Matthew Spriggs
Applied Bioinformatics | S430452
matthew.spriggs.452@cranfield.ac.uk

Table of Contents

Matthew Spriggs
Applied Bioinformatics | S430452
matthew.spriggs.452@cranfield.ac.uk

ASSIGNMENT | **Advanced Sequencing Informatics and Genome Assembly**

MATTHEW SPRIGGS

APPLIED BIOINFORMATICS

ESTIMATED WORD COUNT (EXCLUDING FIGURES AND QUESTIONS): 2041 WORDS.

Matthew Spriggs
Applied Bioinformatics | S430452
matthew.spriggs.452@cranfield.ac.uk

Report

Note: Throughout this report references to files will be mentioned in block capitals e.g. SOAP_ASSEMBLY. Appendix A provides a lookup table for these in-text references and their accompanying file paths relative to the submission zip file.

1. Read QC

1.1 The starting point for this pipeline is paired-end Illumina short reads [ILLUMINA_SR_READ_1, ILLUMINA_SR_READ_2] and some PacBio long reads [PACBIO_READS].

1.2 In what can be considered standard practice, the first stage of this pipeline involved running FastQC (Simon, 2010) followed by MultiQC (Ewels et al., 2016) on the Illumina short reads. *Note: PacBio provided in FASTA format do not have a quality score and could not be assessed in the same manner.*

1.3 The short reads pipeline was executed with 01_QC.sh, producing multiple html output files, notably a fastqc.html file for each read, and an aggregated html file from MultiQC [MULTIQC_REPORT]. The 01_QC.log also provides some high level descriptive statistics for both short read and long read data (Table 1).

Table 1: Summary table for 01_QC.sh descriptive fast file statistics for Illumina short reads and PacBio long reads.*

File	Number Of Reads	Number Of Bases (Mbp:1sf)
ILLUMINA_SR_READ_1	2475000	249.96
ILLUMINA_SR_READ_2	2475000	249.96
PACBIO_READS	33413	49.97

1.4 On interpreting the MULTIQC_REPORT the first initial flag was all sequence quality scores were flagged as failing (Figure 1). By inspecting the Illumina FASTQ files it was noticed that the quality score were all "2". The third line of a FASTQ file is an ASCII encoded error score. The character "2" is a ASCII decimal for "50", which would translate to a Phred score of 17 — or < 0.01 probability the base call is correct (Illumina, n.d.). Interestingly, all the bases in the FASTQ file have the same quality, suggesting this might be a default value or simulated reads of some description.

Matthew Spriggs
Applied Bioinformatics | S430452
matthew.spriggs.452@cranfield.ac.uk

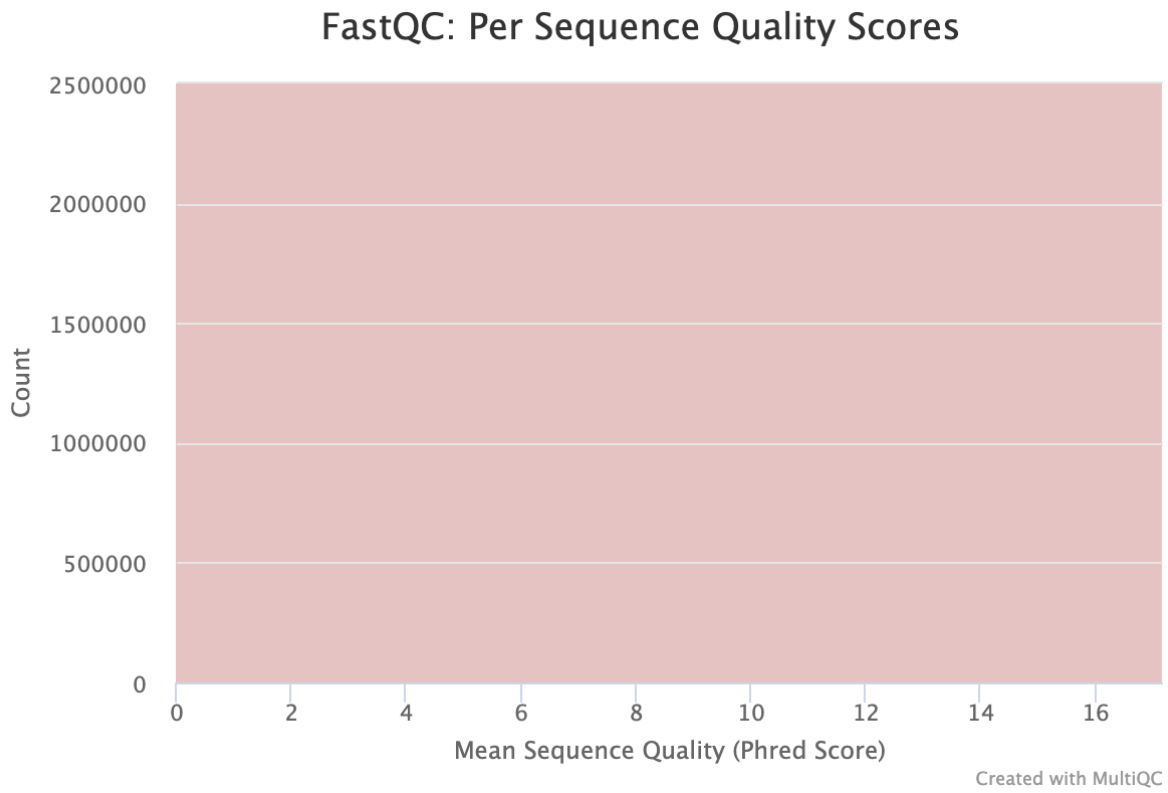

Figure 1: Per sequence quality scores for ILLUMINA_SR_READ_1, ILLUMINA_SR_READ_2 from MULTIQC_REPORT.

1.5 The rest of the MULTIQC_REPORT looked sufficient.

1.6 All reads had a sequence length distribution of 101 bp, and no overrepresented sequencing, implying that the FASTQ files did not have any adapter sequences or artifacts that needed to be removed. For this reason, a decision was taken to omit read trimming.

2. Estimate Genome Size

2.1 The pipeline turns to k-mer analysis and error correction.

2.2 Depending on platform used sequencing data can have different types of systematic bias. For example, substitution errors can be introduced as a result of Illumina sequencing (Yang et al., 2013). For this reason the error correction module from SOAP (Luo et al., 2012). The 00_ERROR_CORRECTION script was run with kmer size 15 and 17 to contrast (Table 2). The remaining parameters were kept as standard.

2.3 On interpretation of SOAP-ec, k-mer 15 produced substantially greater number of k-mers but proportionately less distinct k-mer, increasing k-mer_depth for 15. K-mer 17 produced comparatively less reads and more distinct k-mers, suggesting that the extra reads from k-mer 15 could partly be attributed to noise. The genome size is mostly concordant between k-mers, suggesting a genome size of ~5 Mbp.

2.4 K-mer 17 was chosen as a more conservative choice. The key output from SOAP-ec is a gzipped FASTA file which will be used as input to the assembly tools described below. These error-corrected reads are COR_ILLUMINA_SR_READ_1 and COR_ILLUMINA_SR_READ_2.

Table 2: SOAP-ec Summary Table for kmer 15 and 17.

kmer	kmer_num	kmer_depth	genome_size	base_depth	avg_read_len	unq_kmer_num
17	420750000	60.144	5037036	99.2548	101	92117677
15	430650000	64.7529	5061839	98.7685	101	66172001

2.5 Moving to k-mer spectrum analysis of reads , the tool Jellyfish (Marçais & Kingsford, 2011). K-mer counts are an important step to estimate genome size. If a high proportion of k-mers occurs 'x' times then the sequencing depth can be estimated to be approximately 'x'. Figure 2 shows a histogram distribution of frequency of individual k-mers occurring at a given sequencing depth. The script 02a_KMER_2PASS.sh was used for this. From this plot we can infer some characteristics of our genome. Notably, the sequencing depth of our read is estimated to be >60x. With an estimated genome size of ~5 Mb. A haploid genome is expected, this was chosen in part based on the size of the initial FASTA files (116 Mb). A nucleotide BLAST on the PACBIO_READS suggests

human origin, so it was hypothesised at this point to be a mitochondrial genome, hence haploid is a reasonable choice since the mitochondrial genome is passed down the maternal lineage only.

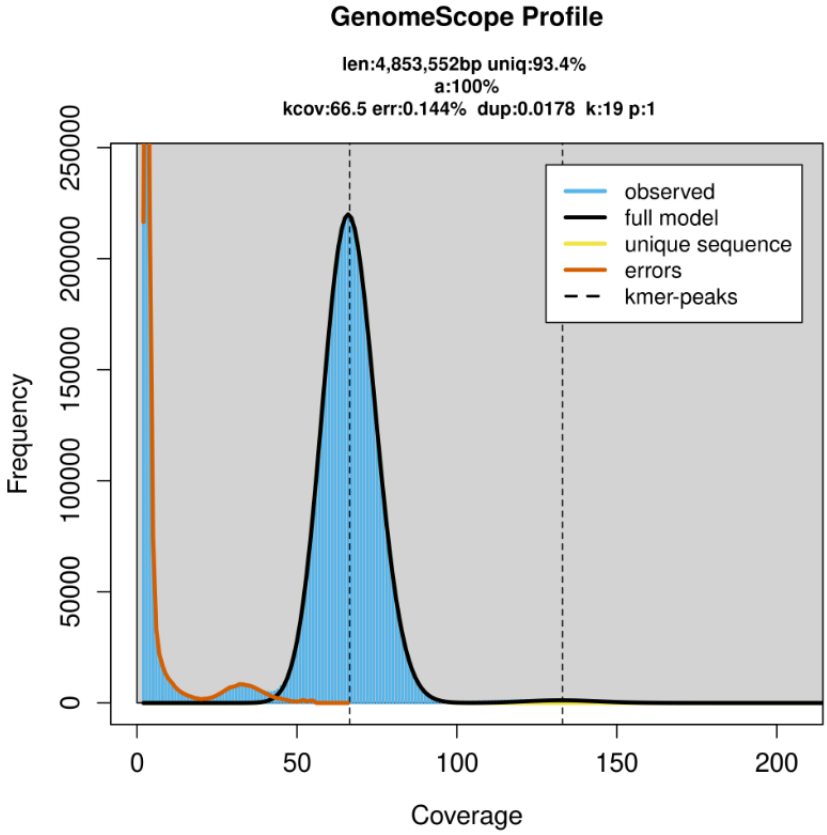

Figure 2: GenoScope Jellyfish profile with k19 using two-pass method. The estimated genome size is 4,853,552bp with 93.4% unique reads and 0.0178 duplicated reads. The error rate percentage is 0.144%. The k-mer coverage is 66.5x for a ploidy genome of 1 — assumed bacterial in origin.

3. Assembly (SOAP)

3.1 The pipeline moves onto assembly of reads. An initial assembly was produced using only the short reads — largely this was a pragmatic choice to have a baseline comparison.

3.2 The tool SOAPdenovo2 (Luo et al., 2012) was used for this using 03_SR_SOAP_ASSEMBLY.sh. This is a qsub batch submission that takes in a required parameter 'input_kmer_size'. This scripting technique was chosen to allow the operator to easily iterate different k-mers without changing the script. Multiple values were tested for k-mer size and the tool gnx was used to assess performance —discussed below. SOAPdenovo-127 program was selected over SOAPdenovo-63 to allow for higher k-mer sizes. The tool was used to build an initial pregraph using the 'z' parameter of 5,000,000 —which is an estimate of genome size (from Figure 2) — its primary use being memory allocation, so the value was set higher than the calculated genome size following the developers' user guide (*GitHub - Aquaskyline/SOAPdenovo2: Next Generation Sequencing Reads de Novo Assembler.*, n.d.). The remaining parameters in the soapPR.config were kept as per the training protocol described in Mohareb (2026). Further exploration of the asm_flag might yield a better assembly using both contig and scaffold assembly — but this was not attempted.

Table 3: SOAP top three assemblies ordered by N50 DESC. The table shows the k-mer size used, the total number of sequences, the total length of sequences, the total number of null bales, the N50 and the number of sequences that make up 50% for the assembly.

kmer_size	num_seqs	len_seqs	NaNs	N50	N50_num_seqs
60	2937	5135216	0	17822	89
64	2767	5139508	0	16151	95
55	3567	5155507	0	15018	101

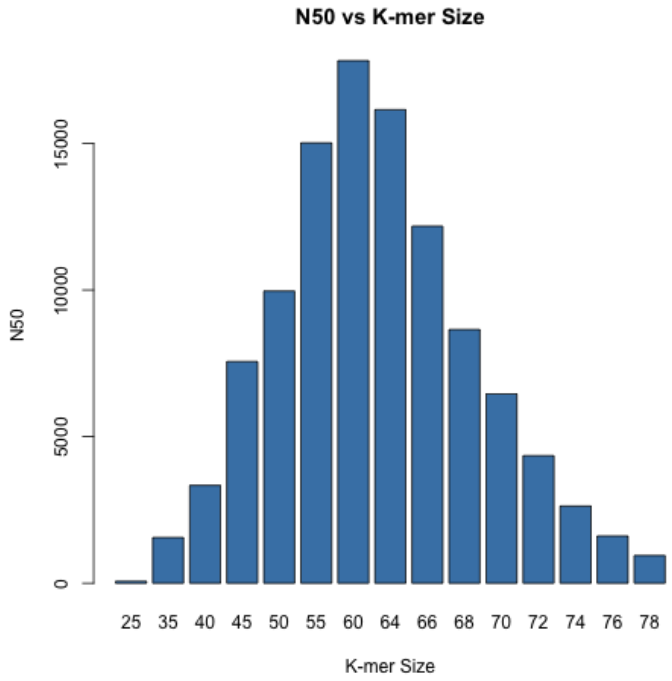

Figure 3: SOAP K-Mer assembly distribution.

3.3 Table 3 shows the descriptive statistics from the top three SOAP assemblies using COR_ILLUMINA_SR_READ_1 and COR_ILLUMINA_SR_READ_2. Of the three assemblies k-mer 60 had the greatest N50 (Figure 3) —although N50 is not the only metric that can be used to perform quality assessment. ALE and KAT were used to explore the assemblies further.

3.4 ALE is a tool designed to give researchers a quantitative and statistically grounded process to determine the quality of their assembly (Clark et al., 2013).

3.5 Submission script 04_ALE.sh was used to generate a quantifiable score for the short read assemblies constructed in section 3 following the tutorial documented in Mohareb (2026a). This pipeline was run for several assemblies [`/assignment/logs/04_ale_k??.log`]. Bash was used to pipe the results to 04_ALE_SUMMARY.log and a LLM was used to summarise this log into csv format 04_ALE_SUMMARY.csv (**Error! Reference source not found.**).

Table 4: ALE Summary Table ordered by ALE_Score DESC.

kmer	ALE_score	numContigs	placeAvg	insertAvg	kmerAvg	depthAvg
64	-7369701	2767	0.984	-1.421	-0.157	81.580
60	-7436427	2932	0.966	-1.348	-0.157	81.626
55	-9363979	3567	0.691	-1.489	-0.157	81.268

3.6 The assembly with the highest ALE score (closest to zero) is the one with the largest probability of being correct (Clark et al., 2013). For this reason the SOAP assembly k-mer 64 was selected for the downstream pipeline — henceforth just referred to as SOAP_ASSEMBLY.

3.7 A final sanity check was carried out using the KAT tool (Mapleson et al., 2017) with the SOAP_ASSEMBLY. KAT outputs a series of stats files [`assignment/assembly_qc/kat/*/*.stats`] and a k-mer comparison plot (Figure 4).

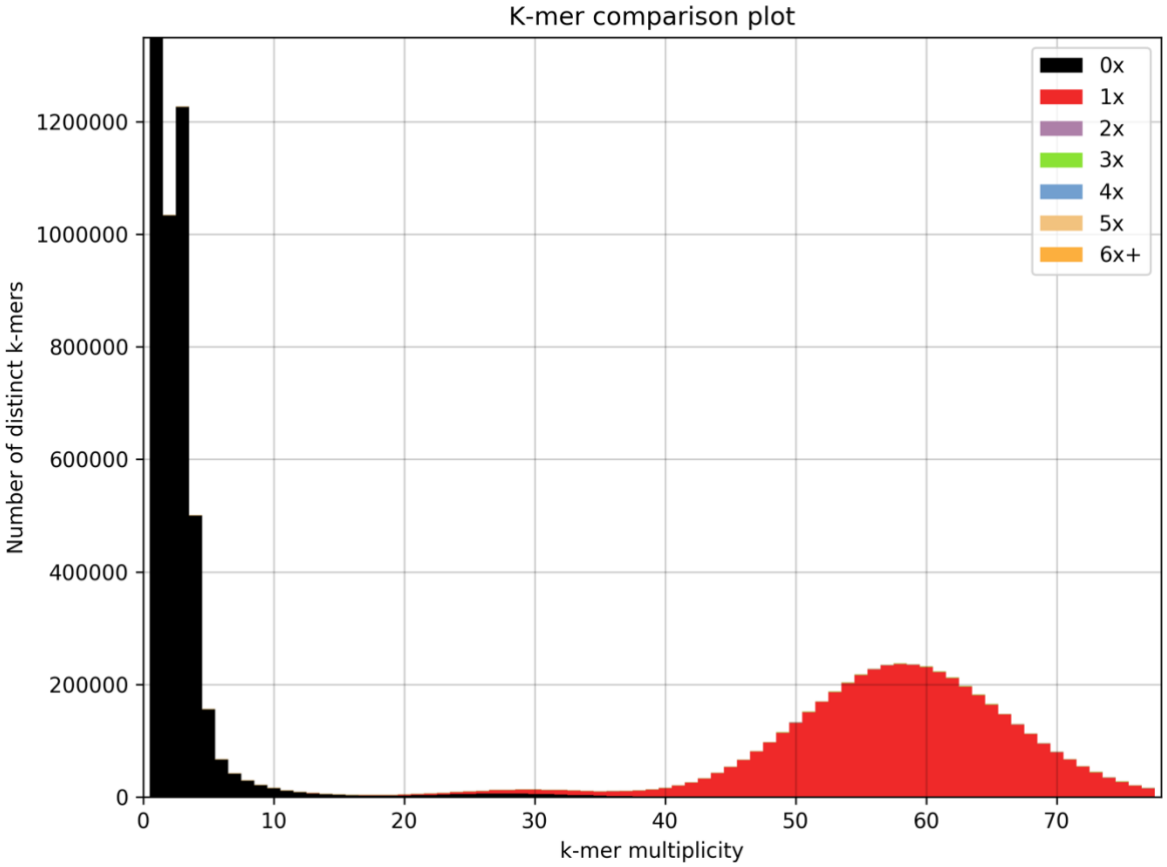

Figure 4: KAT K-mer comparison plot for SOAP_ASSEMBLY using COR_ILLUMINA_SR_READ_1 and COR_ILLUMINA_SR_READ_2.

3.8 This plot shows which k-mers from the reads are incorporated into the assembly and at what frequencies. The sharp peaks at low multiplicity (0–7) correspond to sequencing errors, dimers, or very short fragments that appear only a few times and are unlikely to represent real biological sequence. The main bell-shaped curve reflects the true genomic k-mer content, which in this case appears once in the assembly. This pattern is characteristic of a haploid genome, supporting a bacterial origin (small, haploid), though the nucleotide BLAST hits to human genes suggest the sequence is possibly mitochondrial.

3.9 The plot confirms the viability of the SOAP_ASSEMBLY as the best short-read assembly using short-reads data only.

4. Assembly (DBG2OLC)

4.1 The next stage of the pipeline introduces the long reads from PACBIO_READS, described in Table 1. Several tools were considered to incorporate long reads, namely Canu and Falcon. It was decided that a hybrid assembly can make best use of the high-quality short reads and use long reads as scaffolds to resolve ambiguous alignments.

4.2 The first of these tools is DBG2OLC, a reputable tool published in Nature by Ye et al. (2016). The submission script 06_DBG2OLC.sh was used for this. In a canonical DBG2OLC pipeline the tools SparseAssembler can be used to build a de-Bruijn graph of the short reads. However, any accurate de-Bruijn assembler can be used providing techniques such as gap closure or scaffolding have not been used. The SOAP_ASSEMBLY fit this criterion, as such, was used as the input short-read assembly for DBG2OLC.

4.3 The qsub input parameter 'input_kmer_size' was used to help tune the required k-mers for this pipeline. Values of [50, 58, 60, 64, 70] k-mers were iterated to find the best assembly. The key output file is the which contains the final assembly contigs. Results were aggregated using gnx_wrapper from UTILS.sh into 06_DBG2OLC_SUMMARY.txt which was summarised into 06_DBG2OLC_SUMMARY.csv by an LLM (Table 5).

Table 5: DBG2OLC K-Mer analysis.

kmer_size	num_seqs	len_seqs	NaNs	N50	N50_num_seqs
k50	32	4229677	0	190414	6
K58	32	4229677	0	190414	6
k60	32	4229677	0	190414	6
K64	32	4229677	0	190414	6
K70	32	4229677	0	190414	6

4.4 Table 5 shows that adjusting the '-k' flag does not impact the assembly for this configuration. All of these iterations used the SOAP_ASSEMBLY (kmer_64 from SOAP). *Note: it is a little unclear from the docs what the -k flag represents.*

4.5 Some key parameters to tune for this pipeline would be the AdaptiveTh, KmerCovTh and MinOverlap. However, DBG2OLC produced 32 contigs compared with 2,937 from SOAPdenovo2—a 98.9% reduction in contig count and the final N50 for DBG2OLC was a 10-fold improvement on the SOAPdenovo2 score. So although the pipeline could likely be optimised, it was deemed good enough to continue with other hybrid assembly tools.

5. Assembly (MaSuRCA)

5.1 The final assembly tool considered for this report was MaSuRCA (Zimin et al., 2013). Submission script MASURCA.sh This script creates a boilerplate config file that's needed for execution of the tool, and dynamically updates this configuration to aim for reproducibility of results and avoid human error. After this step the tool was relatively simple to use and a final assembly is produced at the file path: `/assignment/assembly/masurca/HS7_R1.fastq.gz.cor_k64_soap_SR_masurca_hybrid_k64/CA.mr.41/final.genome.scf.fasta`.

kmer_size	num_seqs	len_seqs	NaNs	N50	N50_num_seqs
64	19	4969080	0	1408291	2

5.2 This assembly used the same SOAP_ASSEMBLY as previous tools for short-read inputs. The build is produced from 19 sequences with a 13.5% improvement in N50 over DBG2OLC.

6. Polishing

6.1 A final polishing step was carried out for the final DBG2OLC and MaSuRCA assemblies using 07[A|B]_PILON.sh. These final builds improved the results marginally and the final builds are located at PILON_FASTA_DBG2OLC and PILON_FASTA_MASURCA.

tool	num_seqs	len_seqs	NaNs	N50	N50_num_seqs
polished_dbg2olc	32	4218290	0	189850	6
polished_marsurca	19	4969088	0	1408292	2

Matthew Spriggs
Applied Bioinformatics | S430452
matthew.spriggs.452@cranfield.ac.uk

7. Assembly QC

7.1 The N50 and or ALE score have primarily been used as quantitative measures of assemblies throughout this report. As a final indication of assembly quality the tool Quast was run using 09_QUAST.sh. Quast produces html reports with descriptive statistics and basic plots for the alignments — see QUAST_DBG2OLC.html, QUAST_MASURCA.html.

7.2 The Quast reports contain helpful plots to explore GC content (Figure 5) and a visualisation of the contigs (Figure 6) — shown for the MaSuRCA assembly only.

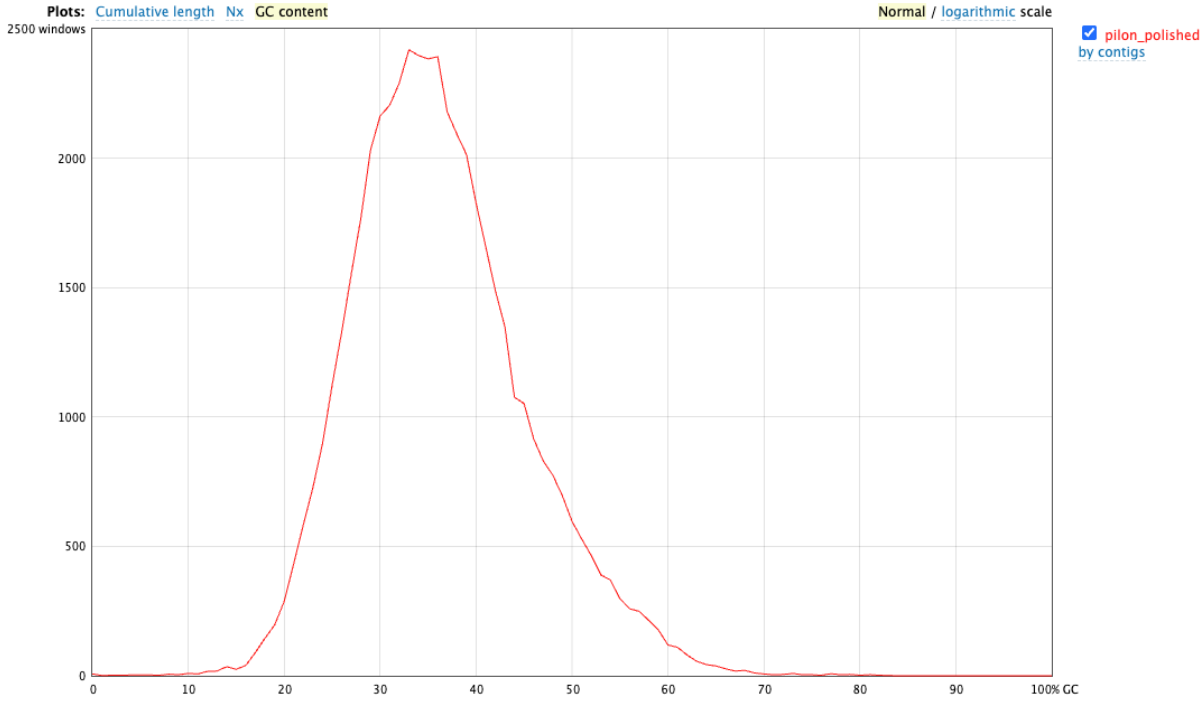

Figure 5: MaSuRCA GC content

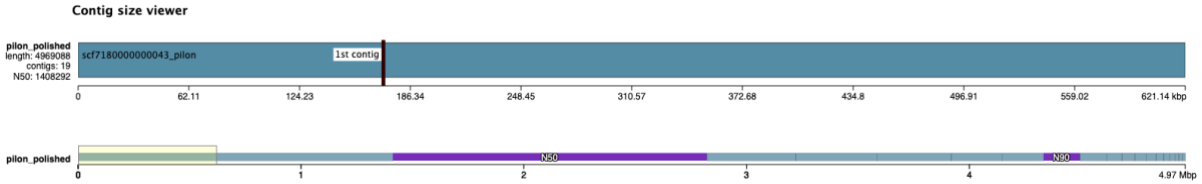

Figure 6: MaSuRCA contig scaffold viewer

7.3 Whilst superficially interesting, the Quast report simply gives me more confidence that the MaSuRCA assembly is the best of the three tools used in this report.

8. Gene Prediction

8.1 Gene prediction was completed using the AUGUSTUS tool on Crescent2. AUGUSTUS is a gene-prediction tool that analyses a DNA sequence and identifies where protein-coding genes are likely to occur (Stanke et al., 2008). The 'species' flag was passed as 'human' after several

nucleotide BLAST searches confirmed human origin. The output of this tool was a GFF file with 26 genes.

8.2 The PILON_FASTA_MASURCA and GFF file were loaded into a custom-built Java tool called geneviz-java [*available at:* https://github.com/ms2206/geneviz-java/]. Using this tool it was possible to get basic statistics on the GFF file (Figure 7). It was also possible to identify candidate CDS sequences that could be used to nucleotide BLAST the NCBI database (Figure 8).

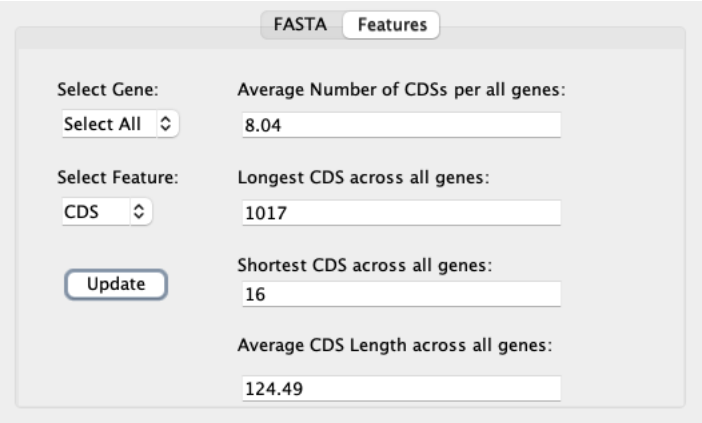

Figure 7: GFF Basic Statistics

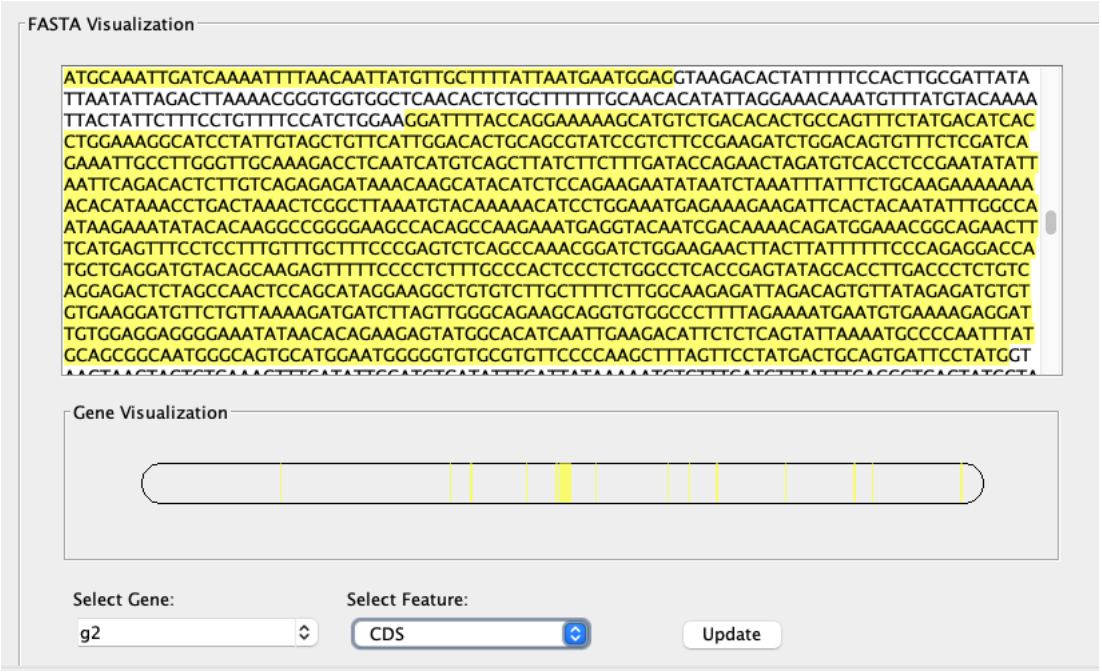

Figure 8: CDS from AUGUSTUS named gene "g2"

8.3 Using a combination of geneviz-java and nucleotide BLAST several CDS regions were investigated further (Table 6).

Table 6: NBCI BLAST Annotation table produced used Geneviz-Java and nBLAST, showing manually curated selection of CDS genes.

Tool	Type	Start	Stop	Strand	Frame	Comments	Query Cover	Per. Ident	SeqID	Chr	Species	NCBI Definition
AUGUSTUS	CDS	114695	115659	+	1	ID=g2.t1.cds;Parent=g2.t1	100	99.79	NM_001346972.2	7	Homo Sapiens	Homo sapiens CDP-L-ribitol pyrophosphorylase A (CRPPA), transcript variant 2, mRNA
AUGUSTUS	CDS	696796	697812	-	0	ID=g5.t1.cds;Parent=g5.t1	100	99.61	OK649233.1	6	Homo Sapiens	Homo sapiens cell line MHC_KAS116 major histocompatibility complex genomic sequence
AUGUSTUS	CDS	743989	744342	-	0	ID=g6.t1.cds;Parent=g6.t1	100	99.72	AC006035.2	7	Homo Sapiens	Homo sapiens BAC clone RP11-196O16 from 7, complete sequence
AUGUSTUS	CDS	32805	33551	+	0	ID=g19.t1.cds;Parent=g19.t1	100	99.87	AC005248.1	7	Homo Sapiens	Homo sapiens BAC clone GS1-83B20 from 7, complete sequence
AUGUSTUS	CDS	1928	2257	-	0	ID=g22.t1.cds;Parent=g22.t1	100	99.09	AC147653.5	7	Pan Troglodytes	Pan troglodytes BAC clone RP43-145M20 from chromosome 7, complete sequence
AUGUSTUS	CDS	48751	48900	+	0	ID=g26.t1.cds;Parent=g26.t1	100	99.64	NM_001101417.4	7	Homo Sapiens	Homo sapiens CDP-L-ribitol pyrophosphorylase A (CRPPA), transcript variant 2, mRNA

8.4 Table 6 indicates that the raw data originates from *Homo sapiens*. It also suggests that the reads likely derive from a region of chromosome 7, with a possible crossover from chromosome 6—although this may reflect translocation events or alignment artefacts. Consequently, the raw data should no longer be considered mitochondrial in origin, but instead represents nuclear genomic sequence. The sequence ID AC147653.5 matches the *Pan troglodytes* (Chimpanzee) genome, indicating this gene is well conserved across humans and chimpanzees.

8.5 The tool OmicsBox was used to further explore gene annotation and GO annotation. The OmicsBox table export file OMICSBOX_TABLE_EXPORT is a challenging to interpret. Largely due to the "black box" nature of the software it is not completely clear how much confidence I can take in the results. The software performed a protein BLAST using Dimond Blast, and Mapping and Annotation using the tools features. One interesting take away from this table is that all proteins returned from the OmicsBox GO annotation were RNA-directed DNA polymerases.

8.6 My ability to interpret these results with confidence is limited.

9. Metabolic Pathways

9.1 Functional KEGG pathway analysis was performed using OmicsBox, and the results are provided in *KEGG.pdf*. As with the GO annotation, confidence in the biological interpretation is limited. The tool identified four pathways within the plant Reactome database, and any further interpretation would be speculative.

Appendix A: File Locations

In-Text Reference	Relative Location
ILLUMINA_SR_READ_1	/assignment/raw_data/HS7_R1.fastq.gz
ILLUMINA_SR_READ_2	/assignment/raw_data/HS7_R2.fastq.gz
PACBIO_READS	/assignment/raw_data/HS7_pacbioData.fastq.gz
MULTIQC_REPORT	/assignment/qc/HS7_R1_multiqc_report/multiqc_report.html
01_QC.sh	/assignment/scripts/01_qc.sh
01_QC.log	/assignment/logs/01_qc.log
02a_KMER_2PASS.sh	/assignment/scripts/02a_kmer_2pass_analysis.sh
00_ERROR_CORRECTION.sh	/assignment/scripts/00_error_correction.sh
00_ERROR_CORRECTION_K17.log	/assignment/logs/00_error_correction_k17.log
COR_ILLUMINA_SR_READ_1	/assignment/kmer/soap_ec/kmer_17/HS7_R1.fastq.gz.cor.pair_1.fq.gz
COR_ILLUMINA_SR_READ_2	/assignment/kmer/soap_ec/kmer_17/HS7_R1.fastq.gz.cor.pair_1.fq.gz
03_SR_SOAP_ASSEMBLY.sh	/assignment/scripts/03_SR_soap_assembly.sh
03_SR_SOAP_ASSEMBLY_K64.log	/assignment/logs/03_SR_soap_assembly_k64.log
04_ALE.sh	/assignment/scripts/04_ale.sh
04_ALE_SUMMARY.log	/assignment/logs/04_ale_summary.log

04_ALE_SUMMARY.csv	/assignment/logs/04_ale_summary.csv
SOAP_ASSEMBLY	/assignment/assembly/soap_denono_2/kmer_64/HS7_R1.fastq.gz.cor/HS7_R1.fastq.gz.cor_pregraph.contig
06_DBG2OLC.sh	/assignment/scripts/06_dbg2olc.sh
UTILS.sh	/assignment/scripts/utils/utils.sh
08_MASURCA.sh	/ assignment/scripts/08_masurca.sh
PILON_FASTA_DBG2OLC	/assignment/pilon/S7_R1.fastq.gz.cor_k64_soap_SR_dbg2olc_hybrid_k64/pilon_polished.fasta
PILON_FASTA_MASURCA	/assignment/pilon/S7_R1.fastq.gz.cor_k64_soap_SR_masuca_hybrid_k64/pilon_polished.fasta
07[A|B]_PILON.sh	/assignment/scripts/07a_pilon.sh | assignment/scripts/07b_pilon.sh
08_QUAST.sh	/assignment/scripts/09_quast.sh
QUAST_DBG2OLC.html	/assignment/assembly_qc/quast/DBG2OLD_quast_report/quast_results/results_2026_01_26_08_12_04/report.html
QUAST_MASURCA.html	/assignment/assembly_qc/quast/MASURCA_quast_report/quast_results/results_2026_01_26_08_11_19/report.html
GFF	/ assignment/annotation/masurca_genes.gff
OMICSBOX_TABLE_EXPORT	/assignment/annotation/omicsBox_blast_mapped_go.xlsx
KEGG.pdf	/assignment/annotation/kegg.pdf

Appendix B: Bibliography

Clark, S. C., Egan, R., Frazier, P. I., & Wang, Z. (2013). ALE: a generic assembly likelihood evaluation framework for assessing the accuracy of genome and metagenome assemblies. *Bioinformatics*, *29*(4), 435–443. https://doi.org/10.1093/BIOINFORMATICS/BTS723

Ewels, P., Magnusson, M., Lundin, S., & Käller, M. (2016). *MultiQC: smmarize analysis results for multiple tools and samples in a single report* (MultiQC/1.14-Foss-2022b). https://github.com/MultiQC/MultiQC/

GitHub - aquaskyline/SOAPdenovo2: Next generation sequencing reads de novo assembler. (n.d.). Retrieved January 30, 2026, from https://github.com/aquaskyline/SOAPdenovo2

Illumina. (n.d.). *Quality Scores | BaseSpace Sequence Hub*. Retrieved January 27, 2026, from https://help.basespace.illumina.com/files-used-by-basespace/quality-scores

Luo, R., Liu, B., Xie, Y., Li, Z., Huang, W., Yuan, J., He, G., Chen, Y., Pan, Q., Liu, Yunjie, Tang, J., Wu, G., Zhang, H., Shi, Y., Liu, Yong, Yu, C., Wang, B., Lu, Y., Han, C., … Wang, J. (2012). SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler. *GigaScience*, *1*(1), 18. https://doi.org/10.1186/2047-217X-1-18

Mapleson, D., Accinelli, G. G., Kettleborough, G., Wright, J., & Clavijo, B. J. (2017). KAT: a K-mer analysis toolkit to quality control NGS datasets and genome assemblies. *Bioinformatics*, *33*(4), 574–576. https://doi.org/10.1093/BIOINFORMATICS/BTW663

Marçais, G., & Kingsford, C. (2011). A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics*, *27*(6), 764–770. https://doi.org/10.1093/BIOINFORMATICS/BTR011

Mohareb, F. (2026a). *Assembly Quality Assessment*. http://mummer.sourceforge.net/manual/

Mohareb, F. (2026b). *Genome size estimation and short reads assembly*.

Simon, A. (2010). *FastQC: A Quality Control Tool for High Throughput Sequence Data*. https://doi.org/10254/464.

Stanke, M., Diekhans, M., Baertsch, R., & Haussler, D. (2008). Using native and syntenically mapped cDNA alignments to improve de novo gene finding. *Bioinformatics*, *24*(5), 637–644. https://doi.org/10.1093/BIOINFORMATICS/BTN013

Yang, X., Chockalingam, S. P., & Aluru, S. (2013). A survey of error-correction methods for next-generation sequencing. *Briefings in Bioinformatics*, *14*(1), 56–66. https://doi.org/10.1093/BIB/BBS015

Ye, C., Hill, C. M., Wu, S., Ruan, J., & Ma, Z. (2016). DBG2OLC: Efficient Assembly of Large Genomes Using Long Erroneous Reads of the Third Generation Sequencing Technologies. *Scientific Reports 2016 6:1*, *6*(1), 31900-. https://doi.org/10.1038/srep31900

Zimin, A. V., Marçais, G., Puiu, D., Roberts, M., Salzberg, S. L., & Yorke, J. A. (2013). The MaSuRCA genome assembler. *Bioinformatics*, *29*(21), 2669. https://doi.org/10.1093/BIOINFORMATICS/BTT476

Matthew Spriggs
Applied Bioinformatics | S430452
matthew.spriggs.452@cranfield.ac.uk

Appendix C: AI Disclaimer

1.1 AI was used to assist the student in this report.

1.2 Throughout the text LLM have been mentioned where the these models have been used to summarize text files and convert into csv files. Typically bespoke python scripts could be used for this which take time and resource.

1.3 AI was used to as a coding parter to troubleshoot the pipeline.