

Reproducible data analysis with Snakemake

Johannes Köster

2019

<https://koesterlab.github.io>

Snakemake is popular

150k downloads since 2015

497 citations (+150 in 2019)



<https://snakemake.readthedocs.io>

Concise DSL

```
rule mytask:
    input:
        "data/{sample}.txt"
    output:
        "result/{sample}.txt"
    shell:
        "some-tool {input} > {output}"
```

Python scripts

```
rule mytask:
    input:
        "data/{sample}.txt"
    output:
        "result/{sample}.txt"
    script:
        "scripts/mytask.py"
```

R scripts

```
rule mytask:
  input:
    "data/{sample}.txt"
  output:
    "result/{sample}.txt"
  script:
    "scripts/mytask.R"
```

Julia scripts

```
rule mytask:
    input:
        "data/{sample}.txt"
    output:
        "result/{sample}.txt"
    script:
        "scripts/mytask.jl"
```

No boilerplate

```
input: mytask:
  output: "data/{sample}.txt"
  script: "result/{sample}.txt"
  "scripts/mytask.py"
```



```
import matplotlib.pyplot as plt
import pandas as pd

d = pd.read_table(snakemake.input[0])

d.hist(bins=snakemake.config["hist-bins"])

plt.savefig(snakemake.output[0])
```

No boilerplate

```
#!/usr/bin/env mytask:
    output: "data/{sample}.txt"
    script: "result/{sample}.txt"
            "scripts/mytask.py"
```



```
import matplotlib.pyplot as plt
import pandas as pd

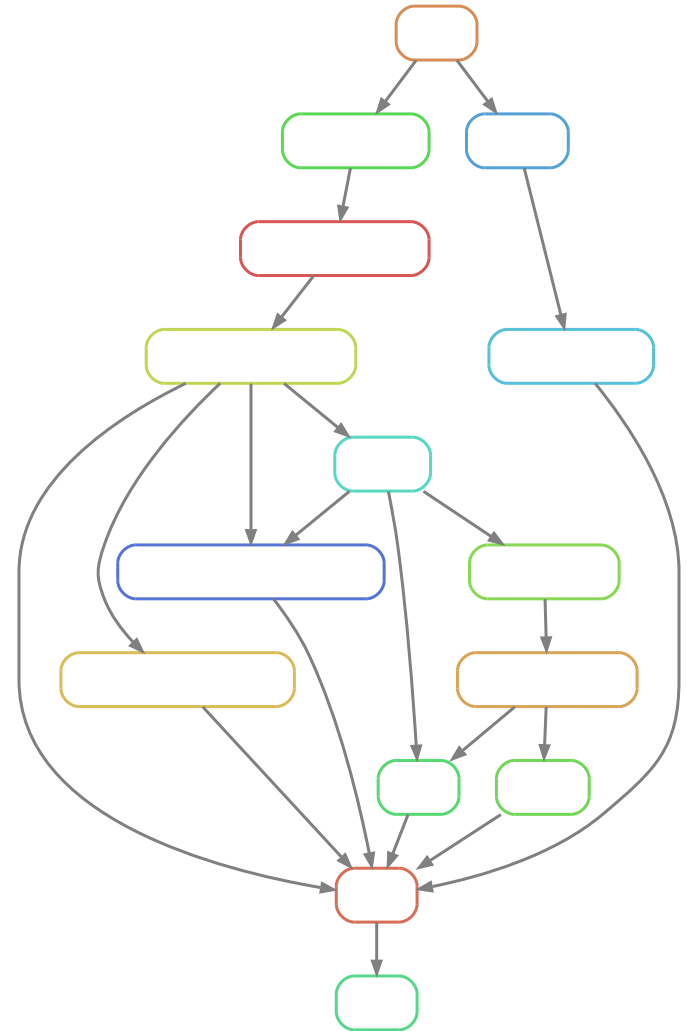
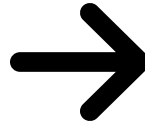
d = pd.read_table(snakemake.input[0])

d.hist(bins=snakemake.config["hist-bins"])

plt.savefig(snakemake.output[0])
```

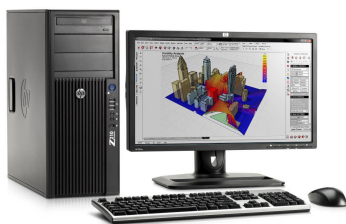

Implicit dependencies

```
rule mytask:  
  input:  
    "path/to/{dataset}.txt"  
  output:  
    "result/{dataset}.txt"  
  script:  
    "scripts/myscript.R"  
  
rule myfiltration:  
  input:  
    "result/{dataset}.txt"  
  output:  
    "result/{dataset}.filtered.txt"  
  shell:  
    "mycommand {input} > {output}"  
  
rule aggregate:  
  input:  
    "results/dataset1.filtered.txt",  
    "results/dataset2.filtered.txt"  
  output:  
    "plots/myplot.pdf"  
  script:  
    "scripts/myplot.R"
```



Scalability

workstation



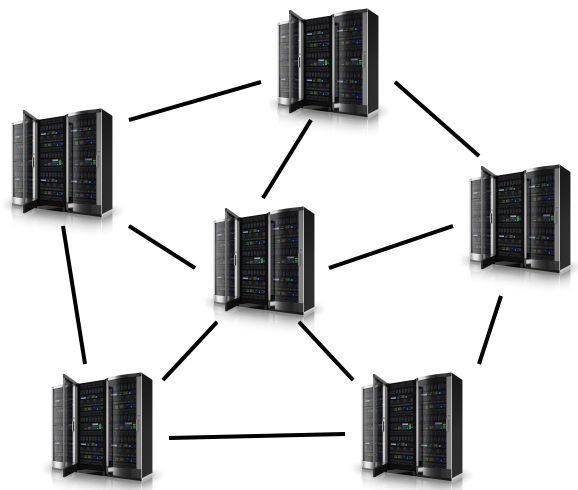
compute server



cluster



grid computing



cloud computing



kubernetes

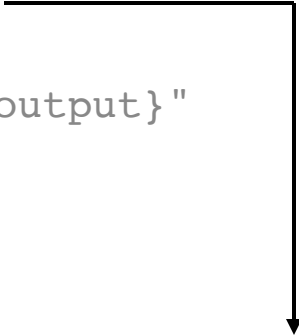


Google Cloud Platform



Conda integration

```
rule mytask:
    input:
        "path/to/{dataset}.txt"
    output:
        "result/{dataset}.txt"
    conda:
        "envs/mycommand.yaml"
    shell:
        "mycommand {input} > {output}"
```



```
channels:
    - bioconda
    - conda-forge
dependencies:
    -mycommand =2.3.1
```

Singularity integration

```
rule mytask:
    input:
        "path/to/{dataset}.txt"
    output:
        "result/{dataset}.txt"
    singularity:
        "docker://some/container"
    shell:
        "mycommand {input} > {output}"
```

Singularity + Conda

```
rule mytask:
    input:
        "path/to/{dataset}.txt"
    output:
        "result/{dataset}.txt"
    conda:
        "envs/mycommand.yaml"
    singularity:
        "docker://some/os"
    shell:
        "mycommand {input} > {output}"
```

Snakemake

Snakemake Report

Fri Jul 19 16:58:33 2019 CET
Snakemake 5.5.3+6.g668fec20

[Workflow](#)

[Statistics](#)

[Configuration](#)

[Rules](#)

RESULTS

Allele Frequency Estimation

Concordance

FDR Control

Precision and Recall

Score Distribution

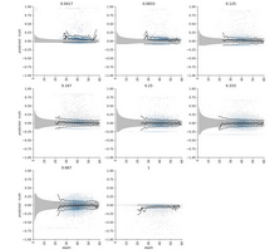
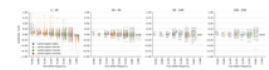
Results

[Allele Frequency Estimation](#)

Show entries

Search:

File	Size	Description	Job properties
simulated-bwa.DEL.svg	328.3 kB	Allele frequency estimation error vs. true allele frequency on simulated data.	Rule plot_allelefreq Wildcards run=simulated-bwa, vartype=DEL Params varlociraptor_callers=['delly', 'lancet', 'manta', 'strelka'], len_ranges=[[1, 30], [30, 50], [50, 100], [100, 250]]
simulated-bwa.DEL.svg	1.9 MB	Allele frequency estimation error vs. true allele frequency on simulated data. The dashed lines depict the standard deviation, solid line depicts the mean. The grey area shows the standard	Rule plot_allelefreq_scatter Wildcards run=simulated-bwa, vartype=DEL Params depth_ranges=[[1, 20], [20, 40]], callers=['delly', 'lancet', 'manta', 'strelka']



Snakemake

Snakemake Report

Fri Jul 19 16:58:33 2019 CET
Snakemake 5.5.3+6.g668fec20

Search:

[Workflow](#)

[Statistics](#)

[Configuration](#)

[Rules](#)

RESULTS

[Allele Frequency Estimation](#)

[Concordance](#)

[FDR Control](#)

[Precision and Recall](#)

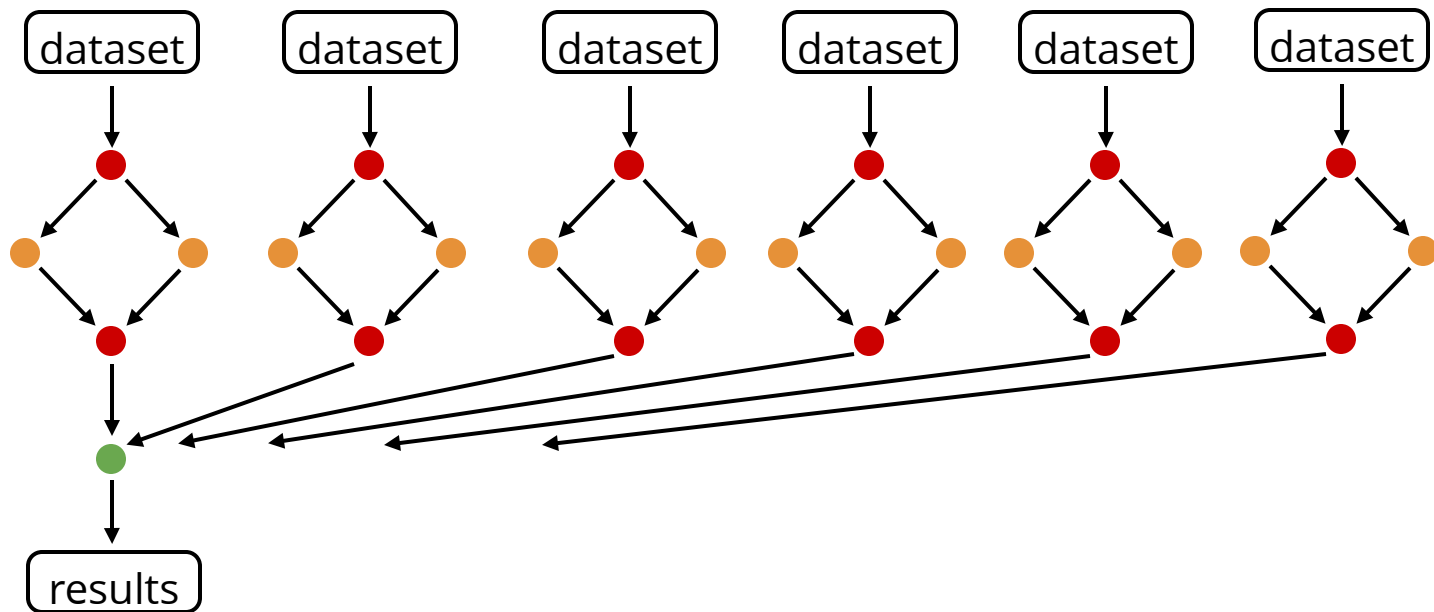
[Score Distribution](#)

Rule	Jobs	Output	Singularity	Conda environment	Code
adhoc_varlociraptor	40				
aggregate_concordance	30				
				python =3.6	1 from common import load_variants 2 import networkx as nx 3 import pandas as pd 4 import numpy as np
				pandas =0.23	5 6 vartype = snakemake.wildcards.vartype 7
				matplotlib =3.0	8 index_cols = ["CHROM", "POS", "SVLEN"] if var 9 10 all_variants = [load_variants(f, vartype=vart 11
				seaborn =0.9.0	12 G = nx.Graph() 13 for calls, (i, j) in zip(all_variants, snakem 14 calls["component"] = None
				pysam =0.13.0	15 for call in calls.itertuples(): 16 a = (i, call.Index) 17 G.add_node(a)
				svgutils =0.2	18 if call.MATCHING >= 0: 19 b = (j, call.MATCHING) 20 G.add_node(b) 21 G.add_edge(a, b) 22
				pybedtools =0.7.10	23 # get a set of calls for each dataset (we dor 24 representatives = {snakemake.params.dataset_c 25
				networkx =2.2	26 if snakemake.wildcards.mode != "varlociraptor" 27 varlociraptor_variants = [load_variants(f 28 for calls in varlociraptor_variants: 29 calls.set_index(index_cols, inplace=1 30 varlociraptor_representatives = {snakemak 31

↑ portability

scalability →

↓ automation/
documentation



<https://snakemake.readthedocs.io>