Reproducibility

in the SKA Challenges and reproducible paper

Javier Moldón
IAA-CSIC

and Laura Darriba (IAA-CSIC)

IAA-CSIC Severo Ochoa SKA Open Science School

Granada. Spain - 10th May 2023


FROM THE PAST

Current You

Future You

The importance of reproducibility

To give the full history of your science

What you did and why

Paper
Documentation
README files

+

How you did it

Actual implementation (methodology)
Data + software + code

↓

You will deliver robust, reliable, trustful science.
Open Science benefits the research community and society
but…

Be realistic

Think about your purpose and skills

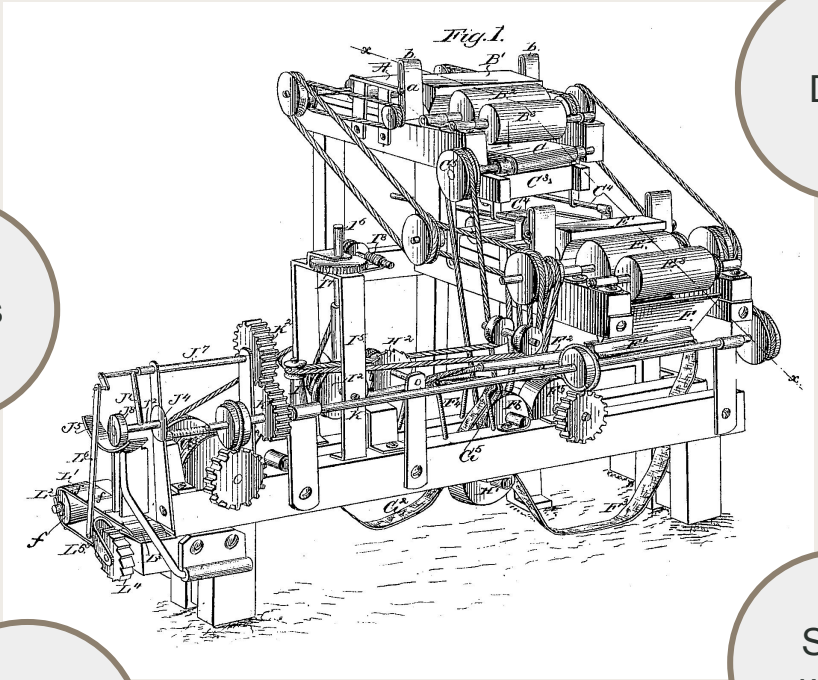
Full end-to-end automatic reproducibility is **hard**

Files

Scripts

Github

Depend encies



Data

Config

Soft- ware

https://thinkbeyondthelogo.wordpress.com/2015/06/19/moving-parts/

Define your scope

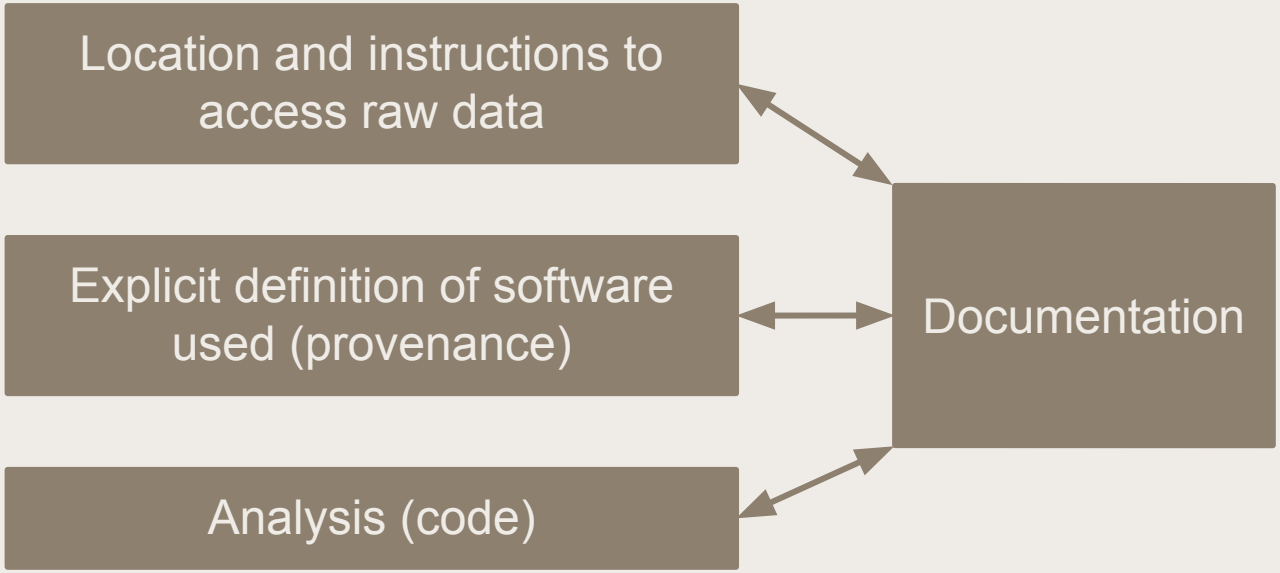
Think about your purpose and skill

Incremental progress is better than frustration

Who is going to reproduce your research?
Potentially, all the future generations of scientist that will ever exist

- Do it for yourself (and your future self)
- For your collaborators
- For the paper referees
- For the community in your field
- For everyone

A reproducible paper



Ramos-Padilla+2021: The viewing angle in AGN SED models, a data-driven analysis

https://github.com/aframosp/AGNView

Where to start?

Get routines

Document what you do (README files)

Use plain text (e.g. **Markdown**)

Use online repositories (**Github**, gitlab)

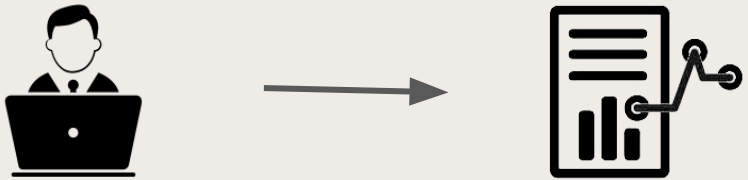
Get used to version control (**git**), at least the basics

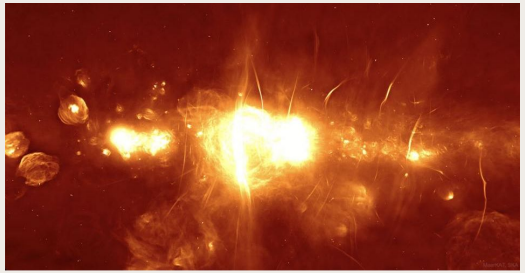
Be **explicit with the software** you use

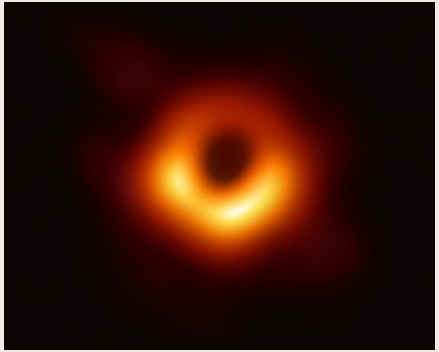
Notebooks can help with collaboration and sharing

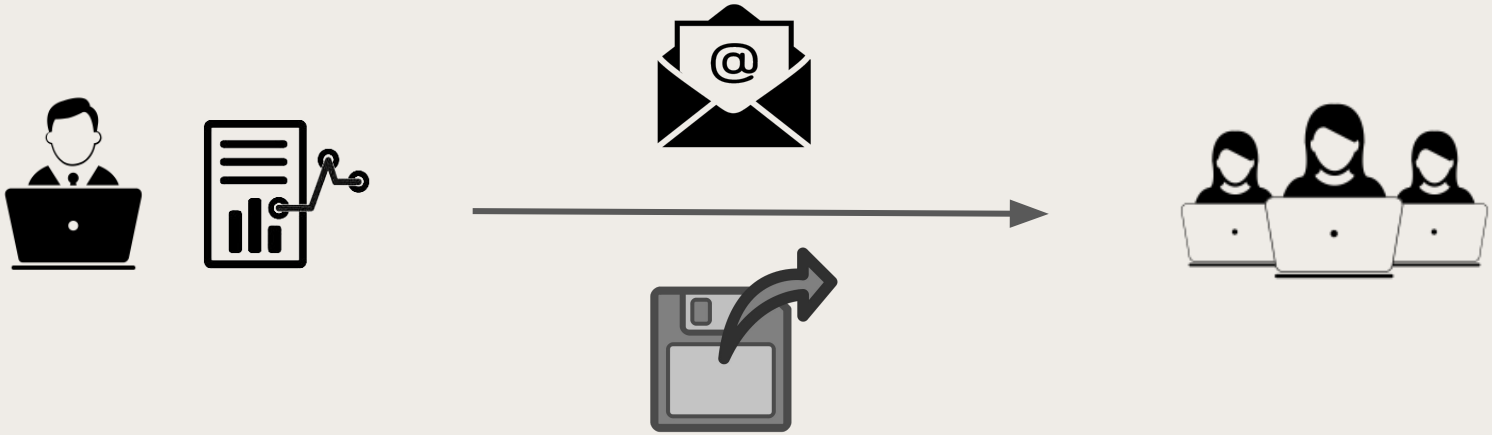
Software repositories

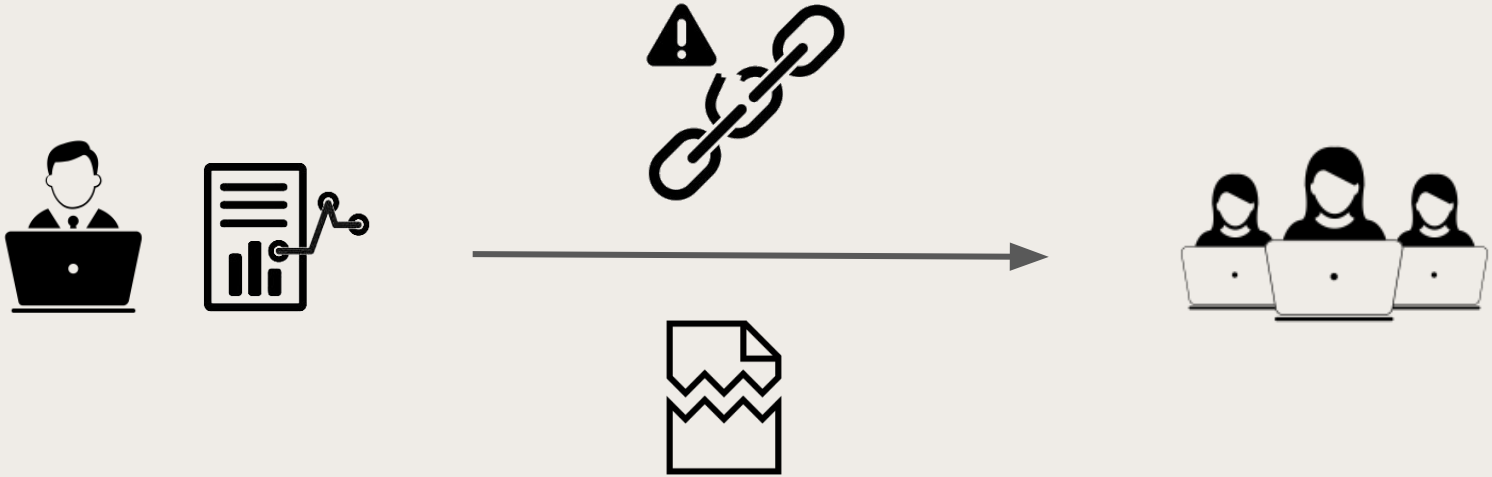
(e.g. Github)

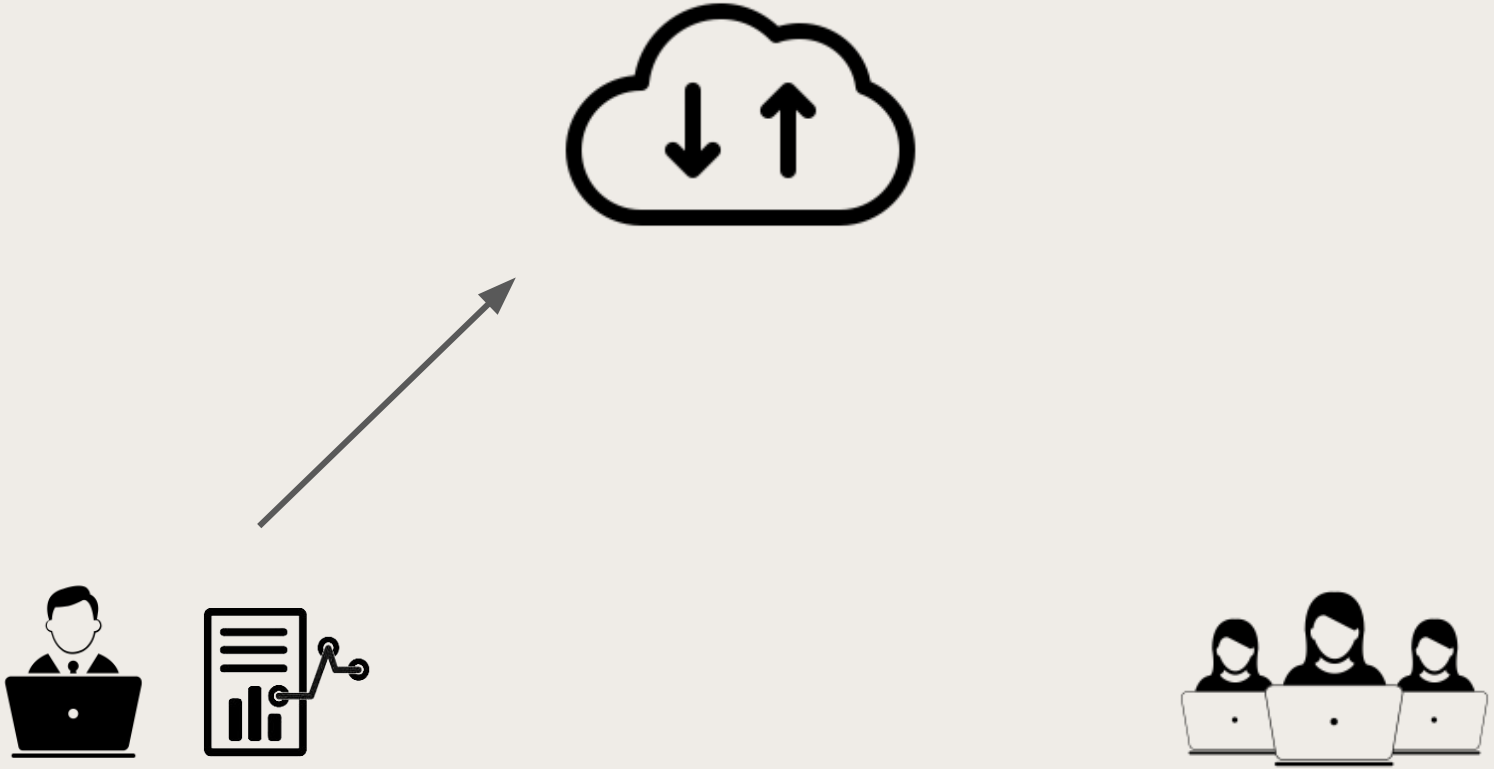


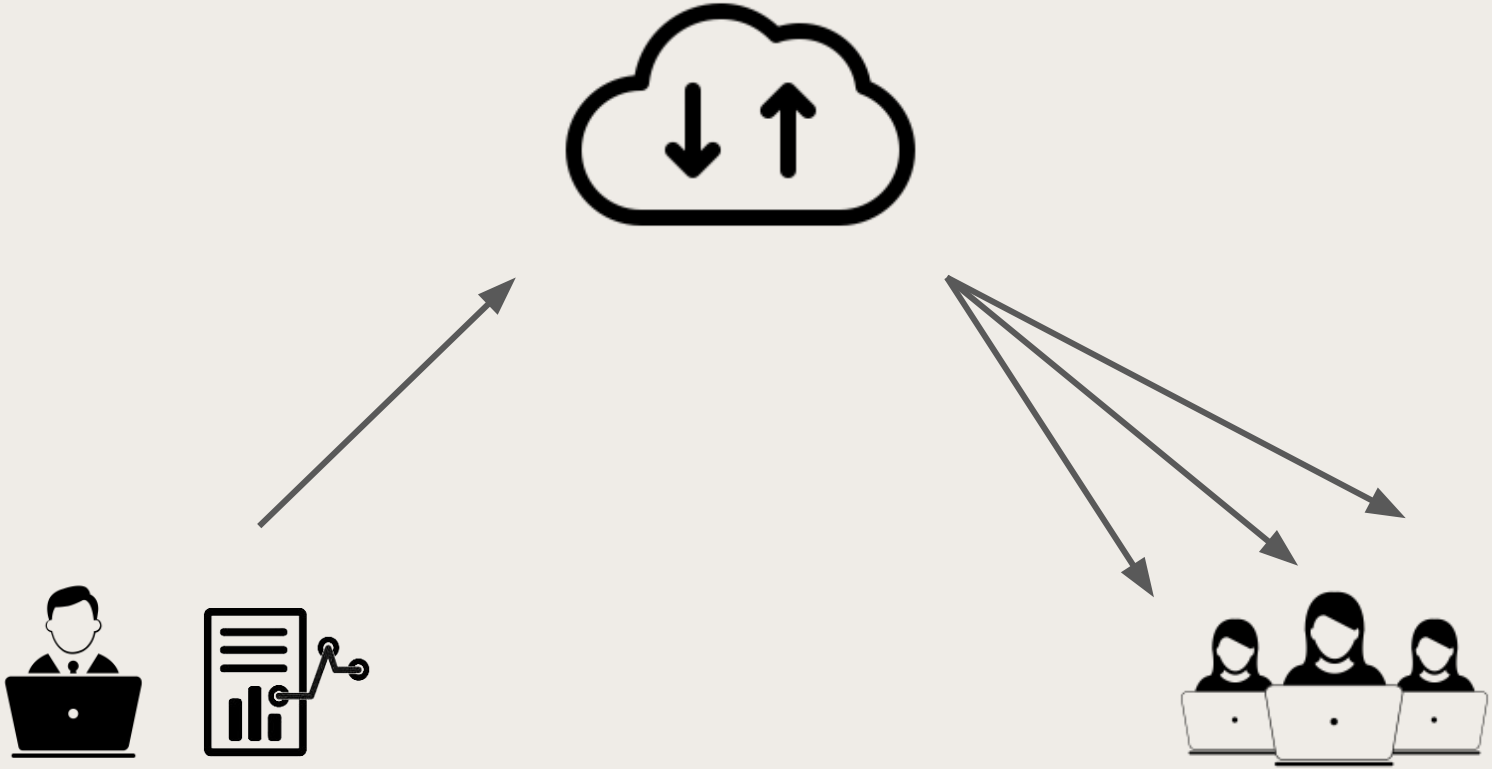


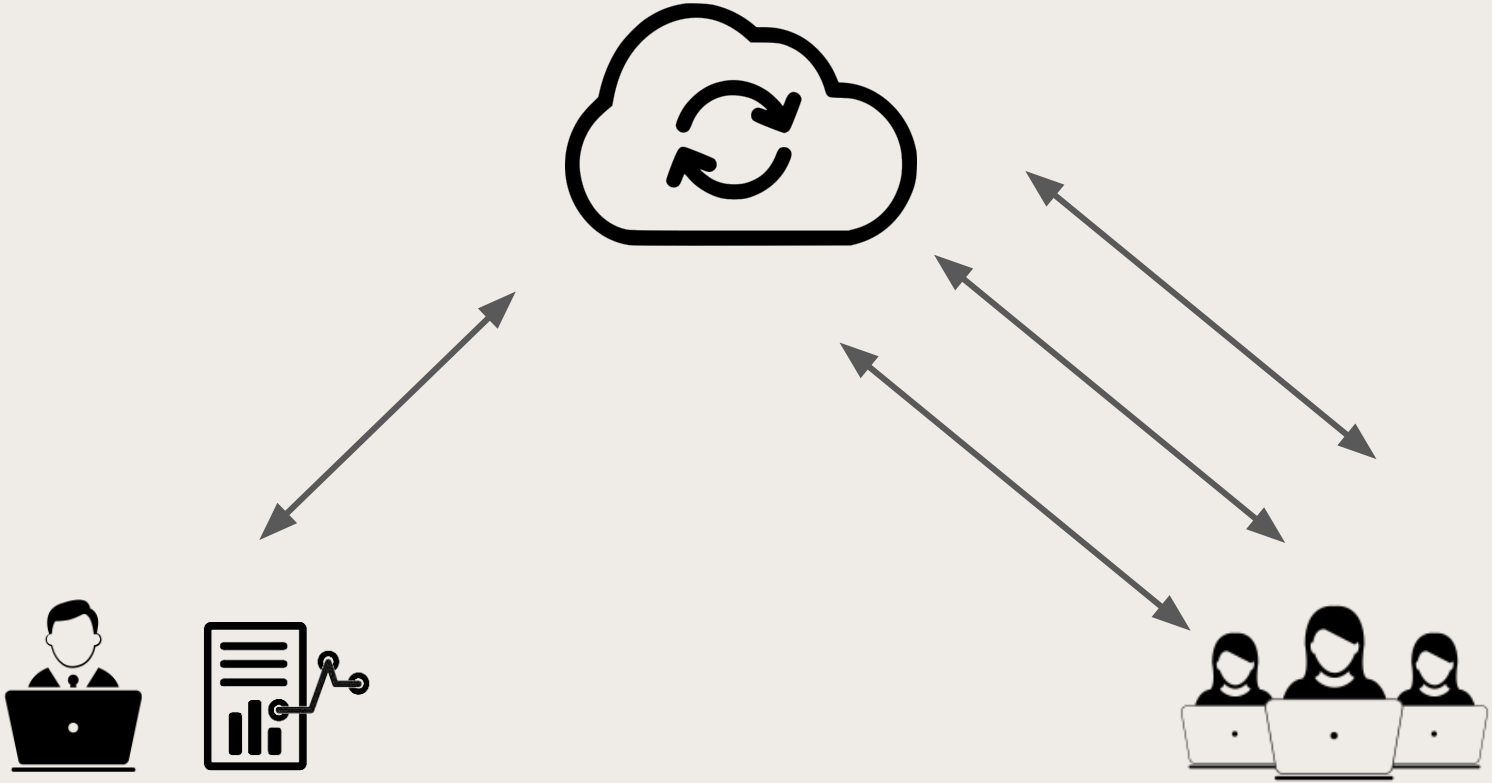






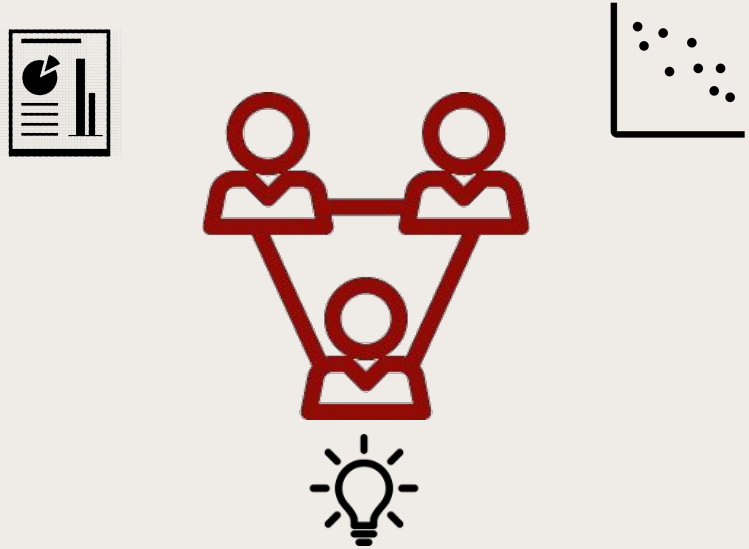






Software repositories

They aim to provide an efficient **collaborative platform** with powerful **git management** under the hood and enabling personal or **social coding**



Github
Gitlab
Bitbucket
…

Basic ingredients

jmoldon / test Private

YOU AND YOUR REPOSITORY

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Actions | Projects 0 | Wiki | Security | Insights | Settings

This project is to share files with my collaborator

YOUR DESCRIPTION

Edit

Manage topics

1 commit | 1 branch | 0 packages | 0 releases | 1 contributor | GPL-3.0

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

jmoldon Initial commit — Latest commit d3d166a 3 minutes ago

.gitignore	Initial commit	3 minutes ago
LICENSE	Initial commit	3 minutes ago
README.md	Initial commit	3 minutes ago

YOUR FILES

README.md

test

YOUR README

test

Automatic rendering of files

Jupyter notebooks
csv
markdown

master droplets / gaia_exploratory / **jupyter_exploratory.ipynb** Go to file ···

jmoldon Add new row limit for gaia with Gaia.ROW_LIMIT Latest commit 09a4ac7 on Apr 9, 2021 **History**

1 contributor

2.49 MB Download

Exploratory analysis of unkown star cluster

We are going to explore the star population of the region of interest by analysing *Gaia* data. *Gaia* Data Release 2 provides high-precision astrometry and three-band photometry for about 1.3 billion sources over the full sky. Theprecision, accuracy, and homogeneity of both astrometry and photometry are unprecedented. To obtain the data we will use the astroquery TAP+ protocol. More information can be found here: https://astroquery.readthedocs.io/en/latest/

In [1]:

```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
import matplotlib

matplotlib.rcParams['figure.figsize'] = (18.0, 10.0)
from astroquery.gaia import Gaia
Gaia.ROW_LIMIT = -1

import astropy.units as u
from astropy.coordinates import SkyCoord
```

Markdown language

Useful and efficient syntax
For .md files, issues, pull requests, etc.

droplets / sessions / droplets-02-conda / droplets-02-conda.md in master

Cancel changes

Edit file | Preview | Spaces | 1 | Soft wrap

```
# Open Science Droplets 02
# Keep your software organized with Conda

Our collaborator shared a [Jupyter notebook](https://github.com/spsrc/droplets/blob/master/gaia_exploratory/jupyter_exploratory.ipynb) with us
and we want to re-run it and change a few things.

## Install Jupyter using conda
Let's work on a temporary folder so at the end of the session we will simply remove the folder to clean everything up:
```
 mktemp --directory
 cd </path/to/temp/folder>
```

Now let's download the git repository containing the notebook:
```
 git clone https://github.com/spsrc/droplets.git
```

First, do we have Jupyter installed?
```
 which jupyter
```

We don't. Instead of using the operating system's package manager (e.g. `apt-get`) or performing a manual installation, let's use `conda`!
```
 curl --output Miniconda.sh https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
 bash Miniconda.sh -b -p conda-install
 source conda-install/etc/profile.d/conda.sh
 conda install mamba --channel conda-forge --yes
```
```

Markdown language

Useful and efficient syntax
For .md files, issues, pull requests, etc.

droplets / sessions / droplets-02-conda / droplets-02-conda.md in master Cancel changes

Edit file Preview Show diff

Open Science Droplets 02

Keep your software organized with Conda

Our collaborator shared a Jupyter notebook with us and we want to re-run it and change a few things.

Install Jupyter using conda

Let's work on a temporary folder so at the end of the session we will simply remove the folder to clean everything up:

```
mktemp --directory
cd </path/to/temp/folder>
```

Now let's download the git repository containing the notebook:

```
git clone https://github.com/spsrc/droplets.git
```

First, do we have Jupyter installed?

```
which jupyter
```

We don't. Instead of using the operating system's package manager (e.g. `apt-get`) or performing a manual installation, let's use `conda`!

README

[Example README editor]

First thing a visitor sees

Describe what is this, for who and why

Describe the structure or add links to relevant files

Include instructions

Project information

Commit history

main

Commits on Jan 30, 2022

Merge pull request #22349 from QuLogic/luatex-kpsewhich ··· timhoffm committed 15 hours ago ✓	Verified	6eba0af
Merge pull request #22025 from anntzer/wxf ··· greglucas committed yesterday ✓	Verified	5722182

Commits on Jan 29, 2022

Deprecate various custom FigureFrameWx attributes/methods. ··· anntzer committed yesterday ✓		b463a3b
Merge pull request #21391 from anntzer/jpegbkg ··· greglucas committed yesterday ✓	Verified	30db833
Merge pull request #22026 from anntzer/wxc ··· greglucas committed yesterday ✓	Verified	f44c225
Merge pull request #22351 from anntzer/tw ··· timhoffm committed yesterday ✗	Verified	d771226
Fix "trailing" whitespace in C docstrings. ··· anntzer committed yesterday ✓		a9d12de
Merge pull request #22342 from anntzer/_qhull ··· timhoffm committed yesterday ✓	Verified	dc1e9c1

Issues

1,462 Open ✓ 6,821 Closed

Author Label Projects Milestones Assignee Sort

- [Doc]: Deprecation note in `get_cmap` is non-actionable Documentation
  #22362 opened 9 hours ago by mwaskom — 1
- [Bug]: text.usetex Vs. DateFormatter topic: date handling topic: text/usetex
  #22350 opened 2 days ago by leocirto — 1 — 3
- [ENH]: Having more customization options for the FancyArrow class (specifically different tails) New feature topic: arrow
  #22344 opened 2 days ago by mapfiable — 4
- [MNT]: Delay (or make pending) the deprecation of set_constrained_layout/set_tight_layout Maintenance
  #22343 opened 3 days ago by anntzer — 1 — 8
- [Bug]: GridSpec or related change between 3.4.3 and 3.5.1
  #22341 opened 3 days ago by KelSolaar — 2
- [Bug]: rcParams['legend.loc'] can't use float-tuple like kwarg legend(loc...) topic: rcparams
  #22338 opened 3 days ago by cphlewis
- [Bug]: Unable to twinx an axis with sharex=True
  #22335 opened 3 days ago by madphysicist — 4
- [Bug]: Data under cursor crashes on QuadMesh status: confirmed bug
  #22334 opened 3 days ago by Stassels v3.5.2 — 3
- [Bug]: First and or last minor ticks sometimes not plotted status: confirmed bug
  #22331 opened 4 days ago by bproxauf — 1

Pull/merge requests

313 Open ✓ 13,745 Closed

Author ▾ Label ▾ Projects ▾ Milestones ▾ Reviews ▾ Assignee ▾ Sort ▾

Revert datetime usetex ticklabels to use default tex font. ✕ topic: date handling topic: text/usetex topic: ticks axis labels 1
#22361 opened 12 hours ago by anntzer • Review required 6 tasks

Let TeX handle multiline strings itself. ✕ status: waiting for other PR topic: text/usetex
#22360 opened 12 hours ago by anntzer • Review required 6 tasks

Slightly refactor TeX source generation. ✕ Maintenance topic: text/usetex
#22359 opened 12 hours ago by anntzer • Approved 6 tasks

Cleanup tripcolor() ✓ Maintenance 1 2
#22356 opened yesterday by timhoffm • Review required v3.6.0

Auto font selection for system locale ✕
#22355 opened yesterday by kmivan • Review required 4 of 6 tasks

correctly treat pan/zoom events of overlapping axes ✕ 13
#22347 opened 2 days ago by raphaelquast • Review required 2 of 6 tasks v3.6.0

MNT: make layout deprecations pending ✓ Maintenance 1 5
#22345 opened 2 days ago by jklymak • Approved 6 tasks v3.6.0

Workflow management systems

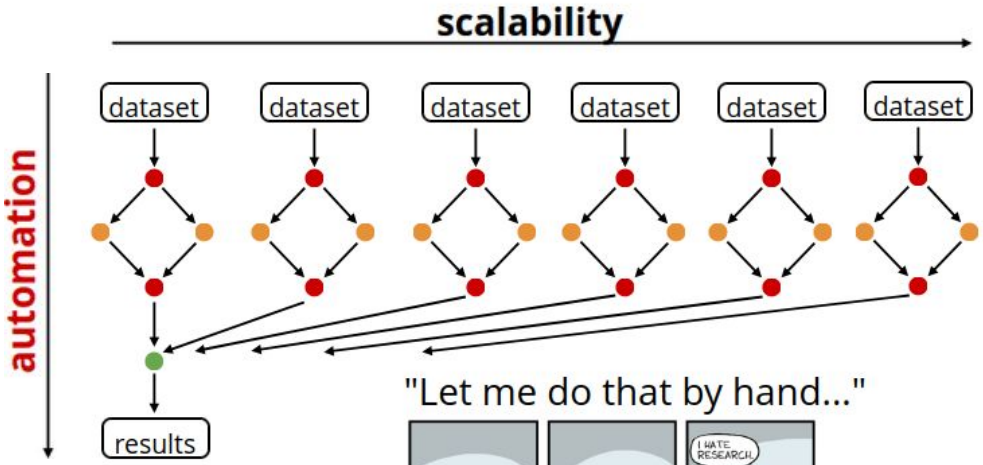
snakemake

Data analysis

"Let me do that by hand..."

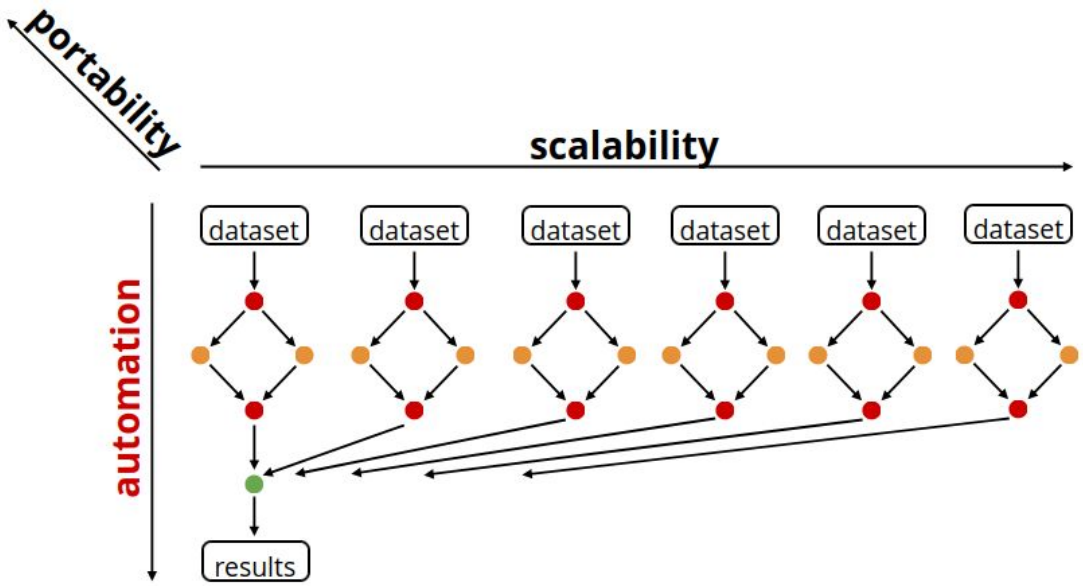
https://slides.com/johanneskoester/ismb-snakemake-tutorial-2019

Data analysis

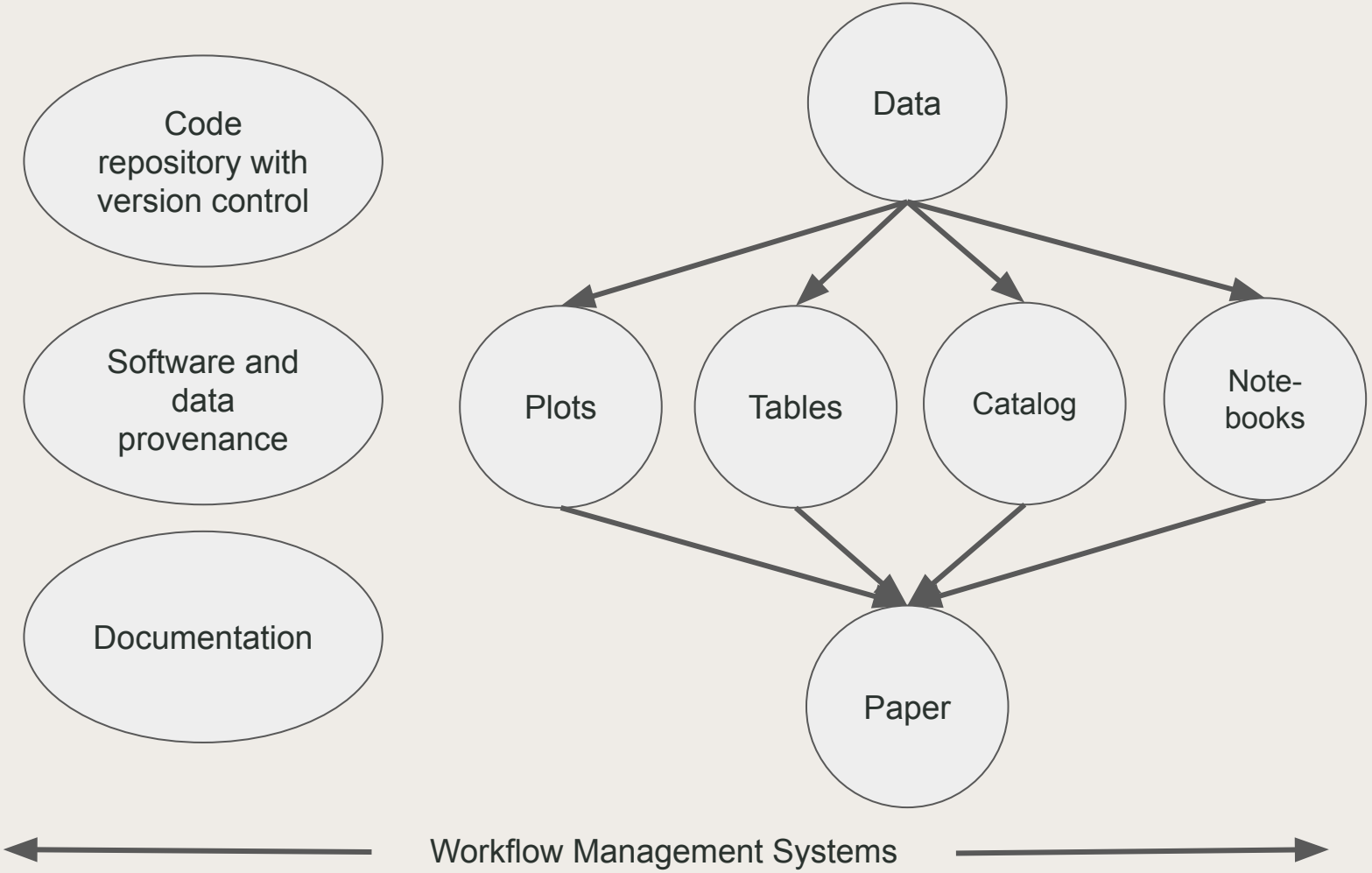


"Let me do that by hand..."

https://slides.com/johanneskoester/ismb-snakemake-tutorial-2019



https://slides.com/johanneskoester/ismb-snakemake-tutorial-2019


Code repository with version control
Software and data provenance
Documentation
Data
Plots
Tables
Catalog
Note-books
Paper
Workflow Management Systems

Workflow management systems

Why workflows?

- Allow you to create, manage, and execute workflows for data analysis
- From raw data to figures/paper automatically
- Atomic steps, more intuitive logic of analysis
- Re-run half-way through
- Re-run same analysis many times. Manage parallelization
- Create workflows that are transparent, shareable, and reusable
- Tools for dependency management (tasks and software packages)

Adapted from S. Luna-Valero https://www.youtube.com/watch?v=ktZf7sze1uq

Existing workflow systems

Existing Workflow systems

Stian Soiland-Reyes edited this page 2 days ago · 343 revisions

Permalink: https://s.apache.org/existing-workflow-systems

Cite as (update dates):

Peter Amstutz, Maxim Mikheev, Michael R. Crusoe, Nebojša Tijanić, Samuel Lampa, et al. (2022): **Existing Workflow systems.** *Common Workflow Language wiki*, GitHub. https://s.apache.org/existing-workflow-systems updated 2023-05-03, accessed 2023-05-03.

Computational Data Analysis Workflow Systems

An incomplete list

Please add new entries at the bottom.

In addition to this list, actively developed free/open-source systems should be registered at https://workflows.community/systems

See also: https://github.com/pditommaso/awesome-pipeline

1. Arvados - CWL-based distributed computing platform for data analysis on massive data sets. https://arvados.org/ https://github.com/arvados/arvados
2. Apache Taverna http://www.taverna.org.uk/ https://taverna.incubator.apache.org/
3. Galaxy http://galaxyproject.org/
4. SHIWA https://www.shiwa-workflow.eu/
5. Apache Oozie https://oozie.apache.org/
6. DNANexus https://wiki.dnanexus.com/API-Specification-v1.0.0/IO-and-Run-Specifications https://wiki.dnanexus.com/API-Specification-v1.0.0/Workflows-and-Analyses
7. ~~BioDT http://www.biodatomics.com/~~ archived at https://web.archive.org/web/20180609011656/http://www.biodatomics.com/
8. Agave http://agaveapi.co/live-docs/
9. DiscoveryEnvironment http://www.iplantcollaborative.org/ci/discovery-environment

https://github.com/common-workflow-language/common-workflow-language/wiki/Existing-Workflow-systems

Why snakemake?

Why?

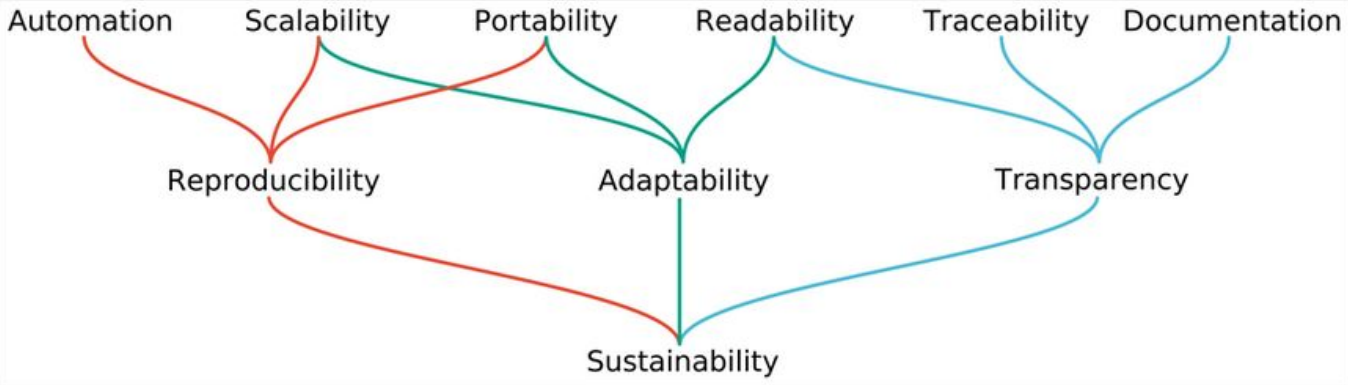
- Snakemake is **based on Python** and uses a domain-specific language to define workflows
- **Easy** to learn (subjective)
- **Scalability**: runs on laptop or multi-core servers (e.g. slurm)
- Easy to **manage software** (specific conda or singularity per task)
- Snakemake provides tools for **visualizing** the job flow
- **Handles boilerplate**: common tasks such as parallelization, suspend/resume, logging

Adapted from S. Luna-Valero https://www.youtube.com/watch?v=ktZf7sze1uq

snakemake

To manage packages, dependencies and execution

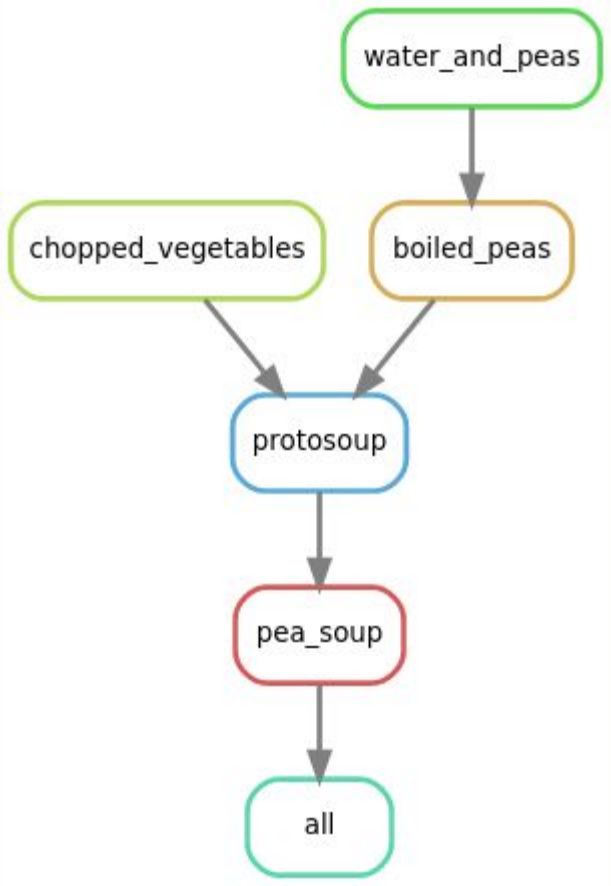
(~make in python)

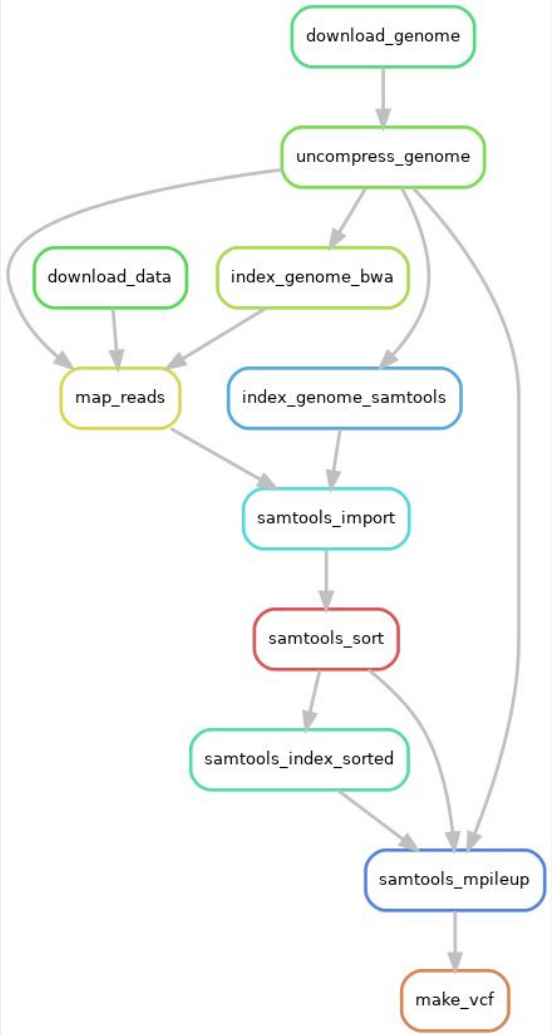


Mölder et al. F1000Research 2021, 10:33

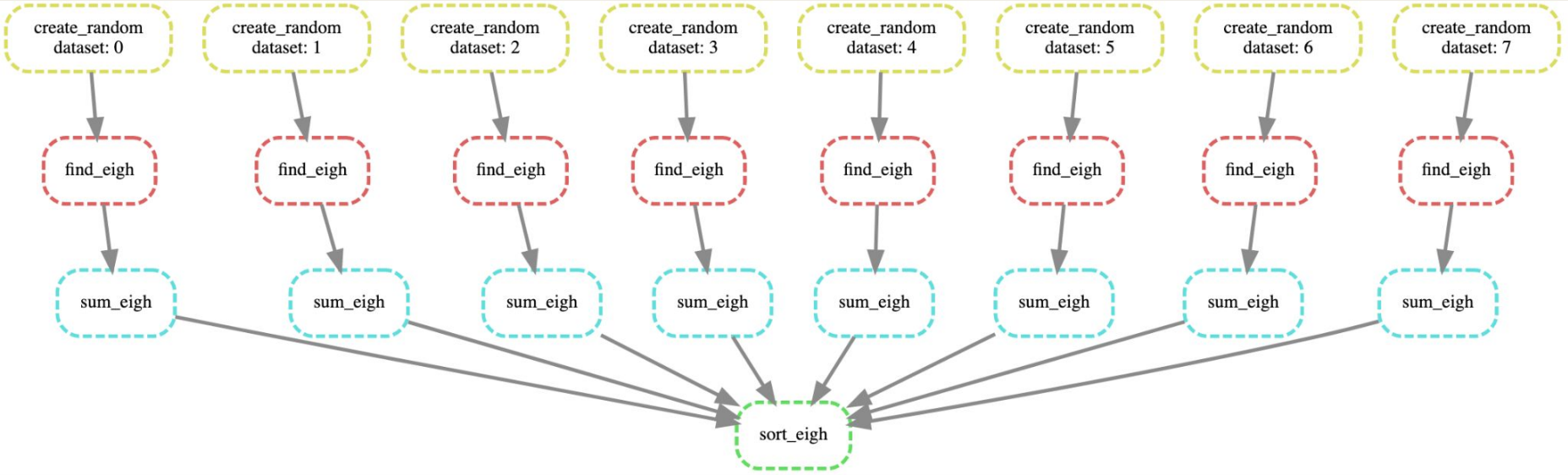
Some DAG examples

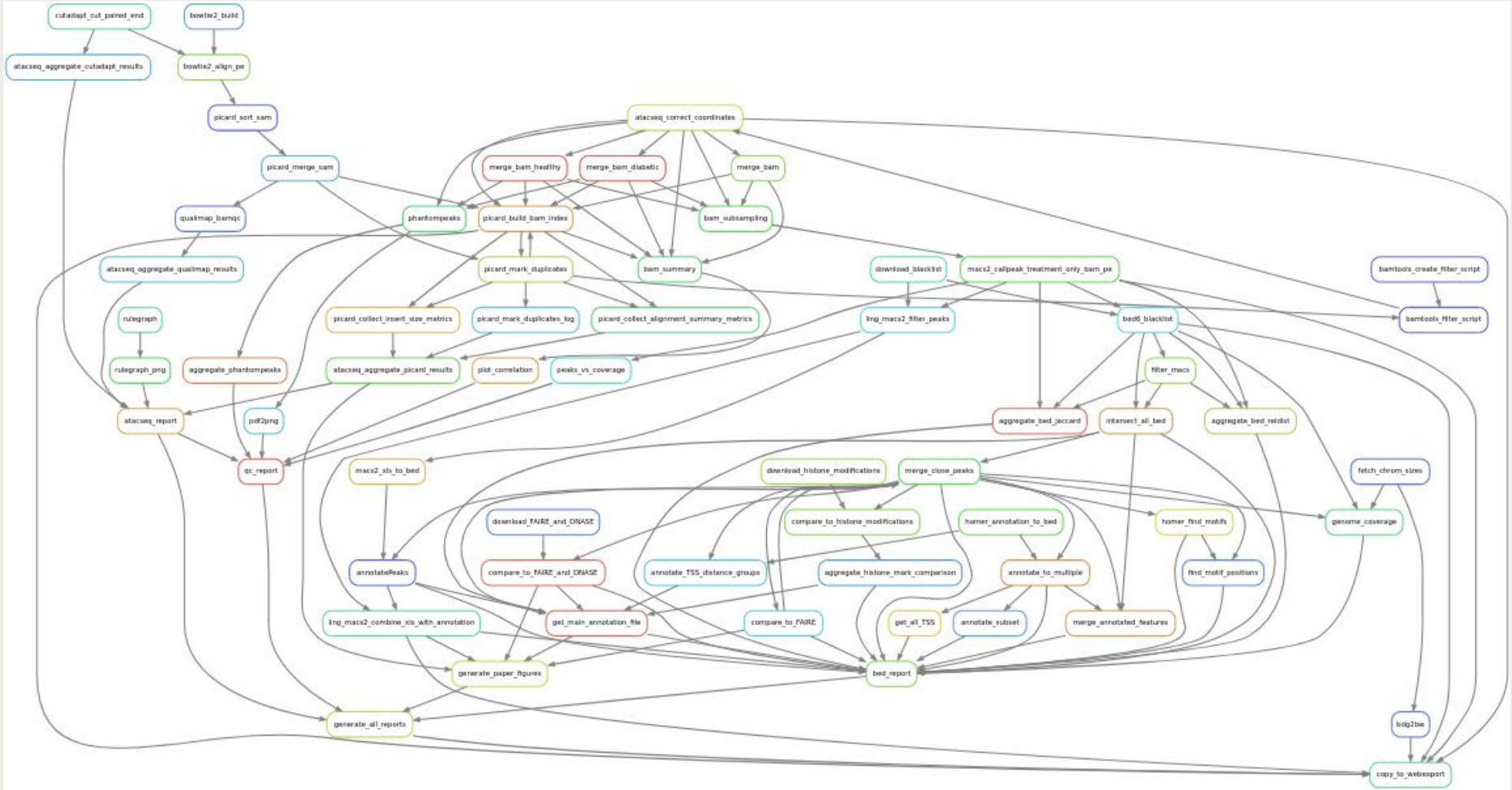
(think input → output)


water_and_peas
chopped_vegetables
boiled_peas
protosoup
pea_soup
all


download_genome
uncompress_genome
download_data
index_genome_bwa
map_reads
index_genome_samtools
samtools_import
samtools_sort
samtools_index_sorted
samtools_mpileup
make_vcf


download_reference
specie: Salmonella-enterica.trim
download_sra
specie: Salmonella-enterica.trim
download_reference
specie: Salmonella-enterica
download_sra
specie: Salmonella-enterica
index_reference
extract_sra
index_reference
extract_sra
bwa_mem
bwa_mem
bam_trim
specie: Salmonella-enterica
bam
bai_trim
decompress_reference
bai
callWithTrimming
callVanilla
all


create_random
dataset: 0
create_random
dataset: 1
create_random
dataset: 2
create_random
dataset: 3
create_random
dataset: 4
create_random
dataset: 5
create_random
dataset: 6
create_random
dataset: 7
find_eigh
find_eigh
find_eigh
find_eigh
find_eigh
find_eigh
find_eigh
find_eigh
sum_eigh
sum_eigh
sum_eigh
sum_eigh
sum_eigh
sum_eigh
sum_eigh
sum_eigh
sort_eigh


cutadapt_cut_paired_end
bowtie2_build
atacseq_aggregate_cutadapt_results
bowtie2_align_pe
picard_sort_sam
atacseq_correct_coordinates
picard_merge_sam
merge_bam_healthy
merge_bam_diabetic
merge_bam
qualimap_bamqc
phantompeaks
picard_build_bam_index
bam_subsampling
atacseq_aggregate_qualimap_results
picard_mark_duplicates
bam_summary
download_blacklist
macs2_callpeak_treatment_only_bam_pe
bamtools_create_filter_script
rulegraph
picard_collect_insert_size_metrics
picard_mark_duplicates_log
picard_collect_alignment_summary_metrics
lmg_macs2_filter_peaks
bed6_blacklist
bamtools_filter_script
rulegraph_png
aggregate_phantompeaks
atacseq_aggregate_picard_results
plot_correlation
peaks_vs_coverage
filter_macs
atacseq_report
pdf2png
aggregate_bed_jaccard
intersect_all_bed
aggregate_bed_reldist
qc_report
macs2_xls_to_bed
download_histone_modifications
merge_close_peaks
fetch_chrom_sizes
download_FAIRE_and_DNASE
compare_to_histone_modifications
homer_annotation_to_bed
homer_find_motifs
genome_coverage
annotatePeaks
compare_to_FAIRE_and_DNASE
annotate_TSS_distance_groups
aggregate_histone_mark_comparison
annotate_to_multiple
find_motif_positions
lmg_macs2_combine_xls_with_annotation
get_main_annotation_file
compare_to_FAIRE
get_all_TSS
annotate_subset
merge_annotated_features
generate_paper_figures
bed_report
generate_all_reports
bdg2bw
copy_to_webexport

```
rule count_countries:
    input:
        "european-countries.txt"
    output:
        "number-of-countries.txt"
    shell:
        "wc --lines european-countries.txt >
    number-of-countries.txt"
```

```
$ cat european-countries.txt
Netherlands
Greece
Spain
Portugal
Italy
Poland
Austria
```

Adapted from S. Luna-Valero https://www.youtube.com/watch?v=ktZf7sze1uq

```
rule count_countries:
    input:
        "european-countries.txt"
    output:
        "number-of-countries.txt"
    shell:
        "wc --lines {input} > {output}"
```

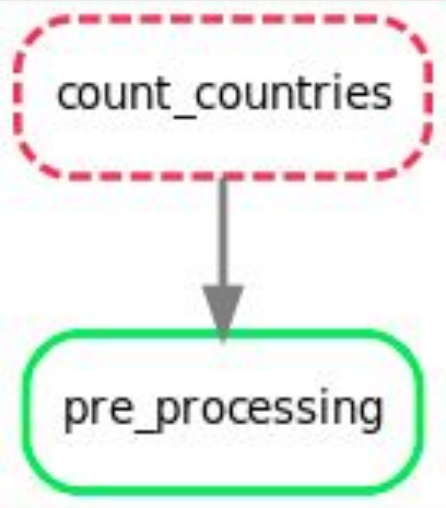
```
$ cat european-countries.txt
Netherlands
Greece
Spain
Portugal
Italy
Poland
Austria
```

```
rule count_countries:
   input:
      "european-countries.txt",
      "other-countries.txt"
   output:
      "number-of-countries.txt"
   shell:
      "wc --lines {input} > {output}"
```

```
$ cat european-countries.txt
Netherlands
Greece
Spain
Portugal
Italy
Poland
Austria
```

```
rule count_countries:
    input:
        "european-countries.txt",
        "other-countries.txt"
    output:
        "number-of-countries.txt"
    shell:
        "wc --lines {input} > {output}"

rule pre_processing:
    input:
        "number-of-countries.txt"
    output:
        "my_plot.png"
    shell:
        "python myscript.py {input}"
```



```
rule count_countries:
    input:
        "european-countries.txt",
        "other-countries.txt"
    output:
        "number-of-countries.txt"
    conda:
          "envs/count_countries.yaml"
    shell:
        "wc --lines {input} > {output}"
```

```
rule count_countries:
   input:
      "european-countries.txt",
      "other-countries.txt"
   output:
      "number-of-countries.txt"
   container:
         "docker://repo/image"
   shell:
      "wc --lines {input} > {output}"
```

```
rule count_countries:
    input:
        "european-countries.txt",
        "other-countries.txt"
    output:
        "number-of-countries.txt"
    shell:
        "wc --lines {input} > {output}"
```

Remote files:
http
ftp
zenodo
Google
Dropbox
AWS

Outputs:
temporary
protected
directory
ensure checksum
pipe

Execution:
shell
script
bash
notebook

Launch:
local
cloud
Kubernetes
slurm
mpi

```
rule count_countries:
    input:
        "european-countries.txt",
        "other-countries.txt"
    output:
        "Number-of-countries.txt"
    container:
        "docker://repo/image"
    log:
        "logs/count_countries.log"
    benchmark:
        "benchmarks/count_countries.txt"
    report:
        "report("fig1.png", caption="fig1.rst")"
    params:
        option="--lines"
    resources:
        mem_mb=100
        nvidia_gpu=1
    threads: 8
    priority: 50
    shell:
        "wc {params.option} {input} > {output}"
```

Show your work!

Workflow management tool for open source scientific articles

- **Based on snakemake**. Final output is ms.pdf
- Github action to generate the pdf
- Can use **Zenodo as archive** for heavy files
- Integration with **overleaf**
- Prepares files for **arXiv** submission
- Links figures in pdf to actual script
- https://show-your.work/en/latest

```
\begin{figure}
    \begin{centering}
        \includegraphics{figures/mandelbrot.pdf}
        \caption{This is a pretty visualization of the Mandelbrot set.}
        \label{fig:mandelbrot}
        \script{mandelbrot.py}
    \end{centering}
\end{figure}
```

Demo time

Example of a simple reproducible paper using snakemake
https://snakemake.readthedocs.io/en/stable/

A quick example

SKA Data Challenge 2

The SKA Data challenges

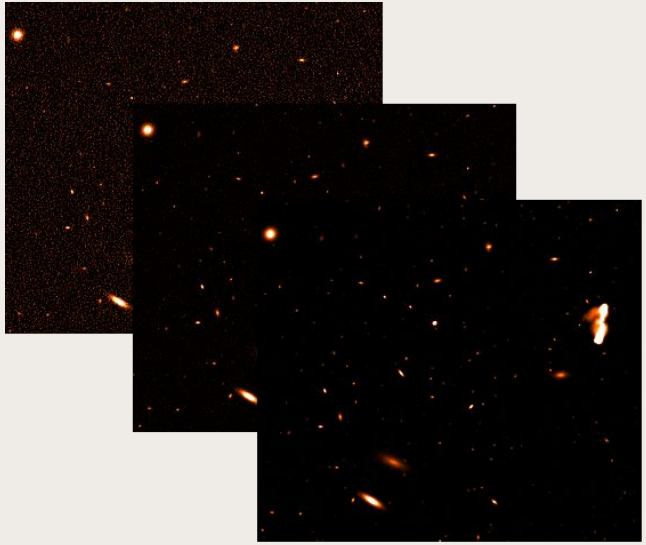
Preparatory activities for the science community. Familiarize with SKA data products and optimize analysis

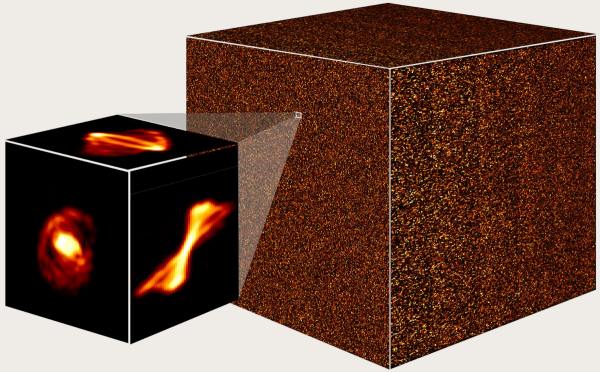
SDC1 (2018)

Continuum images at 3 bands simulating 1000 h of data. Source finding + characterization + identification

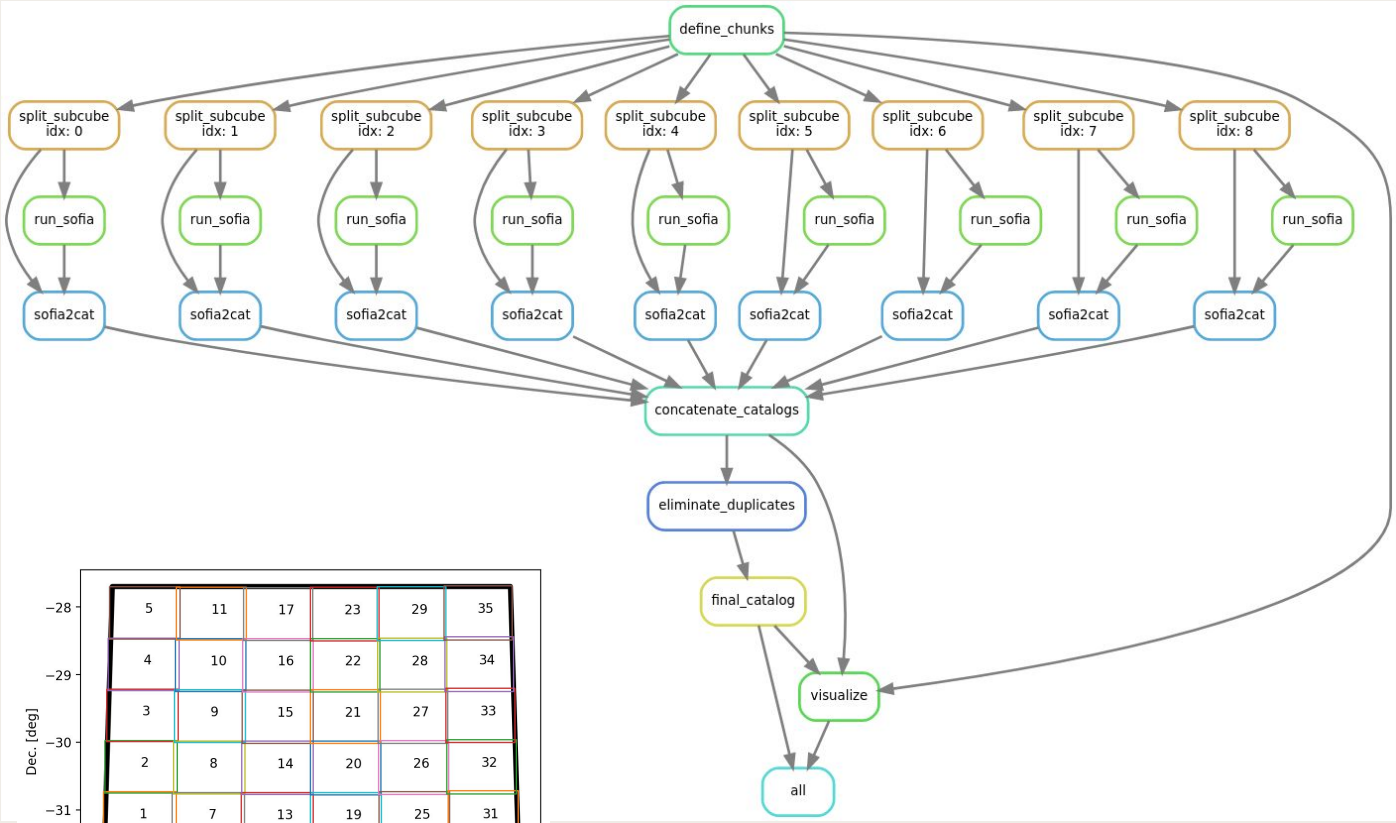
SDC2 (2021)

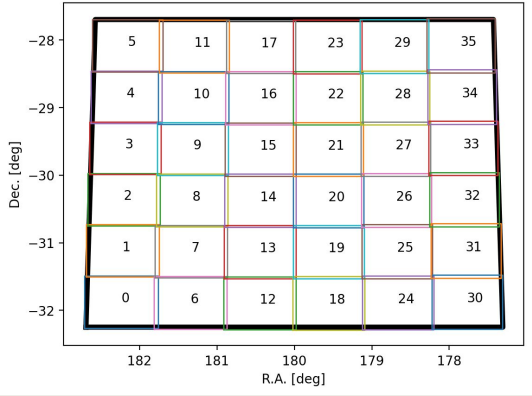
Multifrequency source finding and characterization of HI emitting galaxies. 1 TB data cube





An SKA Data Challenge example

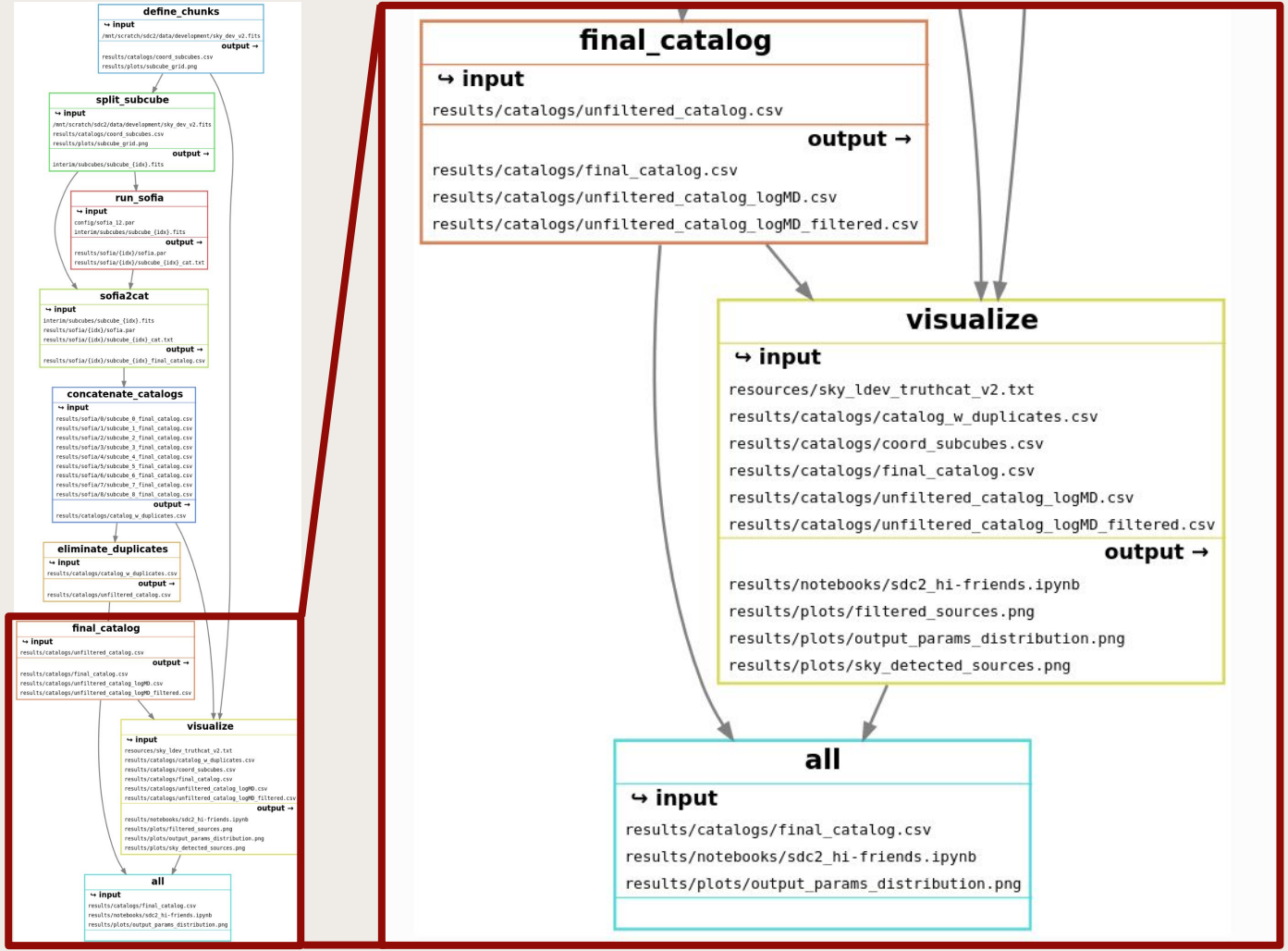




File traceability

Helps anyone to understand the logic and flow of the analysis.

Benefits modularization



File structure

Using a template is very helpful to organize ideas and start the project

Workflow file structure

```
workflow/
├── Snakefile
├── envs
│   ├── analysis.yml
│   ├── chunk_data.yml
│   ├── filter_catalog.yml
│   ├── process_data.yml
│   ├── snakemake.yml
│   └── xmatch_catalogs.yml
├── notebooks
│   └── sdc2_hi-friends.ipynb
├── rules
│   ├── chunk_data.smk
│   ├── concatenate_catalogs.smk
│   ├── run_sofia.smk
│   ├── summary.smk
│   └── visualize_products.smk
├── scripts
│   ├── define_chunks.py
│   ├── eliminate_duplicates.py
│   ├── filter_catalog.py
│   ├── run_sofia.py
│   ├── sofia2cat.py
│   └── split_subcube.py
```

Results file structure

```
results/
├── catalogs
├── logs
├── notebooks
├── plots
└── sofia
```

```
logs/
├── concatenate
│   ├── concatenate_catalogs.log
│   ├── eliminate_duplicates.log
│   └── filter_catalog.log
├── define_chunks
│   └── define_chunks.log
├── run_sofia
│   ├── subcube_0.log
│   ├── subcube_1.log
│   ├── subcube_2.log
│   └── subcube_3.log
├── sofia2cat
│   ├── subcube_0.log
│   ├── subcube_1.log
│   ├── subcube_2.log
│   └── subcube_3.log
├── split_subcube
│   ├── subcube_0.log
│   ├── subcube_1.log
│   ├── subcube_2.log
│   └── subcube_3.log
└── visualize
    └── visualize.log
```

Documentation

Version-controlled documentation deployed with readthedocs

HI-FRIENDS SDC2

latest

Search docs

The SKA Data Challenge 2
Methodology
Workflow Description
Workflow installation
Workflow execution
SDC2 HI-FRIENDS results
SDC2 Reproducibility award
Developers
Acknowledgments

» Welcome to HI-FRIENDS SDC2's documentation!

Edit on GitHub

Welcome to HI-FRIENDS SDC2's documentation!

- The SKA Data Challenge 2
  - The HI-FRIENDS solution to the SDC2
  - Workflow general description
  - The HI-FRIENDS team
- Methodology
  - Data exploration
  - Feedback from the workflow and logs
  - Configuration
  - Unit tests
  - Software managed and containerization
  - Check conformance to coding standards
- Workflow Description
  - Workflow definition diagrams
  - Workflow file structure
  - Output products
  - Snakemake execution and diagrams
- Workflow installation
  - Dependencies
  - Installation

master

hi-friends / docs /

jmoldon fix typo in link to binder in the documentation

..

source

Makefile

make.bat

requirements.txt

sphinx.yml

Developers docs

Populated directly from the scripts using Sphinx

» Developers » filter_catalog module

Edit on GitHub

filter_catalog module

This script filters the output catalog based on some conditions

filter_catalog.arcsec2kpc(*redshift, theta*)

Converts angular size to linear size given a redshift

Parameters:
- **redshift** (*float*) – redshift
- **theta** (*array of floats*) – angular size in arcsec

Returns: **distance_kpc** – linear size in kpc

Return type: array of floats

filter_catalog.compute_d_m(*cat*)

Computes the Mass of HI and linear diameter of the galaxies in a catalog

Parameters: **cat** (*pandas.DataFrame*) – catalog of galaxies

Returns: **cat** – original catalog adding the columns log(M_HI) and log(D_HI_kpc)

Return type: pandas.DataFrame

Use standard Open Science related files

- CITATION.cff
- CODE_OF_CONDUCT.md
- CONTRIBUTING.md
- LICENSE
- README.md

License GPL v3 | snakemake ≥6.5.3 | docs passing | DOI 10.5281/zenodo.5167659

openssf best practices in progress 88% | Contributor Covenant 2.1

launch binder | fair-software.eu

Manuel Parra's talk this afternoon

Interoperable

- Workflow installation
  - Dependencies
  - Installation
  - Deploy in containers
    - Docker
    - Singularity
    - Podman
  - Use tarball of the workflow
  - Use myBinder

WorkflowHub | Browse | Search here... | Search

HI-FRIENDS HI data cube source finding and characterization Version 1

Overview | Files | Related items

Workflow Type: Snakemake

Stable

zenodo | Search | Upload | Communities

Celebrating our 10th anniversary! Send us your birthday greeting here.

August 6, 2021

Software | Open Access

HI-FRIENDS participation in the SKA Data Challenge 2

Moldon, Javier; Darriba, Laura; Verdes-Montenegro, Lourdes; Kleiner, Dane; Sanchez, Susana; Kamphuis, Peter; Parra, Manuel; Józsa, Gyula; Garrido, Julian; Cannon, John; Jones, Michael; Akhlaghi, Mohammad; Sorgho, Amidou; Sabater, José; Pandey-Pommier, Mamta; Alberdi, Antonio; Márquez, Isabel; Gomez, Jose Francisco; Herranz, Diego

This repository contains the workflow used to find and characterize the HI sources in the data cube of the SKA Data Challenge 2. This is developed by the HI-FRIENDS team. The execution of the workflow was conducted in the SP-SRC cluster at the IAA-CSIC. Documentation can be found in HI-FRIENDS SDC2 Documentation. The workflow is maintained in the hi-friends Github repository. It is also published in WorkflowHub.

The repository includes the following files:

- **HI-FRIENDS-SDC2/hi-friends-1.0.0.zip**. A copy of the Github repository https://github.com/HI-FRIENDS-SDC2/hi-friends
- **hi-friends-sdc2-workflow.tar.gz**. Self-contained workflow archive produced by snakemake containing the code, the config files, and all software packages of each defined conda environment. The only dependency to use it is snakemake. Works on Linux, and has been testes on Ubuntu 20.04.
- **hi-friends-wf.sif**. Singularity image of the whole workflow. To execute it, follow instruction in https://hi-friends-sdc2.readthedocs.io/en/latest/installation.html#id1
- **hi-friends-wf.tgz**. Docker container of the whole workflow.
- **hi-friends_solution.tgz**. HI-FRIENDS solution, including code, catalogs, cubelets, etc.

Software Heritage Archive | Browse the archive

Features
- Search
- Downloads
- Save code now
- Add forge now
- Help

https://github.com/HI-FRIENDS-SDC2/hi-friends

13 April 2023, 05:01:25 UTC

Code | Branches (2) | Releases (0) | Visits

Branch: HEAD | 1c56c72 /

Tip revision: d77378f3cdda2a36346f8e8b0abd2603a2ab873f

Update README.md

Reproducibility checklists

SDC2 reproducibility award

List of criteria: link

- Well-documented
- Easy to install
- Easy to use
- Open license
- Accessible code
- Code standards
- Testing

Criteria: https://drive.google.com/file/d/1B2mZ_SYwktvXY-Rbdf0OgIlPuyHBs2UW/view

	Reproducibility of the solution Can the software pipeline be re-run easily to produce the same results? Is it: • Well-documented *Research software documentation best practice* • Easy to install *Top tips for packaging software* • Easy to use *Top tips for documentation*
Well-documented	High-level description of what/who the software is for is available
	High-level description of what the software does is available
	High-level description of how the software works is available
	Documentation consists of clear, step-by-step instructions
	Documentation gives examples of what the user can see at each step e.g. screenshots or command-line excerpt
	Documentation uses `monospace` fonts for command-line inputs and outputs, source code fragments, function names, class names etc
	Documentation is held under version control alongside the code
Easy to install	Full instructions provided for building and installing any software
	All dependencies are listed, along with web addresses, suitable versions, licences and whether they are mandatory or optional
	All dependencies are available
	Tests are provided to verify that the installation has succeeded
	A containerised package is available, containing the code together with all of the related configuration files, libraries, and dependencies required. *Using .e.g. Docker/Singularity*
Easy to use	A getting started guide is provided outlining a basic example of using the software *e.g. a **README file***
	Instructions are provided for many basic use cases
	Reference guides are provided for all command-line, GUI and configuration options

to develop new projects? Does it:

source licence
using a repository for your project
dable source code

	e.g. GNU General Public License (GPL), BSD 3-Clause
Accessible code	Licence is stated in source code repository
	Each source code file has a licence header
	Access to source code repository is available online
	Repository is hosted externally in a sustainable third-party repository *e.g. SourceForge, LaunchPad, GitHub: Introduction to GitHub*
Code standards	Documentation is provided for developers
	Source code is laid out and indented well
	Source code is commented
	There is no commented out code
	Source code is structured into modules or packages
	Source code uses sensible class, package and variable names
Testing	Source code structure relates clearly to the architecture or design
	Source code has unit tests
	Software recommends tools to check conformance to coding standards *e.g. A 'linter' such as PyLint for Python*

Software Sustainability Institute

Conclusions

What to do next?

First things first

Start documenting
(README, code …)

Use version control software repositories

Manual → automatic
Code ←/→ params

Share your work
(Zenodo, ASCL, WorkflowHub, SWH)

Follow a reproducibility checklist

	Reproducibility of the solution Can the software pipeline be re-run easily to produce the same results? Is it: • Well-documented *Research software documentation best practice* • Easy to install *Top tips for packaging software* • Easy to use *Top tips for documentation*	
Well-documented	High-level description of what/who the software is for is available	
	High-level description of what the software does is available	
	High-level description of how the software works is available	
	Documentation consists of clear, step-by-step instructions	
	Documentation gives examples of what the user can see at each step e.g. screenshots or command-line excerpt	
	Documentation uses `monospace` fonts for command-line inputs and outputs, source code fragments, function names, class names etc	
	Documentation is held under version control alongside the code	
Easy to install	Full instructions provided for building and installing any software	
	All dependencies are listed, along with web addresses, suitable versions, licences and whether they are mandatory or optional	
	All dependencies are available	
	Tests are provided to verify that the installation has succeeded	
	A containerised package is available, containing the code together with all of the related configuration files, libraries, and dependencies required. *Using .e.g. Docker/Singularity*	
Easy to use	A getting started guide is provided outlining a basic example of using the software *e.g. a README file*	
	Instructions are provided for many basic use cases	
	Reference guides are provided for all command-line, GUI and configuration options	

	Reusability of the pipeline Can the code be reused easily by other people to develop new projects? Does it: • Have an open licence *Choosing an open source licence* • Have easily accessible source code *Choosing a repository for your project* • Adhere to coding standards *Writing readable source code* • Utilise tests *Testing your software*	
Open licence	Software has an open source licence *e.g. GNU General Public License (GPL), BSD 3-Clause*	
	Licence is stated in source code repository	
	Each source code file has a licence header	
Accessible code	Access to source code repository is available online	
	Repository is hosted externally in a sustainable third-party repository *e.g. SourceForge, LaunchPad, GitHub: Introduction to GitHub*	
	Documentation is provided for developers	
Code standards	Source code is laid out and indented well	
	Source code is commented	
	There is no commented out code	
	Source code is structured into modules or packages	
	Source code uses sensible class, package and variable names	
	Source code structure relates clearly to the architecture or design	
Testing	Source code has unit tests	
	Software recommends tools to check conformance to coding standards *e.g. A 'linter' such as PyLint for Python*	

Get inspired: Workflows

Snakemake workflow catalog

Snakemake workflow catalog

A comprehensive catalog of standards

Standardized usage 155 | All workflows 2214

	Workflow	Description	Topics	QC
Usage	snakemake-workflows/rna-seq-star-deseq2	RNA-seq workflow using STAR and DESeq2	snakemake, sciworkflows, reproducibility, gene-expression-analysis, deseq2	license MIT, last commit last saturday, linting passed, formatting passed
Usage	snakemake-workflows/dna-seq-gatk-variant-calling	This Snakemake pipeline implements the GATK best-practices workflow	reproducibility, snakemake, sciworkflows, genomic-variant-calling, gatk	license MIT, last commit may 2021, linting passed, formatting failed
Usage	franciscozorrilla/metaGEM	:gem: An easy-to-use workflow for generating context specific genome-scale metabolic models and predicting metabolic interactions within microbial communities directly from metagenomic data	metagenomics, computational-biology, metabolic-models, gut-microbiome, snakemake, community-simulations, metagenome-assembled-genomes, microbial-communities, mags, metabolism, bioinformatics, fba, flux-balance-analysis, genome-scale-metabolic-model, genome-scale-model, metabolic-modeling	license MIT, last commit march, linting failed, formatting failed
Usage	snakemake-workflows/dna-seq-varlociraptor	A Snakemake workflow for calling small and structural variants under any kind of scenario (tumor/normal, tumor/normal/relapse, germline, pedigree, populations) via the unified statistical model of Varlociraptor.	varlociraptor, sciworkflows, snakemake, reproducibility, genomic-variant-calling	license MIT, last commit today, linting passed, formatting passed

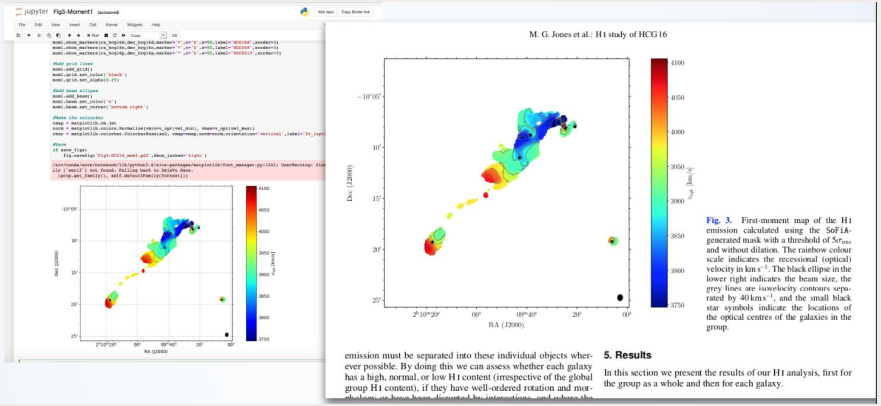
https://snakemake.github.io/snakemake-workflow-catalog/?rules=true

Get inspired: papers with repositories

+ virtually all recent papers from Mohammad Akhlaghi and close collaborators

Jones+2019: Evolution of compact groups from intermediate to final stages. A case study of the H I content of HCG 16

https://github.com/robertdstein/at2019dsg/tree/master



Stein+2020: A tidal disruption event coincident with a high-energy neutrino

https://github.com/robertdstein/at2019dsg/tree/master

Ramos-Padilla+2021: The viewing angle in AGN SED models, a data-driven analysis

https://github.com/aframosp/AGNView

Reprohacks

Events to review the reproducibility of papers

Community

ReproHack Hub: https://www.reprohack.org/

Papers to review: https://www.reprohack.org/paper/

Resources

- Snakemake documentation https://snakemake.readthedocs.io/en/stable/index.html
- Open Science Droplets (IAA) https://droplets-spsrc.readthedocs.io/
- Reproducibility course @ CSIC https://github.com/spsrc/reproducibility-course
- SKA Data Challenge reproducibility criteria: https://drive.google.com/file/d/1B2mZ_SYwktvXY-Rbdf0OgIIPuyHBs2UW/view
- The Turing Way https://the-turing-way.netlify.app/index.html
- A Survey on Adoption Guidelines for the FAIR4RS Principles: Dataset https://zenodo.org/record/6375540
- NBIS Reproducible research course https://nbis-reproducible-research.readthedocs.io/en/course_1811/snakemake/
- Becoming a better scientist with open and reproducible research https://lgatto.github.io/open-and-rr-2/
- Creating an executable paper is a journey through Open Science https://www.nature.com/articles/s42005-020-00403-4
- Chapter 14 Managing Workflows with Snakemake https://eriqande.github.io/eca-bioinf-handbook/snakemake-chap.html
- Five recommendations Endorse for fair software https://fair-software.eu/
- Analysis preservation using Snakemake https://mstamenk.github.io/2017/08/snakemake-tutorial-for-data-analysts.html