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.**

Name	n	N50	Predicted Genome Length
Assembly-untigs.fa	2649	47163	6832951
Assembly-contigs.fa	2525	71311	6923280
Assembly-scaffolds.fa	2503	88458	6923372

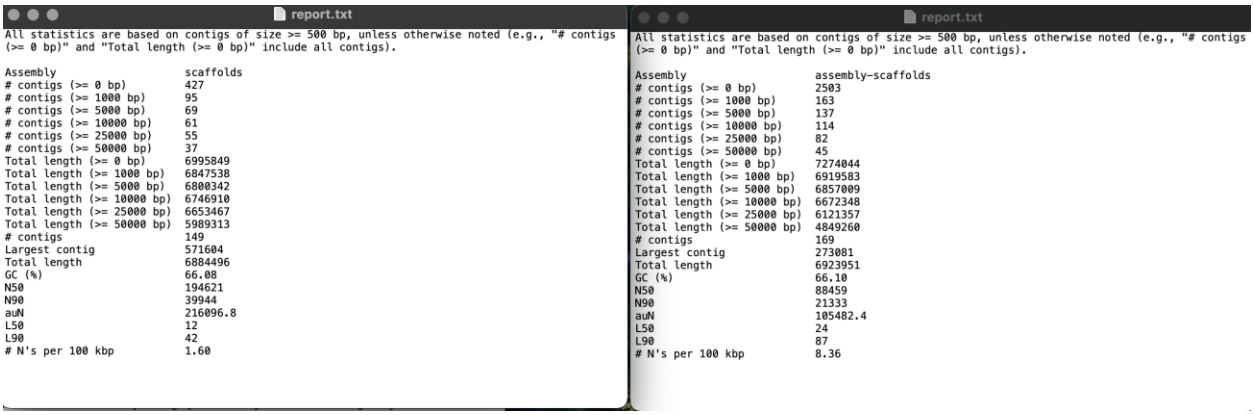
2) https://github.com/bcgsc/abyssLinks to an external site. This is the link to the documentation for ABySS. In your own words, please summarize the function of each of the commands (e.g., abyss-pe, k, B, etc) that you included in your code.

The command line used for this project was abyss-pe name=assembly k=96 B=2G in='SRR32657023_1.fastq.gz SRR32657023_2.fastq.gz'. The command abyss-pe runs the assembly process in ABySS. The command k sets the k-mer length to a specified number. Command B allocates a specified number of memories that are set per thread for the Bloom filter. The command in= specifies what the pair-end reads that will be assembled. In this case the pair-end reads are Sofia_1 and Sofia_2.

3) https://ablab.github.io/spades/index.htmlLinks to an external site. This is the documentation for SPAdes. 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 uses short and long read sequencing data to assemble genomes. This method is used because both techniques have their own strengths so; by combining the techniques it can create a higher quality assembly. I could modify the code used to do a hybrid assembly with nanopore reads by first having two sets of data: the Illumina reads and nanopore reads. Then in spades you would use the command: spades.py -1 illumina_1.fastq -2 illumina_2.fastq --nanopore nanopore.fastq -o hybrid_output. The beginning of this command is the “spades.py -1” which will activate spades. Then you have to specify the two Illumina files which will be assembled. The “-nanopore” will allow for the additional nanopore files. The command will end in “hybrid_output” which will add the files to folder titled ‘hybrid_output.

4) Include a screenshot of the QUAST assembly statistics for the ABySS and SPAdes assembly.



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)	427
# contigs (>= 1000 bp)	95
# contigs (>= 5000 bp)	69
# contigs (>= 10000 bp)	61
# contigs (>= 25000 bp)	55
# contigs (>= 50000 bp)	37
Total length (>= 0 bp)	6995849
Total length (>= 1000 bp)	6847538
Total length (>= 5000 bp)	6800342
Total length (>= 10000 bp)	6746910
Total length (>= 25000 bp)	6653467
Total length (>= 50000 bp)	5989313
# contigs	149
Largest contig	571604
Total length	6884496
GC (%)	66.08
N50	194621
N90	39944
auN	216096.8
L50	12
L90	42
# N's per 100 kbp	1.60

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)	2503
# contigs (>= 1000 bp)	163
# contigs (>= 5000 bp)	137
# contigs (>= 10000 bp)	114
# contigs (>= 25000 bp)	82
# contigs (>= 50000 bp)	45
Total length (>= 0 bp)	7274044
Total length (>= 1000 bp)	6919583
Total length (>= 5000 bp)	6857009
Total length (>= 10000 bp)	6672348
Total length (>= 25000 bp)	6121357
Total length (>= 50000 bp)	4849260
# contigs	169
Largest contig	273081
Total length	6923951
GC (%)	66.10
N50	88459
N90	21333
auN	105482.4
L50	24
L90	87
# N's per 100 kbp	8.36

Spades is on the left and ABySS is on the right

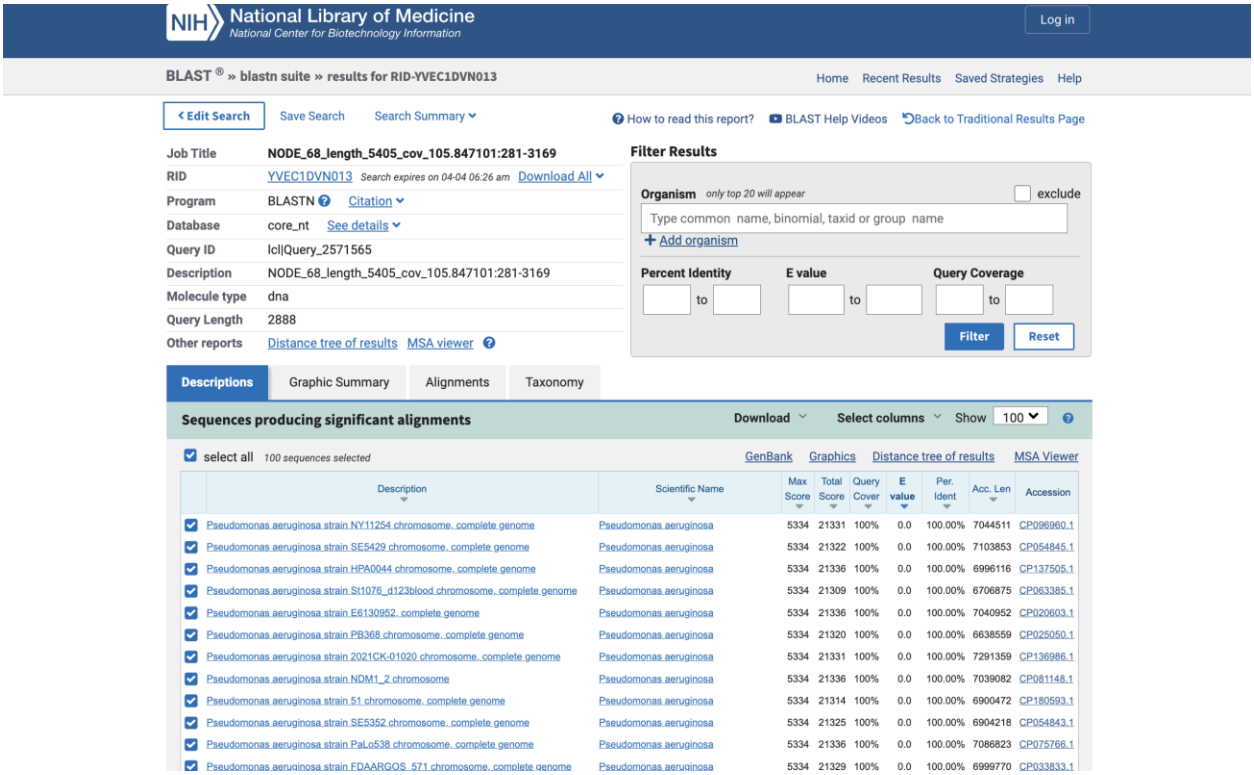
5) Based on the statistics from your genome, which assembly do you think is best? Why? This is the assembly you can use going forward.

Based on the statistics from the genome, the assembly with the highest quality is Spades. The Spades assembly had overall fewer errors as shown in the "# N's per 100 kbp," where spades is 1.60 and Abyss is 8.36. The L50 and L90 results are also lower for the Spades analysis.

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 can be used to figure out what species we have by using barrnap to retreive the 16s rRNA sequence for the species. The 16s rRNA sequence can be copied and pasted into a blastn search on the NCBI database. This will generate the species that most closely match the results.

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



My species is Pseudomonas aeruginosa strain NY11254 chromosome, complete genome.

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

Genome annotation is the process of identifying genes and other key features in an assembled genome. This is important to identify the genome being analyzed, to identify structural components and to assign function. Structural components include ORFs, promoters, operons, introns, exons, various RNAs.

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.

Gene	Location	RAST	Prokka
RecA	Start Base: 85252	RAST Tells Us RAST tells us that the recA gene is a coding sequence that transcribes and translates for the recA protein. It provides the start and end of the gene along with the total length, 1041 bp. RAST also provides the function of the gene and its subsystems. This system gives a more Indepth analysis of the gene's functions and the position of the gene.	Prokka Tells Us Prokka tells us that the RecA gene is located on the locus ICKBCCBO_02443. It is 1041 base pairs long and it's a coding sequence. This means it's a region of the DNA that is transcribed and translated into a protein. This gene is responsible for producing the RecA protein.
		Outputs **ID:** fig|6666666.1445559.peg.6527 **Contig:** NODE_8_length_227180_cov_28.626664 **Type:** CDS **Function:** RecA protein **Subsystem:** DNA repair, bacterial, DNA repair, bacterial RecFOR pathway, DNA repair system including RecA, MutS and a hypothetical protein, RecA and RecX **Start:** 92732 **Stop:** 93772 **Length:** 1041	Outputs **Locus Tag:** ICKBCCBO_02443 **Ftype:** CDS **Length:** 1041 **Gene:** recA **COG:** COG0468 **Product:** Protein RecA
		RAST Tells Us	Prokka Tells Us

rpsB	Start Base: 42800	RAST tells us that rspB is a coding sequence that encodes the ribosomal protein S2p. This protein is in a cluster which aids in ribosome recycling. The RAST system provides a detailed location of the gene including the start points, end points, and the contig. RpsB is positioned on NODE 8 and it's 741 bp long.	Prokka tells us that the rpsB gene is located at locus ICKBCCBO_02404. The length of the gene is 741 and it's a coding sequence. The gene codes for the 30s ribosomal protein S2.
		<u>Outputs</u> **ID:** fig|6666666.1445559.peg.6487 **Contig:** NODE_8_length_227180_cov_28.626664 **Type:** CDS **Function:** SSU ribosomal protein S2p (SAe) **Subsystem:** CBSS-312309.3.peg.1965, Ribosome SSU bacterial, Ribosome recycling related cluster **Start:** 50430 **Stop:** 51170 **Length:** 741	<u>Outputs</u> **Locus Tag:** ICKBCCBO_02404 **Ftype**: CDS **Length**: 741 **Gene**: rpsB **COG**: COG0052 Product: 30S ribosomal protein S2

dnaA	Start Base: 32113	RAST Tells Us RAST tells us that dnaA is a coding sequence which transcribes and translates for dnaA protein. The function of this gene is to produce chromosomal replication initiator protein dnaA. This protein aids in DNA replication. RAST tell sus dnaA is on node 31 and its 1545 bp long.	Prokka Tells Us Prokka tells us that the dnaA gene is located on the locus ICKBCCBO_0526. The gene is 1545 bp long and it's a coding sequence. This gene codes for the chromosomal replication initiator protein called dnaA.
		Outputs **ID:** fig|6666666.1445559.peg.3789 **Contig:** NODE_31_length_64552_cov_37.154381 **Type:** CDS **Function:** Chromosomal replication initiator protein DnaA **Subsystem:** DNA replication cluster 1 **Start:** 40885 **Stop:** 39341 **Length:** 1545	Outputs **Locus Tag:** ICKBCCBO_05260 **Ftype:** CDS **Length**: 1545 **Gene**: dnaA **COG**: COG0593 **Product**: Chromosomal replication initiator protein DnaA

10) Create a table for your ANI results. How do you interpret these results? What do each of the columns represent? Hint: You can refer to the original paper to find that information.

spadesout/scaffolds.fasta	neighbors/putida.fasta	80.1073	929	2232
spadesout/scaffolds.fasta	neighbors/fluorescens.fasta	79.0063	815	2232

The table has two rows and five columns. The first row represents the data for *Pseudomonas putida* and row two represents *Pseudomonas fluorescens*. Column one shows the scaffolds.fasta file for *Pseudomonas aeruginosa*. Column two shows the neighboring species. Column three shows the percentage of shared nucleotide identity. Column four shows the number of aligned nucleotide sequences between the neighboring species and the original species. Column five shows the total number of nucleotides. The table that came from the fastANI results compares the two neighboring species, *Pseudomonas putida* and *Pseudomonas flurescens,* to the scaffolds.fasta file that originated from the *Pseudomonas aeruginosa* data. The first row is comparing *Pseudomonas aeruginosa* to *Pseudomonas putida*. Based on the third column *Pseudomonas putida* shares 80.1073% nucleotide identity. It suggests that 929 out of the 2232 nucleotides are aligned with *Pseudomonas aeruginosa*. The second row is comparing the scaffolds from *Pseudomonas aeruginosa* to *Pseudomonas fluorescens*. Based on the third column *Pseudomonas fluorescens* shares 79.0063% nucleotide identity with *Pseudomonas aeruginosa*. It suggests that 815 out of the 2232 nucleotides are aligned. These results indicate that the two neighboring species are closely related to *Pseudomonas aeruginosa*, and they share several nucleotide sequences.

https://github.com/Samaral7/GenomeAssemblySofia/tree/main/abyssout