Reproducible data analysis with Snakemake

Johannes Köster
University of Duisburg-Essen  https://koesterlab.github.io

The 3 dimensions of reproducibility

Automation/documentation:
- **Document and execute** all steps from raw data to final tables and figures without manual intervention.

Scalability:
- Execute for tens to thousands of datasets.
- Efficiently use any computing platform.

Portability:
- Easily execute analysis on a different system/platform/architecture.

**Automation/documentation**

- **General:**
  - Decompose analysis into rules, written in a Python dialect.
  - Rules define how to obtain output files from input files.
  - Snakemake determines dependencies and execution order in the form of a directed acyclic graph (DAG) of jobs.

- **Automatic reports:**
  - Annotate output files for inclusion.
  - Define categories and (jinja-templated) captions.
  - Obtain self-contained HTML5 document including all files, workflow description, runtime statistics, and provenance information.

```
rule estimate_spike_proportion:
    input:
      "analysis/all.sce.rds"
    output:
      "plots/spike-proportion.svg"
    category="Quality control"
    caption="report/spike-proportion.rst"
    script:
      "scripts/plot-spike-proportion.R"
```

**Scalability**

- **General:**
  - Independent parts of the DAG of jobs can be executed in parallel.
  - Snakemake maximizes parallelism while respecting given resources.
  - Without modification of the workflow definition, Snakemake can scale to any number of cores, compute clusters, the grid, and the cloud.

- **Job groups:**
  - The DAG of jobs can be partitioned into groups.
  - Minimizes queueing and network overhead in cloud and cluster.

```
rule bwa:
    input:
      "genome.fa"
      "reads/{sample}.fastq"
    output:
      "mapped/{sample}.bam"
    group: "mapping"
    threads: 8
    shell:
      "bwa mem -t {threads} {input} | ";
      "samtools view -Sb - > {output}"
```

**Pipe output:**

- Output files can be marked as pipes.
- Consuming jobs will be assigned to the same group.
- Output will not be written to disk but streamed between the jobs.

```
rule bwa:
    input:
      "genome.fa"
      "reads/{sample}.fastq"
    output:
      pipe("mapped/{sample}.bam")
    threads: 8
    shell:
      "bwa mem -t {threads} {input} | ";
      "samtools view -Sb - > {output}"
```

**Portability**

**Software deployment with Conda:**
- Rules can be annotated with (isolated) Conda environments that define a software stack with particular versions to use.
- Jobs are executed within these environments.

**Software deployment with Singularity:**
- Rules/workflows can be annotated with container images.
- Jobs are executed within the container.
- Combination with Conda possible: use container image to define OS, use Conda to define the software stack, let Snakemake perform the composition.

```
singularity: "docker://continuumio/miniconda3"
rule estimate_spike_proportion:
    input:
      "analysis/all.sce.rds"
    output:
      "plots/spike-proportion.org"
    conda:
      "mmtls-cpu=0-{q}/yaml"
    script:
      "scripts/plot-spike-proportion.sh"
```

https://snakemake.readthedocs.io