CSUEB GBI Bioinformatics, Command Line, and AWS Introduction Report

Flint Mitchell

July 11 - 15, 2022

Abstract

The purpose of this project was to learn the basics of the command line, how to use Amazon Web Services EC2 instances, and to take long and short read sequencing data of a sweet potato chloroplast genome and assemble them together into larger, more useful, annotated contigs.

Methods and Results

- *Downloading the Sequencing Data*

We found subsampled short and long read data from the following research paper:

Zhou C, Duarte T, Silvestre R *et al.* Insights into population structure of East African sweetpotato cultivars from hybrid assembly of chloroplast genomes [version 1; peer review: 2 approved with reservations]. *Gates Open Res* 2018, **2**:41 (https://doi.org/10.12688/gatesopenres.12856.1)

We then downloaded this data using the curl command:

```
curl
https://zenodo.org/record/3567224/files/sweet-potato-chloroplast-illumina-reduced.fastq?download=1 > sweet-potato-chloroplast-illumina-reduced.fastq

curl
https://zenodo.org/record/3567224/files/sweet-potato-chloroplast-nanopore-reduced.fastq?download=1 > sweet-potato-chloroplast-nanopore-reduced.fastq
```

After we downloaded the data, we checked to see how many reads we had using the wc command:

```
wc sweet-potato-chloroplast-illumina-reduced.fastq
sweet-potato-chloroplast-nanopore-reduced.fastq
```

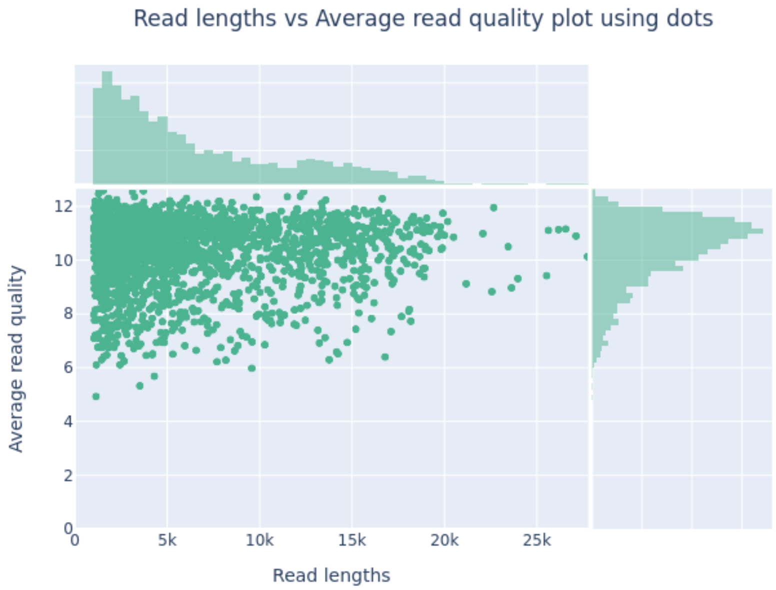
From this we learned that we had 62500 illumina short reads and 2000 Oxford Nanopore (ONT) long reads.

- *Checking the Quality of the Data*

In order to look at the length of our nanopore reads and assess the quality of them, we used a program called NanoPlot. The command to do this was:

```
NanoPlot -t 2 --fastq sweet-potato-chloroplast-nanopore-reduced.fastq
```

This command generated the following plot, showing us the quality scores of the reads with given lengths.



According to NanoPlot, our data had the following average read lengths and qualities.

Mean read length	6,664.9
Mean read quality	10.3
Median read length	4,942.0
Median read quality	10.6

- ***Filtering and Trimming the Data***

This data was already filtered and trimmed, but if I needed to filter and trim this data I would use the following program:

```
trim_galore -q 25 SRR1946554_1_4k.fastq.gz
```

- ***Genome size estimation***

We already had an estimate for the size of the sweet potato chloroplast genome at approximately 160kb (160,000 bases). If we wanted to estimate the size of this genome, we could have used a k-mer counting program called jellyfish and the plotting software on Genome Scope (http://qb.cshl.edu/genomescope/).

- Assembly

With our 2 datasets, we generated 3 assemblies.

- This first used only the nanopore long-read data.
- The second used only the illumina short-read data.
- The third used the result of the first assembly using the long-reads, and polishing it with the short-read data.

For the first assembly using only long-read data, we used a software called flye. This software takes inputs of the estimated genome size, the fastq file with our reads, and a number of CPU threads to speed up the process. The command I used looked like:

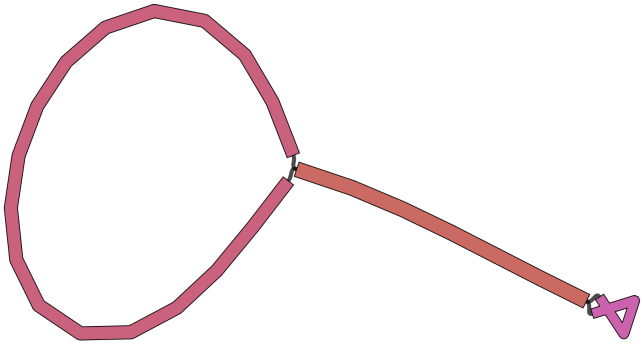
```
flye --genome-size 160000 --nano-raw
sweet-potato-chloroplast-nanopore-reduced.fastq -t 4 --out-dir
sweet-potato-chloroplast-ont-assembly-071322
```

Using the cat command on our CLI, we read out the results of this assembly:

```
nt-assembly-071322$ cat assembly_info.txt
#seq_name  length cov.  circ.  repeat mult.  alt_group  graph_path
contig_1   131851       86     N      N      1          *          -2,1,2
contig_3   4895223      N      N      1      *          2,3,-2
contig_2   29900140     N      Y      1      *          2
```

Using the software tool Bandage, we generated a picture of how the contigs above fit together. In order to do this, we used the following command:

```
Bandage image assembly_graph.gfa sweetpotatogfa.png
```



For the second assembly using only short-read illumina data, we used a program called ABySS. The command to run ABySS that we used is:

```
abyss-pe name=spotato-chloroplast-k50-071422 j=2 v=-v k=31
in="sweet-potato-chloroplast-illumina-reduced.fastq" | tee
sweet-potato-chloroplast-assembly-stdout.log
```

For the third assembly where we polished our long-read assembly with the short read data, we used a series of programs called BWA, samtools, and pilon. BWA was used to index and align our short reads to our long read assembly. Samtools was used to make the output of BWA more compatible with our computer. Pilon was then used to polish the long-read assembly with the higher quality-score short read data.

First, use the following command:

```
bwa index assembly.fasta
```

Then let's copy the short reads to our current directory:

```
cp ~/seqdata/sweet-potato-chloroplast-illumina-reduced.fastq .
```

Next we have a command that creates the alignment between the short reads and the long-read assembly:

```
bwa mem assembly.fasta
../../../seqdata/sweet-potato-chloroplast-illumina-reduced.fastq > aln-se.sam
```

After this we need to convert the output file (which is the `.sam` file) into something thats easier for the computer to read. The SAM file is human-readable and less efficient, where as the BAM file is in binary:

```
samtools view -S -b aln-se.sam > aln-se.bam
```

Then we have to do some further editing to this BAM file with the following two commands:

```
samtools sort aln-se.bam -o aln-se.sorted.bam
```

and

```
samtools index aln-se.sorted.bam
```

And finally we are ready to do the actual polishing of our long read assembly with the short reads using `pilon`:

```
java -Xmx12G -jar pilon-1.24.jar --genome assembly.fasta --bam
aln-se.sorted.bam
```

- Assembly Quality

In order to assess the quality of our assemblies, we used a program called QUAST, Quality Assessment Tool. This generates statistics related to the completeness of our assembly.

The command to run QUAST that we used is:

```
quast.py -o nanopore-chloroplast-assembly --est-ref-size 160000 --min-contig
100 assembly.fasta
```

The outputs of QUAST for the three assemblies were:

For the long-read only assembly using flye:

```
nt-assembly-071322/nanopore-chloroplast-assembly$ cat report.txt
All statistics are based on contigs of size >= 100 bp, unless otherwise noted (e.g., "# contigs (>= 0 bp)" and "Total length (>= 0 bp)" include
all contigs).

Assembly                   assembly
# contigs (>= 0 bp)        3
# contigs (>= 1000 bp)     3
# contigs (>= 5000 bp)     3
# contigs (>= 10000 bp)    3
# contigs (>= 25000 bp)    3
# contigs (>= 50000 bp)    1
Total length (>= 0 bp)     210703
Total length (>= 1000 bp)  210703
Total length (>= 5000 bp)  210703
Total length (>= 10000 bp) 210703
Total length (>= 25000 bp) 210703
Total length (>= 50000 bp) 131851
# contigs                  3
Largest contig             131851
Total length               210703
Estimated reference length 160000
GC (%)                     39.17
N50                        131851
NG50                       131851
N90                        29900
NG90                       48952
auN                        98123.87343796718
auNG                       129218.71565625
L50                        1
LG50                       1
L90                        3
LG90                       2
# N's per 100 kbp          0.00
```

The results of QUAST for the short read illumina data only were:

```
oplast$ cat 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                    spotato-chloroplast-k50-071422-unitigs
# contigs (>= 0 bp)         35
# contigs (>= 1000 bp)      10
# contigs (>= 5000 bp)      7
# contigs (>= 10000 bp)     4
# contigs (>= 25000 bp)     2
# contigs (>= 50000 bp)     0
Total length (>= 0 bp)      131948
Total length (>= 1000 bp)   128852
Total length (>= 5000 bp)   118321
Total length (>= 10000 bp)  99504
Total length (>= 25000 bp)  75047
Total length (>= 50000 bp)  0
# contigs                   10
Largest contig              45529
Total length                128852
GC (%)                      36.89
N50                         29518
N90                         5930
auN                         26403.68987675783
L50                         2
L90                         7
# N's per 100 kbp           0.00
```

The results of QUAST for our long-read assembly polished with high quality short reads is:

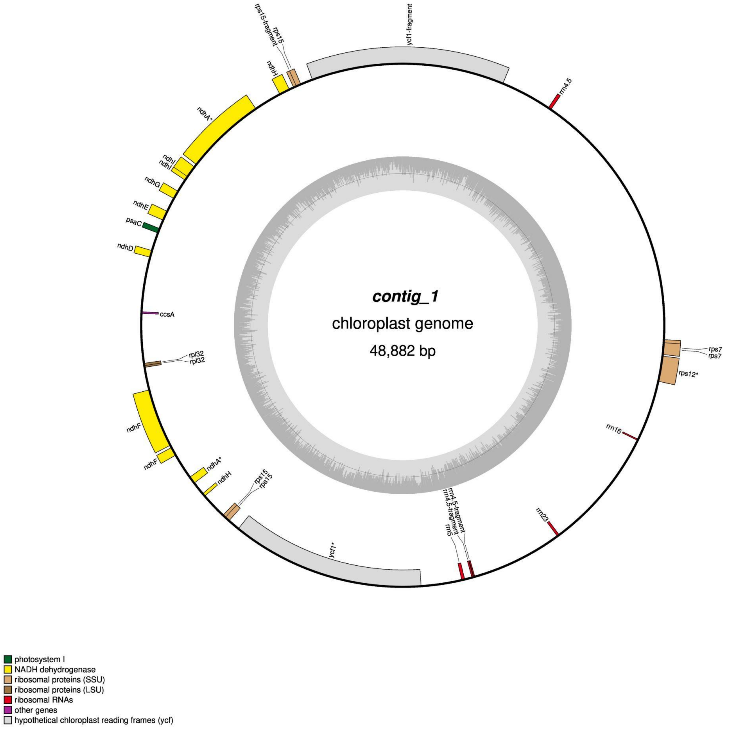
```
ubuntu@ip-172-31-56-123:~/assembly/sweet-potato-nanopore-ONT/nanopore-chloro-assembly-polished$ cat report.txt
All statistics are based on contigs of size >= 100 bp, unless otherwise noted (e.g., "# contigs (>= 0 bp)" and "Total length (>= 0 bp)" include all contigs).

Assembly                    pilon
# contigs (>= 0 bp)         3
# contigs (>= 1000 bp)      3
# contigs (>= 5000 bp)      3
# contigs (>= 10000 bp)     3
# contigs (>= 25000 bp)     3
# contigs (>= 50000 bp)     1
Total length (>= 0 bp)      215274
Total length (>= 1000 bp)   215274
Total length (>= 5000 bp)   215274
Total length (>= 10000 bp)  215274
Total length (>= 25000 bp)  215274
Total length (>= 50000 bp)  135115
# contigs                   3
Largest contig              135115
Total length                215274
Estimated reference length  160000
GC (%)                      39.03
N50                         135115
NG50                        135115
N90                         30411
NG90                        49748
L50                         1
LG50                        1
L90                         3
LG90                        2
# N's per 100 kbp           0.00
```

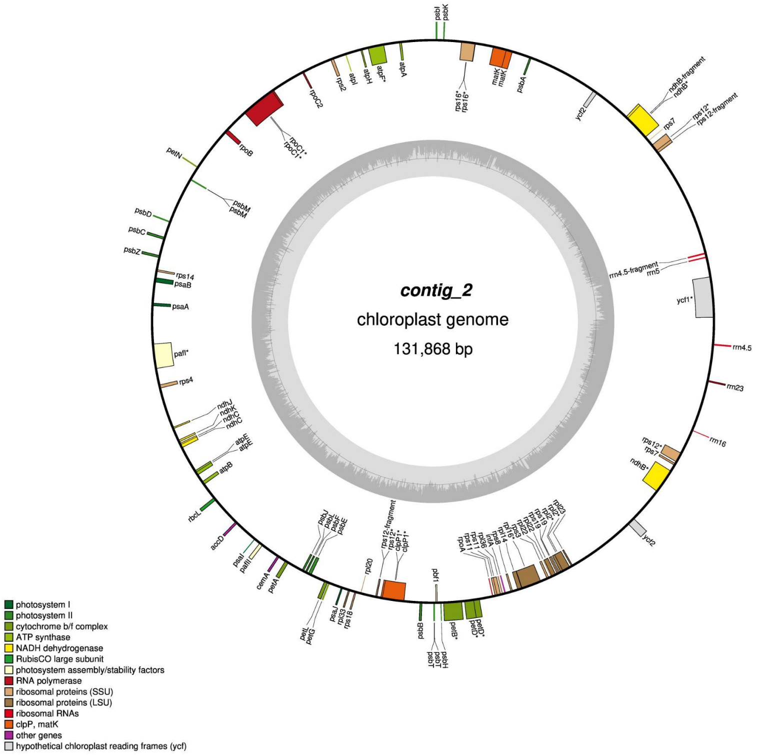
By polishing our long-read assembly with our high quality short reads, the length of our long-read assembly, its longest contig, and its N50 increased in size. This might have to do with the correction of homopolymer deletions in the nanopore data.

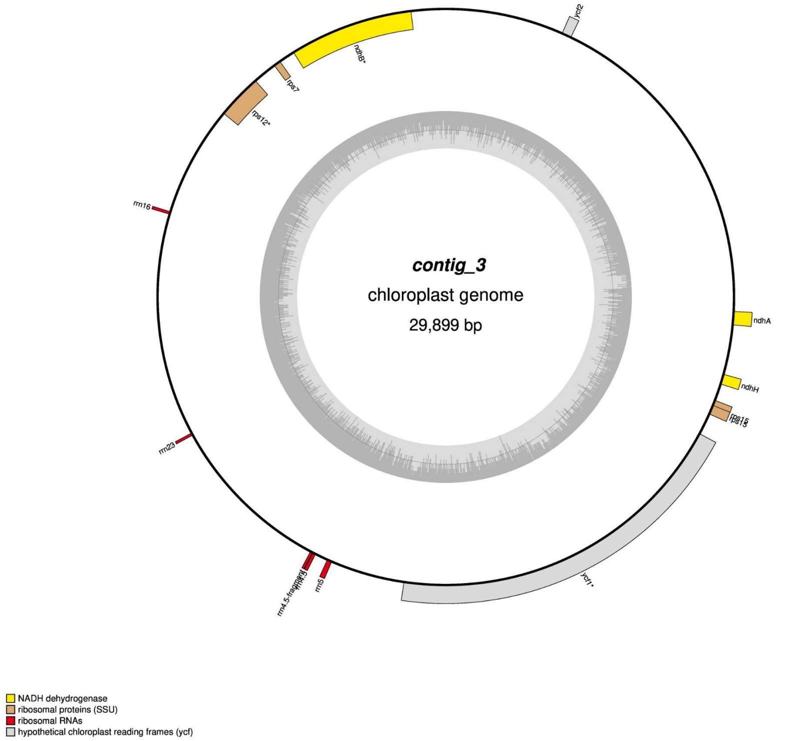
This total number of contigs in our long-read assemblies was less than that of our short-read assemblies. This might be because there is repetitive information in the sweet potato chloroplast genome that the short reads cannot determine how long they are. Since we can't determine the length of the repetitive segments, maybe we can't connect the contigs on either side of those and we end up with gaps.

- Annotation

Using our polished assembly, we then used a tool called GeSeq (https://chlorobox.mpimp-golm.mpg.de/geseq.html) to annotate it with biological information. After uploading our polished assembly, we checked the boxes for "circular" instead of "linear" and for the Sequence Source, we checked "Plastid." We can see identifiable biological elements in the results of this annotation, as shown in the following pictures:


contig_1
chloroplast genome
48,882 bp
ycf1-fragment
rrn4.5
rps7
rps7
rps12*
rrn16
rrn23
rrn4.5-fragment
rrn4.5-fragment
rrn5
ycf1*
rps15
rps15
ndhH
ndhA*
ndhF
ndhF
rpl32
rpl32
ccsA
ndhD
psaC
ndhE
ndhG
ndhI
ndhI
ndhA*
ndhH
rps15-fragment
rps15
🟩 photosystem I
🟨 NADH dehydrogenase
▨ ribosomal proteins (SSU)
▨ ribosomal proteins (LSU)
🟥 ribosomal RNAs
▨ other genes
□ hypothetical chloroplast reading frames (ycf)


contig_2
chloroplast genome
131,868 bp
psbI
psbK
rps16*
rps16*
matK
matK
psbA
ycf2
ndhB-fragment
ndhB*
rps7
rps12*
rps12-fragment
rrn4.5-fragment
rrn5
ycf1*
rrn4.5
rrn23
rrn16
rps12*
rps7
ndhB*
ycf2
rpl23
rpl2*
rpl2*
rps19
rpl22
rps3
rpl16*
rpl14
rps8
infA
rpl36
rps11
rpoA
petD*
petD*
petB*
psbH
psbT
psbT
pbf1
psbB
clpP1*
clpP1*
rps12*
rps12-fragment
rpl20
rps18
rpl33
psaJ
petG
petL
psbE
psbF
psbL
psbJ
petA
cemA
pafII
psaI
accD
rbcL
atpB
atpE
atpE
ndhC
ndhC
ndhK
ndhJ
rps4
pafI*
psaA
psaB
rps14
psbZ
psbC
psbD
psbM
psbM
petN
rpoB
rpoC1*
rpoC1*
rpoC2
rps2
atpI
atpH
atpF*
atpA
🟩 photosystem I
🟩 photosystem II
🟩 cytochrome b/f complex
🟩 ATP synthase
🟨 NADH dehydrogenase
🟩 RubisCO large subunit
□ photosystem assembly/stability factors
🟥 RNA polymerase
▨ ribosomal proteins (SSU)
▨ ribosomal proteins (LSU)
🟥 ribosomal RNAs
🟧 clpP, matK
▨ other genes
□ hypothetical chloroplast reading frames (ycf)



References

All of the references for the data and tools used in this report can be found at https://github.com/Green-Biome-Institute/AWS/wiki/Chloroplast-Assembly-Tutorial!

Conclusion

I now have the knowledge and tools to take noisy data like sequencing data and generate important information from them.