Class 5 – Genome assembly

Goal

- Assemble short reads illumina data into a genomic assembly using Spades Assembler
- Assess the quality of genome assemblies with Quast.
- Generate multiple sample report using a bunch of pre-assembled data as input and visualize the summary report.

roadmap

Execute the following command to copy files for today's exercises to your workshop home directory:

```
> Note: Make sure you change 'username' in the commands below to your 'uniqname'.

wd

#or

cd /scratch/epid582w22_class_root/epid582w22_class/username


> Note: Copy files for today's exercise in your home directory.

cp -r /scratch/epid582w22_class_root/epid582w22_class/shared_data/data/class5 ./
```

Genome Assembly using Spades Pipeline

alt tag

There are a wide range of tools available for the assembly of microbial genomes. These assemblers fall in to two general algorithmic categories, which you can learn more about here. In the end, most assemblers will perform well on microbial genomes, unless there is unusually high GC-content or an over-abundance of repetitive sequences, both of which make accurate assembly difficult.

Here we will use the Spades assembler with default parameters. Because genome assembly is a computationally intensive process, we will submit our assembly jobs to the cluster, and move ahead with some pre-assembled genomes, while your assemblies are running.

spades

> ***i. Create directory to hold your assembly output.***

Create a new directory for the spades output in your class_5 folder

```
> Note: Make sure you change 'username' in the below command with your 'uniqname'.

cd /scratch/epid582w22_class_root/epid582w22_class/username/class5

> We will create a new directory in class_5 to save genome assembly results:

mkdir MSSA_SRR5244781_assembly_result
```

Now, we will use a genome assembly tool called Spades for assembling the reads.

> ***ii. Test out Spades to make sure it's in your path***

Lets load spades module from the Great lakes provided Bioinformatics Modules.

```
> check if spades is working.

module load Bioinformatics
module load spades

spades -h
```

> ***iii. Submit a cluster job to assemble***

Since it takes a huge amount of memory and time to assemble genomes using spades, we will run a slurm script on great lakes cluster for this step.

Now, open the spades.sbat file residing in the class_5 folder with nano and add the following spades command to the bottom of the file. Replace the EMAIL_ADDRESS in spades.sbat file with your actual email-address. This will make sure that whenever the job starts, aborts or ends, you will get an email notification.

```
> Open the spades.sbat file using nano:

nano spades.sbat

> Now replace the EMAIL_ADDRESS in spades.sbat file with your actual email-address.
This will make sure that whenever the job starts, aborts or ends, you will get an
email notification.

> Copy and paste the below command to the bottom of spades.sbat file.

spades -1 forward_paired.fq.gz -2 reverse_paired.fq.gz -o
MSSA_SRR5244781_assembly_result/ --careful
```

> ***iv. Submit your job to the cluster with sbatch***

```
sbatch spades.sbat
```

> ***v. Verify that your job is in the queue with the squeue command***

```
squeue -u username
```

Assembly evaluation using QUAST

The output of an assembler is a set of contigs (contiguous sequences), that are composed of the short reads that we fed in. Once we have an assembly we want to evaluate how good it is. This is somewhat qualitative, but there are some standard metrics that people use to quantify the quality of their assembly. Useful metrics include: i) number of contigs (the fewer the better), ii) N50 (the minimum contig size that at least 50% of your assembly belongs, the bigger the better). In general you want your assembly to be less than 200 contigs and have an N50 greater than 50 Kb, although these numbers are highly dependent on the properties of the assembled genome.

To evaluate some example assemblies we will use the tool quast. Quast produces a series of metrics describing the quality of your genome assemblies.

spades

> ***i. Run quast on a set of previously generated assemblies***

Now to check the example assemblies residing in your class5 folder, run the below quast command. Make sure you are in class5 folder in your home directory using 'pwd'

```
quast.py -o quast SRR5244781_contigs.fasta SRR5244821_contigs.fasta
```

The command above will generate a report file in /scratch/epid582w22_class_root/epid582w22_class/username/class5/quast

> ***ii. Explore quast output***

QUAST creates output in different formats such as html, pdf and text. Now lets check the report.txt file residing in quast folder for assembly statistics. Open report.txt using nano.

```
less quast/report.txt
```

Check the difference between the different assembly statistics. Also check the different types of report it generated.

Generating multiple sample reports using multiqc

alt tag

Let's imagine a real-life scenario where you are working on a project which requires you to analyze and process hundreds of samples. Having a few samples with extremely bad quality is very commonplace. Including these bad samples into your analysis without adjusting their quality threshold can have a profound effect on downstream analysis and interpretations.

- Question: How will you find those bad apples?

In the previous class, we learned how to assess and control the quality of samples as well as screen for contaminants. But the problem with such tools or any other tools is, they work on per-sample basis and produce only single report/logs per sample. Therefore, it becomes cumbersome to dig through each sample's reports and make appropriate quality control calls.

Thankfully, there is a tool called multiqc which parses the results directory containing output from various tools, reads the log report created by those tools (ex: FastQC, FastqScreen, Quast), aggregates them and creates a single report summarizing all of these results so that you have everything in one place. This helps greatly in identifying the outliers and removing or reanalysizing it individually.

Lets take a look at one such mutiqc report that was generated using FastQC results on *C. difficile* samples.

Download the html report Cdiff_multiqc_report.html from your class_5 folder.

```
#Note: Make sure you change 'username' in the below command to your 'uniqname'.

scp username@greatlakes-xfer.arc-
ts.umich.edu:/scratch/epid582w22_class_root/epid582w22_class/username/class5/Cdiff_multi
 ~/Desktop/
```

- Question: Open this report in a browser and try to find the outlier sample/s
- Question: What is the most important parameter to look for while identifying contamination or bad samples?
- Question: What is the overall quality of data?

Lets run multiqc on one such directory where we ran and stored FastQC, FastQ Screen and Quast reports.

if you are not in class_5 folder, navigate to it and change directory to multiqc_analysis

```

cd /scratch/epid582w22_class_root/epid582w22_class/username/class5/

cd multiqc_analysis

multiqc -h

#Run multiqc on sample reports

multiqc ./ --force --filename class5_multiqc

#Check if class5_multiqc.html report was generated

ls -la class5_multiqc.html

#transfer this multiqc report - class5_multiqc.html to your local system and open it
in a browser for visual inspection

scp username@greatlakes-xfer.arc-
ts.umich.edu:/scratch/epid582w22_class_root/epid582w22_class/username/class5/multiqc_ana
 /path-to-local-directory/
```

The report contains the Assembly, Fastq Screen and FastQC report for a mixture of 51 organisms' sequence data. Sample names for Assembly statistics ends with "l500_contigs".

- Question: Which two sample's genome length i.e column Length (Mbp) stand out from all the other genome lengths and what is the reason (hint – check their GC % and their FastQ Screen result)?
- Question: Which sample has the worst N50 value? What do you think must be the reason (hint - check out the general stats)?
- Question: Which sample has the second worst N50 value? Is it the same or a different issue from the worst sample (hint - check out general stats and then the fastQC results)?