Setting up and Running SPEAQeasy

July 11, 2020

Contents

1 Setting up SPEAQeasy

Please note that running SPEAQeasy is an optional step, and the following guide exists to allow one to reproduce our full RNA-seq analysis workflow. Those interested only in utilizing SPEAQeasy outputs in downstream analysis may use the outputs we provide and skip this section.

To set up SPEAQeasy, clone its repository, choose a method for installing dependencies, and identify which "main" script is appropriate for your setup. A "quick start" guide for these steps is here.

```
# Clone SPEAQeasy
git clone git@github.com:LieberInstitute/SPEAQeasy.git

# Enter the repository
cd SPEAQeasy

# Install dependencies: note that this exact command is appropriate only
# for those who cannot use or desire not to use docker to manage software
# dependencies! Replace 'local' with 'docker' if you plan to use docker. See
# the SPEAQeasy documentation linked above for more details.
bash install_software.sh "local"
```

2 Modify the main script

For the purposes of the below example, we will consider a user who wishes to run SPEAQeasy on a computing cluster using SLURM, and managing dependencies with docker. However, the process is nearly identical regardless of the choice of "main" script. Below we examine the original content of `run_pipeline_slurm.sh`, our "main" script, which we will modify as appropriate for the particular example data set.

```
$ORIG_DIR/Software/nextflow main.nf \
    --sample "single" \
    --reference "hg19" \
    --strand "unstranded" \
```

```
    --small_test \
    --annotation "$ORIG_DIR/Annotation" \
    -with-report execution_reports/pipeline_report.html \
    -with-dag execution_DAGs/pipeline_DAG.html \
    -profile slurm
```

We note that the example data are paired-end human samples, which we plan to align to the "hg38" reference genome. Reads were sequenced such that FASTQ files are "reverse"-stranded. Thus, we will change the `--sample` flag to have the value "paired", the `--reference` flag to be "hg38", and the `--strand` flag to be "reverse".

Next, we will direct SPEAQeasy to the input files of interest and choose a directory in which to place outputs. In the downloading the example data section, you should've selected a directory in which to place FASTQ files and the associated `samples.manifest` file. This same directory should be passed to the `--input` flag, which will replace the `--small_test` option. Finally, choose any desired output directory for SPEAQeasy files, to pass to the `--output` flag. Combining these changes, the resulting `nextflow` command in your main script should look like:

```
$ORIG_DIR/Software/nextflow main.nf \
    --sample "paired" \
    --reference "hg38" \
    --strand "reverse" \
    --input "/some/dir/containing/samples.manifest/file" \
    --output "/some/dir/for/outputs" \
    --annotation "$ORIG_DIR/Annotation" \
    -with-report execution_reports/pipeline_report.html \
    -with-dag execution_DAGs/pipeline_DAG.html \
    -profile docker_slurm # in this example, docker is used
```

3 Run the pipeline

Full execution of SPEAQeasy can take several hours or up to a few days, and it is in general not recommended to run the "main" script interactively if possible. Those with access to a cluster managed by a resource manager like Sun Grid Engine (SGE) or Simple Linux Utility for Resource Management (SLURM) should submit the main script as a job.

```
# SLURM users may use the following
sbatch run_pipeline_slurm.sh

# SGE users may use
qsub run_pipeline_sge.sh

# Those using the JHPCE cluster may use
qsub run_pipeline_jhpce.sh

# Those running SPEAQeasy locally must run interactively
bash run_pipeline_local.sh
```

When SPEAQeasy completes, one may find the `RangedSummarizedExperiment` objects and other primary outputs in the `count_objects` directory relative to the output folder specified. The genotype data as a single VCF file can be found under `merged_variants`. Together these outputs will be used to resolve identity issues in the samples, then perform differential expression and visualization of results.