Snakemake hands-on exercises

Preparation

Log in to Euler. Set the default software stack on Euler to new. Edit .software_stack_default and place “new” as the content of the file, to get such results in your bash session:

```
[michalo@eu-login-08 ~]$ set_software_stack.sh -i
The global default is set to: old
You have set the new software stack as your personal default,
which supersedes the global default
[michalo@eu-login-08 ~]$ cat .software_stack_default
new
```

Use the module to have Snakemake available.

```
module load gcc/8.2.0 python/3.11.2
which snakemake
snakemake –version
```

Simple exercise

Get the simple workflow that runs without a specific software stack into your Euler scratch space and run it there:

```
cd /cluster/scratch/username
git clone https://github.com/michalogit/snakemaketax.git
cd snakemaketax
```

Have a look around the workflow. Follow the instructions at
https://github.com/michalogit/snakemaketax
and create the DAG graph output

Extra task:

- Convert locally the graph to a PDF, using the dot utility

RNA-seq workflow exercise

The goal is to run the full RNA-sequencing primary analysis using the template of Snakemake workflow from github:

```
git clone https://github.com/michalogit/snake_hisat
```

Prepare the running folder in the scratch:

```
cd /cluster/scratch/yourusername
mkdir myRNAseq
mkdir myRNAseq/data
cd /cluster/scratch/yourusername/myRNAseq/data
cp /cluster/scratch/michalo/data/*fastq.gz .
```

Copy the genome with hisat2 index:

```
cd /cluster/scratch/yourusername
mkdir grch38
cd /cluster/scratch/yourusername/grch38
cp /cluster/scratch/michalo/grch38/* .
```

Get current genome annotation GFF file

```
wget https://ftp.ensembl.org/pub/release-
110/gff3/homo_sapiens/Homo_sapiens.GRCh38.110.chr.gff3.gz

gunzip Homo_sapiens.GRCh38.110.chr.gff3.gz
```

Copy the Snakefile and other required files from the github repository:

```
cd /cluster/scratch/yourusername
cp snake_hisat/Snakefile myRNAseq/
cp snake_hisat/cluster.json myRNAseq/
cp snake_hisat/adapters.fa myRNAseq/
```

Edit the Snakefile:

- Remove the rules for splicing analysis: cufflinks and stringtie
- Remove the marker files for cufflinks and stringtie from “rule all”
- Adjust the genome paths to /cluster/scratch/yourusername/genomes

Execute the workflow:

- Create the screen session, take note of the login node name
- Activate snakemake using the python module call in the screen session

- Do the dry run with: `snakemake -np`
- Fix the Snakefile syntax if needed
- Do the proper run of Snakemake, configured in cluster.json

```
snakemake -p -j 999 --cluster-config cluster.json --cluster
"sbatch --time {cluster.time} -n 1 --cpus-per-
task={cluster.n}"
```

Go out of the screen session, check at times with "squeue" how Slurm runs the jobs
Go in the screen session, check if Snakemake is running without a crash.

Extra task:
Get the counts.csv into R and find most differentially expressed genes by fold change.

Additional exercises

1) Change the configuration of Snakemake from cluster.json into "Simple Slurm" https://github.com/jdblischak/smk-simple-slurm
using the instructions in github Readme.

2) Change the rules in the Snakefile to those available in Snakefile_containers that use Galaxy software stack with Singularity. See also: https://scicomp.ethz.ch/wiki/Galaxy_Depot_Software_Stack

- Run the dry run and a proper run as above.
- Check the difference in rule parameters
- Find where the containers have been downloaded.
- Check if the results in counts.csv are the same.