PacBio Assembly Exercises

Starting Note: Please do not copy and paste the commands. Characters in this document may not be copied correctly. Please type the commands and use **<tab> complete** for commands, directories and long names.

Loading Modules:

First do	**`module load bioinfo-tools`**	and then
miniasm:	**`module load miniasm`**	
minimap:	**`module load minimap`**	
Canu:	**`module load Canu/1.3`**	
Falcon:	**`module load FALCON-integrate/20161112`**	
SMRT:	**`module load SMRT/2.3.0`**	
Quast:	**`module load quast/3.2`**	

E. coli Assembly

1. Copy the exercise data to your working directory:

   ```
   cp –vr /proj/g2016024/nobackup/PacBio_assembly .
   ```

   Use miniasm to assemble the **`p6_25x_ecoli.fastq.gz`**.

   ```
   minimap/minimap -Sw5 -L100 -m0 -t8
   p6_25x_ecoli.fastq.gz p6_25x_ecoli.fastq.gz |
   gzip -1 > minimap_ecoli_reads.paf.gz;
   miniasm/miniasm -f p6_25x_ecoli.fastq.gz
   minimap_ecoli_reads.paf.gz >
   miniasm_ecoli_contigs.gfa;
   awk ‘/^S/{print “>”++seq”\n”$3}’
   miniasm_ecoli_contigs.gfa > contigs.fasta
   ```

2. Use Canu to assemble the **`p6_25x_ecoli.fastq.gz`** (Assembly takes approximately 20 mins, so write the falcon config for the next exercise while you wait).

   ```
   canu -p canu_ecoli –d canu_ecoli genomeSize=46m
   maxThreads=8 useGrid=false -pacbio-raw
   p6_25x_ecoli.fastq.gz
   ```

3. Make a directory for your Falcon assembly.
   Create a Falcon config file and then run Falcon (approx. 5 mins).
   See https://github.com/pb-cdunn/FALCON-

examples/blob/master/run/ecoli/fc_run.cfg for an example cfg.

```
mkdir Falcon_Ecoli_assembly;
seqtk seq -l 5000 -A p6_25x_ecoli.fastq.gz >
p6_25x_ecoli.fasta;
readlink -f *.fasta > input.fofn;
<write falcon config file>;
fc_run.py falcon.cfg
```

4. Compare each assembly to the reference using Quast

```
quast.py -R ecoli_reference.fasta <draft_assembly1>
<draft_assembly2> …
```

5. Make dot plots to compare assembly structure using Gepard. Gepard is located in **/proj/g2016024/nobackup/Tools/Gepard-1.40.jar**.

```
GHOME=/proj/g2016024/nobackup/Tools;
java -cp $GHOME/Gepard-1.40.jar
org.gepard.client.cmdline.CommandLine -seq1 <fasta1>
-seq2 <fasta2> -outfile <comparison>.png -matrix
$GHOME/matrices/edna.mat
```

BAC Assembly

Advanced usage of bax/bas.h5 files (pacbio's native format). The full tutorial can be found here: https://github.com/PacificBiosciences/Bioinformatics-Training/wiki/HGAP-Whitelisting-Tutorial. Until PacBio switches to BAM format, it is important to know how to use the bax/bas.h5 files as they contain the pulsefield information necessary for polishing, which is not present in the fastq files.

6. The aim here is to assemble a BAC using HGAP. The first step is to filter out E. coli contamination using **blasr** (pacbio's read alignment tool). First complete loading of the **SMRT/2.3.0** environment and then run **blasr**.

```
source $SMRT_SETUP_SCRIPT;
blasr
m120729_040044_42134_c100384402550000001523033010171
256_s1_p0.bas.h5 ecoli_reference.fasta -bestn 1 -
header > ecoli.align;
```

7. The alignment shows there are some reads that match to E. coli. The whitelist.py script is used to find the names of reads that map to E.coli and print out the read names of those not mapped to E. coli. This script is specific to this exercise, and would need to be modified for general use with bax.h5 files (the current format). Use the whitelist.py script to create a list of reads to keep for HGAP.

```
python whitelist.py ecoli.align 81740 >
whitelist.txt
```

8. Edit the HGAP_protocol.xml file to include the whitelist (add the part in red to the file), and change the genome size to 200kb (the size of the BAC).

```
<moduleStage name="filtering" editable="true">
  <module label="PreAssembler Filter v1" id="P_Filter"
editableInJob="true">
    <param>...</param>

    <param name="whiteList" label="Read IDs to whitelist">
      <value>/path/to/whitelist.txt</value>
      </param>

  </module>
  <module label="PreAssemblerSFilter Reports v1" id="P_FilterReports"
editableInJob="false"/>
</moduleStage>

....

<param name="genomeSize" label="Genome Size (bp)">
      <title>Approximate genome size in base pairs</title>
      <value>200000</value>
      <input type="text"/>
      <rule type="digits" required="true" min="1.0" message="Must be a
value between 1 and 10000000" max="1.0E7"/>
    </param>
```

9. Create the input file and run HGAP. This assembly takes longer than 1 hour. Once HGAP looks like it's running without errors, stop the program using **ctrl + c** and look at the output provided for you.

```
readlink -f
m120729_040044_42134_c100384402550000001523033010171
256_s1_p0.bas.h5 > input.fofn;
fofnToSmrtpipeInput.py input.fofn > input.xml;
smrtpipe.py --params=HGAP_protocol.xml xml:input.xml
```

Extra

10. If the assembly from Falcon isn't in 1 contig, modify parameters until you find a combination that works to make a single contig.