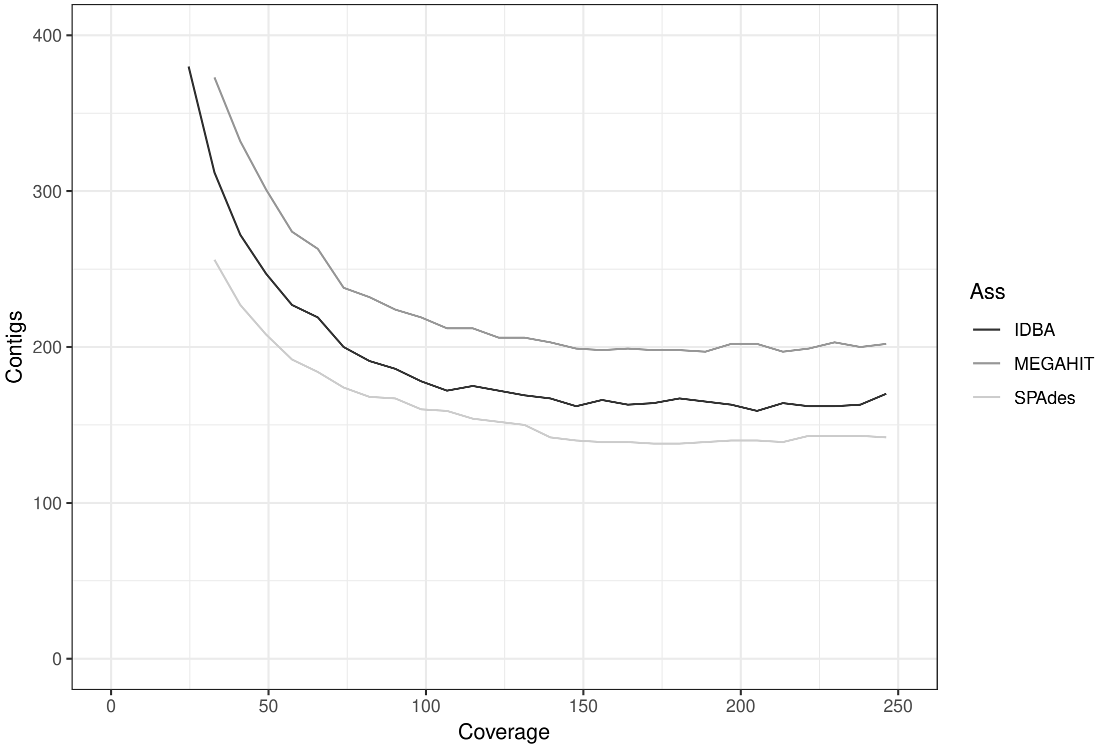

Figure B.5: Contiguity levels off at certain depth of coverage independent of assembly algorithm. Such leveling off of contiguity of these assemblies (*E. coli*, SRA # SRR3989809) is due to limitations surrounding short reads' lengths and biases and not simply the failings of one specific assembly algorithm.

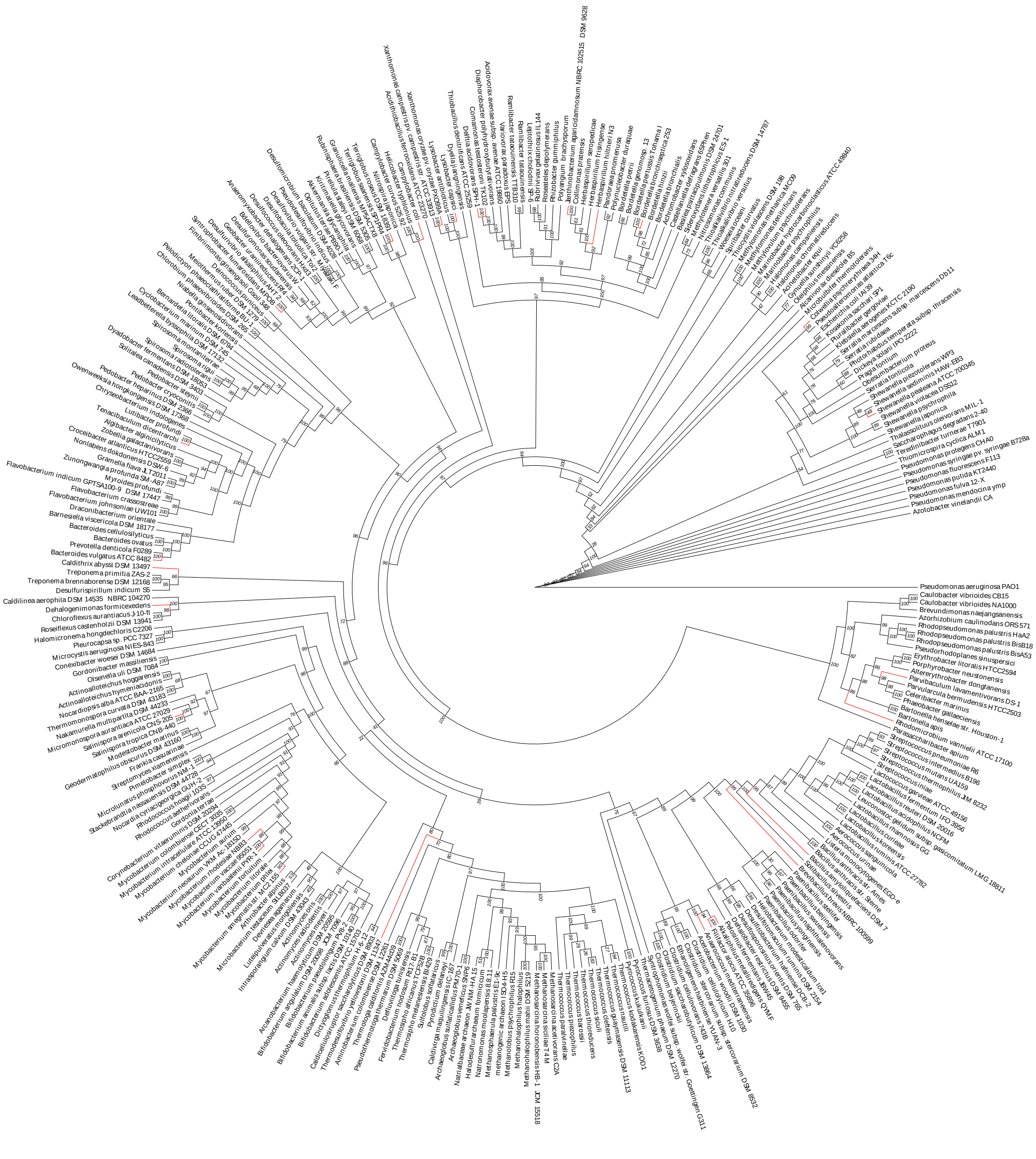
Pseudomonas aeruginosa PAO1
Caulobacter vibrioides CB15
Caulobacter vibrioides NA1000
Brevundimonas naejangsanensis
Azorhizobium caulinodans ORS571
Rhodopseudomonas palustris HaA2
Rhodopseudomonas palustris BisB18
Rhodopseudomonas palustris BisA53
Pseudorhodoplanes sinuspersici
Erythrobacter litoralis HTCC2594
Porphyrobacter neustonensis
Altererythrobacter dongtanensis
Parvibaculum lavamentivorans DS-1
Parvularcula bermudensis HTCC2503
Celeribacter marinus
Phaeobacter gallaeciensis
Bartonella henselae str. Houston-1
Bartonella apis
Rhodomicrobium vannielii ATCC 17100
Parasaccharibacter apium
Streptococcus pneumoniae R6
Streptococcus intermedius B196
Streptococcus mutans UA159
Streptococcus thermophilus JIM 8232
Streptococcus iniae
Lactococcus garvieae ATCC 49156
Lactobacillus fermentum IFO 3956
Lactobacillus reuteri DSM 20016
Lactobacillus acidophilus NCFM
Leuconostoc gelidum subsp. gasicomitatum LMG 18811
Lactobacillus rhamnosus GG
Lactobacillus curieae
Serratia rubidaea
Photorhabdus temperata subsp. thracensis
Dickeya solani IPO 2222
Pragia fontium
Obesumbacterium proteus
Serratia fonticola
Shewanella piezotolerans WP3
Shewanella sediminis HAW-EB3
Shewanella pealeana ATCC 700345
Shewanella violacea DSS12
Shewanella psychrophila
Shewanella japonica
Thalassolituus oleivorans MIL-1
Saccharophagus degradans 2-40
Teredinibacter turnerae T7901
Thiomicrospira cyclica ALM1
Pseudomonas protegens CHA0
Pseudomonas syringae pv. syringae B728a
Pseudomonas fluorescens F113
Pseudomonas putida KT2440
Pseudomonas fulva 12-X
Pseudomonas mendocina ymp
Azotobacter vinelandii CA
Zobellia galactanivorans
Croceibacter atlanticus HTCC2559
Nonlabens dokdonensis DSW-6
Gramella flava JLT2011
Zunongwangia profunda SM-A87
Myroides profundi
Flavobacterium indicum GPTSA100-9 DSM 17447
Flavobacterium crassostreae
Flavobacterium johnsoniae UW101
Draconibacterium orientale
Barnesiella viscericola DSM 18177
Bacteroides cellulosilyticus
Bacteroides ovatus
Prevotella denticola F0289
Bacteroides vulgatus ATCC 8482
Caldithrix abyssi DSM 13497
Treponema primitia ZAS-2
Treponema brennaborense DSM 12168
Desulfurispirillum indicum S5
Caldilinea aerophila DSM 14535 NBRC 104270
Dehalogenimonas formicexedens
Chloroflexus aurantiacus J-10-fl
Roseiflexus castenholzii DSM 13941
Halomicronema hongdechloris C2206
Pleurocapsa sp. PCC 7327
Microcystis aeruginosa NIES-843
Conexibacter woesei DSM 14684
Gordonibacter massiliensis
Olsenella uli DSM 7084
Actinoalloteichus hoggarensis
Actinoalloteichus hymeniacidonis
Nocardiopsis alba ATCC BAA-2165
Thermomonospora curvata DSM 43183
Nakamurella multipartita DSM 44233
Micromonospora aurantiaca ATCC 27029
Salinispora arenicola CNS-205
Salinispora tropica CNB-440

Figure B.6: A maximum-likelihood unrooted tree containing bootstrap values and taxonomic lables of the 16S rRNA sequences of the 311 selected prokaryotic strains. Red branches represent organisms that were unable to be fully resolved with x8 multiplexed PacBio simulated data.

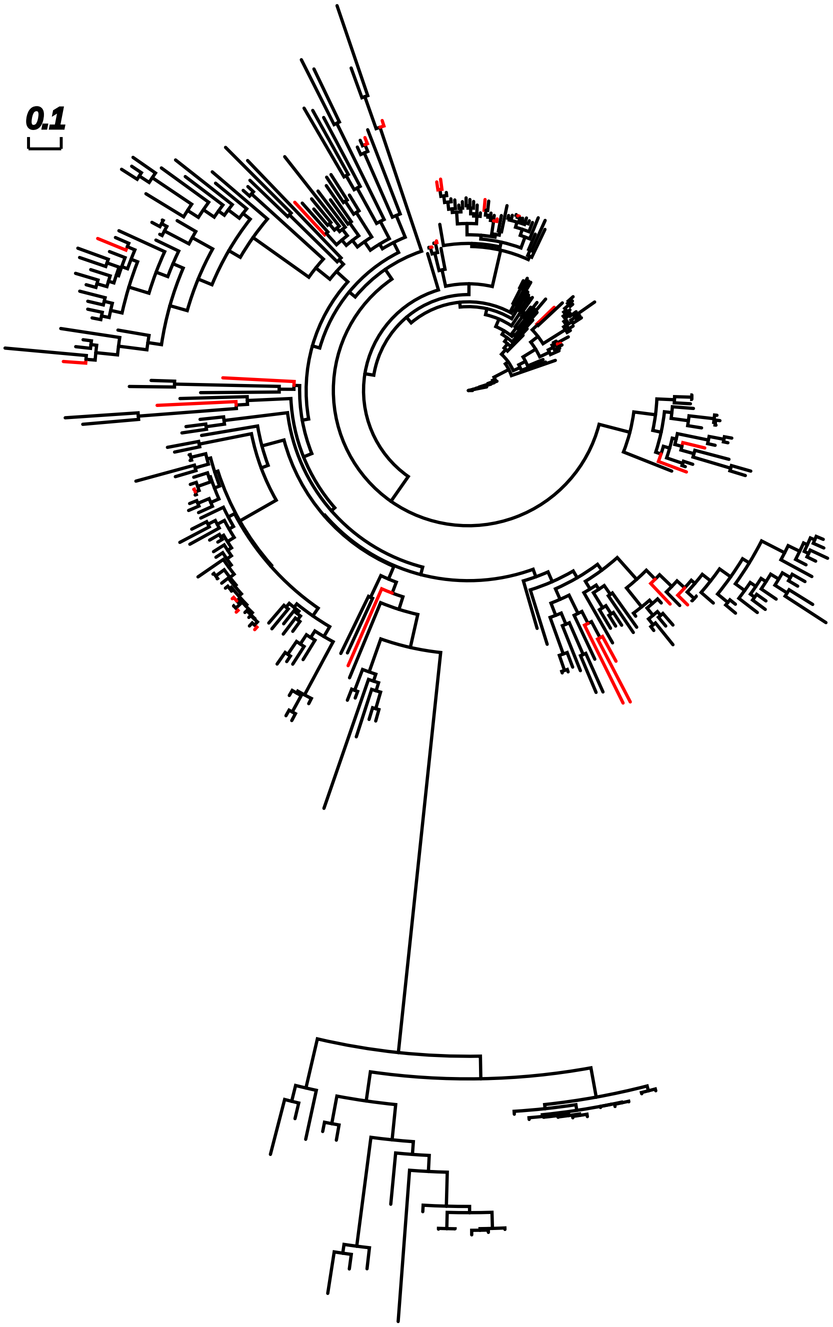

Figure B.7: A maximum-likelihood unrooted tree containing branch lengths drawn to scale of the 16S rRNA sequences of the 311 selected prokaryotic strains. Red branches represent organisms that were unable to be fully resolved with x8 multiplexed PacBio simulated data.

Appendix C

Pipeline Code

```
#!/bin/bash
# This bash pipeline downloads sequencing data from NCBI in fastq format, subsets data across a coverage increment, assembles all of the subset sequencing data, assesses the quality of all the assemblies, and generates a csv file with all of the statistics of the organisms over the selected coverage increment. Read comments in the script to change SRA numbers to analyze, output directory, and coverage increment at which to run this script.
# Dependencies for this script available at the below links:
# SRAtoolkit \url{https://www.ncbi.nlm.nih.gov/sra/docs/toolkitsoft/}
# seqtk \url{https://github.com/lh3/seqtk}
# SPAdes \url{http://cab.spbu.ru/software/spades/}
# QUAST \url{http://bioinf.spbau.ru/quast}
# For this script to run, all dependencies must be in your path
# Exit function and trap to clean up files
function CLEAN_UP {
        echo "We're done!"
        exit
}
trap CLEAN_UP SIGHUP SIGINT EXIT

# Set the variable $LIST to a list of SRA numbers you would
   like to analyze
LIST='SRR3989779 SRR3989780'

# Set output directory you would like all of the work to be
   done in. Default is Coverage_analysis
OUTPUT=Coverage_analysis
mkdir ${OUTPUT}

# Downloads list of SRA numbers
prefetch -v ${LIST}

# For loop for running entire pipeline
for ACCESSION_NUMBER in $LIST; do
 # Converts .sra files to .fastq
 echo "Fastq-dumping, this may take some time..."
 fastq-dump --outdir ./${OUTPUT}/ --split-files ~/ncbi/public/
```

```
    sra/${ACCESSION_NUMBER}.sra
 # Randomly subsets fastq files. If desired, change the
    sequence increment corresponding to which coverage
    increment is desired
 for INCREMENT in $(seq 250000 250000 15000000); do
  seqtk sample -s100 ./${OUTPUT}/${ACCESSION_NUMBER}_1.fastq
     $INCREMENT > ./${OUTPUT}/${ACCESSION_NUMBER}_sub_${
     INCREMENT}_1.fastq
  seqtk sample -s100 ./${OUTPUT}/${ACCESSION_NUMBER}_2.fastq
     $INCREMENT > ./${OUTPUT}/${ACCESSION_NUMBER}_sub_${
     INCREMENT}_2.fastq
  # Running the SPAdes genome assembler
  spades.py -1 ./${OUTPUT}/${ACCESSION_NUMBER}_sub_${INCREMENT}
     _1.fastq -2 ./${OUTPUT}/${ACCESSION_NUMBER}_sub_${
     INCREMENT}_2.fastq -o ./${OUTPUT}/${ACCESSION_NUMBER}
     _sub_${INCREMENT}_spades_assembly
  # Running QUAST to generate statistics regarding the SPAdes
     assemblies
  quast.py -o ./${OUTPUT}/${ACCESSION_NUMBER}_sub_${INCREMENT}
     _quast --no-plots ./${OUTPUT}/${ACCESSION_NUMBER}_sub_${
     INCREMENT}_spades_assembly/contigs.fasta
  # Adding SRA numbers into quast report files
  sed -i "4i  SRA_number  ${x}" ./${OUTPUT}/${ACCESSION_NUMBER}
     _sub_${INCREMENT}_quast/report.txt
}
```

```
  done
done

# Converting quast results to a transposed csv file. This file is now optimized for importation into R, the ggplot package, ect. Thanks to ghostdog74 for help with the awk command! \url{https://stackoverflow.com/questions/1729824/an-efficient-way-to-transpose-a-file-in-bash}  and ValeriyKr for help with the sed command! \url{https://unix.stackexchange.com/questions/335276/grep-v-how-to-exclude-only-the-first-or-last-n-lines-that-match}
paste ./${OUTPUT}/*_quast/report.txt | tail -n +4 |  sed ’s/\(.\) /\1/g’ | awk ’
{
    for (i=1; i<=NF; i++)  {
        a[NR,i] = $i
    }
}
NF>p { p = NF }
END {
    for(j=1; j<=p; j++) {
        str=a[1,j]
        for(i=2; i<=NR; i++){
            str=str" "a[i,j];
        }
        print str
    }
}’ | sed ’2 {h; s/.*/iiii/; x}; /contigs/ {x; s/ˆi//; x; td; b;
    :d; d}’ | tr ’ ’ ’,’ > Total_results.csv

exit
```

Bibliography

[1] Robert D Fleischmann, Mark D Adams, Owen White, Rebecca A Clayton, Ewen F Kirkness, Anthony R Kerlavage, Carol J Bult, Jean-Francois Tomb, Brian A Dougherty, Joseph M Merrick, et al. Whole-genome random sequencing and assembly of *Haemophilus influenzae* Rd. *Science*, 269(5223):496–512, 1995. 1.1, 2.1

[2] Brian P Hedlund, Jeremy A Dodsworth, Senthil K Murugapiran, Christian Rinke, and Tanja Woyke. Impact of single-cell genomics and metagenomics on the emerging view of extremophile "microbial dark matter". *Extremophiles*, 18(5):865–875, 2014. 1.1

[3] Klaus Neuhaus, Richard Landstorfer, Lea Fellner, Svenja Simon, Andrea Schafferhans, Tatyana Goldberg, Harald Marx, Olga N Ozoline, Burkhard Rost, Bernhard Kuster, et al. Translatomics combined with transcriptomics and proteomics reveals novel functional, recently evolved orphan genes in *Escherichia coli* O157: H7 (EHEC). *BMC Genomics*, 17(1):133, 2016. 1.1, 1.3.3

[4] Rebecca A Weingarten, Ryan C Johnson, Sean Conlan, Amanda M Ramsburg, John P Dekker, Anna F Lau, Pavel Khil, Robin T Odom, Clay Deming, Morgan Park, et al. Genomic analysis of hospital plumbing reveals diverse reservoir of bacterial plasmids conferring carbapenem resistance. *MBio*, 9(1):e02011–17, 2018. 1.1, 1.3.1

[5] Leigh J Manley, Duanduan Ma, and Stuart S Levine. Monitoring error rates in Illumina sequencing. *Journal of Biomolecular Techniques: JBT*, 27(4):125, 2016. 1.1

[6] Sara Goodwin, John D McPherson, and W Richard McCombie. Coming of age: ten years of next-generation sequencing technologies. *Nature Reviews Genetics*, 17(6):333, 2016. 1.1

[7] James H Lan, Yuxin Yin, Elaine F Reed, Kevin Moua, Kimberly Thomas, and Qiuheng Zhang. Impact of three Illumina library construction methods on GC bias and HLA genotype calling. *Human Immunology*, 76(2-3):166–175, 2015. 1.1, 2.1

[8] Sergey Koren, Gregory P Harhay, Timothy PL Smith, James L Bono, Dayna M Harhay, Scott D Mcvey, Diana Radune, Nicholas H Bergman, and Adam M Phillippy. Reducing assembly complexity of microbial genomes with single-molecule sequencing. *Genome Biology*, 14(9):R101, 2013. 1.1, 1.4, 1.5, 2.1, 3.1.2, 3.2.2, 3.2.2.5

[9] Timothy J Krause and Joshua R Herr. Genomic variation among highly similar bacterial strains results in a disparity of genome assembly quality metrics, 2018. Submitted to the *Journal of Bioinformatics and Computational Biology* on September 12th, 2018. 1.1, 1.4, 1.5, 3.1.3

[10] Martin O Pollard, Deepti Gurdasani, Alexander J Mentzer, Tarryn Porter, and Manjinder S Sandhu. Long reads: their purpose and place. *Human Molecular Genetics*, 2018. 1.1

[11] Miten Jain, John R Tyson, Matthew Loose, Camilla LC Ip, David A Eccles, Justin O'Grady, Sunir Malla, Richard M Leggett, Ola Wallerman, Hans J Jansen, et al. MinION analysis and reference consortium: phase 2 data release and analysis of R9. 0 chemistry. *F1000Research*, 6, 2017. 1.1

[12] Miten Jain, Sergey Koren, Karen H Miga, Josh Quick, Arthur C Rand, Thomas A Sasani, John R Tyson, Andrew D Beggs, Alexander T Dilthey, Ian T Fiddes, et al. Nanopore sequencing and assembly of a human genome with ultra-long reads. *Nature Biotechnology*, 36(4):338, 2018. 1.1, 1.5

[13] Oxford Nanopore Technologies. Flow cells, 2018. Available at: https://store.nanoporetech.com/flowcells.html. 1.1

[14] Ivo Elliott, Damien Ming, Matthew T Robinson, Pruksa Nawtaisong, Paul N Newton, Mariateresa de Cesare, Rory Bowden, and Elizabeth M Batty. MinION sequencing enables rapid whole genome assembly of *Rickettsia typhi* in a resource-limited setting. *bioRxiv*, page 292102, 2018. 1.1

[15] Chen-Shan Chin, David H Alexander, Patrick Marks, Aaron A Klammer, James Drake, Cheryl Heiner, Alicia Clum, Alex Copeland, John Huddleston, Evan E Eichler, et al. Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data. *Nature Methods*, 10(6):563, 2013. 1.1, 3.1.1, 3.1.2, 3.2.2, 3.2.2.9

[16] Nicholas J Loman, Joshua Quick, and Jared T Simpson. A complete bacterial genome assembled de novo using only nanopore sequencing data. *Nature Methods*, 12(8):733, 2015. 1.1, 3.1.1, 3.2.2

[17] Miriam Land, Loren Hauser, Se-Ran Jun, Intawat Nookaew, Michael R Leuze, Tae-Hyuk Ahn, Tatiana Karpinets, Ole Lund, Guruprased Kora, Trudy Wassenaar, et al. Insights from 20 years of bacterial genome sequencing. *Functional & Integrative Genomics*, 15(2):141–161, 2015. 1.2

[18] Claire M Fraser, Jonathan A Eisen, Karen E Nelson, Ian T Paulsen, and Steven L Salzberg. The value of complete microbial genome sequencing (you get what you pay for). *Journal of Bacteriology*, 184(23):6403–6405, 2002. 1.2

[19] Paul A Kitts, Deanna M Church, Françoise Thibaud-Nissen, Jinna Choi, Vichet Hem, Victor Sapojnikov, Robert G Smith, Tatiana Tatusova, Charlie Xiang, Andrey Zherikov, et al. Assembly: a resource for assembled genomes at NCBI. *Nucleic Acids Research*, 44(D1):D73–D80, 2015. 1.1, 1.2

[20] Carl Kingsford, Michael C Schatz, and Mihai Pop. Assembly complexity of prokaryotic genomes using short reads. *BMC Bioinformatics*, 11(1):21, 2010. 1.2, 1.4, 2.1, 3.1.1, 3.1.2, 3.2.2

[21] ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57, 2012. 1.2

[22] Quan Zhang and Yuzhen Ye. Not all predicted CRISPR–Cas systems are equal: isolated cas genes and classes of CRISPR like elements. *BMC Bioinformatics*, 18(1):92, 2017. 1.2

[23] Pernille Gymoese, Gitte Sørensen, Eva Litrup, John Elmerdal Olsen, Eva Møller Nielsen, and Mia Torpdahl. Investigation of outbreaks of *Salmonella enterica* serovar Typhimurium and its monophasic variants using whole-genome sequencing, Denmark. *Emerging Infectious Diseases*, 23(10):1631, 2017. 1.3, 1.3.1

[24] N Ricker, H Qian, and RR Fulthorpe. The limitations of draft assemblies for understanding prokaryotic adaptation and evolution. *Genomics*, 100(3):167–175, 2012. 1.3.1, 2.1

[25] James B Kaper, James P Nataro, and Harry LT Mobley. Pathogenic *Escherichia coli*. *Nature Reviews Microbiology*, 2(2):123, 2004. 1.3.1

[26] Vinita Periwal and Vinod Scaria. Insights into structural variations and genome rearrangements in prokaryotic genomes. *Bioinformatics*, 31(1):1–9, 2014. 1.3.2, 1.5, 3.1.3

[27] Philip J Hastings, James R Lupski, Susan M Rosenberg, and Grzegorz Ira. Mechanisms of change in gene copy number. *Nature Reviews Genetics*, 10(8):551, 2009. 1.3.2

[28] Bart PHJ Thomma, Michael F Seidl, Xiaoqian Shi-Kunne, David E Cook, Melvin D Bolton, Jan AL van Kan, and Luigi Faino. Mind the gap; seven reasons to close fragmented genome assemblies. *Fungal Genetics and Biology*, 90:24–30, 2016. 1.3.2

[29] Longzhu Cui, Hui-min Neoh, Akira Iwamoto, and Keiichi Hiramatsu. Coordinated phenotype switching with large-scale chromosome flip-flop inversion observed in bacteria. *Proceedings of the National Academy of Sciences*, 109(25):E1647–E1656, 2012. 1.3.2

[30] Daven C Presgraves. Evolutionary genomics: new genes for new jobs. *Current Biology*, 15(2):R52–R53, 2005. 1.3.2

[31] Konstantin Khalturin, Georg Hemmrich, Sebastian Fraune, René Augustin, and Thomas CG Bosch. More than just orphans: are taxonomically-restricted genes important in evolution? *Trends in Genetics*, 25(9):404–413, 2009. 1.3.3

[32] Diethard Tautz and Tomislav Domazet-Lošo. The evolutionary origin of orphan genes. *Nature Reviews Genetics*, 12(10):692, 2011. 1.3.3

[33] Cyrus Chothia. One thousand families for the molecular biologist. *Nature*, 357:543–544, 1992. 1.3.3

[34] Dennis Vitkup, Eugene Melamud, John Moult, and Chris Sander. Completeness in structural genomics. *Nature Structural and Molecular Biology*, 8(6):559, 2001. 1.3.3

[35] Michael Levitt. Nature of the protein universe. *Proceedings of the National Academy of Sciences*, 106(27):11079–11084, 2009. 1.3.3

[36] Chuan-Yun Li, Yong Zhang, Zhanbo Wang, Yan Zhang, Chunmei Cao, Ping-Wu Zhang, Shu-Juan Lu, Xiao-Mo Li, Quan Yu, Xiaofeng Zheng, et al. A human-specific de novo protein-coding gene associated with human brain functions. *PLoS Computational Biology*, 6(3):e1000734, 2010. 1.3.3

[37] Adrian J Verster, Erin B Styles, Abigail Mateo, W Brent Derry, Brenda J Andrews, and Andrew Fraser. Taxonomically restricted genes with essential functions frequently play roles in chromosome segregation in *Caenorhabditis elegans* and *Saccharomyces cerevisiae*. *G3: Genes, Genomes, Genetics*, pages g3–300193, 2017. 1.3.3

[38] Cory Weller and Martin Wu. A generation-time effect on the rate of molecular evolution in bacteria. *Evolution*, 69(3):643–652, 2015. 1.3.4

[39] Samuel K Sheppard and Martin CJ Maiden. The evolution of *Campylobacter jejuni* and *Campylobacter coli*. *Cold Spring Harbor Perspectives in Biology*, page a018119, 2015. 1.3.4

[40] John W Drake. A constant rate of spontaneous mutation in DNA-based microbes. *Proceedings of the National Academy of Sciences*, 88(16):7160–7164, 1991. 1.3.4

[41] Javier Alonso Iserte, Betina Ines Stephan, Sandra Elizabeth Goñi, Cristina Silvia Borio, Pablo Daniel Ghiringhelli, and Mario Enrique Lozano. Family-specific de-

generate primer design: a tool to design consensus degenerated oligonucleotides. *Biotechnology Research International*, 2013, 2013. 1.3.5

[42] Felix Francis, Michael D Dumas, and Randall J Wisser. ThermoAlign: a genome-aware primer design tool for tiled amplicon resequencing. *Scientific Reports*, 7:44437, 2017. 1.3.5

[43] Sergey Koren and Adam M Phillippy. One chromosome, one contig: complete microbial genomes from long-read sequencing and assembly. *Current Opinion in Microbiology*, 23:110–120, 2015. 1.4, 1.4, 2.1, 3.1.2

[44] Hsin-Hung Lin and Yu-Chieh Liao. Evaluation and validation of assembling corrected PacBio long reads for microbial genome completion via hybrid approaches. *PLoS One*, 10(12):e0144305, 2015. 1.4, 3.1.2

[45] Lambert C, Harting J, and Baybayan P. Multiplexing strategies for microbial whole genome sequencing using the Sequel System. In *11th Annual DOE Joint Genome Institute Genomics of Energy & Environment Meeting*. DOE Joint Genome Institute, 2016. 1.4, 3.1.4, 3.2.2

[46] C Heiner, Kim M, Ferrao H, Wallace VJ, Eng K, Fedak R, Wong J, Kilburn D, Ashby M, Baybayan P, Burke JM, Bjornson K, and Liu KJ. Single chromosomal genome assemblies on the Sequel System with circulomics high molecular weight DNA extraction for microbes. In *DOE Joint Genome Institute UGM*. DOE Joint Genome Institute, 2018. 1.4, 1.5, 2.3, 3.1.4, 3.2.2

[47] Pacific Biosystems. Microbial multiplexing workflow on the Sequel System, 2018. Available at: https://www.pacb.com/wp-content/uploads/Application-Note-Microbial-Multiplexing-Workflow-on-the-Sequel-System.pdf. 1.4, 3.2.2

[48] Ryan R Wick, Louise M Judd, Claire L Gorrie, and Kathryn E Holt. Completing bacterial genome assemblies with multiplex MinION sequencing. *Microbial Genomics*, 3(10), 2017. 1.4, 1.5, 2.3, 3.1.2, 3.1.4, 3.2.2

[49] Niranjan Nagarajan and Mihai Pop. Sequence assembly demystified. *Nature Reviews Genetics*, 14(3):157, 2013. 1.4

[50] Robert Ekblom and Jochen BW Wolf. A field guide to whole-genome sequencing, assembly and annotation. *Evolutionary Applications*, 7(9):1026–1042, 2014. 1.5, 3.1

[51] Guy Bresler, Ma'ayan Bresler, and David Tse. Optimal assembly for high throughput shotgun sequencing. *BMC Bioinformatics*, 14(5):S18, 2013. 1.5, 2.1, 2.2.5, 3.1.2, 3.1.3, 3.3

[52] David Sims, Ian Sudbery, Nicholas E Ilott, Andreas Heger, and Chris P Ponting. Sequencing depth and coverage: key considerations in genomic analyses. *Nature Reviews Genetics*, 15(2):121, 2014. 1.5, 3.1.3

[53] Guillaume Marçais and Carl Kingsford. A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics*, 27(6):764–770, 2011. 1.5, 2.1, 2.2.3

[54] Crusoe MR, Alameldin HF, Awad S, Boucher E, Caldwell A, and Cartwright R et al. The khmer software package: enabling efficient nucleotide sequence analysis. *F1000Research*, 4, 2015. 1.5

[55] Tanja Magoc, Stephan Pabinger, Stefan Canzar, Xinyue Liu, Qi Su, Daniela Puiu, Luke J Tallon, and Steven L Salzberg. GAGE-B: an evaluation of genome assemblers for bacterial organisms. *Bioinformatics*, 29(14):1718–1725, 2013. 1.5, 2.1, 2.3, 3.1, 3.1.3

[56] Francesca Giordano, Louise Aigrain, Michael A Quail, Paul Coupland, James K Bonfield, Robert M Davies, German Tischler, David K Jackson, Thomas M Keane, Jing Li, et al. De novo yeast genome assemblies from MinION, PacBio and MiSeq platforms. *Scientific Reports*, 7(1):3935, 2017. 1.5, 3.1

[57] Ruichao Li, Miaomiao Xie, Ning Dong, Dachuan Lin, Xuemei Yang, Marcus Ho Yin Wong, Edward Wai-Chi Chan, and Sheng Chen. Efficient generation of complete sequences of MDR-encoding plasmids by rapid assembly of MinION barcoding sequencing data. *GigaScience*, 7(3):gix132, 2018. 1.5

[58] W Florian Fricke and David A Rasko. Bacterial genome sequencing in the clinic: bioinformatic challenges and solutions. *Nature Reviews Genetics*, 15(1):49, 2014. 1.5

[59] Steven L Salzberg, Adam M Phillippy, Aleksey Zimin, Daniela Puiu, Tanja Magoc, Sergey Koren, Todd J Treangen, Michael C Schatz, Arthur L Delcher, Michael Roberts, et al. GAGE: A critical evaluation of genome assemblies and assembly algorithms. *Genome Research*, 22(3):557–567, 2012. 2.1, 2.3, 3.1

[60] Sebastian Jünemann, Karola Prior, Andreas Albersmeier, Stefan Albaum, Jörn Kalinowski, Alexander Goesmann, Jens Stoye, and Dag Harmsen. GABenchToB: a genome assembly benchmark tuned on bacteria and benchtop sequencers. *PLoS One*, 9(9):e107014, 2014. 2.1, 2.3

[61] Derrick Scott and Bert Ely. Comparison of genome sequencing technology and assembly methods for the analysis of a GC-rich bacterial genome. *Current microbiology*, 70(3):338–344, 2015. 2.1, 3.1

[62] Aleksey V Zimin, Guillaume Marçais, Daniela Puiu, Michael Roberts, Steven L Salzberg, and James A Yorke. The MaSuRCA genome assembler. *Bioinformatics*, 29(21):2669–2677, 2013. 2.1

[63] Shaun D Jackman, Benjamin P Vandervalk, Hamid Mohamadi, Justin Chu, Sarah Yeo, S Austin Hammond, Golnaz Jahesh, Hamza Khan, Lauren Coombe, Rene L Warren, et al. ABySS 2.0: resource-efficient assembly of large genomes using a Bloom filter. *Genome Research*, pages gr–214346, 2017. 2.1

[64] Yonatan H Grad, Marc Lipsitch, Michael Feldgarden, Harindra M Arachchi, Gustavo C Cerqueira, Michael FitzGerald, Paul Godfrey, Brian J Haas, Cheryl I Murphy, Carsten Russ, et al. Genomic epidemiology of the *Escherichia coli* O104: H4 outbreaks in Europe, 2011. *Proceedings of the National Academy of Sciences*, 109(8):3065–3070, 2012. 2.1

[65] Mahdi Heydari, Giles Miclotte, Piet Demeester, Yves Van de Peer, and Jan Fostier. Evaluation of the impact of Illumina error correction tools on de novo genome assembly. *BMC Bioinformatics*, 18(1):374, 2017. 2.1

[66] Yen-Chun Chen, Tsunglin Liu, Chun-Hui Yu, Tzen-Yuh Chiang, and Chi-Chuan Hwang. Effects of GC bias in next-generation-sequencing data on de novo genome assembly. *PLoS One*, 8(4):e62856, 2013. 2.1, 2.2.1

[67] Erwin L Van Dijk, Yan Jaszczyszyn, and Claude Thermes. Library preparation methods for next-generation sequencing: tone down the bias. *Experimental Cell Research*, 322(1):12–20, 2014. 2.1

[68] Leho Tedersoo, Ave Tooming-Klunderud, and Sten Anslan. PacBio metabarcoding of fungi and other eukaryotes: errors, biases and perspectives. *New Phytologist*, 217(3):1370–1385, 2018. 2.1, 3.1.3

[69] Marie-Ka Tilak, Fidel Botero-Castro, Nicolas Galtier, and Benoit Nabholz. Illumina library preparation for sequencing the GC-rich fraction of heterogeneous genomic DNA. *Genome Biology and Evolution*, 10(2):616–622, 2018. 2.1

[70] Michael G Ross, Carsten Russ, Maura Costello, Andrew Hollinger, Niall J Lennon, Ryan Hegarty, Chad Nusbaum, and David B Jaffe. Characterizing and measuring bias in sequence data. *Genome Biology*, 14(5):R51, 2013. 2.1, 3.1.3, 3.2.2

[71] Todd J Treangen and Steven L Salzberg. Repetitive DNA and next-generation sequencing: computational challenges and solutions. *Nature Reviews Genetics*, 13(1):36, 2012. 2.1, 3.2.2.5

[72] Paul Medvedev, Konstantinos Georgiou, Gene Myers, and Michael Brudno. Computability of models for sequence assembly. In *International Workshop on Algorithms in Bioinformatics*, pages 289–301. Springer, 2007. 2.1

[73] Esko Ukkonen. Approximate string-matching with q-grams and maximal matches. *Theoretical Computer Science*, 92(1):191–211, 1992. 2.1, 2.2.5, 3.1.2

[74] Christoph Bleidorn. Third generation sequencing: technology and its potential impact on evolutionary biodiversity research. *Systematics and Biodiversity*, 14(1):1–8, 2016. 2.1

[75] Changsheng Li, Feng Lin, Dong An, Wenqin Wang, and Ruidong Huang. Genome sequencing and assembly by long reads in plants. *Genes*, 9(1):6, 2017. 2.1

[76] Binghang Liu, Yujian Shi, Jianying Yuan, Xuesong Hu, Hao Zhang, Nan Li, Zhenyu Li, Yanxiang Chen, Desheng Mu, and Wei Fan. Estimation of genomic characteristics by analyzing k-mer frequency in de novo genome projects. *arXiv*, 1308.2012, 2013. 2.1, 2.2.3

[77] Gregory W Vurture, Fritz J Sedlazeck, Maria Nattestad, Charles J Underwood, Han Fang, James Gurtowski, and Michael C Schatz. GenomeScope: fast reference-free

genome profiling from short reads. *Bioinformatics*, 33(14):2202–2204, 2017. 2.1, 2.2.3, 3.3

[78] Marc W. Allard, Errol Strain, David Melka, Kelly Bunning, Steven M. Musser, Eric W. Brown, and Ruth Timme. Practical value of food pathogen traceability through building a whole-genome sequencing network and database. *Journal of Clinical Microbiology*, 54(8):1975–1983, 2016. 2.2.1

[79] Simon Andrews. FastQC: a quality control tool for high throughput sequence data, 2010. Available at: `http://www.bioinformatics.babraham.ac.uk/projects/fastqc`. 2.2.1

[80] Francesco Vezzi, Giuseppe Narzisi, and Bud Mishra. Feature-by-feature–evaluating de novo sequence assembly. *PLoS One*, 7(2):e31002, 2012. 2.2.2, 3.2.2

[81] James F Denton, Jose Lugo-Martinez, Abraham E Tucker, Daniel R Schrider, Wesley C Warren, and Matthew W Hahn. Extensive error in the number of genes inferred from draft genome assemblies. *PLoS Computational Biology*, 10(12):e1003998, 2014. 2.2.2

[82] Matthew W Hahn, Simo V Zhang, and Leonie C Moyle. Sequencing, assembling, and correcting draft genomes using recombinant populations. *G3: Genes, Genomes, Genetics*, pages g3–114, 2014. 2.2.2

[83] Jonathan L Klassen and Cameron R Currie. Gene fragmentation in bacterial draft genomes: extent, consequences and mitigation. *BMC Genomics*, 13(1):14, 2012. 2.2.2, 2.2.4

[84] Guillaume Marçais, Arthur L Delcher, Adam M Phillippy, Rachel Coston, Steven L Salzberg, and Aleksey Zimin. MUMmer4: A fast and versatile genome alignment system. *PLoS Computational Biology*, 14(1):e1005944, 2018. 2.2.2, 3.2.2.5

[85] Anton Bankevich, Sergey Nurk, Dmitry Antipov, Alexey A Gurevich, Mikhail Dvorkin, Alexander S Kulikov, Valery M Lesin, Sergey I Nikolenko, Son Pham, Andrey D Prjibelski, et al. SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. *Journal of Computational Biology*, 19(5):455–477, 2012. 2.2.3, 3.1.1, 3.2.1.2

[86] Alexey Gurevich, Vladislav Saveliev, Nikolay Vyahhi, and Glenn Tesler. QUAST: quality assessment tool for genome assemblies. *Bioinformatics*, 29(8):1072–1075, 2013. 2.2.3, 3.2.1.2, 3.2.2.1

[87] Stefano Lonardi, Hamid Mirebrahim, Steve Wanamaker, Matthew Alpert, Gianfranco Ciardo, Denisa Duma, and Timothy J Close. When less is more: 'slicing' sequencing data improves read decoding accuracy and de novo assembly quality. *Bioinformatics*, 31(18):2972–2980, 2015. 2.2.3, 3.2.1.2, 3.2.2.4

[88] Yu Peng, Henry CM Leung, Siu-Ming Yiu, and Francis YL Chin. IDBA-UD: a de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth. *Bioinformatics*, 28(11):1420–1428, 2012. 2.2.3, 3.2.1.2

[89] Torsten Seemann. Prokka: rapid prokaryotic genome annotation. *Bioinformatics*, 30(14):2068–2069, 2014. 2.2.3, 2.2.4

[90] Konstantinos T Konstantinidis and James M Tiedje. Trends between gene content and genome size in prokaryotic species with larger genomes. *Proceedings of the National Academy of Sciences*, 101(9):3160–3165, 2004. 2.2.4

[91] Yubo Hou and Senjie Lin. Distinct gene number-genome size relationships for eukaryotes and non-eukaryotes: gene content estimation for dinoflagellate genomes. *PLoS One*, 4(9):e6978, 2009. 2.2.4, 2.2.6

[92] Lin Xu, Hong Chen, Xiaohua Hu, Rongmei Zhang, Ze Zhang, and ZW Luo. Average gene length is highly conserved in prokaryotes and eukaryotes and diverges only between the two kingdoms. *Molecular Biology and Evolution*, 23(6):1107–1108, 2006. 2.2.4

[93] Igor B Rogozin, Kira S Makarova, Darren A Natale, Alexey N Spiridonov, Roman L Tatusov, Yuri I Wolf, Jodie Yin, and Eugene V Koonin. Congruent evolution of different classes of non-coding DNA in prokaryotic genomes. *Nucleic Acids Research*, 30(19):4264–4271, 2002. 2.2.4

[94] Sebastian E Ahnert, Thomas MA Fink, and Andrei Zinovyev. How much non-coding DNA do eukaryotes require? *Journal of Theoretical Biology*, 252(4):587–592, 2008. 2.2.4

[95] JR Herr, M Öpik, and DS Hibbett. Towards the unification of sequence–based classification and sequence–based identification of host–associated microorganisms. *New Phytologist*, 205(1):27–31, 2015. 2.2.4

[96] Aarti Desai, Veer Singh Marwah, Akshay Yadav, Vineet Jha, Kishor Dhaygude, Ujwala Bangar, Vivek Kulkarni, and Abhay Jere. Identification of optimum sequencing depth especially for de novo genome assembly of small genomes using next generation sequencing data. *PLoS One*, 8(4):e60204, 2013. 2.2.4, 3.1.3, 3.3

[97] Todd J Treangen, Anne-Laure Abraham, Marie Touchon, and Eduardo PC Rocha. Genesis, effects and fates of repeats in prokaryotic genomes. *FEMS Microbiology Reviews*, 33(3):539–571, 2009. 2.2.5, 3.1, 3.2.2.5, 3.2.2.9

[98] Guillaume Achaz, Eduardo PC Rocha, P Netter, and Eric Coissac. Origin and fate of repeats in bacteria. *Nucleic Acids Research*, 30(13):2987–2994, 2002. 2.2.5

[99] Dent A Earl, Keith Bradnam, John St John, Aaron Darling, Dawei Lin, Joseph Faas, Hung On Ken Yu, Buffalo Vince, Daniel R Zerbino, Mark Diekhans, et al. Assemblathon 1: a competitive assessment of de novo short read assembly methods. *Genome Research*, pages gr–126599, 2011. 2.2.5

[100] Zheng Chang, Zhenjia Wang, and Guojun Li. The impacts of read length and transcriptome complexity for de novo assembly: A simulation study. *PLoS One*, 9(4):e94825, 2014. 2.3

[101] Stephen Richards. Full disclosure: Genome assembly is still hard. *PLoS Biology*, 16(4):e2005894, 2018. 2.3

[102] Filipe Ribeiro, Dariusz Przybylski, Shuangye Yin, Ted Sharpe, Sante Gnerre, Amr Abouelleil, Aaron M Berlin, Anna Montmayeur, Terrance P Shea, Bruce J Walker, et al. Finished bacterial genomes from shotgun sequence data. *Genome Research*, pages gr–141515, 2012. 3.1

[103] Ivica Letunic and Peer Bork. Interactive tree of life (iTOL) v3: an online tool for the display and annotation of phylogenetic and other trees. *Nucleic Acids Research*, 44(W1):W242–W245, 2016. 3.1, 3.2.2.9

[104] Euler L. Solutio problematis ad geometriam situs pertinentis. *Commentarii Academiae Scientiarum Petropolitanae*, 8:128–140, 1741. 3.1.1

[105] Zhenyu Li, Yanxiang Chen, Desheng Mu, Jianying Yuan, Yujian Shi, Hao Zhang, Jun Gan, Nan Li, Xuesong Hu, Binghang Liu, et al. Comparison of the two ma-

jor classes of assembly algorithms: overlap–layout–consensus and de-bruijn-graph. *Briefings in Functional Genomics*, 11(1):25–37, 2012. 3.1.1

[106] Eugene W Myers Jr. A history of DNA sequence assembly. *It-Information Technology*, 58(3):126–132, 2016. 3.1.1, 3.3

[107] Yao-Ting Huang and Chen-Fu Liao. Integration of string and de Bruijn graphs for genome assembly. *Bioinformatics*, 32(9):1301–1307, 2016. 3.1.1

[108] Claire M Fraser, Jeannine D Gocayne, Owen White, Mark D Adams, Rebecca A Clayton, Robert D Fleischmann, Carol J Bult, Anthony R Kerlavage, Granger Sutton, Jenny M Kelley, et al. The minimal gene complement of *Mycoplasma genitalium*. *Science*, 270(5235):397–404, 1995. 3.1.1

[109] Granger G Sutton, Owen White, Mark D Adams, and Anthony R Kerlavage. TIGR assembler: A new tool for assembling large shotgun sequencing projects. *Genome Science and Technology*, 1(1):9–19, 1995. 3.1.1

[110] Frederick Sanger, Steven Nicklen, and Alan R Coulson. DNA sequencing with chain-terminating inhibitors. *Proceedings of the national academy of sciences*, 74(12):5463–5467, 1977. 3.1.1

[111] Kari-Jouko Räihä and Esko Ukkonen. The shortest common supersequence problem over binary alphabet is NP-complete. *Theoretical Computer Science*, 16(2):187–198, 1981. 3.1.1

[112] René L Warren, Granger G Sutton, Steven JM Jones, and Robert A Holt. Assembling millions of short DNA sequences using SSAKE. *Bioinformatics*, 23(4):500–501, 2006. 3.1.1

[113] Jason R Miller, Sergey Koren, and Granger Sutton. Assembly algorithms for next-generation sequencing data. *Genomics*, 95(6):315–327, 2010. 3.1.1

[114] Hannu Peltola, Hans Söderlund, and Esko Ukkonen. SEQAID: A DNA sequence assembling program based on a mathematical model. *Nucleic Acids Research*, 12(1 Pt 1):307–321, 1984. 3.1.1

[115] Eugene W Myers, Granger G Sutton, Art L Delcher, Ian M Dew, Dan P Fasulo, Michael J Flanigan, Saul A Kravitz, Clark M Mobarry, Knut HJ Reinert, Karin A Remington, et al. A whole-genome assembly of *Drosophila*. *Science*, 287(5461):2196–2204, 2000. 3.1.1

[116] Sergey Koren, Brian P Walenz, Konstantin Berlin, Jason R Miller, Nicholas H Bergman, and Adam M Phillippy. Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. *Genome Research*, pages gr–215087, 2017. 3.1.1, 3.1.2, 3.2.2.1, 3.2.2.4, 3.3, 3.2.2.6, 3.3

[117] David R Bentley, Shankar Balasubramanian, Harold P Swerdlow, Geoffrey P Smith, John Milton, Clive G Brown, Kevin P Hall, Dirk J Evers, Colin L Barnes, Helen R Bignell, et al. Accurate whole human genome sequencing using reversible terminator chemistry. *Nature*, 456(7218):53, 2008. 3.1.1

[118] Jared T Simpson and Mihai Pop. The theory and practice of genome sequence assembly. *Annual Review of Genomics and Human Genetics*, 16:153–172, 2015. 3.1.1

[119] Daniel Zerbino and Ewan Birney. Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Research*, pages gr–074492, 2008. 3.1.1

[120] Ramana M Idury and Michael S Waterman. A new algorithm for DNA sequence assembly. *Journal of Computational Biology*, 2(2):291–306, 1995. 3.1.1

[121] Pavel A Pevzner. 1-tuple DNA sequencing: computer analysis. *Journal of Biomolecular Structure and Dynamics*, 7(1):63–73, 1989. 3.1.1

[122] Pavel A Pevzner, Haixu Tang, and Michael S Waterman. An Eulerian path approach to DNA fragment assembly. *Proceedings of the National Academy of Sciences*, 98(17):9748–9753, 2001. 3.1.1

[123] John Eid, Adrian Fehr, Jeremy Gray, Khai Luong, John Lyle, Geoff Otto, Paul Peluso, David Rank, Primo Baybayan, Brad Bettman, et al. Real-time DNA sequencing from single polymerase molecules. *Science*, 2008. 3.1.1

[124] Alexander S Mikheyev and Mandy MY Tin. A first look at the Oxford Nanopore MinION sequencer. *Molecular Ecology Resources*, 14(6):1097–1102, 2014. 3.1.1

[125] Dmitry Antipov, Anton Korobeynikov, Jeffrey S McLean, and Pavel A Pevzner. HybridSPAdes: an algorithm for hybrid assembly of short and long reads. *Bioinformatics*, 32(7):1009–1015, 2015. 3.1.1

[126] Yu Lin, Jeffrey Yuan, Mikhail Kolmogorov, Max W Shen, Mark Chaisson, and Pavel A Pevzner. Assembly of long error-prone reads using de Bruijn graphs. *Proceedings of the National Academy of Sciences*, 113(52):E8396–E8405, 2016. 3.1.1

[127] Konstantin Berlin, Sergey Koren, Chen-Shan Chin, James P Drake, Jane M Landolin, and Adam M Phillippy. Assembling large genomes with single-molecule sequencing and locality-sensitive hashing. *Nature Biotechnology*, 33(6):623, 2015. 3.1.1

[128] Adam C English, Stephen Richards, Yi Han, Min Wang, Vanesa Vee, Jiaxin Qu, Xiang Qin, Donna M Muzny, Jeffrey G Reid, Kim C Worley, et al. Mind the gap:

upgrading genomes with Pacific Biosciences RS long-read sequencing technology. *PLoS One*, 7(11):e47768, 2012. 3.1.1

[129] Ali Bashir, Aaron A Klammer, William P Robins, Chen-Shan Chin, Dale Webster, Ellen Paxinos, David Hsu, Meredith Ashby, Susana Wang, Paul Peluso, et al. A hybrid approach for the automated finishing of bacterial genomes. *Nature Biotechnology*, 30(7):701, 2012. 3.1.1

[130] Marten Boetzer and Walter Pirovano. SSPACE-LongRead: scaffolding bacterial draft genomes using long read sequence information. *BMC Bioinformatics*, 15(1):211, 2014. 3.1.1

[131] Mahul Chakraborty, James G Baldwin-Brown, Anthony D Long, and JJ Emerson. Contiguous and accurate de novo assembly of metazoan genomes with modest long read coverage. *Nucleic Acids Research*, 44(19):e147–e147, 2016. 3.1.1

[132] Sergey Koren, Michael C Schatz, Brian P Walenz, Jeffrey Martin, Jason T Howard, Ganeshkumar Ganapathy, Zhong Wang, David A Rasko, W Richard McCombie, Erich D Jarvis, et al. Hybrid error correction and de novo assembly of single-molecule sequencing reads. *Nature Biotechnology*, 30(7):693, 2012. 3.1.1

[133] Heng Li. Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences. *Bioinformatics*, 32(14):2103–2110, 2016. 3.1.1, 3.3

[134] Ole K Tørresen, Bastiaan Star, Sissel Jentoft, William B Reinar, Harald Grove, Jason R Miller, Brian P Walenz, James Knight, Jenny M Ekholm, Paul Peluso, et al. An improved genome assembly uncovers prolific tandem repeats in Atlantic cod. *BMC Genomics*, 18(1):95, 2017. 3.1.1

[135] Sara Goodwin, James Gurtowski, Scott Ethe-Sayers, Panchajanya Deshpande, Michael C Schatz, and W Richard McCombie. Oxford Nanopore sequencing, hybrid error correction, and de novo assembly of a eukaryotic genome. *Genome Research*, 2015. 3.1.1

[136] Leena Salmela and Eric Rivals. LoRDEC: accurate and efficient long read error correction. *Bioinformatics*, 30(24):3506–3514, 2014. 3.1.1

[137] Leena Salmela, Riku Walve, Eric Rivals, and Esko Ukkonen. Accurate self-correction of errors in long reads using de Bruijn graphs. *Bioinformatics*, 33(6):799–806, 2016. 3.1.1

[138] Hayan Lee, James Gurtowski, Shinjae Yoo, Maria Nattestad, Shoshana Marcus, Sara Goodwin, W Richard McCombie, and Michael Schatz. Third-generation sequencing and the future of genomics. *bioRxiv*, page 048603, 2016. 3.1.1

[139] Justin Chu, Hamid Mohamadi, René L Warren, Chen Yang, and Inanc Birol. Innovations and challenges in detecting long read overlaps: an evaluation of the state-of-the-art. *Bioinformatics*, 33(8):1261–1270, 2016. 3.1.1

[140] Nicholas J Loman, Raju V Misra, Timothy J Dallman, Chrystala Constantinidou, Saheer E Gharbia, John Wain, and Mark J Pallen. Performance comparison of benchtop high-throughput sequencing platforms. *Nature Biotechnology*, 30(5):434, 2012. 3.1.3

[141] Eric S Lander and Michael S Waterman. Genomic mapping by fingerprinting random clones: a mathematical analysis. *Genomics*, 2(3):231–239, 1988. 3.1.3

[142] International Human Genome Sequencing Consortium. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860, 2001. 3.1.3

[143] Raga Krishnakumar, Anupama Sinha, Sara W Bird, Harikrishnan Jayamohan, Harrison S Edwards, Joseph S Schoeniger, Kamlesh D Patel, Steven S Branda, and Michael S Bartsch. Systematic and stochastic influences on the performance of the MinION nanopore sequencer across a range of nucleotide bias. *Scientific Reports*, 8(1):3159, 2018. 3.1.3

[144] Phil Green. 2× genomes–does depth matter? *Genome Research*, 17(11):1547–1549, 2007. 3.1.3, 3.3

[145] Ivan Sović, Krešimir Križanović, Karolj Skala, and Mile Šikić. Evaluation of hybrid and non-hybrid methods for de novo assembly of nanopore reads. *Bioinformatics*, 32(17):2582–2589, 2016. 3.1.3

[146] Dinghua Li, Chi-Man Liu, Ruibang Luo, Kunihiko Sadakane, and Tak-Wah Lam. MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. *Bioinformatics*, 31(10):1674–1676, 2015. 3.2.1.2

[147] Bianca K Stöcker, Johannes Köster, and Sven Rahmann. SimLoRD: simulation of long read data. *Bioinformatics*, 32(17):2704–2706, 2016. 3.2.2.1, 3.2.2.5, 3.2.2.9

[148] William S Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74(368):829–836, 1979. 3.2.2.4

[149] Robert Vaser, Ivan Sović, Niranjan Nagarajan, and Mile Šikić. Fast and accurate de novo genome assembly from long uncorrected reads. *Genome Research*, 2017. 3.2.2.4

[150] E Lerat. Identifying repeats and transposable elements in sequenced genomes: how to find your way through the dense forest of programs. *Heredity*, 104(6):520, 2010. 3.2.2.5

[151] Falk Hildebrand, Axel Meyer, and Adam Eyre-Walker. Evidence of selection upon genomic gc-content in bacteria. *PLoS genetics*, 6(9):e1001107, 2010. 3.2.2.7

[152] Jon Bohlin, Vegard Eldholm, John HO Pettersson, Ola Brynildsrud, and Lars Snipen. The nucleotide composition of microbial genomes indicates differential patterns of selection on core and accessory genomes. *BMC genomics*, 18(1):151, 2017. 3.2.2.7

[153] Karin Lagesen, Peter Hallin, Einar Andreas Rødland, Hans-Henrik Stærfeldt, Torbjørn Rognes, and David W Ussery. RNAmmer: consistent and rapid annotation of ribosomal RNA genes. *Nucleic Acids Research*, 35(9):3100–3108, 2007. 3.2.2.8

[154] Lam-Tung Nguyen, Heiko A Schmidt, Arndt von Haeseler, and Bui Quang Minh. IQ-TREE: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. *Molecular Biology and Evolution*, 32(1):268–274, 2014. 3.2.2.8

[155] Subha Kalyaanamoorthy, Bui Quang Minh, Thomas KF Wong, Arndt von Haeseler, and Lars S Jermiin. ModelFinder: fast model selection for accurate phylogenetic estimates. *Nature Methods*, 14(6):587, 2017. 3.2.2.8

[156] Diep Thi Hoang, Olga Chernomor, Arndt von Haeseler, Bui Quang Minh, and Le Sy Vinh. UFBoot2: improving the ultrafast bootstrap approximation. *Molecular Biology and Evolution*, 35(2):518–522, 2017. 3.2.2.8

[157] Zilong He, Huangkai Zhang, Shenghan Gao, Martin J Lercher, Wei-Hua Chen, and Songnian Hu. Evolview v2: an online visualization and management tool for cus-

tomized and annotated phylogenetic trees. *Nucleic Acids Research*, 44(W1):W236–W241, 2016. 3.2.2.8

[158] Tamara Münkemüller, Sébastien Lavergne, Bruno Bzeznik, Stéphane Dray, Thibaut Jombart, Katja Schiffers, and Wilfried Thuiller. How to measure and test phylogenetic signal. *Methods in Ecology and Evolution*, 3(4):743–756, 2012. 3.2.2.9

[159] Matthew W Pennell and Luke J Harmon. An integrative view of phylogenetic comparative methods: connections to population genetics, community ecology, and paleobiology. *Annals of the New York Academy of Sciences*, 1289(1):90–105, 2013. 3.2.2.9

[160] Joe Parker, Andrew Rambaut, and Oliver G Pybus. Correlating viral phenotypes with phylogeny: accounting for phylogenetic uncertainty. *Infection, Genetics and Evolution*, 8(3):239–246, 2008. 3.2.2.9

[161] Liam J Revell, Luke J Harmon, and David C Collar. Phylogenetic signal, evolutionary process, and rate. *Systematic Biology*, 57(4):591–601, 2008. 3.2.2.9

[162] Fritz J Sedlazeck, Hayan Lee, Charlotte A Darby, and Michael C Schatz. Piercing the dark matter: bioinformatics of long-range sequencing and mapping. *Nature Reviews Genetics*, page 1, 2018. 3.3

[163] Benedikt Rauscher, Florian Heigwer, Marco Breinig, Jan Winter, and Michael Boutros. GenomeCRISPR-a database for high-throughput CRISPR/Cas9 screens. *Nucleic Acids Research*, page gkw997, 2016. 3.3

[164] Akshay Kumar Avvaru, Saketh Saxena, Divya Tej Sowpati, and Rakesh Kumar Mishra. MSDB: A comprehensive database of simple sequence repeats. *Genome Biology and Evolution*, 9(6):1797–1802, 2017. 3.3

[165] Paul Muir, Shantao Li, Shaoke Lou, Daifeng Wang, Daniel J Spakowicz, Leonidas Salichos, Jing Zhang, George M Weinstock, Farren Isaacs, Joel Rozowsky, et al. The real cost of sequencing: scaling computation to keep pace with data generation. *Genome Biology*, 17(1):53, 2016. 3.3

[166] Justin Jee, Aviram Rasouly, Ilya Shamovsky, Yonatan Akivis, Susan R Steinman, Bud Mishra, and Evgeny Nudler. Rates and mechanisms of bacterial mutagenesis from maximum-depth sequencing. *Nature*, 534(7609):693, 2016. 3.3

[167] Michael Schmid, Daniel Frei, Andrea Patrignani, Ralph Schlapbach, Juerg E Frey, Mitja NP Remus-Emsermann, and Christian H Ahrens. Pushing the limits of de novo genome assembly for complex prokaryotic genomes harboring very long, near identical repeats. *bioRxiv*, page 300186, 2018. 3.3

[168] Ivan Erill. Information theory and biological sequences: Insights from an evolutionary perspective. In *Information Theory: New Research*, pages 1–28. Nova Science Publishers, Inc, 2012. 3.3

[169] Sally R Partridge, Stephen M Kwong, Neville Firth, and Slade O Jensen. Mobile genetic elements associated with antimicrobial resistance. *Clinical Microbiology Reviews*, 31(4):e00088–17, 2018. 3.3

[170] Aaron E Darling, István Miklós, and Mark A Ragan. Dynamics of genome rearrangement in bacterial populations. *PLoS Genetics*, 4(7):e1000128, 2008. 3.3