1) From the ABySS output, create a table for the unitigs, contigs, and scaffolds with the number of each, N50 for each, and predicted genome length.

ABySS output	Number of sequences (n)	N50 (bp)	Predicted genome length (sum)
assembly-unitigs.fa	2,203	11,193	4,012,179
assembly-contigs.fa	1,734	14,566	4,060,444
assembly-scaffolds.fa	1,553	17,809	4,063,103

2) In your own words, please summarize the function of each of the commands (e.g., abyss-pe, k, B, etc) that you included in your code.

- Abyss-pe assembles a genome from short reads. It follows the stages: building unitigs → contigs → scaffolds.
- name=assembly sets the prefix for output files (e.g., assembly-unitigs.fa, assembly-contigs.fa, and assembly-scaffolds.fa).
- k=96 sets the k-mer size to 96, which defines how long each k-mer is when building the assembly graph.
- B=2G allocates 2 gigabytes of memory for the bloom filter, which efficiently tracks k-mers during construction and saves memory.
- in='Julian_1.fastq.gz Julian_2.fastq.gz' specifies the input paired-end read files used for assembly.

3) Based on this manual, can you identify how you could modify the code you used to do a hybrid assembly with nanopore reads? Please explain what a hybrid assembly is and why someone might want to do that.

- A hybrid assembly includes combining short reads and long reads in order to assemble a genome.
- One would perform a hybrid assembly since short reads are accurate but too short to resolve repeats/complex regions and long reads are more error-prone. Combining them gives the accuracy of short reads and leads to more complete/correct genome assemblies due to the long-range information of long reads.

- To perform a hybrid assembly with nanopore reads, modify the code as follows:

  spades.py \

  -1 Julian_1.fastq.gz \

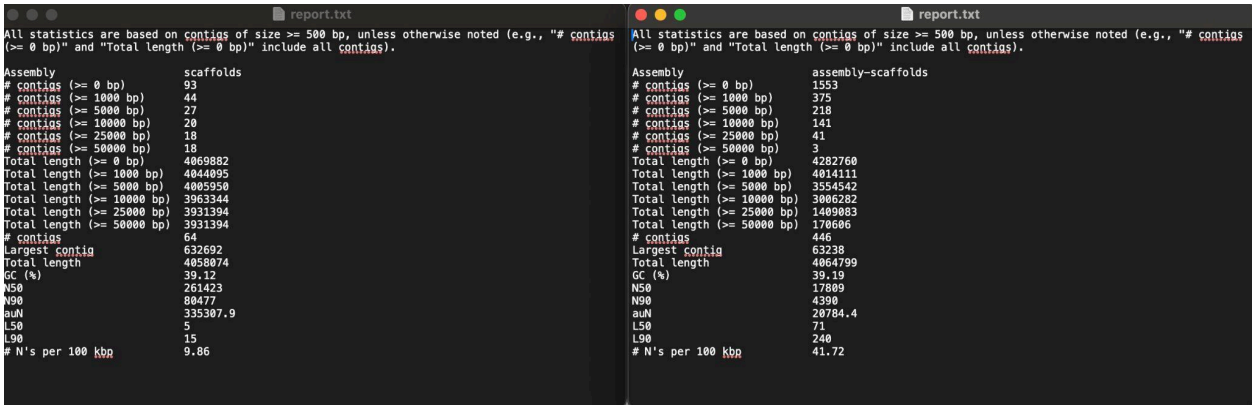
  -2 Julian_2.fastq.gz \

  --nanopore nanopore_reads.fastq \

  -o spades_hybrid_out

4) Include a screenshot of the QUAST assembly statistics for the ABySS and SPAdes assembly. This is a demo from the sample files we worked on.

- SPAdes on left; ABySS on right



report.txt

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).

Assembly	scaffolds
# contigs (>= 0 bp)	93
# contigs (>= 1000 bp)	44
# contigs (>= 5000 bp)	27
# contigs (>= 10000 bp)	20
# contigs (>= 25000 bp)	18
# contigs (>= 50000 bp)	18
Total length (>= 0 bp)	4069882
Total length (>= 1000 bp)	4044095
Total length (>= 5000 bp)	4005950
Total length (>= 10000 bp)	3963344
Total length (>= 25000 bp)	3931394
Total length (>= 50000 bp)	3931394
# contigs	64
Largest contig	632692
Total length	4058074
GC (%)	39.12
N50	261423
N90	80477
auN	335307.9
L50	5
L90	15
# N's per 100 kbp	9.86

report.txt

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).

Assembly	assembly-scaffolds
# contigs (>= 0 bp)	1553
# contigs (>= 1000 bp)	375
# contigs (>= 5000 bp)	218
# contigs (>= 10000 bp)	141
# contigs (>= 25000 bp)	41
# contigs (>= 50000 bp)	3
Total length (>= 0 bp)	4282760
Total length (>= 1000 bp)	4014111
Total length (>= 5000 bp)	3554542
Total length (>= 10000 bp)	3006282
Total length (>= 25000 bp)	1409083
Total length (>= 50000 bp)	170606
# contigs	446
Largest contig	63238
Total length	4064799
GC (%)	39.19
N50	17809
N90	4390
auN	20784.4
L50	71
L90	240
# N's per 100 kbp	41.72

5) Based on the statistics from your genome, which assembly do you think is best? Why? This is the assembly you can use going forward.

- SPAdes is better overall. There are fewer contigs, which means a more contiguous genome. There is a much larger N50 and lower L50, indicating longer and fewer contigs to cover 50% of the genome. The N90 is larger, showing more robust small contigs. The L90 is much lower, meaning SPAdes reaches 90% of the genome with way fewer contigs. There is a larger max contig size. There are fewer ambiguous bases (N's). Overall better quality assembly when compared to ABySS.

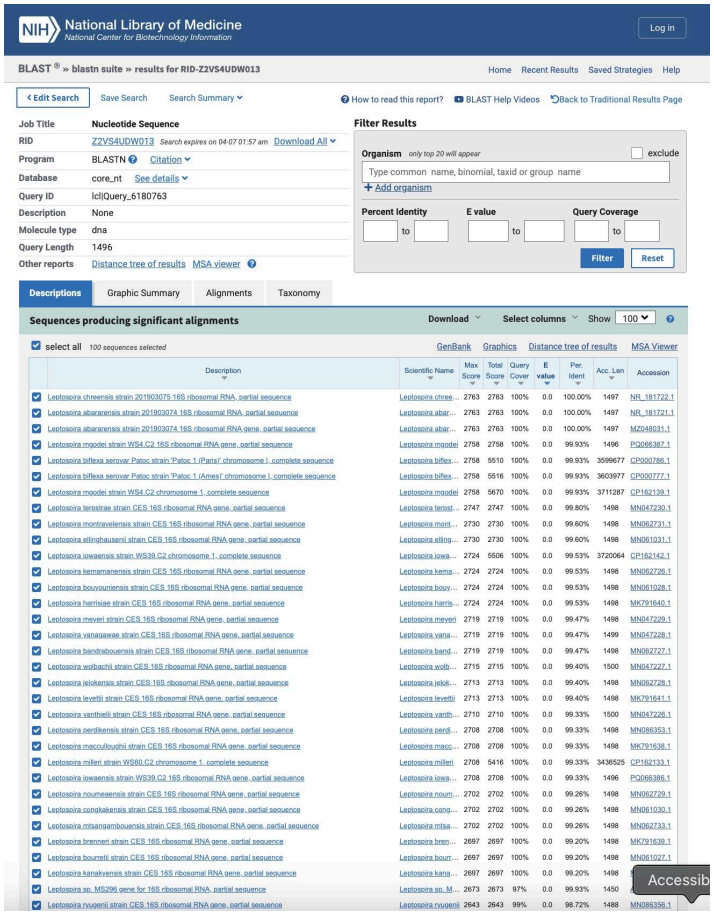
6) How can we use barrnap to figure out what species we have? Why is using the 16S rRNA sequence a good, but imperfect, tool for identifying species identity?

- Barrnap is a tool that identifies ribosomal RNA genes in genome assemblies. It helps you locate the rRNA genes in your data and then using BLAST helps you

identify the species by comparing to known references. Using the 16S rRNA sequence is good, but imperfect, tool for identifying species identity because it is great for a general ID, however less accurate for distinguishing closely related species. The 16S rRNA is highly conserved, has extensive databases, and easy to amplify. On the other hand, there is limited resolution between closely related species, some bacteria contain several slightly different 16S genes, and there are some incomplete databases.

7) What species do you have? Include a screenshot of your top NCBI results.

Leptospira chreensis



8) What is genome annotation? Why is it important to do that?

- Genome annotation is the process of identifying and labeling the functional elements within a DNA sequence. This annotation tells us where genes are and what they likely do. It is important because it makes raw DNA meaningful,

identifies key traits, supports comparative genomics, enables downstream research, and is essential for databases.

9) Perform a genome annotation using two different programs. Find 3 of the 5 genes/features in your results file and create a table of those results: recA, gyrA, 16S rRNA, rpsB, dnaA. What is the location of the genes you chose? What does each program tell you about the gene? How are the outputs different between the two programs.

Prokka	**Location**	**Function**
recA	Reverse(minus) strand: 144,849-145,910 bp	DNA repair, Homologous recombination, SOS response regulation
gyrA	Forward(positive) strand: 374,133-376,769 bp	Introduces negative supercoils into DNA, essential for DNA replication, transcription, and repair
dnaA	Forward(positive) strand: 50,669-52,069 bp	Master initiator of bacterial DNA replication, binds the origin of replication, recruits other proteins to start replication, essential for cell division and chromosome copy number control

dfast	**Location**	**Function**
recA	Reverse(minus) strand: 93,203-94,366 bp	DNA repair, Homologous recombination, antibiotic resistance mechanisms
gyrA	Reverse(minus) strand: 8,337-10,862 bp	Catalyzes ATP-dependent negative supercoiling of DNA, critical for DNA replication and transcription

dnaA	Reverse(minus) strand: 15,670-16,995 bp	Master regulator of bacterial DNA replication initiation, binds the origin of replication and unwinds DNA, controls cell cycle timing and chromosome copy number

- Prokka and DFAST differ in their annotation outputs. Gene locations may vary due to different prediction tools (Prodigal vs. MetaGeneAnnotator) or assembly versions. Functional annotations are simpler in Prokka (e.g., "Protein RecA"), while DFAST adds details like EC numbers, antibiotic resistance links, and cell cycle roles. Evidence in Prokka relies on UniProt/COG, whereas DFAST uses RefSeq with identity percentages and e-values. DFAST also includes extra metadata (e.g., internal IDs), making it more detailed but less concise than Prokka. DFAST used for in-depth analysis and Prokka used for quick reviews.

10) Create a table for your ANI results. How do you interpret these results? What do each of the columns represent?

Query Genome	Reference Genome	ANI (%)	Fragments Used	Total Fragments
spadesout/scaffolds.fasta	neighbors/typhimurium.fasta	98.9482	1,483	1,610
spadesout/scaffolds.fasta	neighbors/bongori.fasta	90.0417	1,241	1,610

- Column representation: Query Genome- the assembled genomes, Reference Genome- genomes compared against *Salmonella* strains, ANI(%)- average nucleotide identity percent used to determine genetic similarity between two genomes, Fragments used- number of conserved DNA fragments aligned for ANI calculation, and Total fragments- the total fragments sampled from the query genome.
- Result interpretation: Genome was identified as *Salmonella enterica* subsp. *enterica* serovar Typhimurium, supported by a 98.95% ANI match with the *typhimurium* reference genome, which confirms species-level identity. In contrast, the 90.04% ANI with *Salmonella bongori* demonstrates the expected genus-level divergence between *S. enterica* and *S. bongori*.