THE UNIVERSITY OF UTAH

Workflow Managers
Snakemake and Nextflow

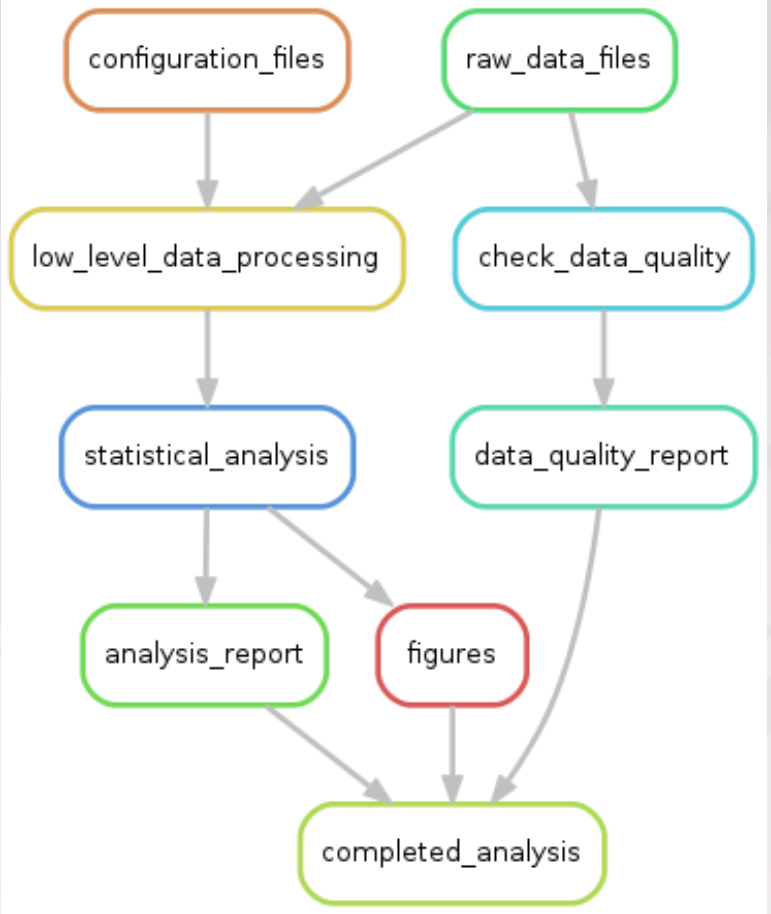
Brett Milash

Center for High Performance Computing

University of Utah

A workflow manager is software that:

- Conducts a complex work flow or analysis
- Tracks dependencies from results back to configuration and data files
- Executes statements step-by-step, in parallel where possible, to carry out work flow



https://snakemake.readthedocs.io/en/stable/

Why use a workflow manager?

- Human efficiency and convenience
- Computational efficiency – only the required steps are executed
- Reproducibility
- Portability between clusters, institutions
- Modularity – re-use and standardization

(My) Selection criteria

Over 100 different workflow managers:
https://github.com/pditommaso/awesome-pipeline

- Actively used and developed
- Can be configured for local and/or cluster execution
- Native SLURM support
- No significant system administration support required
- General purpose (i.e. not just for a single research area)
- Significant functionality bang for your learning buck

Good candidates: snakemake and nextflow

https://snakemake.readthedocs.io/en/stable/

Simple snakefile example

```
rule link:
        input: "hello_world.o"
        output: "hello_world"
        shell: """
                module load gcc/6.1.0
                gcc -o {output} {input}
                """

rule compile:
        input: source="hello_world.c",headers=["hello_world.h",]
        output: "hello_world.o"
        shell: """
                module load gcc/6.1.0
                gcc -c {input.source}
                """
```

Rules have:

- names
- inputs
- outputs
- actions (shell or python)

Rules are:

- linked implicitly (or explicitly)
- executed in parallel if possible
- executed locally or on a cluster

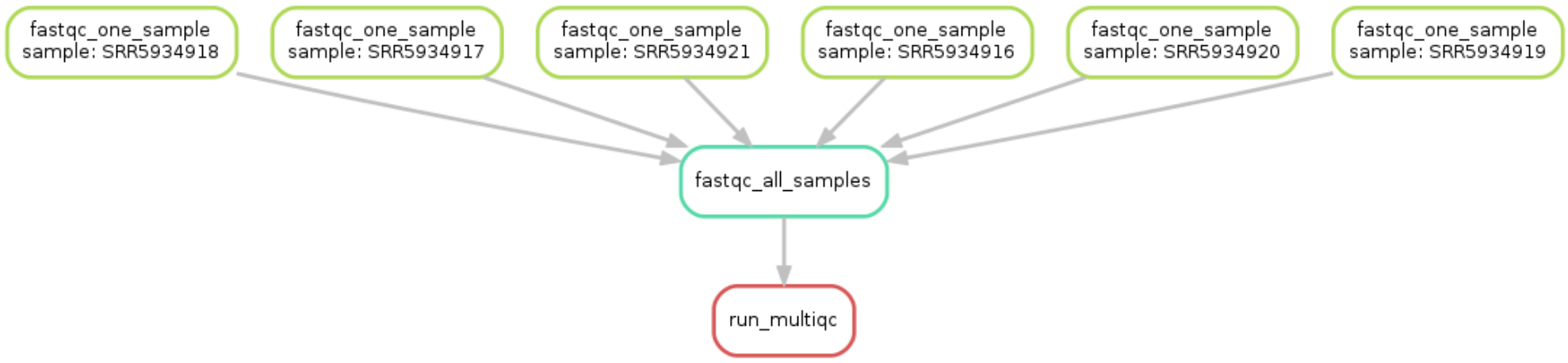
More snakemake features

- Modular workflows

```
include: "path/to/other/snakefile"
```

- Graphical output

```
snakemake –s snakefile --rulegraph | dot –Tpng > rulegraph.png
snakemake –s snakefile --dag | dot –Tpng > dag.png
```



Snakemake may not be right for you

- What if your inputs and outputs aren't files?
- What if your cluster doesn't use SLURM or LSF?
  - HTCondor (Open Science Grid: > 1.2 billion core hours last year)
- What if your workflow changes?
- nextflow: https://www.nextflow.io/
  - non-file inputs and outputs
  - support for HTCondor (OSG) and many other schedulers
  - workflow file is part of the workflow – when a rule changes, it gets re-run

Nextflow basics

Nextflow is based on the dataflow programming model.

Imagine your workflow as an assembly line built from:

- Channels
  - Carry data around
  - Not just files – also values, streaming data (pipes), “sets” of items -> data types
  - “Operators” allow channel filtering, forking, combining, transforming, …
- Processes
  - Input from channel(s)
  - Output to channel(s)
  - Scripts: shell, python, perl, R, or any other interpreted language
- Executors
  - “Plug-ins” that control where the work is done
  - Local (by default), also SLURM, Open Science Grid , SGE, LSF, PBS, etc …

Nextflow compile example

compile2.nf

```
#!/usr/bin/env nextflow
c_files_channel = Channel.fromPath( "*.c" )

process compile {
        module 'gcc/4.9.2'
        input: file c_file from c_files_channel
        output: file '*.o' into result
        script: "gcc -c -g ${c_file}"
}

result.subscribe { println "Compiled $it" }
```