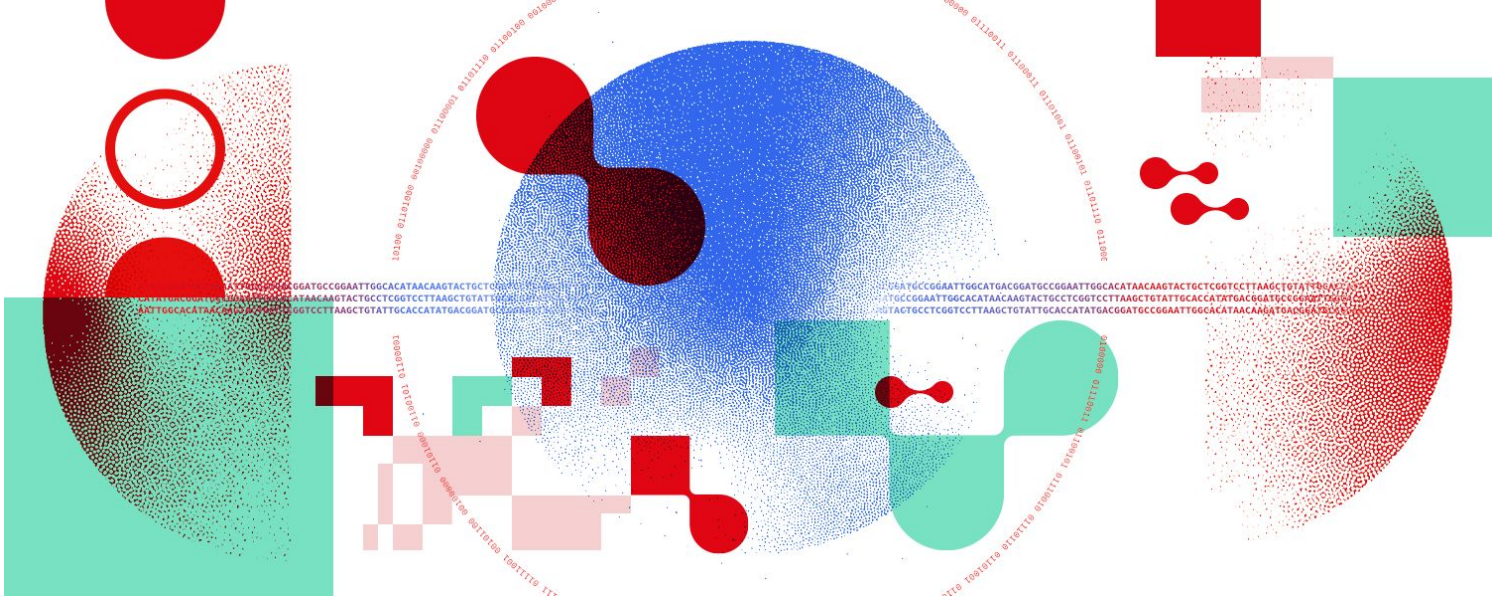


Snakemake for reproducible research

Additional advanced concepts

Centre hospitalier universitaire vaudois

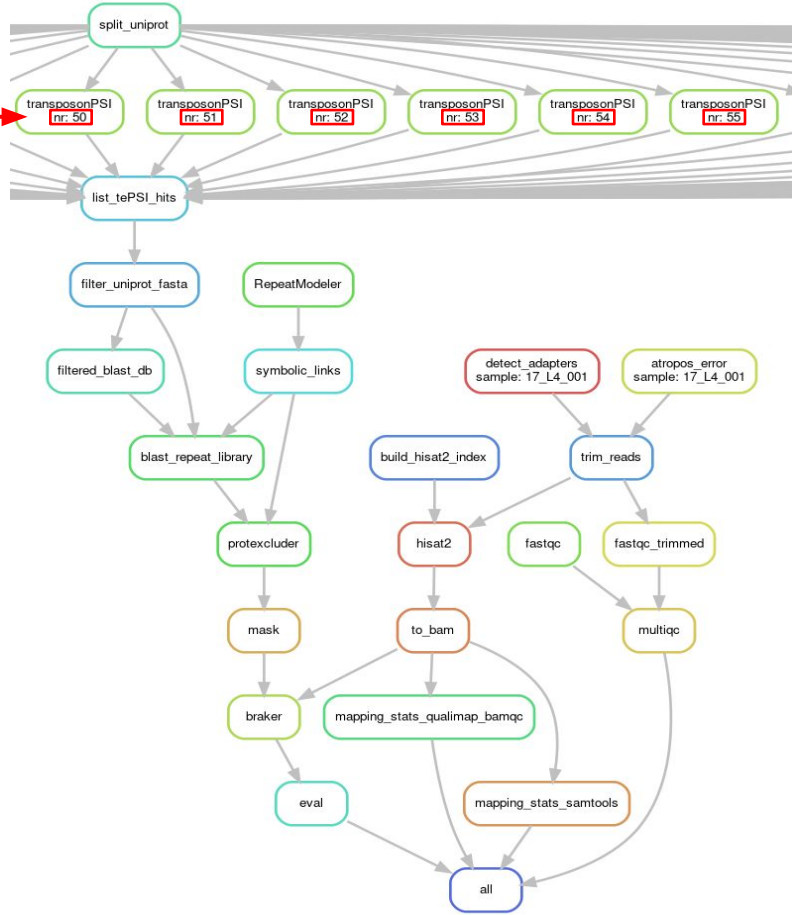
Empa
Materials Science and Technology

Antonin Thiébaut & Rafael Riudavets Puig
antonin.thiebaut@chuv.ch
Rafael.RiudavetsPuig@empa.ch

SIB
Swiss Institute of Bioinformatics

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?



Using non-conventional outputs

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

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

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