Whole genome assembly workshop

Exercise 1. Estimate genome size by kmer distribution of Illumina sequencing data

1. Using Putty (Windows) or Terminal (Mac) to connect to your assigned computer.
   Create a directory /workdir/myUserID (replace myUserID with you BioHPC ID), copy the fastq.gz file to the working directory.

```
mkdir /workdir/myUserID

cd /workdir/myUserID

cp /shared_data/assembly_workshop_2018/*.fastq.gz  ./
```

2. Run ErrorCorrectReads.pl from ALLPATHS-LG package to get the genome size estimate. The tool was developed for Illumina read error correction. From the tools we have tested, it is the most accurate tool for genome size estimation.

```
export PATH=/programs/allpathslg/bin:$PATH

ErrorCorrectReads.pl PAIRED_READS_A_IN=R1.fastq.gz PAIRED_READS_B_IN=R2.fastq.gz
KEEP_KMER_SPECTRA=1  PHRED_ENCODING=33 PLOIDY=1  READS_OUT=correct_out >&
report.log &
```

- PLOIDY=1: it is a haploid bacteria genome. Use 2 for diploid genome.
- PHRED_ENCODING=33: Always use 33, unless your data set is extremely old.
- KEEP_KMER_SPECTRA=1: Output kmer size distribution
- From the output, we are only interested in two files:
  - report.log. There is a line that tells you the genome size:
    3361319 estimated genome size in bases.
  - correct_out.fastq.kspec/frag_reads_edit.24mer.kspec
    Plot the first two columns in Excel or R, which gives you the Kmer size distribution.

Exercise 2. Assemble a small genome with de bruijn graph.

It is a bacterial genome. The files should have been copied to the /workdir/myUserID from last step.

```
cd /workdir/myUserID

ls SRR1982238_*.fastq.gz
```

3. Trim low quality data and adapters from the two fastq files. (the following commands are in a single line)

```
java -jar /programs/trimmomatic/trimmomatic-0.36.jar PE -phred33 SRR1982238_1.fastq.gz SRR1982238_2.fastq.gz r1.fastq u1.fastq r2.fastq u2.fastq ILLUMINACLIP:/programs/trimmomatic/adapters/TruSeq3-PE-2.fa:2:30:10 LEADING:10 TRAILING:10 SLIDINGWINDOW:4:15 MINLEN:150
```

There will be 4 new files created after this step, r1.fastq, r2.fastq, u1.fastq, u2.fastq. We will use the r1.fastq and r2.fastq for next step.

4. Run assembly with SOAPdenovo.

   Make the following config.txt file with a text editor:

```
#maximal read length

max_rd_len=250

[LIB]

#average insert size

avg_ins=500

#if sequence needs to be reversed

reverse_seq=0

#in which part(s) the reads are used

asm_flags=3

#in which order the reads are used while scaffolding

rank=1

# cutoff of pair number for a reliable connection (at least 3 for short insert size)

pair_num_cutoff=3

#minimum aligned length to contigs for a reliable read location (at least 32 for short
insert size)

map_len=45

#a pair of fastq file, read 1 file should always be followed by read 2 file

q1=r1.fastq

q2=r2.fastq
```

Note: If you have multiple libraries, you can repeat the [LIB] section many times in this file.

Run soapdenovo with this command:

```
/programs/SOAPdenovo2/SOAPdenovo-127mer all -s config.txt -K 127 -R -o soap-assembly
```

Among the files produced:

1. soap-assembly.scafSeq: Fasta file of the scaffold. This is the file you will use for annotation.
2. soap-assembly.contig: Fasta file of the contig.
3. soap-assembly. scafStatistics: statistics report of the assembly.

Examine the soap-assembly. scafStatistics to evaluate the assembly.

4. Run assembly with ABySS.

```
export PATH=/programs/abyss-1.5.2-128/bin:$PATH

/programs/abyss-1.5.2-128/bin/abyss-pe  k=128 name=abyss_assembly lib='pe1'
pe1='r1.fastq r2.fastq'
```

This step will produce a contig fasta file: abyss_assembly-contigs.fa

Run the quast.py tool to get statistic report of the file:

```
/programs/quast-2.2/quast.py abyss_assembly-contigs.fa
```

The report is in the file “quast_results/latest/report.txt”

5. Run exercise 1 steps on the file r1.fastq and r2.fastq to estimate genome size, and compare with the assembled genome.