Molli Coombs

Genome Assembly Final Report

Introduction

The genome assembled for this experiment was from a gram-negative species of bacillus. It was isolated from a patient with a blood infection and the bacteria was showing signs of anti-biotic resistance. Assembling and analyzing this genome will give insight into the species and may help us to understand where this particular strain is getting its resistance from. Looking at the makeup of the genome can give us information on how the organism operates and what functions it prioritizes. Having the ability to identify the unknown strain quickly allows for better understanding and treatment to future patients that experience a similar problem. The main goal of this experiment was to fully assemble the genome of the bacteria collected from the sick patient and analyze the composition of the genome to allow for further study. The genome was quality checked and then assembled using abyss. It was then uploaded to RAST for identification and analysis after it was sequenced.

Changes from proposal

The original proposal stated the genome would be assembled using Velvet. During the project Abyss was used instead of velvet to assemble the data. Fastqc was used on the raw data to see if trimming was needed before assembling. Trimmomatic was planned to be used to trim the raw data but was unable to be run for the project. The original data was previously assembled and not available as raw data, but raw unassembled data of the species was able to be found and used instead. In the proposal the main goal was to identify the bacteria that was collected and using RAST for analysis you can look at the closest genetic neighbors, but it relies more on you to provide taxonomy information to be more accurate. RAST is more accurate with already known taxonomy data and may not be the best resource for identifying totally unknown species. It does, however, give good insight into the features and subsystems the organism has in its genome.

Results

The Fasta files were successfully retrieved from NCBI to be processed. With the data downloaded the first step was to run the sequences through Fastqc to check the quality of the sequences. Figure 1 shows the qc report of the per base sequence quality. The quality deteriorates at the end which is normal for Illumina MySeq data.Chart, histogram

Description automatically generated

Figure 1. QC report quality scores across all bases.

The next step in the process was to crop the sequences to 220 base pairs using trimmomatic. This step was unable to be executed because of an issue with the Java script that couldn’t be fixed. Since the QC report looked good, we were able to proceed with the untrimmed data and assemble the genome.

The genome was assembled using Abyss as the assembler. It was run at seven different K values to see which one worked best. Figures 2-4 show the graphs made to evaluate the k-mer lengths and decide which assembly length worked best with the data.

Figure 2. N50 values plotted against k-mer values Figure 3. Max BP value plotted against k-mer values.

Figure 4. Total base pairs of sequence plotted against k-mer values.

Using the three figures above I decided the assembly using the k-mer value of 115 had the best results and this is the one I chose to analyze.

Using rast.nmpdr.org the scaffolds file of the assembly was uploaded and analyzed. Figure 5 shows the graph of the subsystem content in the genome. This is a good way to look at the general makeup of the genome and can also be used to compare the genome to close genetic neighbors. Looking at the closest genetic neighbors can help confirm the identity of the organism with the genome that was assembled.

Discussion

Chart, pie chart

Description automatically generated With the genome fully assembled it was uploaded to RAST and figure 5 can be used to look at the way the genome is broken up in the organism. The largest portions are amino acids and carbohydrates, but a large portion is also membrane transport. A feature that was found in the sequence was for biotin synthesis and fatty acid metabolism. When looking at the closest genetic

Figure 5. RAST subsystem breakdown of genome

neighbors they are made up of other Cupriavidus bacteria, suggesting the identification of this bacteria as Cupriavidus basilensis is correct.

Due to the fact that trimming was unable to be done on the raw data that could have an impact on the full sequencing assembly. This could make identification and analysis harder if the raw data being used is messy, but since the QC report for this

data looked good; I am confident that the assembly without trimming is fine for this sequence.

This was my first experience assembling a full genome and overall the process wasn’t as intimidating as I thought it would be. I now have a better understanding of how to write pipelines for assemblies like this which will be helpful in the future. Analyzing the outputs of the pipelines and examining quality throughout the project showed me what to look for in good and bad sequences when assembling genomes.

A few challenges faced during the project were not having the raw data to use in the beginning, not being able to execute the trimming commands, and running into the k-mer value limit at the beginning of assembling. To deal with the raw data issue we were able to find unassembled raw sequences that closely matched what was used in the original paper, and this was used for assembly. The QC that was run on the raw unpaired data was good enough that not being able to crop the sequences wasn’t detrimental to the assembly, and the data was able to be used. When running the assembly script, the first few times some of the k-values I was trying to use wouldn’t run. It was found out that the program has a k-value cap of 128, so I was able to re-run the scripts at lower k-values and get successful assemblies.

Pipeline Code

Assembly prep

# run fastqc on unpaired raw data

fastqc SRR12739615\_1.fastq

fastqc SRR12739615\_2.fastq

Assembly trimming

# crop each sequence to 220 base pairs each to get rid of the lower quality data at the end

trimmomatic PE -threads 6 -trimlog trim.log SRR12739615\_1.fastq SRR12739615\_2.fastq -baseout trimmed.fastq CROP:220

Assembling

# assemble the unpaired fastq files at different k-values using abyss assembler

abyss-pe np=8 name=SRR12739615\_123 k=123 in='SRR12739615\_1.fastq

SRR12739615\_2.fastq'

abyss-pe np=8 name=SRR12739615\_121 k=121 in='SRR12739615\_1.fastq

SRR12739615\_2.fastq'

abyss-pe np=8 name=SRR12739615\_128 k=128 in='SRR12739615\_1.fastq

SRR12739615\_2.fastq'