BINF 6110 Assignment 3:

Introduction:

The assembly of genome is an essential process in bioinformatics analysis, which reconstructs an organism's genome. Accurately assembled genome can prove to be a vital step in downstream analysis, such as gene annotation, functional characterization, and learning the genomic and proteomic structure to gain biological insights. In computational biology emergence of many assemblers have occurred, for the purposes of this study, three commonly used assemblers were compared, such as Flye, SPAdes, Unicycler. This was done in effort to analyze which assembler can provide an accurate reconstruction genome and evaluate their performance relative to other assemblers used during this project.

The first assembler used was Flye, designed for long-read sequencing technologies such as Oxford Nanopore Technologies, and PacBio (1). Flye can generate arbitrary paths in unknown repeat graphs and construct accurate repeat graphs from error riddled disjoints, indicating that it generates better or comparable assemblies (1). The second assembler used was SPAdes, which is utilized in short-read sequencing and uses a multi-step repetitive approach for assembly refinement (2). The third assembler used was Unicycler which is utilized for both long and short-read sequencing, acting as a hybrid assembler, aiming for contiguity and accuracy (3).

For the comparative analysis Quast and Prokka were used to determine the performance of assemblers being compared. Quast is a prevalent tool that provides essential assembly metrics such as contig length, N50, and mis assembly rates (4). Prokka was used for gene annotation, as Prokka is widely used for genome annotation and provides functional predictions (5). These tools were used to examine how different and accurate results three assemblers can produce.

Methods:

Assembly Pipeline:

The 136x2 fastq raw data was assembled using three assemblers: Unicycler, Flye, and SPAdes. The 136x2_R1&R2 fastq (short read) paired ends raw data was assembled through SPAdes and 136x2 fastq (long read) data was assembled through Flye and Unicycler.

The assemblies were performed on cedar computing cluster using the default setting for each tool. All of the output contigs were retained for subsequent analysis.

Assembly Assessment with QUAST:

The three assemblies utilized were assessed using QUAST, which provided with various range of assembly metrics such as, largest contig, GC content, N50, L50, and total assembly length. These metrics are key indictors of completeness of genome structure and contiguity. Each of the assembly output files were input into QUAST and the results obtained were compared and analysed to evaluate the differences in the genome assemblies being compared.

Gene Annotation with Prokka:

Prokka was utilized to carry out the functional gene annotation. Each of the output generated from the assembled genome was ran on Prokka to identify features such as rRNA, tRNA, and CDS, and were assigned to COG categories. The resulting annotation output files were TSV and GFF files, which were used to compare the number of genes identified by each assembler.

Visualization:

The QUAST and Prokka resulted were visualized through R Studio using packages such as, tidyr, dplyr, and ggplot2. The QUAST report table and Nx statistics plot were generated through python. The genome assembly comparison and gene annotation plot were generated in R studio.

Analysis Log:

# Loaded necessary environment and software modules

```
Module load StdEnv/2020

Module load gcc/9.3.0

Module load spades

Module load quast/5.0.2

Module load prokka

Module load abricate
```

# Performed genome assembly using SPAdes with paired-end FASTQ files

```
spades.py -1 136x2_R1.fastq -2 136x2_R2.fastq -o spades_output --threads 16 --memory 100
```

# Checked that the contigs file was created successfully

```
ls -lh spades_output/contigs.fasta
```

# Counted the number of contigs in the assembly

```
grep ">" spades_output/contigs.fasta | wc -l
```

# Printed the lengths of the top 10 longest contigs

```
awk '{if($0 ~ ">") {if (seqlen) print seqlen; seqlen=0} else {seqlen += length($0)}} END {print seqlen}' spades_output/contigs.fasta | sort -nr | head -10
```

# Ran QUAST to evaluate the assembly quality

```
quast.py -o quast_results \
  -m 500 \

  --gene-finding \

  --threads 8 \
```

# Annotate the SPAdes assembly using Prokka

```
  prokka --outdir prokka_spades --prefix spades_annotation spades_output/scaffolds.fasta
```

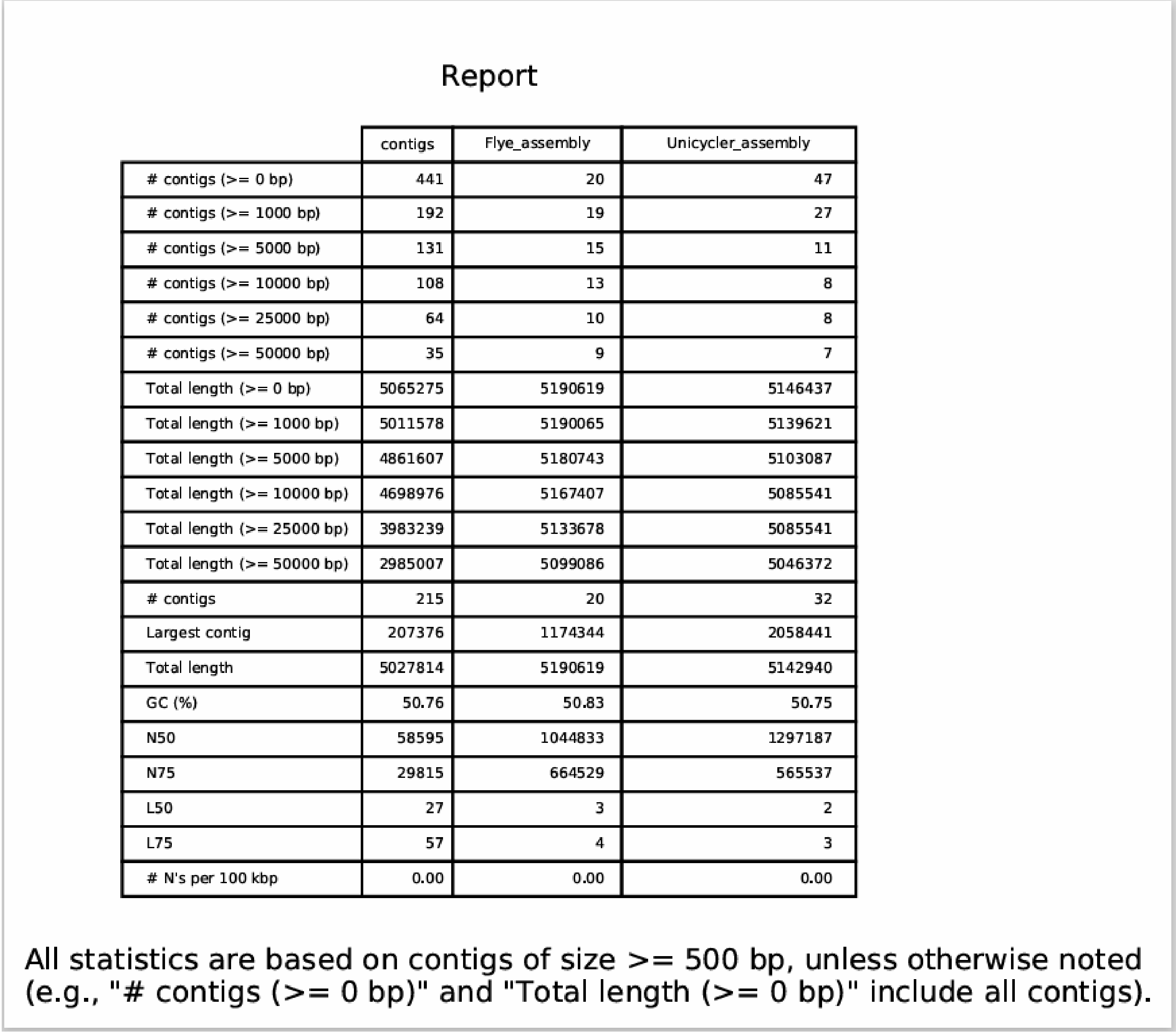
# Annotate Flye assembly using Prokka

```
prokka --outdir prokka_flye --prefix flye_annotation Flye_assembly.fasta
```

# Annotate Unicycler assembly using Prokka

```
prokka --outdir prokka_unicycler --prefix unicycler_annotation Unicycler_assembly.fasta
```

Results:



Report

	contigs	Flye_assembly	Unicycler_assembly
# contigs (>= 0 bp)	441	20	47
# contigs (>= 1000 bp)	192	19	27
# contigs (>= 5000 bp)	131	15	11
# contigs (>= 10000 bp)	108	13	8
# contigs (>= 25000 bp)	64	10	8
# contigs (>= 50000 bp)	35	9	7
Total length (>= 0 bp)	5065275	5190619	5146437
Total length (>= 1000 bp)	5011578	5190065	5139621
Total length (>= 5000 bp)	4861607	5180743	5103087
Total length (>= 10000 bp)	4698976	5167407	5085541
Total length (>= 25000 bp)	3983239	5133678	5085541
Total length (>= 50000 bp)	2985007	5099086	5046372
# contigs	215	20	32
Largest contig	207376	1174344	2058441
Total length	5027814	5190619	5142940
GC (%)	50.76	50.83	50.75
N50	58595	1044833	1297187
N75	29815	664529	565537
L50	27	3	2
L75	57	4	3
# N's per 100 kbp	0.00	0.00	0.00

All statistics are based on contigs of size >= 500 bp, unless otherwise noted (e.g., "# contigs (>= 0 bp)" and "Total length (>= 0 bp)" include all contigs).

Figure 1: QUAST Summary Table of Assembly Metrics

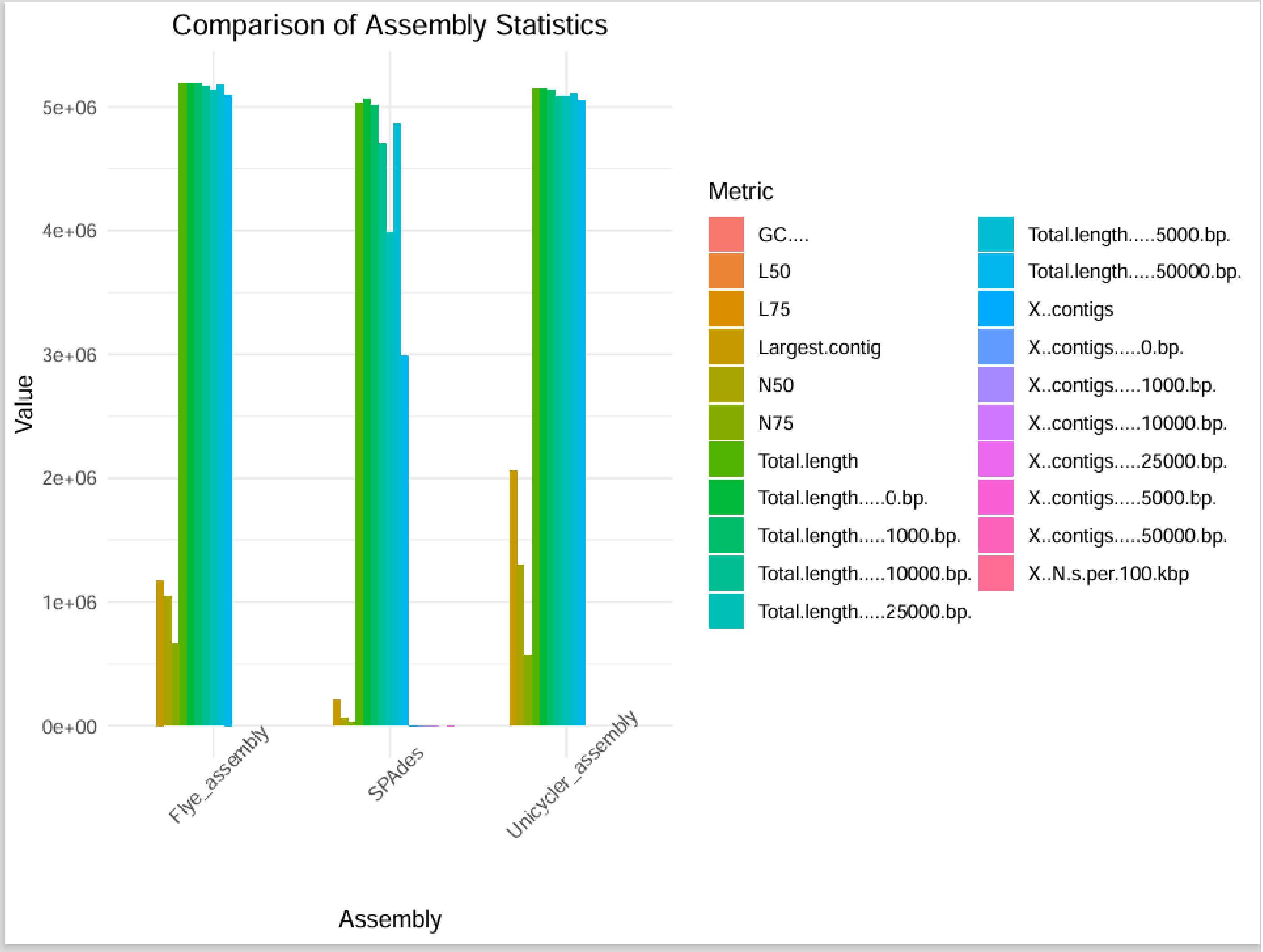


Figure 2: Bar Plot of Assembly Statistics across Three Genome Assemblers (SPAdes, Flye, and Unicycler)

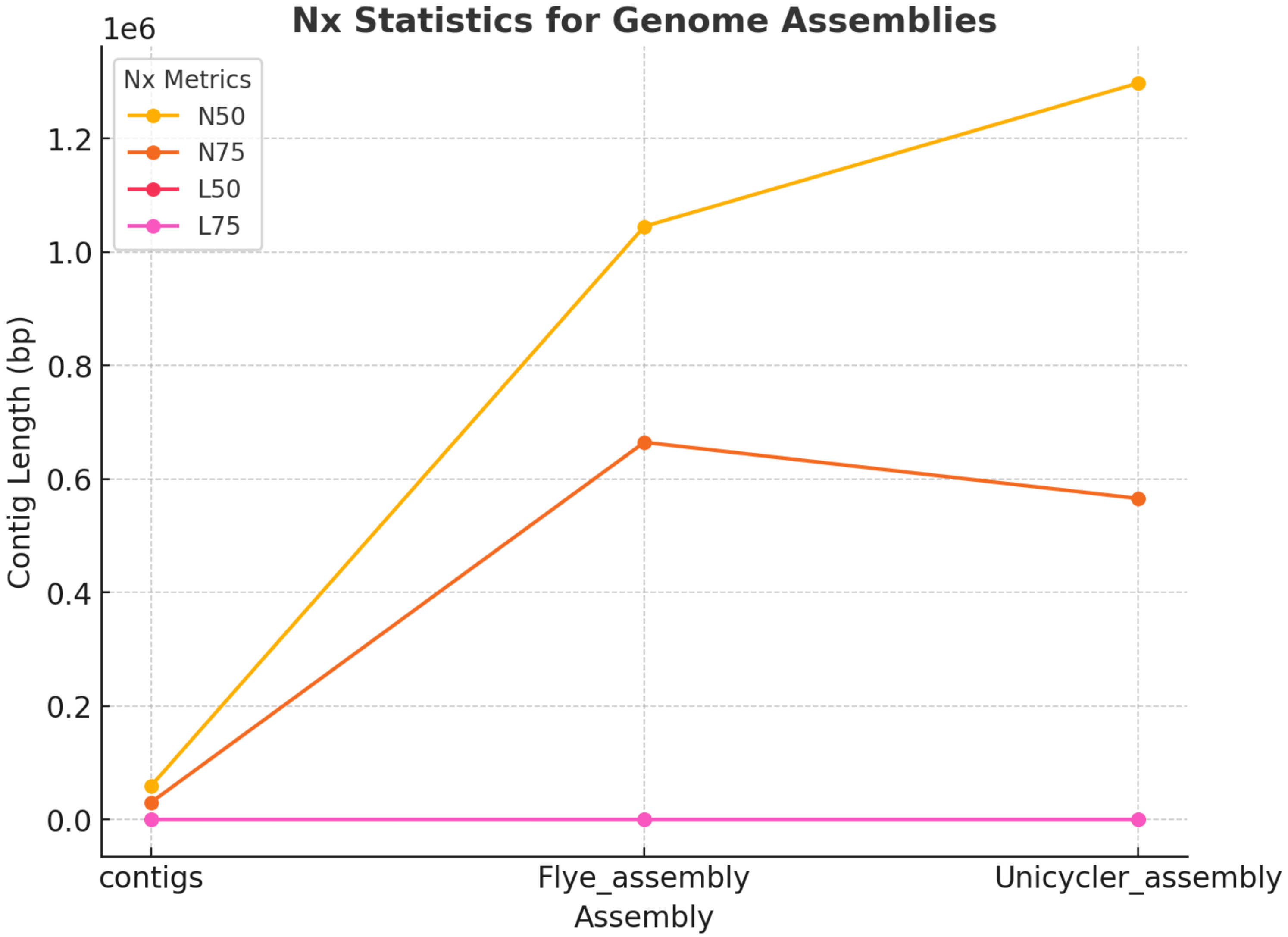


Figure 3: Nx Statistics Plot for SPAdes (contigs), Flye, and Unicycler Genome Assemblies

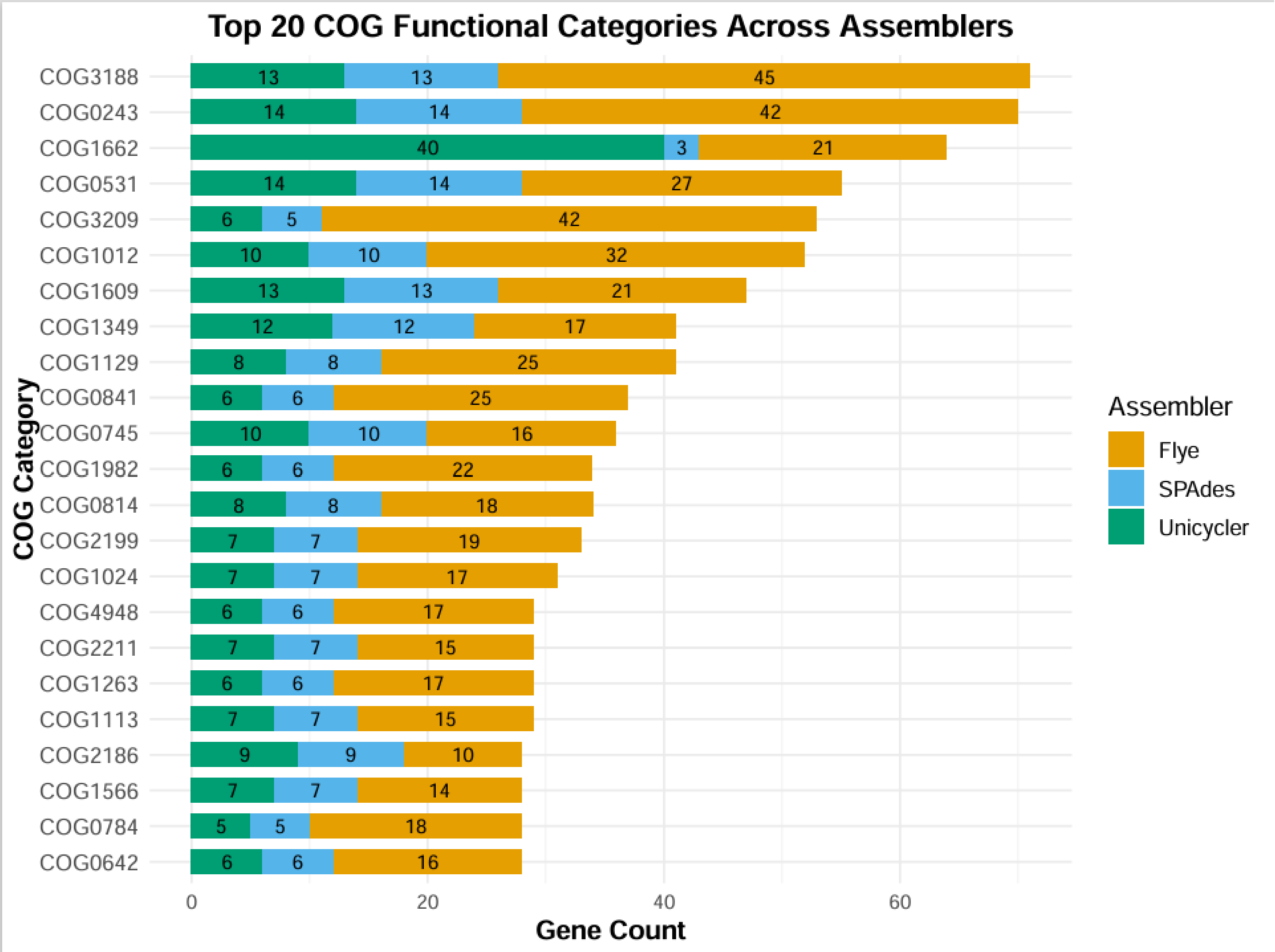


Figure 4: The Top 20 COG Functional Categories Across Three Genome Assemblers

Discussion:

The Quast analysis performed, provided a clear insight into the difference in assembly quality amongst the three-assemblers tested. The key parameters that were noted, included N50, total length, and number of contigs. As Flye is optimally used for long-read sequencing, produced fewer but longer contigs, which was evident by the large N50 and contig values. The Quast analysis also suggests that Flye is capable of generating more contiguous assemblies, which can be vital for downstream genome analysis. SPAdes, produced higher number of contigs, however had lowest N50 value. This aligns with the expectation of SPAdes assemblies, as short read assemblers struggle with complex genome structures and repeat regions, which results in fragmented assemblies. The increased number of contigs can be indicative of misassemblies. Unicycler demonstrated a balanced performance between the two assemblers, as it maintained a reasonable N50 value and a slightly shorter assembly length in comparison to Flye. Unicycler and Flye demonstrated most complete and contiguous assemblies with higher N50 values, while SPAdes produced more fragmented assembly with highest number of contigs and lowest N50 value, which suggests that its less optimal for genome reconstruction.

The Prokka analysis conducted indicates that the genes assigned to Clusters of Orthologous Groups (COG), across the three assemblers, Flye and Unicycler identified more complete genes compared to SPAdes. Unicycler demonstrated to have the highest number of genes mapped to COG, which is indicative of a comprehensive gene annotation. SPAdes, in contrast depicted the lowest count of identified genes, potentially indicating an incomplete gene prediction.

During this study it was found that Flye was the most optimal in generating long and contiguous assemblies and is useful for structural genome studies. Unicycler can provide a balance between complete and contiguity assembly, which can be used as a hybrid for short and long sequencing reads. While SPAdes can be useful for short read sequences, in this project it proved to be relatively the weakest of the assemblers used. The choices of these analysis were made in effort to provide a comprehensive understanding of assemblers, so scientists can determine which assembler can be utilized for different types of data.

Abricate analysis were attempted, however, the tool did not seem to be compatible and did not download in the environment server used, therefore AMR profiling could not be performed.

References:

1. 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
2. Bankevich, A., Nurk, S., Antipov, D., Gurevich, A. A., Dvorkin, M., Kulikov, A. S., Lesin, V. M., Nikolenko, S. I., Pham, S., Prjibelski, A. D., Pyshkin, A. V., Sirotkin, A. V., Vyahhi, N., Tesler, G., Alekseyev, M. A., & Pevzner, P. A. (2012). SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. *Journal of computational biology : a journal of computational molecular cell biology*, *19*(5), 455–477. https://doi.org/10.1089/cmb.2012.0021
3. 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. https://doi.org/10.1371/journal.pcbi.1005595

4. Gurevich, A., Saveliev, V., Vyahhi, N., & Tesler, G. (2013). QUAST: quality assessment tool for genome assemblies. *Bioinformatics (Oxford, England)*, *29*(8), 1072–1075. https://doi.org/10.1093/bioinformatics/btt086

5. Seemann T. (2014). Prokka: rapid prokaryotic genome annotation. *Bioinformatics (Oxford, England)*, *30*(14), 2068–2069. https://doi.org/10.1093/bioinformatics/btu153