

Introduction to Snakemake

Dillon Barker

2021-10-22

Lesson 3

2021-10-22

Lesson 3

1. Assignment 2 Answer
2. Fixing when things go wrong
3. Priorities
4. Script files (Python, R, and Julia)
5. Modularizing workflows
6. Visualizing Workflows
7. Linting
8. Clinic

Assignment 2 Answer

Conda Environment Files

```
snakemake-intro-data/  
├─ asn-2.smk  
└─ envs/  
    ├─ pirate.yaml  
    └─ prokka.yaml
```

```
# envs/prokka.yaml  
name: prokka  
channels:  
  - conda-forge  
  - bioconda  
dependencies:  
  - prokka
```

```
# envs/pirate.yaml  
name: pirate  
channels:  
  - bioconda  
  - conda-forge  
dependencies:  
  - pirate
```

Assignment 2 Answer

Snakemake Workflow

https://github.com/dorbarker/snakemake-intro/blob/main/answer_keys/asn-2.smk

When Things Go Wrong

Locked Working Directories

- Snakemake locks its working directory when running
 - Prevents other snakemake instances from running in the same place
- Snakemake removes the lock when it completes (success *or* failure)
- Lockfile may not be removed when if snakemake crashes or is killed by slurm
 - `scancel --user your_username`

```
Unable to lock working directory.
```

Unlocking a Stale Lock

The Correct Way: `--unlock`

```
sbatch -c 1 --mem 2G --wrap=\n"snakemake --unlock -s path/to/your/workflow.smk -d your/locked/analysis/ -j 1"
```

💀💀💀 Nuclear Option 💀💀💀

```
rm -r your/locked/analysis/.snakemake
```

When Things Go Wrong

It Should Have Worked But It Didn't!

- Sometimes a job finishes successfully but snakemake doesn't detect the outputs
- Filesystem latency *especially* on a cluster like Waffles can be a factor
 - It takes some non-zero amount of time to write results to disk
- Consider setting `--latency-wait` with larger number (default: 5 seconds)

Priorities

- Rule priorities may be set
- These **are not** the same as slurm partitions
- Only determine the priority of execution *within* the workflow
- Rules default to priority 0
- Bigger numbers are higher priority

```
rule my_example:  
  input: ...  
  output: ...  
  priority: 50  
  shell: ...
```

Script Files

- Not to be confused with an external command that happens to be a script
- Like a `run` block, but stored in another file
 - Path is specified relative to the workflow
- Available languages:
 - Python
 - R
 - Julia
- Snakemake will use special variables in the script
 - These scripts won't be general purpose!
 - Tied to snakemake

Script Files

```
# workflow.smk
rule scriptfile_example:
    input:
        "path/to/inputfile",
        "path/to/other/inputfile"
    output:
        "path/to/outputfile",
        "path/to/another/outputfile"
    script:
        "scripts/script.py"
```

```
# scripts/script.py
def do_something(data_path, out_path, threads, myparam):
    # python code here

do_something(snakemake.input[0], snakemake.output[0],
             snakemake.threads, snakemake.config["myparam"])
```

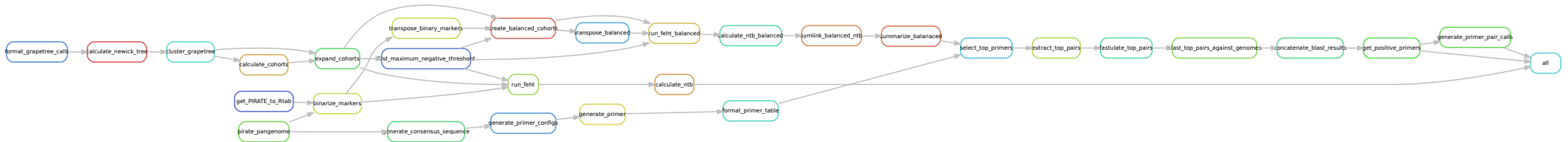
Including Workflows

- Existing workflows can be included with `include`
- Allows more modular design of workflows
- Like copy/pasting another workflow in

```
rule all:  
    input: "summary_report.txt"  
  
include: "summary_report.smk"  
include: "determine_amr.smk"  
include: "assemble_genomes.smk"
```

Visualizing Workflows

```
# Shows just the abstract graph of rules
snakemake -s your/workflow.smk --rulegraph | dot -Tpng > workflow.png
# OR
# Show every sample's journey through the workflow
snakemake -s your/workflow.smk --dag | dot -Tpng > workflow.png
```



Linting

Lint, or a **linter**, is a static code analysis tool used to flag programming errors, bugs, stylistic errors and suspicious constructs.

— Wikipedia

- Get automated *suggestions* for your workflow!

```
snakemake --lint -s path/toyour/workflow.smk
```

Colophon

- This presentation is written in RMarkdown + Xaringan
 - Built with Snakemake

GitHub Repository

<https://github.com/dorbarker/snakemake-intro/>

Today's Lecture

<https://github.com/dorbarker/snakemake-intro/blob/main/lessons/snakemake-intro-lesson-3.pdf>

Clinic

Please feel free to ask any questions!