For your 3rd homework exercise (due 14 Mar @ 12:45), please:

- Read the following documentation:
  - Canu: https://canu.readthedocs.io/en/stable/quick-start.html#assembling-pacbio-or-nanopore-data
  - SPAdes: https://github.com/ablab/spades?tab=readme-ov-file#sec3.2
  - QUAST: http://quast.sourceforge.net/docs/manual.html#sec2.1
  - Mummer: http://mummer.sourceforge.net/manual/#mummerplot
- Open a new issue on Github, and give it the title "Homework_3".
- Using VScode, develop a BASH script that uses at least 24 Gb of memory (SBATCH --mem) and 6 cores (SBATCH --cpus-per-task) to perform the following operations:
  - Sets up a variable at the top of the body of the script for an output directory with the name /work/gene8940/your_myid/homework_3 into which you will store results;
  - Using PacBio CLR data in /work/gene8940/instructor_data, Assembles the *E. coli* MG1655 genome with Canu (canu/2.2-GCCcore-11.2.0) as follows (NOTE: the command below is a single command split over multiple lines of code):

```
canu -p <canu_output_prefix> -d <canu_output_dir> genomeSize=4.8m useGrid=false -pacbio-raw \
<pacbio.fq.gz>
```

  - Using Illumina paired end short read data in /work/gene8940/instructor_data, assembles the *E. coli* MG1655 genome using with SPAdes (SPAdes/3.15.5-GCC-11.3.0) as follows (NOTE: the command below is a single command split over multiple lines of code):

```
spades.py -t <num_cores> -k 21,33,55,77 --isolate --memory <gb_memory> --pe1-1 \
<illumina_read1.fq.gz> --pe1-2 <illumina_read2.fq.gz> -o <spades_output_dir>
```

  - NOTE: Be sure to match the <num_cores> in your SPAdes assembly to the `#SBATCH --cpus-per-task` SLURM value and the <gb_memory> to the `#SBATCH --mem` SLURM value.
  - Runs QUAST (QUAST/5.2.0-foss-2022a) to generate assembly quality assessment statistics (number of contigs, N50 and L50) for the PacBio/Canu contigs and Illumina/Spades scaffolds using the RefSeq assembly as a reference:

```
quast.py -o <quast_output_dir> -t <num_cores> -r <MG1655_RefSeq.fasta> \
<canu_output_dir/canu_output_prefix.contigs.fasta> <spades_output_dir/scaffolds.fasta>
```

  - NOTE: Be sure to match the <num_cores> in your QUAST analysis to the `#SBATCH --cpus-per-task` SLURM value.
  - Generates mummerplots (MUMmer/4.0.0rc1-GCCcore-11.3.0) for the PacBio/Canu and Illumina/SPAdes assemblies versus the unmasked release 58 version of the *E. coli* MG1655 genome sequence (*Chromosome.fa.gz) from Ensembl as follows (NOTE: The mummer analysis requires 3 distinct commands -- nucmer, delta-filter, and mummerplot -- to generate the mummerplot. The first two commands below are on a single line and the third command is a single command split over multiple lines of code. Also, fasta sequences used as input for nucmer must be unzipped.):

```
nucmer -t <num_cores> <MG1655_RefSeq_genome.fa> <assembly.fasta> -p <mummer_output_prefix>
delta-filter -1 <mummer_output_prefix.delta>  >  <mummer_output_prefix.1delta>
mummerplot --size large -layout --color -f --png <mummer_output_prefix.1delta> -p \
<mummer_output_prefix>
```

- Be sure that you script includes a SLURM header in this script that sends stderr/stdout to log files & code your script so that output files are sent to a "scratch" directory (in /work/gene8940/yourusername).
- Commit this script into your local git repository, push the script to github, pull the script down on the teaching cluster, and submit the script to the teaching cluster scheduler. When the script has executed successfully, post a comment with the following information:

1) Report the N50 and L50 for both assemblies and state what these values mean;
2) Upload .pngs of mummerplots for both assemblies and describe what these plots show;
3) The URL to the location of the script on GitHub;
4) The git SHA revision of the script used for this analysis.

Good luck and please post a comment in Slack or reach out to me/Oliver if you have any questions.