# Comparing bacterial genome assembly via Unicycler and Flye

Emily Maier

BINF 6110

# INTRODUCTION

The present work compares the efficacy of two genome assembly software, Unicycler and Flye. It is interesting to compare these software because of their opposing approaches to genome assembly, specifically in handling repeated elements. Unicycler uses long reads to optimize a path through a De Bruijn graph created from short reads (Wick et al., 2017). To resolve repeated elements, paired-end short reads and long reads are used to create bridge contigs, resulting in a single path (Wick et al., 2017). Conversely, Flye first arbitrarily joins long reads into disjointigs used to construct assembly and then repeat graphs (Kolmogorov et al., 2019). Repeat graphs are analyzed topologically to resolve repeats (Kolmogorov et al., 2019). Of note, Unicycler was specifically designed to assemble bacterial genomes (Wick et al., 2017), where Flye can assemble a broader range of genomes (Kolmogorov et al., 2019).

To evaluate the software, the genome of a strain of E. coli will be assembled. E. coli have a single, double-stranded, circular chromosome which is known to contain bacterial interspersed mosaic elements (BIME) (Verma et al., 2019). BIME are repetitive sequences that are palindromic in nature and can be found in between genes (Verma et al., 2019). The genome can also include plasmids which vary in presence and copy number (Johnson & Nolan, 2009).

Despite the deliberate handling of repeated elements, it is expected that if the software do not produce similar assemblies it will be due to an inability to assemble repeated elements accurately. Specifically, the chromosome may not be assembled into a single contig due to difficulty resolving the BIME. Since Unicycler was made to assemble bacterial genomes, it is expected to perform better because its approach to resolving repeats should be tailored to repeated elements that are characteristic of bacteria.

To compare the efficacy of each software, each assembly will first be run through BLAST. Although a perfect match is not expected due to the dynamic nature of bacterial genomes (Johnson & Nolan, 2009), this step should confirm that both assemblies reflect an E. coli genome. Additionally, number of contigs, total length of assembly, N50, largest fragment and N50 vs. largest fragment will be compared to draw conclusions regarding software efficacy.

## MATERIALS AND METHODS

Short and long read sequence data of bacteria cultured in Dr. Ricker's lab were produced by Illumina and Oxford Nanopore sequencing respectively. Prior to genome assembly, FastQC 0.11.9 verified the quality of runs through statistic and quality scores. Trimmomatic 0.39 trimmed artifacts of Illumina sequencing. Following genome assembly with Unicycler 0.4.4 or Flye 2.6, each assembly was run through blastn 2.12.0 against the “complete.1.1.genomic.fna” Refseq database to ensure that both assemblies matched to E. coli.

## RESULTS AND DISCUSSION

Both software generated assemblies that matched closest with E. coli assemblies documented in Refseq. As seen in Table 1, Flye yielded many fewer contigs over a longer assembly than Unicycler, which can be interpreted as a more complete assembly with fewer or no fragmented replicons. Since Unicycler had more contigs and a shorter total assembly length, it is probable that the software yielded an assembly with gaps that could not be bridged to form complete replicons. Although Unicycler had a higher N50, it was only 63% of the assembly’s largest fragment compared to Flye’s N50 which was 89 % of its largest fragment. These proportions corroborate the notion that the Unicycler assembly produced many small contigs.

The main difference between Unicycler and Flye that may have impacted their performance is their approach to resolving assembly graphs. The Unicycler algorithm relies on accurately determining the multiplicity of each contig in the De Bruijn graph to generate the best path (Wick et al., 2017). Determining multiplicity is hampered by varying plasmid copy number in each cell (Wick et al., 2017), as is very possible in E. coli. If multiplicity was determined incorrectly then the following steps would result in an inaccurate assembly. As this is not a component of the Flye algorithm, it could explain the suspected superior performance.

It is not possible to definitively conclude which assembler performed better without further analysis to gain a better understanding of the features of each the assembly. For example, PlasmidFinder 1.3 could be used to identify complete and incomplete plasmids among the contigs in each assembly. Additionally, CRISPRfinder could be used to identify repeated elements in each assembly and compare them to each other and to other E. coli assemblies. Doing so would provide more information to understand which components of the genome were assembled differently and could validate the previously stated expectations.

Overall, both software ultimately yielded assemblies that were consistent with the E. coli genome and have pros and cons that should be considered prior to use. Strengths of the Unicycler algorithm include automatic optimization of default settings so that the final assembly is less impacted by user ability, as well as the generation and scoring of 10 k-mer graphs to use the one with fewest contigs and dead ends (Wick et al., 2017). However, the reliance on sufficiently long read depth and accurately determining multiplicity of contigs may be weaknesses of the algorithm (Wick et al., 2017). Flye seems to have a stronger ability to accurately resolve repeated regions but may have more mismatch and indel errors due to a lack of short read data (Kolmogorov et al., 2019).

**Table 1** Summary of quality measures from Unicycler and Flye assembly of E. coli genome.

| | Unicycler | Flye |
|---|---|---|
| Number of Contigs | 47 | 20 |
| Total Length | 5147264 | 5190619 |
| N50 | 1297127 | 1044833 |
| Largest fragment | 2058636 | 1174344 |
| N50/largest fragment (%) | 63 | 89 |

REFERENCES

Johnson, T. J., & Nolan, L. K. (2009). Pathogenomics of the Virulence Plasmids of Escherichia coli. Microbiology and Molecular Biology Reviews, 73(4), 750–774. https://doi.org/10.1128/MMBR.00015-09

Kolmogorov, M., Yuan, J., Lin, Y., & Pevzner, P. A. (2019). Assembly of long, error-prone reads using repeat graphs. Nature Biotechnology, 37(5), 540–546. https://doi.org/10.1038/s41587-019-0072-8

Refseq. (2022). complete.1.1.genomic.fna. Retrieved from NCBI FTP site. https://ftp.ncbi.nlm.nih.gov/refseq/release/complete/complete.1.1.genomic.fna

Ricker, N., Qian, H., & Fulthorpe, R. . (2012). The limitations of draft assemblies for understanding prokaryotic adaptation and evolution. Genomics (San Diego, Calif.), 100(3), 167–175. https://doi.org/10.1016/j.ygeno.2012.06.009

Verma, S. C., Qian, Z., & Adhya, S. L. (2019). Architecture of the Escherichia coli nucleoid. PLoS Genetics, 15(12), e1008456–e1008456. https://doi.org/10.1371/journal.pgen.1008456

Wick, R. R., Judd, L. M., Gorrie, C. L., & Holt, K. E. (2017). Unicycler: Resolving bacterial genome assemblies from short and long sequencing reads. PLoS Computational Biology, 13(6), e1005595–e1005595. https://doi.org/10.1371/journal.pcbi.1005595