

Snakemake for reproducible research

Additional advanced concepts

Antonin Thiébaut & Rafael Riudavets Puig

antonin.thiebaut@chuv.ch

Rafael.RiudavetsPuig@empa.ch

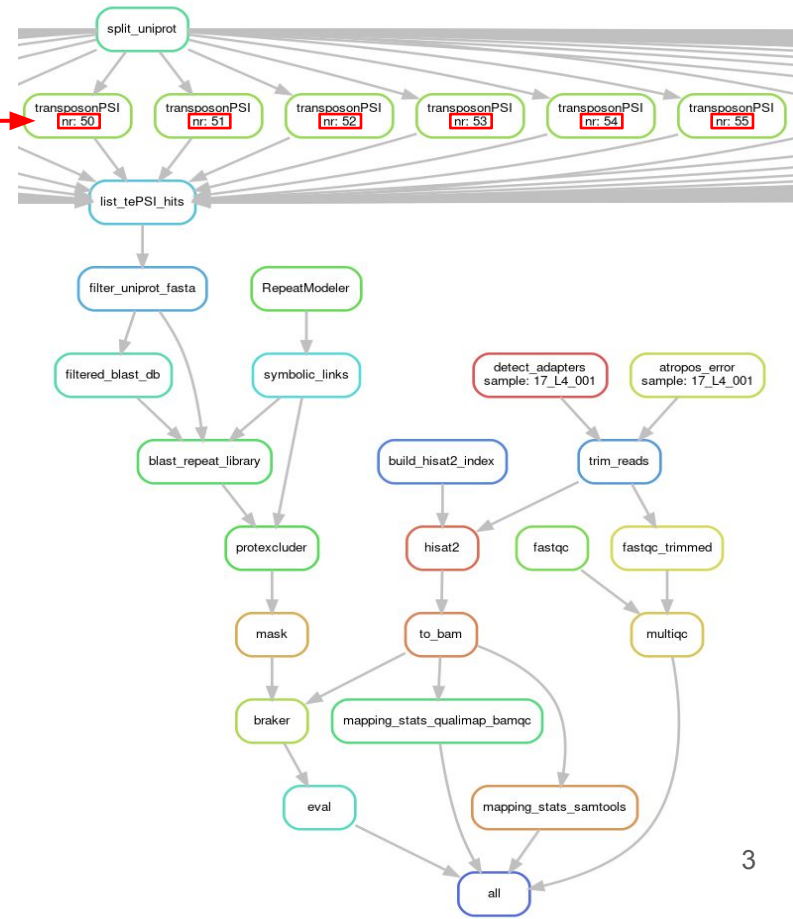
Using non-conventional outputs

- Snakemake has built-in utilities to assign properties to 'special' outputs

Property	Syntax	Function
Temporary	<code>temp('path/to/file.txt')</code>	File is deleted as soon as it is not required by any future jobs
Protected	<code>protected('path/to/file.txt')</code>	File cannot be overwritten after the job ends (useful to prevent erasing a file by mistake, for example files requiring heavy computation)
Ancient	<code>ancient('path/to/file.txt')</code>	Ignore file timestamp and assume file is older than any outputs: file will not be re-created when re-running the workflow, except when <code>--force</code> parameters are used
Directory	<code>directory('path/to/directory')</code>	Output is a directory instead of a file (use 'touch' instead if possible)
Touch	<code>touch('path/to/file.txt')</code>	Create an empty flag file 'file.txt' regardless of the shell command (if the command finished without errors)

Building a Directed Acyclic Graph (DAG)

- Snakemake determines which jobs to run to produce desired **outputs**
- **Rule** can appear more than once, with different **wildcards**
 - 1 rule + 1 wildcard values = 1 job
- **Arrows** = dependency between jobs
 - Snakemake runs jobs in any order that doesn't break dependency
- **DAG** = work list, ≠ flowchart
 - No if/else decisions or loops
 - Snakemake runs every job in the DAG exactly once
- **DAG** ≠ checking **shell** directives
 - Shell commands are tested during execution
 - Works? Produces expected outputs?



Reminder on best practices

- One repository = one workflow
- Use Conda environments / Docker containers when possible
- Break out large workflow into modules with extension ".smk"
- Specify parameters in a config file located in a 'config' folder
- If you have many samples with information, use a sample sheet located in the 'config' folder
- Follow the official directory structure
- Use explicit rule and variable names
- Comment to explain your workflow; use docstring comments in rules

