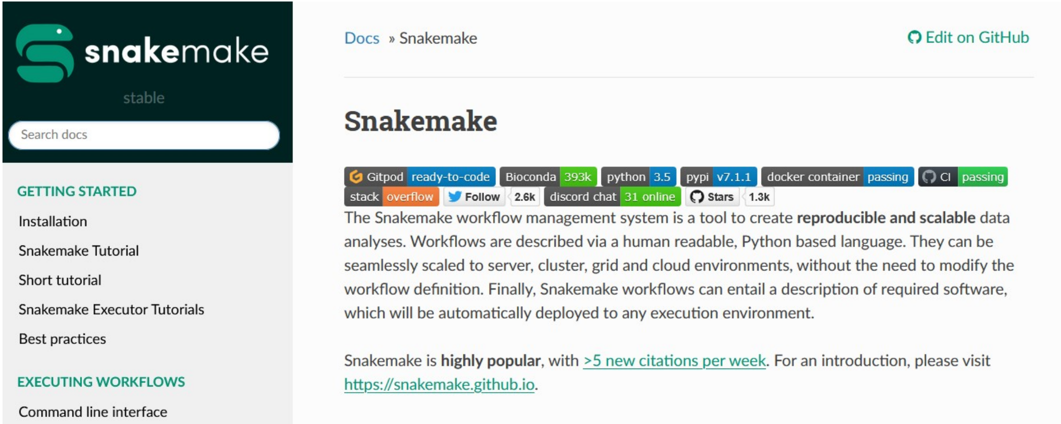
Snakemake: Making data workflows easier and more reproducible

Johannes Hampp

16 March 2022

Center for International Development and Environmental Research (ZEU)
Justus-Liebig University Giessen



snakemake

stable

Search docs

GETTING STARTED

- Installation
- Snakemake Tutorial
- Short tutorial
- Snakemake Executor Tutorials
- Best practices

EXECUTING WORKFLOWS

- Command line interface

Docs » Snakemake

Edit on GitHub

Snakemake

Gitpod ready-to-code | Bioconda 393k | python 3.5 | pypi v7.1.1 | docker container passing | CI passing | stack overflow | Follow 2.6k | discord chat 31 online | Stars 1.3k

The Snakemake workflow management system is a tool to create **reproducible and scalable** data analyses. Workflows are described via a human readable, Python based language. They can be seamlessly scaled to server, cluster, grid and cloud environments, without the need to modify the workflow definition. Finally, Snakemake workflows can entail a description of required software, which will be automatically deployed to any execution environment.

Snakemake is **highly popular**, with >5 new citations per week. For an introduction, please visit https://snakemake.github.io.

Source: Screenshot snakemake.readthedocs.io

Outline

Is it relevant for you and me?

Making research more FAIR?

Quick overview

Live presentation

Is it relevant for you and me?

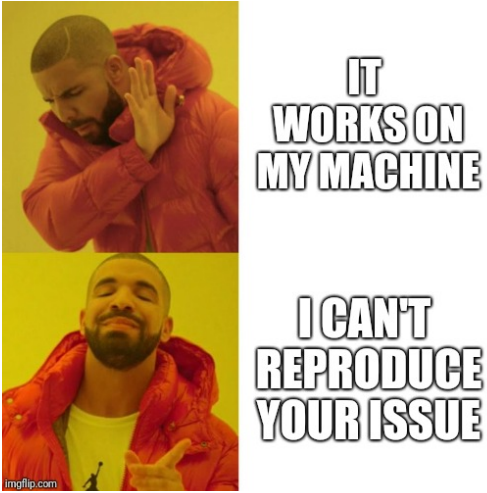
Significance

Who am I?

- B./M.Sc. in experimental physics (modelling + experiment data crunching)
- PhD student in Energy System Modelling
- Proponent of Open Source Software, Data, Access
- Co-maintainer of multiple OSS packages and models with 10+ core devs
- I've seen a lot of multi-generation models (physics, economics, energy systems)

`snakemake` turned my way of working upside down.

Research: Can (not) reproduce

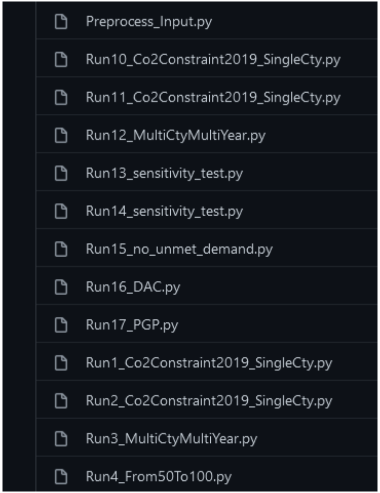


Challenges (e.g. model or data pipeline for experimental data):

- Legacy work (previous PhD student, own work, student assistants, other researchers)
- Which data goes in, which data goes out? (documentation is nice, is it comprehensible and up-to-date?)
- How to execute the pipeline? (same data or new data)
- Something is not working, but what? (thousands of lines of monolithic code)
- How to extent on the work? ("I'll just put this in here...")

Source: devrant.com

Documentation TL;DR



Preprocess_Input.py
Run10_Co2Constraint2019_SingleCty.py
Run11_Co2Constraint2019_SingleCty.py
Run12_MultiCtyMultiYear.py
Run13_sensitivity_test.py
Run14_sensitivity_test.py
Run15_no_unmet_demand.py
Run16_DAC.py
Run17_PGP.py
Run1_Co2Constraint2019_SingleCty.py
Run2_Co2Constraint2019_SingleCty.py
Run3_MultiCtyMultiYear.py
Run4_From50To100.py

"Documentation too lax; did not reproduce"?

A common workflow example

1. Order preserved by file names
2. When to run `PreProcess.py`?
3. No additional documentation (Except for paper: "We did so-and-so…")
4. Which parts do I need to run if I change external (input) data?

Source: Screenshot GitHub Repository

Making research more FAIR?

“R” stands for “Reusable”

What does that really entail?

- Repeat & Rerun
- Reproduce
- Replicate
- Reliable & Robust
- Rapport building

Source: Loosely based on Agapow (2017)

Quick overview

What it is

Snakemake is a workflow management system.

It is a system to manage workflows.

Core concept: Rules

```
rule do_research:
    input:
        # define input dependencies
        ’raw_data.csv’
    output:
        # files created through this rule
        ’research_results.csv’
    run:
        # your magic converting <input> to <output>
        ’research.py’
```

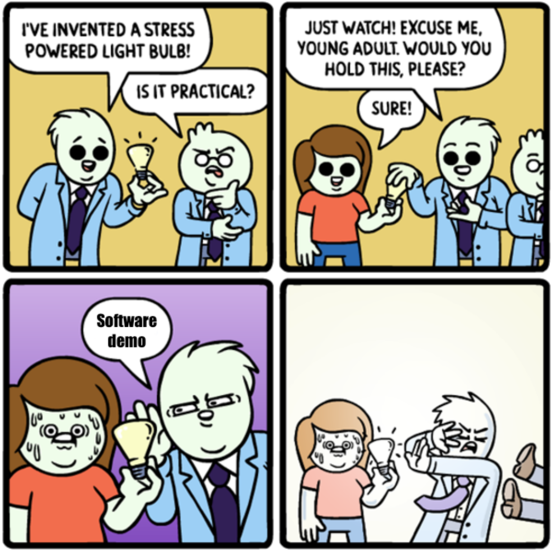
Support for: Python, R, R Markdown, Julia, Rust, Jupyter notebooks and any shell command (!)

Advantages (highly opiniated selection)

What it does	How it helps
Human readable workflow definition	Easy and fast to learn
	Define (and implicitly document) dependencies
	Faster onboarding of new students & staff
Explicit dependencies	Reduces mishaps and mistakes from manual execution
“Rules” (Dependencies) defined and monitored	Automatic re-run if input or code is updated
Scales well	Independent rules run as such
	Rules can be kept small (good for collab., error tracking, re-running)

Live presentation

Hoping this works...

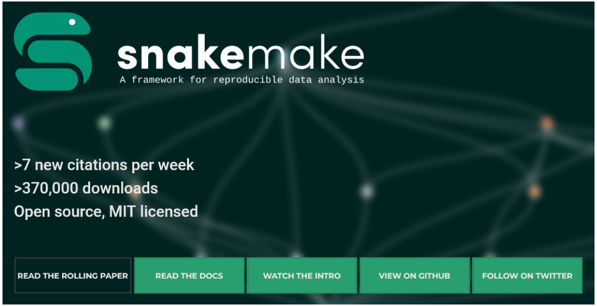


I'VE INVENTED A STRESS POWERED LIGHT BULB!
IS IT PRACTICAL?
JUST WATCH! EXCUSE ME, YOUNG ADULT. WOULD YOU HOLD THIS, PLEASE?
SURE!
Software demo

Source: Reddit

How to get started

- Website, Docs, Tutorials, Videos, Best Practices: `https://snakemake.github.io`
- Rolling paper: `https://f1000research.com/articles/10-33/v1`
- Code from live demo: `https://github.com/euronion/snakemake-demo`
- Download and install (with Anaconda): `conda install -c bioconda snakemake`



snakemake

A framework for reproducible data analysis

>7 new citations per week

>370,000 downloads

Open source, MIT licensed

Source: `https://snakemake.github.io`

License