SHORT READ ASSEMBLY

Quality control:

The quality of raw fastq reads are assessed using FASTQC. The graphical representations generated by FASTQC are evaluated, and sequence trimming is performed to remove any contamination, such as adapter contamination.

```
fastp --detect_adapter_for_pe -i Cow_1.fastq -I Cow_2.fastq -o dm1_trim.fq -O dm2_trim.fq
-q 30 --json=fastp.json --html=fastp.html -w 30

conda activate assembly
```

Before assembly, the genome size is estimated at 17 kmer using frequency-based Jellyfish and GenomeScope. It provides an overall statistics at a particular kmer value of the input data including depth, heterozygosity, repetition and so on.

```
jellyfish count -C -m 17 -s 1000000000 -t 32 ../dm1_trim.fq ../dm2_trim.fq -o reads_17.jf

jellyfish histo -t 32 reads_17.jf > reads_17.histo

conda deactivate

/home/nanobioinfo22/shailesh_nipgr/app/genomescope/genomescope.R reads_17.histo 17 150
genomescope_dm_stats
```

Genome Assembly:

MaSuRCA is whole genome assembly software. It combines the efficiency of the de Bruijn graph and Overlap-Layout-Consensus (OLC) approaches. MaSuRCA can assemble data sets containing only short reads from Illumina sequencing or a mixture of short reads and long reads.

```
masurca -i dm1_trim.fq,dm2_trim.fq -t 32
```

```
Verifying PATHS...
jellyfish OK
runCA OK
createSuperReadsForDirectory.perl OK
creating script file for the actions...done.
execute assemble.sh to run assembly
[Wed May 15 16:20:24 IST 2024] Processing pe library reads
[Wed May 15 16:20:33 IST 2024] Average PE read length 148
[Wed May 15 16:20:33 IST 2024] Using kmer size of 99 for the graph
[Wed May 15 16:20:33 IST 2024] MIN_Q_CHAR: 33
[Wed May 15 16:20:33 IST 2024] Creating mer database for Quorum
[Wed May 15 16:20:46 IST 2024] Error correct PE
[Wed May 15 16:21:20 IST 2024] Estimating genome size
[Wed May 15 16:21:34 IST 2024] Estimated genome size: 106623927
[Wed May 15 16:21:34 IST 2024] Creating k-unitigs with k=99
[Wed May 15 16:22:28 IST 2024] Computing super reads from PE
[Wed May 15 16:23:21 IST 2024] Using linking mates
[Wed May 15 16:23:22 IST 2024] Celera Assembler
[Wed May 15 16:27:37 IST 2024] Overlap/unitig success
[Wed May 15 16:27:37 IST 2024] Recomputing A-stat for super-reads
[Wed May 15 16:28:25 IST 2024] Filtering overlaps
[Wed May 15 16:31:13 IST 2024] Recomputing A-stat for super-reads
[Wed May 15 16:34:55 IST 2024] CA success
[Wed May 15 16:34:55 IST 2024] Gap closing
[Wed May 15 16:44:00 IST 2024] Gap close success
[Wed May 15 16:44:00 IST 2024] Removing redundant scaffolds
[Wed May 15 16:44:29 IST 2024] Assembly complete, primary scaffold sequences are in CA/primary.genome.scf.fasta
[Wed May 15 16:44:29 IST 2024] redundant or haplotype variant scaffold sequences are in CA/alternative.genome.scf.fasta
[Wed May 15 16:44:29 IST 2024] All done, final assembly is in CA/primary.genome.scf.fasta
[Wed May 15 16:44:29 IST 2024] Final stats for CA/primary.genome.scf.fasta
N50 1322
Sequence 69742932
Average 1046.33
E-size 1715.06
Count 66655
```

Creates several files and outputs. The final assembled genome is present in the CA folder named “scaffolds.ref.fa”. The assembled genome is further scaffolded using RagTag. Homology-based assembly

scaffolding is an approach of ordering and orienting the draft assembly (query) sequences into longer sequences by comparing against the closely related genome.

```
ragtag.py scaffold <reference.fa> scaffold.fasta
```

The scaffold.fasta generated from the above step is then utilized to close the gaps emerging during the scaffolding process via GapCloser, further improving the overall quality. Make sure to change the path of the input fq reads in the example.config file.

```
GapCloser -a scaffold.fasta -b <example.config> -o Draft_genome.fa -t 224 -l 150

conda deactivate
```

In order to describe the completeness and contiguity of a genome assembly, several summary statistics and in-silico validations are performed. Quast i.e Quality Assessment Tool for Genome Assemblies is one such tool for genome assembly evaluation.

```
quast.py Draft_genome.fa -o DG_evalulate -t 224

OR

quast.py Draft_genome.fa -o DG_evalulate -r <ref.fa> -g ref.gff -t 224
```

Assembly	Draft_genome
# contigs (>= 0 bp)	2299
# contigs (>= 1000 bp)	1030
# contigs (>= 5000 bp)	92
# contigs (>= 10000 bp)	14
# contigs (>= 25000 bp)	7
# contigs (>= 50000 bp)	7
Total length (>= 0 bp)	69648141
Total length (>= 1000 bp)	68768727
Total length (>= 5000 bp)	66946009
Total length (>= 10000 bp)	66416173
Total length (>= 25000 bp)	66329781
Total length (>= 50000 bp)	66329781
# contigs	2298
Largest contig	17846273
Total length	69647827
GC (%)	41.01
N50	15187024
N90	14549498
auN	14355760.0
L50	3
L90	4
# N's per 100 kbp	1867.20

BUSCO (Benchmarking Universal Single-Copy Orthologs) is yet another correctness measure that provides measures for quantitative assessment of genome assembly based on evolutionarily informed expectations of gene content from near-universal single-copy orthologs.

```
busco -f -i Draft_genome.fa -m genome -l diptera_odb10 -o output_busco -c 124
```

```
|Results from dataset diptera_odb10                                                    |
|C:34.4%[S:34.2%,D:0.2%],F:21.1%,M:44.5%,n:3285,E:6.8%                                 |
|1129    Complete BUSCOs (C)    (of which 77 contain internal stop codons)             |
|1123    Complete and single-copy BUSCOs (S)                                           |
|6    Complete and duplicated BUSCOs (D)                                               |
|693    Fragmented BUSCOs (F)                                                         |
|1463    Missing BUSCOs (M)                                                           |
|3285    Total BUSCO groups searched                                                  |
```