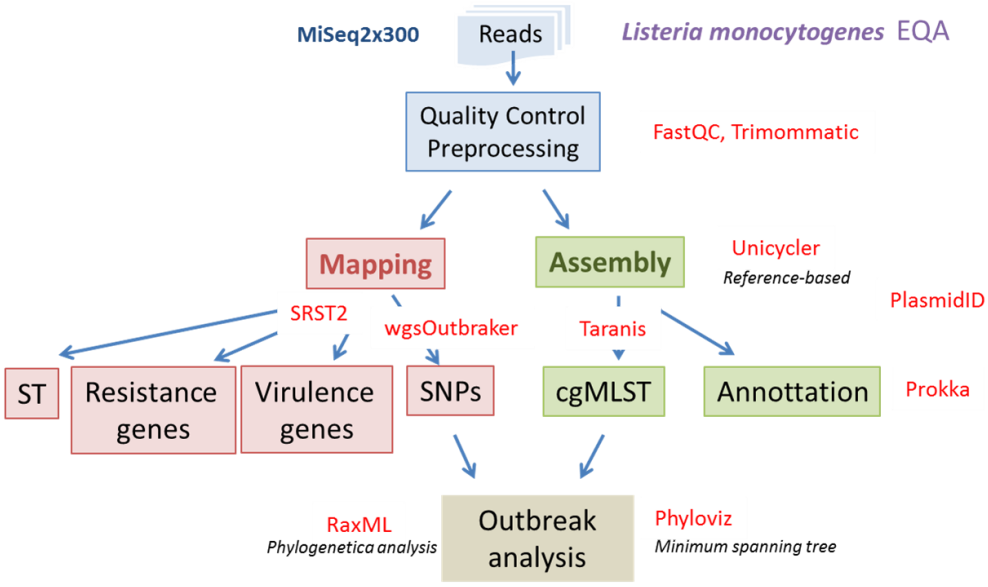
Bacterial WGS training : Exercise 3

Title	**Sequence quality and assambly.**
Training dataset:	
Questions:	• How do I know if my data was correctly sequenced? • How can I improve my data quality? • How do I assamble the reads? • How do I know if my reads were correctly assembled
Objectives:	• Check quality of sequenced data. • Trimm low quality segments and adapters. • Assamble mapped reads.
Time estimation:	1 h 30 min
Key points:	• Analysis of sequence quality. • Assembly.

Introduction

Training summary



Training dataset description

This dataset was used for external quality assessment (EQA-5) scheme for typing of Listeria monocytogenes (*L. monocytogenes*) organised for laboratories providing data to the Food and Waterborne Diseases and Zoonoses Network (FWD-Net) managed by ECDC. Since 2012, the Section for Foodborne Infections at the Statens Serum Institut (SSI) in Denmark has arranged this EQA under a framework contract with ECDC. The EQA-5 contain serotyping and molecular typing-based cluster analysis.

Human listeriosis is a relatively rare but serious zoonotic disease with an EU notification rate of 0.47 cases per 100 000 population in 2016. The number of human listeriosis cases in the EU has increased since 2008, with the highest annual number of deaths since 2009 reported in 2015 at 270.

The objectives of the EQA are to assess the quality and comparability of the typing data reported by public health national reference laboratories participating in FWD-Net. Test isolates for the EQA were selected to cover isolates currently relevant to public health in Europe and represent a broad range of clinically relevant types for invasive listeriosis. Two separate sets of 11 test isolates were selected for serotyping and molecular typing-based cluster analysis. The expected cluster was based on a pre-defined categorisation by the organiser. Twenty-two *L. monocytogenes* test isolates were selected to fulfil the following criteria:

- cover a broad range of the common clinically relevant types for invasive listeriosis
- include closely related isolates
- remain stable during the preliminary test period at the organising laboratory.

The 11 test isolates for cluster analysis were selected to include isolates with different or varying relatedness isolates and different multi locus sequence types.

How do I know if my data was correctly sequenced?

Despite the improvement of sequencing methods, there is no error-free technique. The Phred quality score (Ewing et al., 1998) has been used since the late 90s as a measure of the quality of each sequenced nucleotide. Phred quality scores not only allow us to determine the accuracy of sequencing and of each individual position in an assembled consensus sequence, but it is also used to compare the efficiency of the sequencing methods.

Phred quality scores Q are defined as a property which is logarithmically related to the base-calling error probabilities P . The Phred quality score is the negative ratio of the error probability to the reference level of P = 1 expressed in Decibel (dB):

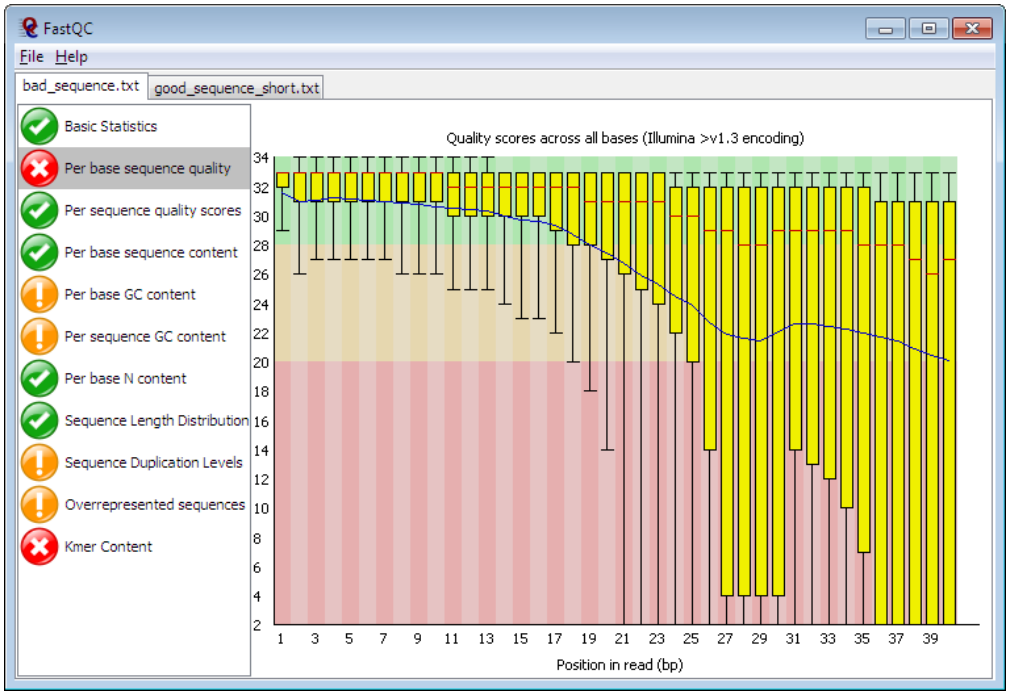
$$Q = -10 \log_{10} \mathrm{P}$$

A correct measuring of the sequencing quality is essential for identifying problems in the sequencing and removal of low-quality sequences or sub sequences. Conversion of typical Phred scores used for quality thresholds into accuracy can be ead in the following table:

Phred score	Error probability	Accuracy
10	1/10	90%
20	1/100	99%
30	1/1000	99.9%

Phred score	Error probability	Accuracy
40	1/10000	99.99%

There are multiple software to read and generate statistics to help with the interpretation of the quality of a sequence. One of the most commonly used methods for this task is FastQC (Andrews, 2010), a java program that run on any system and has both command line and graphic interface.



FastQC aims to provide a simple way to do some quality control checks on raw sequence data coming from high throughput sequencing. It provides a modular set of analyses which you can use to give a quick impression of whether your data has any problems of which you should be aware before doing any further analysis. Here you can compare examples of a good sequencing output and a bad one.

How can I improve my data quality?

Most modern aligners can filter out low quality reads and clip off low quality ends and adapters. In case it has to be done manually, because the sequencing was poor but you still need to use that data or because you want to have more control on the trimming of reads or use a particular method, there are standalone applications that allow you to do it. Trimmomatic (Bolger, Lohse, & Usadel, 2014) is one of the most broadly used along with fastp (Chen S. et al, 2018). Fastp is the one we will use in the training . It is written in C++, has multi-threading support and performs a variety of useful trimming tasks for illumina paired-end and single ended data.

Common trimming includes removal of short reads, and cut off adapters and a number of bases, if below a threshold quality. Modern algorithms also include more complex methods, as the sliding window trimming. This is the method we will use in the exercises, and it allows to trimm a variable number of bases in each read, cutting once the average quality within the window falls below a threshold.

De novo or reference-based assembly?

After preprocessing, the next step is aligning the reads to rebuild the genomic sequence. There are two main ways of doing this:

- Reference-based assembly

  For each of the short reads in the FASTQ file, a corresponding location in the reference sequence is determined. A mapping algorithm will locate a location in the reference sequence that matches the read, while tolerating a certain amount of mismatch to allow subsequence variation detection tath correspond to the actual difference between the reference and de assembled genome.

- *De novo* assembly

  De novo genome assembly consists in taking a collection of short sequencing reads and reconstruct the genome sequence, source of all these fragments.The output of an assembler is decomposed into contigs: contiguous regions of the genome which are resolved, and/or scaffolds: longer sequences formed by reordered and oriented contigs with positional information but without sequence resolution.

Exercise

Preprocessing

As we have seen in the introduction, the first step is to know the quality of our sequences. Those with an unnaceptable quality will be trimmed in order to remove the nucleotides with bad quality to ease future analysis algorithms such assembly. In order to check quality and trim the reads wee have to execute this command:

```
cd
pwd
#Output: /home/alumno
cd wgs/bacterial_wgs_training_dataset/ANALYSIS/
pwd
#Output: /home/alumno/wgs/bacterial_wgs_training_dataset/ANALYSIS/
ls
#Output: 01-handsonlinux
```

Now we only have the folder from the first day of training. And here in `ANALYSIS` folder is where we will run the nextflow. First we activate the nextflow conda environment

```
micromamba deactivate
micromamba activate nextflow
```

An now run the main nextflow command for pre-processing

```
nextflow run ../../bacterial_wgs_training/main.nf \
  --reads '../RAW/DOWNSAMPLED/*_R{1,2}.fastq.gz' \
```

```
    -profile conda \
    --fasta ../REFERENCES/listeria_NC_021827.1_NoPhagues.fna \
    --step preprocessing  \
    --outdir 01-preprocessing
```

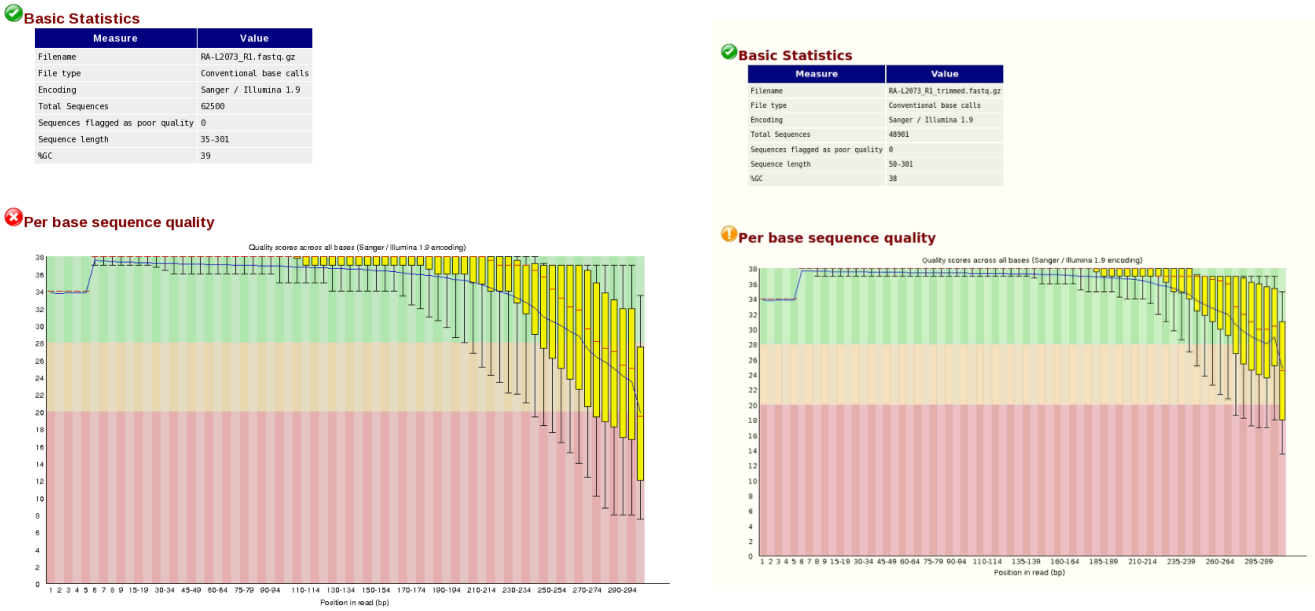
This execution runs internally three programs: FastQC, fastp and MultiQC as follow:

For each sample those command are executed:

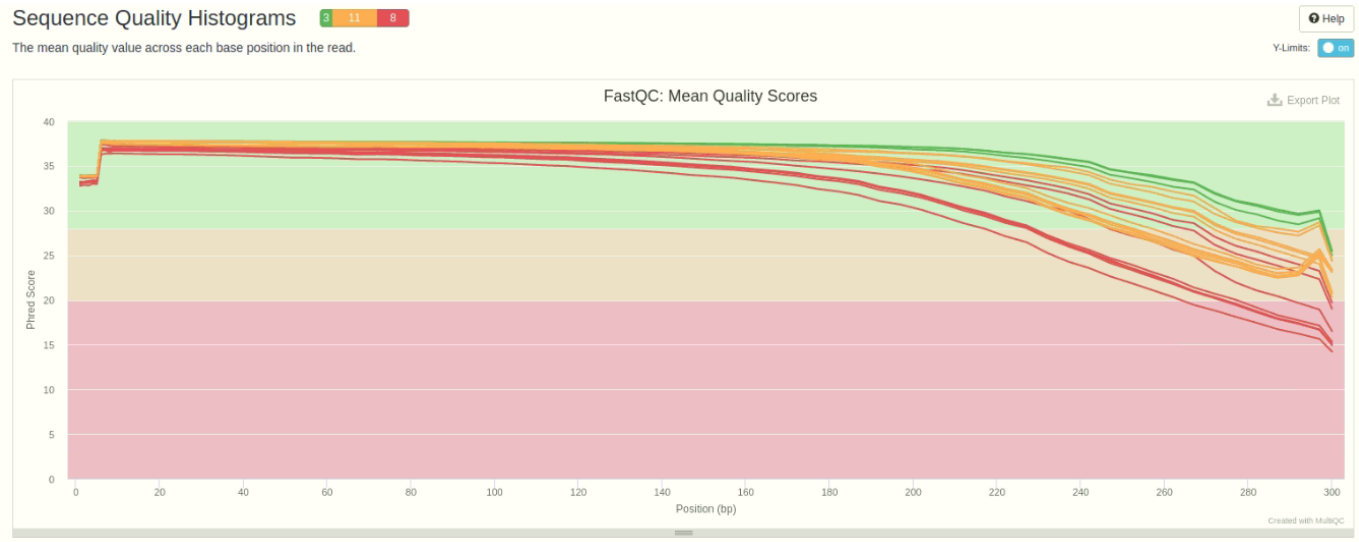
- `fastqc reads_R1.fastq.gz reads_R2.fastq.gz`
    - reads_R1.fastq.gz and reads_R2.fastq.gz are the input Illumina reads which quality is analyzed
- `fastp --in1 reads_R1.fastq.gz --in1 reads_R2.fastq.gz --out1 reads_trimmed_R1.fastq --out2 reads_trimmed_R2.fastq --unpaired1 reads_unpaired_R1.fastq --unpaired2 reads_unpaired_R2.fastq --detect_adapter_for_pe --cut_front --cut_tail --cut_mean_quality 20 --qualified_quality_phred 20 --unqualified_percent_limit 10 --length_required 50 --trim_poly_x --thread 1 --json sample.fastp.json --html sample.fastp.html 2> sample.fastp.log`
    - reads_R1.fastq.gz and reads_R2.fastq.gz are the input Illumina reads which will be trimmed
    - reads_trimmed_R[1|2].fastq reads_fail_R[1|2].fastq
        - trimmed refer to sequences trimmed that passed the quality filter for both R1 and R2
        - fail refer to sequences that did not pass the quality filter
    - detect_adapter_for_pe: Detects automatically adapters for pair-end data
    - cut_front and cut_tail: remove bases with low quality from 5' to tail and 3' to front respectively using a sliding window.
    - cut_mean_quality: mean quality requirement for cut_front and cut_tail. The bases in the sliding window with mean quality below 20 will be cut.
    - qualified_quality_phred: The quality value that a base is qualified. Bases below 20Q Phred are unqualified.
    - unqualified_percent_limit: How many percents of bases are allowed to be unqualified. Maximum 10% of bases can be unqualified.
    - length_required: reads shorter than 50 will be discarded
    - trim_poly_x: enable polyX trimming in 3' ends.
- `fastqc reads_trimmed_R[1|2].fastq reads_fail_R[1|2].fastq`
    - reads_trimmed_R[1|2].fastq reads_fail_R[1|2].fastq are sequences after trimming step which quality will be assesed
- `multiqc RESULTS_DIRECTORY`
    - MultiQC will automatically search for raw and trimmed reads quality results and will compare them in user-friendly graphs

Final results should look like those

Before trimming **After trimming**



Here we can see the quality of the R1 reads before and after trimming



This is the MultiQC output comparing the quality of trimmed and raw reads

Assembly

Reconstruct the source genome is a mandatory step for latter analysis such annotation, comparative analysis and cgMLST. In order to assemble all samples we need to run this command:

First check that you are in the correct folder:

```
pwd
cd
pwd
#Output: /home/alumno
cd wgs/bacterial_wgs_training_dataset/ANALYSIS/
```

```
pwd
#Output: /home/alumno/wgs/bacterial_wgs_training_dataset/ANALYSIS/
ls
#Output: 01-handsonlinux 02-assembly work
```

Now we can run the assembly process

```
nextflow run ../../bacterial_wgs_training/main.nf \
  -resume \
  -profile conda \
  --reads '../RAW/DOWNSAMPLED/*_R{1,2}.fastq.gz' \
  --fasta ../REFERENCES/listeria_NC_021827.1_NoPhagues.fna \
  --gtf ../REFERENCES/listeria_NC_021827.1_NoPhagues.gff \
  --step assembly \
  --outdir 02-assembly
```

This execution runs internally four programs: FastQC, fastp, **Unicycler** , MultiQC and Quast:

Software for preprocessing are executed as noted before. This step includes an aditional assembly process that uses Unicycler to assemble all samples. Nextflow runs Unicycler for each sample as follow:

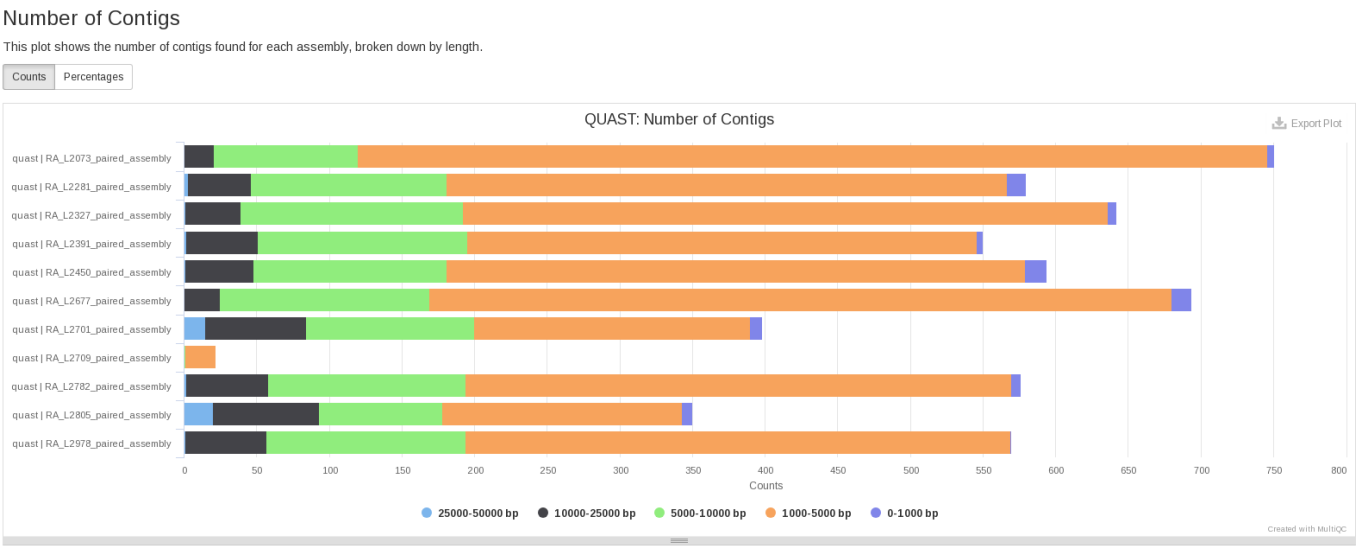
```
unicycler -1 reads_R1.fastq.gz -2 reads_R2.fastq.gz
```

Once assembled, the file containing the contigs (SAMPLE_paired_assembly.fasta) will be checked by Quast, using a reference fasta and gff from *Listeria monocytogenes* strain J1817. Quast was executed as follow:

```
quast.py -R reference.fasta -G reference_genes.gff *_paired_assembly.fasta
```

- R is the fasta file with the reference sequence
- G refers to the gff file for the same sequence
- All sequences are used as input with a wildcard that includes all fasta files with assembled contigs

Final results should look like these



This is the MultiQC output sumarizing the number of contigs assembled for each sample, sorting them by size

QUAST

QUAST is a quality assessment tool for genome assemblies, written by the Center for Algorithmic Biotechnology.

Assembly Statistics

Copy table | Configure Columns | Plot | Showing 11/11 rows and 10/10 columns.

Sample Name	N50 (Kbp)	N75 (Kbp)	L50 (K)	L75 (K)	Largest contig (Kbp)	Length (Mbp)	Misassemblies	Mismatches/100kbp	Indels/100kbp	Genome Fraction
quast | RA_L2978_paired_assembly	6.5Kbp	3.9Kbp	0.1K	258.0K	25.1Kbp	2.7Mbp	2.0	16.71	1.94	90.2%
quast | RA_L2805_paired_assembly	13.2Kbp	6.7Kbp	0.1K	138.0K	42.1Kbp	2.8Mbp	2.0	14.31	1.05	93.5%
quast | RA_L2782_paired_assembly	6.4Kbp	3.8Kbp	0.1K	258.0K	26.5Kbp	2.7Mbp	8.0	329.39	7.43	87.1%
quast | RA_L2709_paired_assembly	1.4Kbp	1.1Kbp	0.0K	14.0K	6.2Kbp	0.0Mbp	0.0	1 204.82	0.00	0.0%
quast | RA_L2701_paired_assembly	10.2Kbp	5.8Kbp	0.1K	168.0K	40.5Kbp	2.8Mbp	4.0	13.49	0.87	92.5%
quast | RA_L2677_paired_assembly	5.1Kbp	3.0Kbp	0.2K	329.0K	21.3Kbp	2.6Mbp	14.0	15.00	1.87	84.9%
quast | RA_L2450_paired_assembly	6.4Kbp	3.6Kbp	0.1K	268.0K	30.3Kbp	2.6Mbp	12.0	15.75	1.45	88.2%
quast | RA_L2391_paired_assembly	7.0Kbp	4.0Kbp	0.1K	249.0K	33.0Kbp	2.7Mbp	1.0	18.78	1.25	88.8%
quast | RA_L2327_paired_assembly	5.9Kbp	3.2Kbp	0.1K	294.0K	26.8Kbp	2.7Mbp	7.0	327.44	7.17	85.5%
quast | RA_L2281_paired_assembly	6.3Kbp	3.6Kbp	0.1K	262.0K	30.9Kbp	2.6Mbp	17.0	341.39	7.27	85.7%
quast | RA_L2073_paired_assembly	4.2Kbp	2.5Kbp	0.2K	373.0K	16.6Kbp	2.4Mbp	23.0	17.00	1.21	81.1%

This is the MultiQC output sumarizing basic assembly statistics for each sample

Worst Median Best — Show heatmap

Genome statistics	RA_L2073_paired_assembly	RA_L2391_paired_assembly	RA_L2677_paired_assembly	RA_L2978_paired_assembly	RA_L2281_paired_assembly	RA_L2450_paired_assembly	RA_L2701_paired_assembly
Genome fraction (%)	81.079	88.828	84.92	90.172	85.733	88.172	92.463
Duplication ratio	1	1	1.001	1.001	1.001	1	1
# genomic features	1736 + 824 part	2113 + 600 part	1881 + 768 part	2157 + 611 part	1992 + 637 part	2073 + 643 part	2368 + 412 part
Largest alignment	16 612	33 033	21 336	25 068	29 638	30 305	40 471
Total aligned length	2 405 510	2 635 297	2 519 300	2 675 166	2 543 440	2 615 874	2 743 222
NGA50	3176	6162	4234	5948	5104	5358	9519
LGA50	267	151	219	153	166	166	96
Misassemblies							
# misassemblies	23	1	14	2	17	12	4
Misassembled contigs length	84 193	9611	45 868	6390	111 490	72 879	37 962
Mismatches							
# mismatches per 100 kbp	17	18.78	15	16.71	341.39	15.75	13.49
# indels per 100 kbp	1.21	1.25	1.87	1.94	7.27	1.45	0.87
# N's per 100 kbp	0	0	0	0	0	0	0
Statistics without reference							
# contigs	748	546	684	569	569	584	392
Largest contig	16 612	33 033	21 336	25 068	30 915	30 305	40 471
Total length	2 440 656	2 676 227	2 562 578	2 714 287	2 629 607	2 618 624	2 787 129
Total length (>= 1000 bp)	2 439 127	2 676 227	2 559 569	2 714 287	2 628 029	2 615 105	2 785 415
Total length (>= 10000 bp)	257 236	739 181	320 638	811 392	700 516	658 319	1 419 641
Total length (>= 50000 bp)	0	0	0	0	0	0	0

Extended report

This is the Quast output sumarizing all detailed assembly statistics, comparing all the assemblies in one table

This is Quast Icarus viewer where contigs are aligned to the reference supplied in order to check how similar the assembled contigs are to the genome from the database.