

FAIR Research Data Management in a git-like manner with Datalad

HZDR Data and Software Management Days 2024
2024-10-29 // Andreas Knüpfer, Timothy Callow / CASUS, HZDR



Intro

Why is it important? Why is it hard?

README metadata concept

Version Management

Datalad

Dataset inventory

Hands-on Tutorial

Tutorial Appendix

Next steps: Integration with HPC

Summary & Outlook

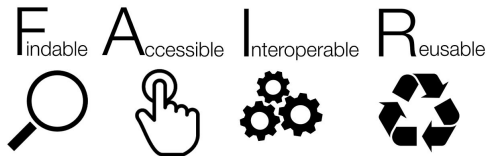
Scientific Data Management / Research Data Management is no *rocket science* but still a practical challenge in many fields and groups

Why is it important?

- Part of DFG's "good scientific practice" and you sign it with every funding project
- Reproducibility of your results
- Knowledge management in your team
- **Your** future self will profit from it

Why is it hard?

- There are very sophisticated (read "complicated") solutions (it is its own research field after all)
- ... no intuitive, generic/traditional community standards (yet)
- Easy to postpone task
- *At times it may seem you are doing this for other people only*



SangyaPundir, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=53414062>

- 0 Reliable storage
 - F1. Unique and persistent identifier
 - F2. Rich metadata
 - F3. Metadata include the identifier
 - F4. (Meta)data searchable
 - A1. (Meta)data retrievable by identifier via standardised protocol
 - A2. Metadata accessible, even when data no longer available
- 1 Collaborative access
 - I1. Formal, accessible, shared, and broadly applicable language
 - I2. (Meta)data use FAIR vocabularies
 - I3. Qualified references to other (meta)data
 - R1.1. Data usage license
 - R1.2. Detailed provenance
 - R1.3. Meet domain-relevant community standards

²<https://www.go-fair.org/fair-principles/>

- Personally, I've been searching and evaluating data management solutions for years, still finding new approaches and new software
- ... from buzzword bingo to ridiculously expensive to more or less Open Source

Outline: *Generic* proposal addressing the *practical-but-not-too-complex* challenge

- Start with the **README type metadata concept**
- Combine with very well established **git** and **gitlab** workflows
- Extend it for large and binary files with **git-annex**
- Simplify tooling with **Datalad**
- Add reproducibility features (also **Datalad**)
- Add tailored dataset inventory with **NocoDB**

There are many solutions available. Any is better than not using structured data + metadata management!

Science community portals (free)

- Dataverse <https://dataverse.org/>
- German NFDI portals (several)
- Materials Net
<https://materialsproject.org>
- Nomad <https://nomad-lab.eu/>
- ... many more

Commercial solutions

- Arcitecta MediaFlux <https://www.arcitecta.com/mediaflux/>
- Stronglink <https://stronglink.com/>
- Grau Data Metadata Hub
<https://www.graadata.com/>
- StarFish
<https://starfishstorage.com/>
- LakeFS <https://lakefs.io/>
- ...

README metadata concept

Step 1: Basic README metadata concept³

- For collections of uniform datasets
- Dataset == smallest self-contained subset (per specimen, per simulation run, ...)
- Each dataset in separate subdirectory (flat or tree but not nested)
- Place a README file in every dataset (plain text or markdown)
- ... use uniform structure, same categories, adopt community standards

Step 2: More advanced variants

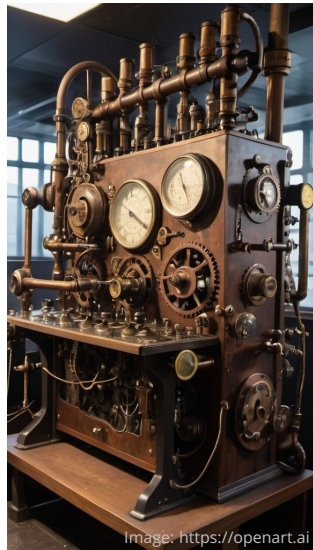
- Make README.md good/robust for automatic parsing
- Instead of README.md use DATASET.json (or any other structured format)
- Adhere to a JSON schema (community standards or self-defined)

▶ The index file represents the whole dataset dir with all files and subdirs

³<https://data.research.cornell.edu/data-management/sharing/readme/>

Automatically create the metadata files (DATASET.json or other) for all datasets in the data collection?

- Not manually, no complicated contraption either ...
- Custom extraction script as part of data collection
- Include every metadata field you want to search for
- Extract from input/output files and logs (JSON, YAML, TOML, ...)
- Extract from env. variables, Slurm log files, ...
- Date, time, location (hostname, cluster name)
- Software from modules, git repos, Spack, --version
- Store as key-value pairs, nested structures okay





- well established and well known
- Clear basic steps:
commit, push/pull, merge, ...
- Command line tools + graphical
interfaces + IDE integrations
- **But git is not for binary data
and not for very large data**



- Central repositories and web GUI
- Authentication and authorization
- Several alternatives like github,
Gogs, Gitea, Gin, ...

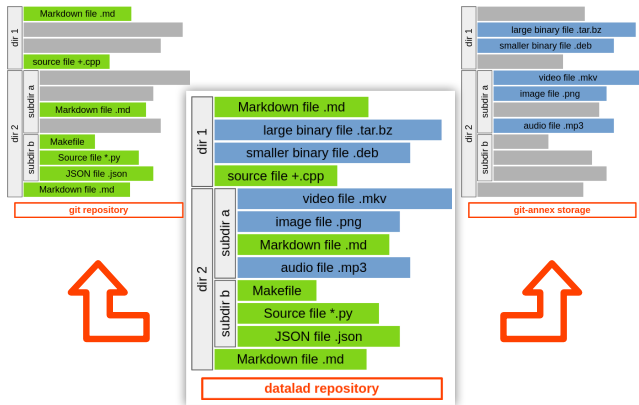


- git plugin adding new subcommands
- Manage large binary files via git, yet don't store file contents in git⁴
- All **text files** go to central git repository
- All **binary files** go to special git-annex storages (default with `text2git` template, can be configured, e.g. for large text files)
- **Tracking information** for annexed files goes to git repository, too
- Work almost as with normal git repository: all the known subcommands and IDE integrations work like expected
- Allows multiple special git-annex storages with different subsets of the files

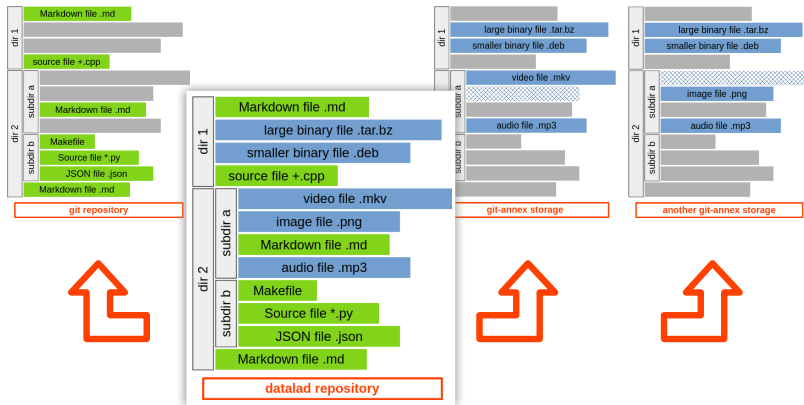
(I have no idea why I didn't come across git-annex earlier)

⁴<https://git-annex.branchable.com/>

git + git-annex: Text files and binary files go to separate storages:



git + git-annex: Text files and binary files go to separate storages:



```
knue@fwu082:~/data/atomistic_foundation_model.datalad > git annex list
here
|origin
||web
|||bittorrent
|||minio
||||
X__X A_/Au_/Au_/Au/exported_calculation_nodes.aiida
X__X A_/Au_/Au_/Au/training_data.json.gz
X__X B_/B_/B_/B/exported_calculation_nodes.aiida
X__X B_/B_/B_/B/training_data.json.gz
____X S_/SiH_/SiH4_/SiH4/exported_calculation_nodes.aiida
____X S_/SiH_/SiH4_/SiH4/training_data.json.gz
knue@fwu082:~/data/atomistic_foundation_model.datalad > git annex get
get S_/SiH_/SiH4_/SiH4/exported_calculation_nodes.aiida (from minio...)
(checksum...) ok
get S_/SiH_/SiH4_/SiH4/training_data.json.gz (from minio...)
(checksum...) ok
(recording state in git...)
knue@fwu082:~/data/atomistic_foundation_model.datalad > git annex list
here
|origin
||web
|||bittorrent
|||minio
||||
X__X A_/Au_/Au_/Au/exported_calculation_nodes.aiida
X__X A_/Au_/Au_/Au/training_data.json.gz
X__X B_/B_/B_/B/exported_calculation_nodes.aiida
X__X B_/B_/B_/B/training_data.json.gz
X__X S_/SiH_/SiH4_/SiH4/exported_calculation_nodes.aiida
X__X S_/SiH_/SiH4_/SiH4/training_data.json.gz
knue@fwu082:~/data/atomistic_foundation_model.datalad > |
```

- Well, a bit more complicated than basic git. Yet, nothing really compared with the merge vs. rebase wars of the early 3rd millennium.
- git-annex can use various storages: S3, AWS Glacier, webdav, git-lfs, ... and many more
- also FTP, IMAP or a bunch of USB sticks⁵
- git-annex can use **rcclone** ... and thus each and every storage over the network, really

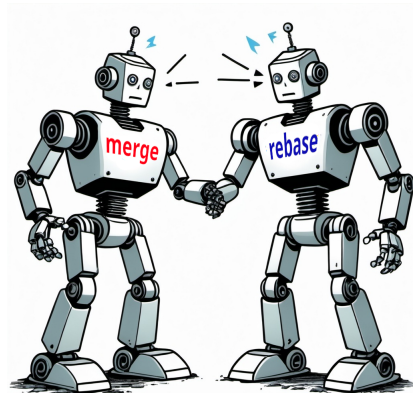
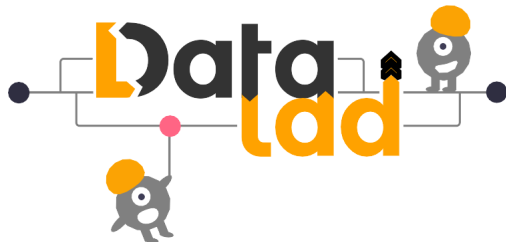


Image: <https://openart.ai>

⁵<https://handbook.datalad.org/en/latest/basics/101-117-sharelocal2.html#where-s-waldo>

- distributed data management⁶
- on top of git and git-annex
- free and Open Source
- ▶ simplifies some tasks
- ▶ additional functionality



⁶<https://helmholtz.software/software/datalad>

Datalad subcommands:

create new datasets locally ... like `git init`

clone other datasets ... like `git clone`, same URL, some extras

save changes to datasets ... like `git commit` without staging

push changes to a remote location ... like `git push`

update changes to a remote location ... like `git pull`

get / drop content on-demand ... download or forget annex files

create-sibling-XYZ new siblings == remote repositories or annex storages

► One can always use the git / git annex counterparts, so can IDEs.

- Create dataset == repository with `-c text2git`, then add a text file

```
$> datalad create -c text2git datalad-demo
$> cd datalad-demo
$> vim README.md
$> datalad status # optional
$> datalad save -m "Add README"
```

- Add central gitlab repository with datalad convenience command

```
$> datalad create-sibling-gitlab --site hzdr-gitlab --layout \
collection --access ssh --project casus-data/datalad-demo
```

- Add remote annex storage with `git annex`, then push there

```
$> git annex initremote minio type=external \
externaltype=rclone chunk=100MiB encryption=none \
target=minio prefix=datalad-demo
$> datalad push --to minio . # or selectively per subdir
```

- ... Datalad adds convenience and gives helpful hints

create-sibling Create a dataset sibling on a UNIX-like shell
... local or SSH accessible machine

create-sibling-github Create dataset sibling on GitHub.org

create-sibling-gitlab Create dataset sibling at a GitLab site

create-sibling-gogs Create a dataset sibling on a GOGS site

create-sibling-gin Create a dataset sibling on a GIN site
with annex file hosting

create-sibling-gitea Create a dataset sibling on a Gitea site

create-sibling-ria Creates a sibling to a dataset in a **RIA store**⁷
... Remote Indexed Archives for dataset archiving

⁷https://handbook.datalad.org/en/latest/beyond_basics/101-147-riastores.htm ▶ ◀ ≡ ≡ ≡ 🔍 ↺

“YODAs Organigram on Data Analysis”⁸

- P1** One thing, one dataset
- P2** Record where you got it from, and where it is now
- P3** Record what you did to it, and with what

⁸<https://handbook.datalad.org/en/latest/basics/101-127-yoda.html>

“YODAs Organigram on Data Analysis”

P1 One thing, one dataset

1. Store input data for an analysis in dedicated inputs/ sub-repo
2. Store scripts or code in a dedicated `code/` sub-repo (pure git!)
3. Leave the input data of an analysis untouched
4. Collect results of an analysis in a dedicated place
5. Include a place for complete execution environments
6. Separate repositories for different analysis projects

"YO

P1 One thing, one

1. Store input
2. Store script
3. Leave the
4. Collect res
5. Include a
6. Separate

```
├── ci/                                # continuous integration configuration
│   └── .travis.yml
├── code/                             # your code
│   ├── tests/                       # unit tests to test your code
│   │   └── test_myscript.py
│   └── myscript.py
├── docs                             # documentation about the project
│   ├── build/
│   └── source/
├── envs                             # computational environments
│   └── Singularity
├── inputs/                          # dedicated inputs/, will not be changed
│   └── data/                        # one stand-alone data component
│       ├── dataset1/
│       │   └── datafile_a
│       └── dataset2/
│           └── datafile_a
├── important_results/               # outputs away from the input data
│   └── figures/
├── CHANGELOG.md                    # notes for fellow humans about your
├── HOWTO.md
└── README.md
```

“YODAs Organigram on Data Analysis”

P2 Record where you got it from, and where it is now

1. Track data's provenance for other datasets:
Include other datasets as datalad subdatasets
via `datalad clone` instead of `git clone`
2. Track data's provenance for public files:
To insert external files, use `datalad download-url`,
then datalad will track the origin
3. Datalad makes it easy to share the entire conglomeration of
subdatasets and external sources for you

YODA principles P2: Let Datalad download files for you

Instead of

```
$> curl -O https://imgs.xkcd.com/comics/git_2x.png  
$> datalad save -m "download manually from XKCD"
```

use

```
&> datalad download-url -m "download from XKCD" \  
    https://imgs.xkcd.com/comics/git_2x.png
```

then Datalad / git-annex track where things came from:

```
knue@fwu082:~/data/datalad-demo > git annex list  
here  
|hzdr-gitlab  
|web  
||bittorrent  
|||minio  
||||  
X_X_X images/git_2x.png  
X_X_X images/git_commit_2x.png  
X__X images/git_commit_2x.small.png  
knue@fwu082:~/data/datalad-demo > █
```

“YODAs Organigram on Data Analysis”

P3 Record what you did to it, and with what

1. Use `datalad run` to execute scripts and code
2. ... preferably with scripts / sources also in the repository
3. ... with input and output files specified (relative paths)
4. Also: `datalad containers-run`

```
$>datalad run -m "Resize image" \  
--input git_commit_2x.png --output git_commit_2x.small.png \  
"convert {inputs} -geometry 439x439 {outputs}"
```

```
knue@fwu082:~/data/datalad-demo/images > git log -1
commit 4f7a1e6d301e3bdd03aeb6c9e355fb5f2ceb9a23 (HEAD -> master)
Author: Andreas Knüpfer <knue@posteo.de>
Date:   Fri Jul 5 13:19:01 2024 +0200
```

```
[DATALAD RUNCMD] Resize image
```

```
=== Do not change lines below ===
```

```
{
  "chain": [],
  "cmd": "convert {inputs} -geometry 439x439 {outputs}",
  "dsid": "be528a88-4333-49cf-9705-0be6e3f84a23",
  "exit": 0,
  "extra_inputs": [],
  "inputs": [
    "git_commit_2x.png"
  ],
  "outputs": [
    "git_commit_2x.small.png"
  ],
  "pwd": "images"
}
```

```
^^^ Do not change lines above ^^^
```

```
knue@fwu082:~/data/datalad-demo/images > datalad push --to hzdr-gitlab
publish(ok): . (dataset) [refs/heads/git-annex->hzdr-gitlab:refs/heads/git-annex da03dda..7da1063]
publish(ok): . (dataset) [refs/heads/master->hzdr-gitlab:refs/heads/master 32bc50d..4f7a1e6]
action summary:
  publish (ok: 2)
knue@fwu082:~/data/datalad-demo/images > █
```

- Someone else clones the repository somewhere else
- ... then all text files and tracking information included but annex files not yet

One way:

- `datalad get` for selected / all binary files

Other way:

- `datalad rerun` with git commit id

```
$> datalad rerun 4f7a1e6d301e3bdd03aeb6c9e355fb5f2ceb9a23
```

- ... automatically execute all recorded steps again to produce the file anew
- ... including `datalad get` from web or from siblings

► **Added value** when doing all steps the **datalad way**

```
knue@fwu082:~/data/datalad-demo-retrieve/images > git pull
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (8/8), done.
Entpackte Objekte: 100% (9/9), 1.01 KiB | 206.00 KiB/s, fertig.
remote: Total 9 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
```

```
Von codebase.helmholtz.cloud:casus-data/datalad-demo/project
```

```
32bc50d..4f7a1e6 master -> origin/master
```

```
da03dda..7da1063 git-annex -> origin/git-annex
```

```
Aktualisiere 32bc50d..4f7a1e6
```

```
Fast-forward
```

```
images/git_commit_2x.small.png | 1 +
```

```
1 file changed, 1 insertion(+)
```

```
create mode 120000 images/git_commit_2x.small.png
```

```
knue@fwu082:~/data/datalad-demo-retrieve/images > l
```

```
insgesamt 12K
```

```
drwxrwxr-x 1 knue knue 246  5. Jul 13:23 .
```

```
drwxrwxr-x 1 knue knue 250  5. Jul 11:34 ..
```

```
lrwxrwxrwx 1 knue knue 127  5. Jul 11:34 git_2x.png -> ../.git/annex/objects/1g/jp/MD5E-s82807--81bf1f649a76c89c8d35afbea902664a.png/MD5E-s82807--81bf1f649a76c89c8d35afbea902664a.png
```

```
lrwxrwxrwx 1 knue knue 129  5. Jul 13:02 git_commit_2x.png -> ../.git/annex/objects/9Q/x7/MD5E-s114552--451016cbf7ad1b82b51fe0d26765b57d.png/MD5E-s114552--451016cbf7ad1b82b51fe0d26765b57d.png
```

```
lrwxrwxrwx 1 knue knue 127  5. Jul 13:23 git_commit_2x.small.png -> ../.git/annex/objects/42/3V/MD5E-s98067--9262d53579202120ee3296576cd27ef3.png/MD5E-s98067--9262d53579202120ee3296576cd27ef3.png
```

```
knue@fwu082:~/data/datalad-demo-retrieve/images > git annex list
```

```
(merging origin/git-annex into git-annex...)
```

```
(recording state in git...)
```

```
here
```

```
|origin
```

```
||web
```

```
|||bittorrent
```

```
||||minio
```

```
|||||
```

```
X__X git_2x.png
```

```
X_X__ git_commit_2x.png
```

```
----- git_commit_2x.small.png
```

```

knue@fwu082:~/data/datalad-demo-retrieve/images > datalad rerun 4f7a1e6d301e3bdd03aeb6c9e355fb5f2ceb9a23
[INFO ] run commit 4f7a1e6; (Resize image)
[INFO ] Making sure inputs are available (this may take some time)
run.remove(ok): images/git_commit_2x.small.png (file) [Removed file]
[INFO ] == Command start (output follows) =====
[INFO ] == Command exit (modification check follows) =====
run(ok): /home/knue/data/datalad-demo-retrieve (dataset) [convert git_commit_2x.png -geometry 439x...]
add(ok): images/git_commit_2x.small.png (file)
save(ok): . (dataset)
action summary:
  add (ok: 1)
  get (notneeded: 1)
  run (ok: 1)
  run.remove (ok: 1)
  save (ok: 1)
knue@fwu082:~/data/datalad-demo-retrieve/images > l
insgesamt 12K
drwxrwxr-x 1 knue knue 246  5. Jul 13:25 .
drwxrwxr-x 1 knue knue 250  5. Jul 11:34 ..
lrwxrwxrwx 1 knue knue 127  5. Jul 11:34 git_2x.png -> ../.git/annex/objects/1g/jp/MD5E-s82807--81bf1f64
9a76c89c8d35afbea902664a.png/MD5E-s82807--81bf1f649a76c89c8d35afbea902664a.png
lrwxrwxrwx 1 knue knue 129  5. Jul 13:02 git_commit_2x.png -> ../.git/annex/objects/9Q/x7/MD5E-s114552--
451016cbf7ad1b82b51fe0d26765b57d.png/MD5E-s114552--451016cbf7ad1b82b51fe0d26765b57d.png
lrwxrwxrwx 1 knue knue 127  5. Jul 13:25 git_commit_2x.small.png -> ../.git/annex/objects/26/f1/MD5E-s98
067--148b5827cf340aa6eaf1d62241df0c.png/MD5E-s98067--148b5827cf340aa6eaf1d62241df0c.png
knue@fwu082:~/data/datalad-demo-retrieve/images > git annex list
here
|origin
||web
|||bittorrent
|||minio
||||
X___X git_2x.png
X_X__ git_commit_2x.png
X____ git_commit_2x.small.png
knue@fwu082:~/data/datalad-demo-retrieve/images >

```


- Use gitlab webhooks to feed data collection inventory from gitlab repo
- Each `DATASET.json` is representative of entire dataset
- ... when added, altered, deleted → update metadata record in inventory
- Link to gitlab project points to the very dataset subdirectory or the old commits for past versions
- Additional `README.md` give summary of every dataset for convenience

[Live example](#)

NOCODB

- Interactive database web front-end
- Searching, sorting, filtering, grouping ... without a learning curve
- Export + sharing of metadata records
- Feed via REST-API
- Open Source and locally hosted



casus-datalad / atomistic_foundation_model.datalad / project - GitLab - Mozilla Firefox

casus-datalad / atomistic_foundation_model.datalad / project

133%

Project

- project
- Pinned
- Issues
- Merge requests
- Manage
- Plan
- Code
- Build
- Secure
- Deploy
- Operate
- Monitor
- Analyze
- Settings

Name	Last commit	Last update
.datalad	[DATA Lad] new dataset	2 months ago
A_/Au_/Au_/Au	git-annex in knue@fwu082:~/enc/priv...	1 week ago
B_/B_/B_/B	test change 2	2 days ago
S_/SIH_/SIH4_/SIH4	new pps	1 week ago
scripts	Initial commit with some description, s...	2 months ago
subdirs	more description, prepare create_dirs....	2 months ago
.gitattributes	[DATA Lad] exclude paths from annex'i...	2 weeks ago
.gitignore	Initial commit with some description, s...	2 months ago
README.md	Remove my user name from examples	2 months ago
SUBDIRS.md	Initial commit with some description, s...	2 months ago
extract.py	new pps	1 week ago

README.md

Collection of Atomistic Structure Data

... for unary, binary, ternary compounds.

Create proper subdir structure

This data repository will become very large in terms of data size and in terms of datasets (more than 59.000). Thus, a

Project information

- 55 Commits
- 4 Branches
- 0 Tags
- 523 KIB Project Storage

README

- Add LICENSE
- Add CHANGELOG
- Add CONTRIBUTING
- Add Kubernetes cluster
- Set up CI/CD
- Add Wiki
- Configure Integrations

Created on

June 17, 2024

casus-datalad / atomistic

datasets | Atomistic F

Au/Au/Au/Au - master

https://data.casus.science/dashboard/#/nc/p7gl0acndm2wsul/m9bh2roy32qfwmp

120%

NOCODB

Quick Actions

Team & Settings

Integrations

Create Base

Bases

Atomistic Foundation Model

materials

Create View

Hossein_suggestion

datasets

Create View

Getting Started

Materials

AlpakaBenchmarks

Tmp Base

Test-Create-Base

datasets

Default View

Data

Details

Share

Fields 27

Filter

Group 1

Sort

#

stat...

URL

T_Aut...

T_Autho...

T_Auth...

T_Ch...

#_n...

#_n...

#_...

code_version

pps

CreatedAt

Au

Count 3

1	replaced	https://codebase.helmholtz...	PC2		CASUS	Au	206	201	5	[CP2K version 2024.1 (De...	Au-UPF Au_ON...	2024-08-20
2	deleted	https://codebase.helmholtz...	PC2		CASUS	Au	206	201	5	[CP2K version 2024.1 (De...	[Au]: Au_ON...	2024-08-21
3	current	https://codebase.helmholtz...	PC2		CASUS	Au	206	201	5	[CP2K version 2024.1 (De...	[Au]: Au_ON...	2024-08-21

New record

B

Count 11

SiH4

Count 2

a.knuepfer@hzdr.de

RDM with Datalad Andreas Knüpfer, Timothy Callow / CASUS, HZDR | 34

A./Au./Au./Au. - master · casus-datalad / atomistic_foundation_model.datalad / project - GitLab - Mozilla Firefox

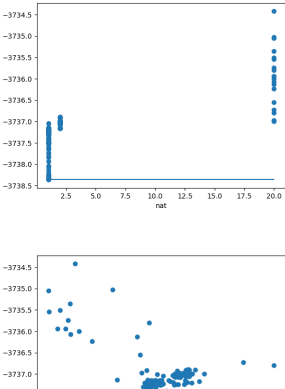
casus-datalad / atomistic_foundation_model.datalad / project - tree/master/A./Au./Au./Au

casus-datalad / atomistic_foundation_model.datalad / project / Repository

README.md

Dataset Au

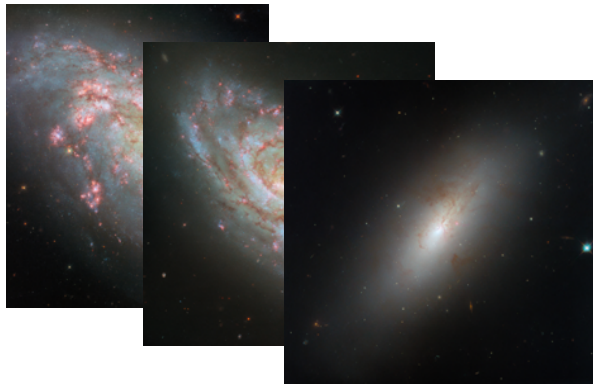
number of data: 206, number of bulks: 201, number of clusters: 5
Created with SIRIUS_CP2K



The top scatter plot displays a dense vertical cluster of points at $\text{nat} \approx 2.0$ and a sparse vertical cluster at $\text{nat} \approx 20.0$. The bottom scatter plot shows a more dispersed distribution of points across the nat range from 2.0 to 20.0.

Let's work with a Datalad repository for **ESA/Hubble Pictures of the Week**
(<https://esahubble.org/images/potw/>)

- Look around in the gitlab project
- Clone a data repository
- Look at existing data and metadata
- Add new datasets following our project's conventions
- Extract metadata and add it in a structured form
- Make it available to our project partners
- Watch the dataset inventory update automatically



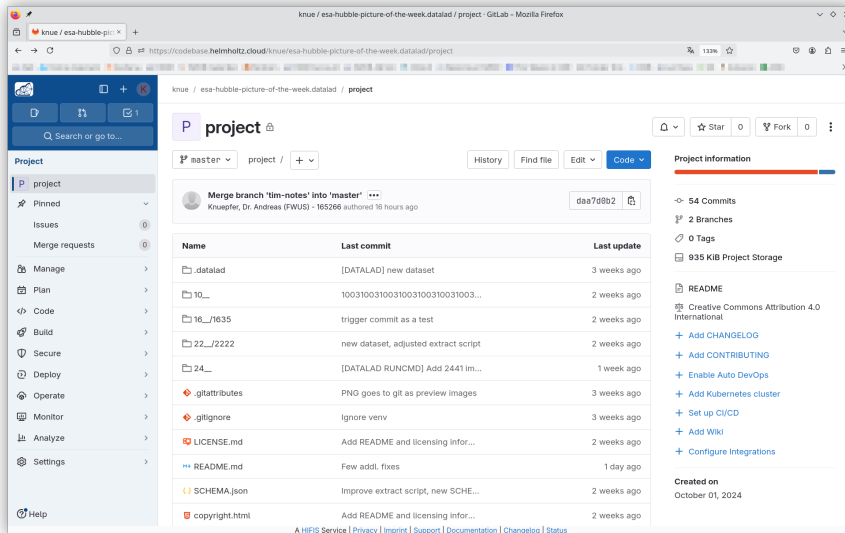
In the tutorial project

- Work with the HZDR gitlab project
<https://codebase.helmholtz.cloud/knue/esa-hubble-picture-of-the-week.datalad/project>
- Datasets in sub-sub-dirs ab__/abcd/
- Presence of DATASET.json file marks a subdir it as a dataset
- README.md file for convenience
- **Invite** all tutorial participants as gitlab project members

In general

- Central gitlab repository for collaborative work
- One repository per data collection
- Data collection owner grants access, defines roles etc.
- Many datasets in a data collection, same or similar enough structure
- Always exclusive subdir per dataset
- Subdir convention: never unlimited number of subdirs in any dir
- ... arbitrary naming scheme for intermediate subdirs

Tutorial: The gitlab project – top level



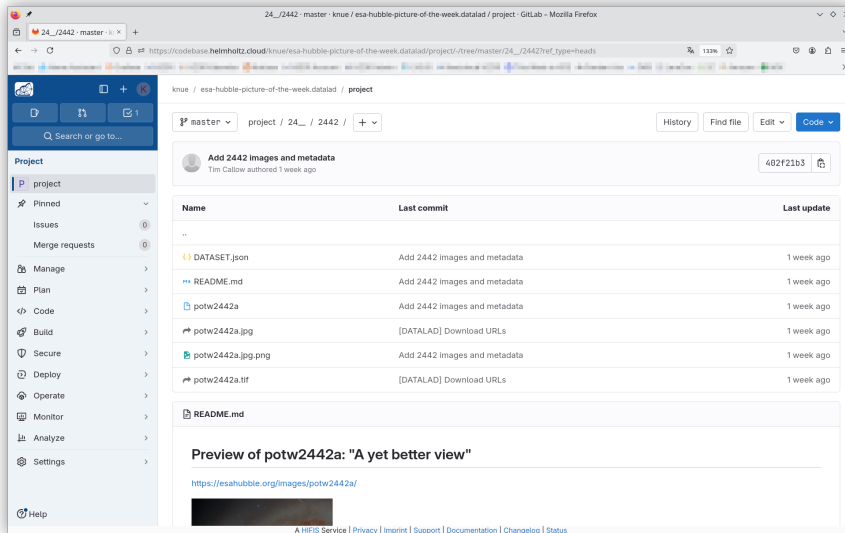
The screenshot shows the GitLab web interface for a project named "project". The browser address bar shows the URL: `https://codebase.helmholtz.cloud/knue/esa-hubble-picture-of-the-week.datalad/project`. The interface includes a sidebar with navigation options like "Project", "Pinned", "Issues", "Merge requests", "Manage", "Plan", "Code", "Build", "Secure", "Deploy", "Operate", "Monitor", "Analyze", "Settings", and "Help". The main content area displays the project's commit history, including a recent merge of branch "tim-notes" into "master" by Knuepfer, Dr. Andreas (FWUS) - 165266, 16 hours ago. Below this, a table lists the project's files and their last commit details.

Name	Last commit	Last update
<code>.datalad</code>	[DATALAD] new dataset	3 weeks ago
<code>10_</code>	1003100310031003100310031003...	2 weeks ago
<code>16_/_1635</code>	trigger commit as a test	2 weeks ago
<code>22_/_2222</code>	new dataset, adjusted extract script	2 weeks ago
<code>24_</code>	[DATALAD RUNCMD] Add 2441 im...	1 week ago
<code>.gitattributes</code>	PNG goes to git as preview images	3 weeks ago
<code>.gitignore</code>	Ignore venv	3 weeks ago
<code>LICENSE.md</code>	Add README and licensing infor...	2 weeks ago
<code>README.md</code>	Few addl. fixes	1 day ago
<code>SCHEMA.json</code>	Improve extract script, new SCHE...	2 weeks ago
<code>copyright.html</code>	Add README and licensing infor...	2 weeks ago

Project information sidebar:

- 54 Commits
- 2 Branches
- 0 Tags
- 935 KiB Project Storage
- README
- Creative Commons Attribution 4.0 International
- + Add CHANGELOG
- + Add CONTRIBUTING
- + Enable Auto DevOps
- + Add Kubernetes cluster
- + Set up CI/CD
- + Add Wiki
- + Configure Integrations
- Created on: October 01, 2024

Tutorial: The gitlab project – sub-sub-dir with files of a dataset



The screenshot shows a web browser displaying a GitLab project page. The browser's address bar shows the URL: `https://codebase.helmholtz.cloud/knue/esa-hubble-picture-of-the-week.datalad/project/-/tree/master/24_/2442?ref_type=heads`. The page title is "24_/2442 · master · knue / esa-hubble-picture-of-the-week.datalad / project · GitLab - Mozilla Firefox".

The left sidebar contains a navigation menu with the following items: Project, Pinned, Issues (0), Merge requests (0), Manage, Plan, Code, Build, Secure, Deploy, Operate, Monitor, Analyze, Settings, and Help.

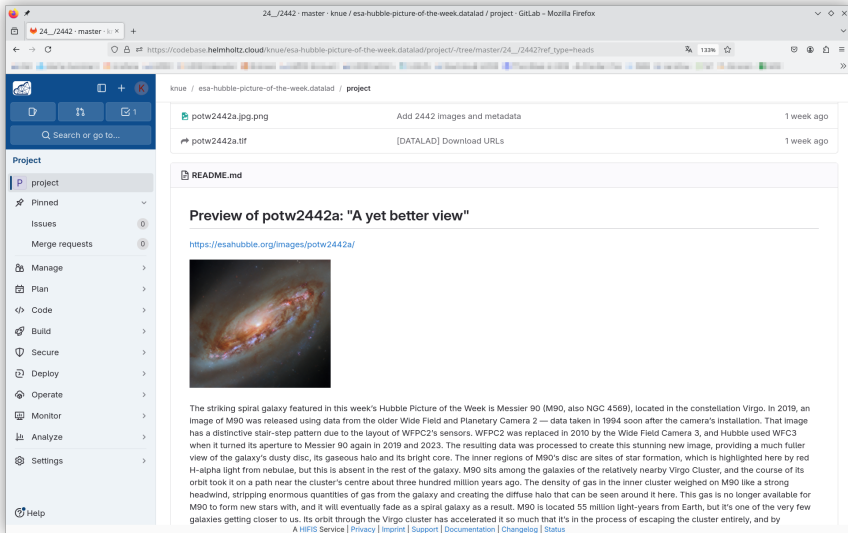
The main content area shows the project structure for "project". The breadcrumb navigation is "project / 24_ / 2442". The commit hash is "402f21b3". The commit message is "Add 2442 Images and metadata" by "Tim Callow" 1 week ago.

Name	Last commit	Last update
..		
DATASET.json	Add 2442 Images and metadata	1 week ago
README.md	Add 2442 Images and metadata	1 week ago
potw2442a	Add 2442 Images and metadata	1 week ago
potw2442a.jpg	[DATALAD] Download URLs	1 week ago
potw2442a.jpg.png	Add 2442 Images and metadata	1 week ago
potw2442a.tif	[DATALAD] Download URLs	1 week ago

Below the table, there is a section for "README.md" with the title "Preview of potw2442a: 'A yet better view'" and a link to `https://esahubble.org/images/potw2442a/`. A small image placeholder is visible below the link.

At the bottom of the page, there is a footer with the text: "A HIFIS Service | Privacy | Imprint | Support | Documentation | Changelog | Status".

Tutorial: The gitlab project – the README.md shown for convenience



The screenshot shows a web browser displaying the GitLab interface for a project named '24_/2442 - master · knue / esa-hubble-picture-of-the-week.datalad / project'. The left sidebar contains navigation links for Project, Pinned, Issues, Merge requests, Manage, Plan, Code, Build, Secure, Deploy, Operate, Monitor, Analyze, and Settings. The main content area shows a list of files: 'potw2442a.jpg.png' (Add 2442 Images and metadata, 1 week ago) and 'potw2442a.tif' ([DATALAD] Download URLs, 1 week ago). Below this is the 'README.md' file, which features a preview of a Hubble image of Messier 90 (M90, also NGC 4569) with the title 'Preview of potw2442a: "A yet better view"'. The image is a spiral galaxy with a bright core and a diffuse halo. The text below the image describes the galaxy's history and its location in the Virgo Cluster.

24_/2442 - master · knue / esa-hubble-picture-of-the-week.datalad / project · GitLab - Mozilla Firefox

https://codebase.heimholtz.cloud/knue/esa-hubble-picture-of-the-week.datalad/project/-/tree/master/24_/2442?ref_type=heads

knue / esa-hubble-picture-of-the-week.datalad / project

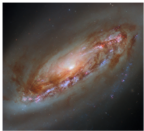
potw2442a.jpg.png Add 2442 Images and metadata 1 week ago

potw2442a.tif [DATALAD] Download URLs 1 week ago

README.md

Preview of potw2442a: "A yet better view"

<https://esahubble.org/images/potw2442a/>



The striking spiral galaxy featured in this week's Hubble Picture of the Week is Messier 90 (M90, also NGC 4569), located in the constellation Virgo. In 2019, an image of M90 was released using data from the older Wide Field and Planetary Camera 2 — data taken in 1994 soon after the camera's installation. That image has a distinctive stair-step pattern due to the layout of WFPC2's sensors. WFPC2 was replaced in 2010 by the Wide Field Camera 3, and Hubble used WFC3 when it turned its aperture to Messier 90 again in 2019 and 2023. The resulting data was processed to create this stunning new image, providing a much fuller view of the galaxy's dusty disc, its gaseous halo and its bright core. The inner regions of M90's disc are sites of star formation, which is highlighted here by red H-alpha light from nebulae, but this is absent in the rest of the galaxy. M90 sits among the galaxies of the relatively nearby Virgo Cluster, and the course of its orbit took it on a path near the cluster's centre about three hundred million years ago. The density of gas in the inner cluster weighed on M90 like a strong headwind, stripping enormous quantities of gas from the galaxy and creating the diffuse halo that can be seen around it here. This gas is no longer available for M90 to form new stars with, and it will eventually fade as a spiral galaxy as a result. M90 is located 55 million light-years from Earth, but it's one of the very few galaxies getting closer to us. Its orbit through the Virgo cluster has accelerated it so much that it's in the process of escaping the cluster entirely, and by

A HIFIS Service | Privacy | Imprint | Support | Documentation | Changelog | Status

In the tutorial project

- We want a nice way to browse, search, filter, group, ... our datasets
- Dataset inventory via NOCODB
... click the [public link in gitlab project README file near the bottom](#)
- Nice, interactive, and intuitive way to explore the datasets in the collection
- Works with thousands of metadata keys (columns) and many millions of entries (rows)
- Link in URL column points to precise subdir in the gitlab project

In general

- F.A.I.R. principles demand metadata to be available in a searchable way
- Needs some integration via gitlab webhooks and a NOCODB instance
- ... or other web portals or scientific community portals, e.g. NOMAD for theoretical chemistry

Tutorial: The dataset inventory

Browser: pictures | ESA Hubble potw - Mozilla Firefox
URL: https://data.casus.science/dashboard/#/nc/p123rrieuutloo/mvcljif3k105wv

NOCODB / pictures / Default View

88 Data % Details

Fields 3 Filter Group Sort

#	URL	status	branch	EXIF.SourceFile	EXIF.FileSize	EXIF.FileModifyD...	EXIF.FileA...
1	https://codebase.helmholtz...	current		10___/1001/potw1001a.jpg	11688221	2024-10-03T23:11:36+01...	2024-10-03T11...
2	https://codebase.helmholtz...	current		10___/1002/potw1002a.jpg	2467474	2024-10-03T23:11:38+01...	2024-10-03T11...
3	https://codebase.helmholtz...	current		./potw1003a.jpg	104256	2024-10-03T23:11:39+01...	2024-10-03T11...
4	https://codebase.helmholtz...	current		16___/1635/potw1635a.jpg	107856	2024-10-03T23:23:07+01...	2024-10-03T10...
5	https://codebase.helmholtz...	current	master	24___/2438/potw2438a.jpg	6272617	2024-10-03T18:06:13+02...	2024-10-03T18...
6	https://codebase.helmholtz...	current	master	24___/2440/potw2440a.jpg	6703781	2024-10-03T11:47:51+02...	2024-10-03T17...
7	https://codebase.helmholtz...	current	master	24___/2439/potw2439a.jpg	6784522	2024-10-03T15:25:25+02...	2024-10-03T15...
8	https://codebase.helmholtz...	current	master	22___/2222/potw2222a.jpg	5143114	2024-10-04T23:35:39+01...	2024-10-04T16...
9	https://codebase.helmholtz...	current	master	24___/2442/potw2442a.jpg	5829069	2024-10-23T23:19:35+02...	2024-10-23T11...
10	https://codebase.helmholtz...	current	master	24___/2441/potw2441a.jpg	7783865	2024-10-23T15:23:33+02...	2024-10-23T13...

+ New record

10 records

Try NocoDB Cloud

Left sidebar menu:

- Quick Actions
- Team & Settings
- Integrations
- Create Base
- Bases
 - Atomistic Foundation Model
 - materials
 - datasets
 - datasets-new
 - Getting Started
 - CP2k Benchmarks
 - Materials
 - Base
 - ESA Hubble Potw
 - pictures** (selected)
 - + Create View
 - Tmp Base
 - AlpakaBenchmarks
 - Test-Create-Base

Tutorial: The dataset inventory

Atomistic Foundation Model - Mozilla Firefox

knue / esa-hubble-pic x Atomistic Foundation x +

https://data.casus.science/dashboard/#/nc/p123rieiuttoo/mvrcj13k105wv 132W

NOCODB << / pictures ▾ / Default View ▾ 88 Data % Details Share

Quick Actions (K) Team & Settings Integrations + Create Base

Bases

- Atomistic Foundation Model
 - materials
 - datasets
 - datasets-new
- Getting Started
- CP2k Benchmarks
- Materials
- Base
- ESA Hubble Potw
- pictures**
 - + Create View
- Tmp Base
- AlpakaBenchmarks
- Test-Create-Base

a.knuepfer@hzdr.de + New record ^

Try NocoDB Cloud 10 records

Fields 3 Filter Group Sort

Search fields

- URL
- T branch
- T EXIF.SourceFile
- # EXIF.ExifTool-ExifToolVersion
- T EXIF.File.FileName
- T EXIF.File.Directory
- # EXIF.File.FileSize
- T EXIF.File.FileModifyDate
- T EXIF.File.FileAccessDate
- T EXIF.File.FileinodeChangeDate
- # EXIF.File.FilePermissions
- T EXIF.File.FileType

Show all fields Show system fields

s	T branch	T EXIF.SourceFile	# EXIF.File.FileSize	T EXIF.File.FileModifyDate	T EXIF.File.FileAccessDate
		10___/1001/potw1001a.jpg	11688221	2024-10-03T23:11:36+01...	2024-10-03T11:36+01...
		10___/1002/potw1002a.jpg	2467474	2024-10-03T23:11:38+01...	2024-10-03T11:38+01...
		./potw1003a.jpg	104256	2024-10-03T23:11:39+01...	2024-10-03T11:39+01...
		16___/1635/potw1635a.jpg	107856	2024-10-03T23:23:07+01...	2024-10-03T11:23:07+01...
	master	24___/2438/potw2438a.jpg	6272617	2024-10-03T18:06:13+02...	2024-10-03T18:06:13+02...
	master	24___/2440/potw2440a.jpg	6703781	2024-10-03T11:47:51+02...	2024-10-03T11:47:51+02...
	master	24___/2439/potw2439a.jpg	6784522	2024-10-03T15:25:25+02...	2024-10-03T15:25:25+02...
	master	22___/2222/potw2222a.jpg	5143114	2024-10-04T23:35:39+01...	2024-10-04T16:35:39+01...
	master	24___/2442/potw2442a.jpg	5829069	2024-10-23T23:19:35+02...	2024-10-23T11:19:35+02...
	master	24___/2441/potw2441a.jpg	7783865	2024-10-23T15:23:33+02...	2024-10-23T11:23:33+02...

Tutorial: The dataset inventory

Atomistic Foundation Model - Mozilla Firefox

knue / esa-hubble-pic x Atomistic Foundation x +

https://data.casus.science/dashboard/#/nc/p123rieuut00/mvcljif3k105wv 132w

NOCODB << / pictures ▾ / Default View ▾ 88 Data % Details Share

Quick Actions (K) Team & Settings Integrations Create Base

Bases

- Atomistic Foundation Model
 - materials
 - datasets
 - datasets-new
- Getting Started
- CP2k Benchmarks
- Materials
- Base
- ESA Hubble Potw
 - pictures** + Create View
- Tmp Base
- AlpakaBenchmarks
- Test-Create-Base

a.knuepfer@hzdr.de

Try NocoDB Cloud

2 groups

Fields 3 Filter Group 1 Sort

URL status T branch T EXIF.SourceFile # EXIF.File.FileSize T EXIF.File.FileModifyD... T EXIF.File.File...

Germany Count 4

1	https://codebase.helmholtz...	current		10___/1001/potw1001a.jpg	11688221	2024-10-03T23:11:36+01...	2024-10-03T11:36+01...
2	https://codebase.helmholtz...	current		10___/1002/potw1002a.jpg	2467474	2024-10-03T23:11:38+01...	2024-10-03T11:38+01...
3	https://codebase.helmholtz...	current		/potw1003a.jpg	104256	2024-10-03T23:11:39+01...	2024-10-03T11:39+01...
4	https://codebase.helmholtz...	current		16___/1635/potw1635a.jpg	107856	2024-10-03T23:23:07+01...	2024-10-03T11:23:07+01...

New record ^

United States Count 6

1	https://codebase.helmholtz...	current	master	24___/2438/potw2438a.jpg	6272617	2024-10-03T18:06:13+02...	2024-10-03T11:06:13+02...
2	https://codebase.helmholtz...	current	master	24___/2440/potw2440a.jpg	6703781	2024-10-03T11:47:51+02...	2024-10-03T11:47:51+02...
3	https://codebase.helmholtz...	current	master	24___/2439/potw2439a.jpg	6784522	2024-10-03T15:25:25+02...	2024-10-03T11:25:25+02...
4	https://codebase.helmholtz...	current	master	22___/2222/potw2222a.jpg	5143114	2024-10-04T23:35:39+01...	2024-10-04T11:35:39+01...
5	https://codebase.helmholtz...	current	master	24___/2442/potw2442a.jpg	5829069	2024-10-23T23:19:35+02...	2024-10-23T11:19:35+02...
6	https://codebase.helmholtz...	current	master	24___/2441/potw2441a.jpg	7783865	2024-10-23T15:23:33+02...	2024-10-23T11:23:33+02...

New record ^

<< < 1 > >>

Tutorial: The dataset inventory

Browser window: pictures | ESA Hubble potw - Mozilla Firefox

URL: https://data.casus.science/dashboard/#/nc/p123rieiutdoo/mvcljfr3k105wv

NOCODB

Quick Actions [X] [K]

Team & Settings

Integrations

+ Create Base

Bases

- Atomistic Foundation Model
 - materials
 - datasets
 - datasets-new
- Getting Started
- CP2k Benchmarks
- Materials
- Base
- ESA Hubble Potw
 - pictures**
 - + Create View
 - Tmp Base
 - AlpakaBenchmarks
 - Test-Create-Base

Share

Share View Default View

Enable Public Viewing ☒

https://data.casus.science/dashboard/#/nc/view/25e

Restrict access with password ☒

Enter the password

Allow Download ☒

Share Base ESA Hubble Potw

Enable Public Access ☐

Close Manage Base Access

10 records

Install datalad version $\geq 1.1.0$ & co via package manager

```
$> sudo apt install datalad git-annex git-annex-remote-rclone
```

or via

```
$> curl -Ls https://micro.mamba.pm/api/micromamba/linux-64/latest | tar \
    -xvj bin/micromamba
$> mv bin/micromamba $HOME/.local/bin
$> micromamba config append channels conda-forge
$> micromamba config set channel_priority strict
$> micromamba activate
$> micromamba search datalad
$> micromamba install datalad git-annex rclone git-annex-remote-rclone
```

Specific for this tutorial example also install exiftool and its Python libs

```
$> sudo apt install libimage-exiftool-perl
$> pip install -r requirements.txt
```

In the tutorial project

- Clone the gitlab repository via datalad
- ... use the link which gitlab gives you under "Clone with SSH"

```
$> datalad clone git@codebase.helmholtz.cloud:knue/  
    esa-hubble-picture-of-the-week.datalad/project.git  
    [esa-hubble-picture-of-the-week.datalad]
```

- Look around inside the local copy: files in top level dir, sub-dir structure
- Access the *.png files as well as the *.jpg and *.tif files
- Try `datalad status`, `datalad siblings`, `git annex list`

In general

- Always use the `datalad` subcommands to clone and to manage "siblings"

- When you clone the repository, only the regular git files are created
- All “annex files” are there only as broken symlinks
- Use `datalad get 24__/2441/` or `datalad get 24__/` to download some or `datalad get .` to download all
- Compare output of `git annex list` afterwards
- Use `datalad drop [subdirs]` or `git annex drop [subdirs]` to delete binary files from the local repo, yet retaining their tracking information

You successfully mastered part 1 of the tutorial!

Now you can work with the existing data, analyze it, or use it as read-only input data for your own projects on top of it.

In the tutorial project

- Browse ESA/Hubble Picture of the Week and pick a picture which is not yet in this data repository: <https://esahubble.org/images/potw/>
- Go to the specific page like <https://esahubble.org/images/potw2422a/> and note the 4-digit number `abcd` near the end of the URL.
- Create the proper subdirectory according to our convention:

```
mkdir -p ab__/abcd/ , then
```

```
cd ab__/abcd/ .
```

In general

- Create a dataset sub-dir according to the repository convention first
- ... either for generating data from simulations, experiments etc. or for downloading from somewhere else
- You may want to use a branch instead of working with `main`. Branching, merging, forking, merge requests etc. work like usual.

In the tutorial project

- Download the web page (the HTML file) to the dataset sub-dir e.g., with

```
curl -O https://esahubble.org/images/potw2422a/
```


(in case of issues with older versions of curl use

```
curl -o potw2422a https://esahubble.org/images/potw2422a/
```

).
- This is one file in our dataset that is mandatory to be there. In this example lets pretend it is a file that we generated ourselves like an input file.

In general

- Copy, create, keep all files that belong to the dataset like input files for a simulation code, output files produced by a simulation, log files created as a byproduct, ...

In the tutorial project

- On the specific web page use right-click and "copy link" to get the links to the "Fullsize Original" in TIF format and the "Large JPEG" in JPG format ... don't download them via the web browser, though
- Instead use `datalad download-url` from within the correct dataset sub-dir:

```
$> datalad download-url \  
    https://esahubble.org/media/archives/images/original/potw2422a.tif \  
    https://cdn.esahubble.org/archives/images/large/potw2422a.jpg
```

Comment: Of course we could have used `datalad download-url` instead of `curl` in step 3 before. This would be the more datalad-ish way if it was not for the example purpose.

In general

- Always use `datalad download-url` to download data from public URLs so that datalad can track the origin and re-download it on demand.

In the tutorial project

The `DATASET.json` file with the structured metadata is still missing. It is an essential file because it promotes the mere **sub-dir** to a **dataset**.

- Go to the root dir of the data repository: `cd ../../`
- Run `./extract_metadata.py ab_/_abcd/` with the correct digits `abcd`
- This automatically extracts all metadata and produces the `DATASET.json` and `README.md` files together with the small preview image in PNG format.

In general

- Always have a project specific `./extract_metadata.py` script or similar ... need to maintain one such script per data repository.
- Run it for every new dataset before adding it to the repository.

In the tutorial project

- Now `datalad status` or `git status` will show changes
- Use `datalad save \`
`-m "<commit message>"[subdir]`
to commit it locally

In general

- Local commits, log, branching, ... like usual
- Corresponding git commands will also work ... so will git support integrated in your favorite code editor or git GUI tools

In the tutorial project

- Use `datalad push [--to <name-of-gitlab-sibling>]` to make your local changes known to the rest of the team (like `git push`)
- `datalad update --how=merge` updates the local repo from the central gitlab repo (like `git pull`)

In general

Nothing special here, work like you would with normal git repositories and use either the `datalad` or `git` subcommands.

Inventory

After a successful `datalad save` + `datalad push` resp. `git commit` and `git push` the NOCODB inventory gets updates automatically. Check public link <https://data.casus.science/dashboard/#/nc/view/25e463e8-7617-4090-b698-54a5f286ac45> == link in gitlab project README file

Finishing line

You successfully mastered part 2 of the tutorial!

Now you now all steps how to contribute to datalad repositories together with the generic DATASET.json metadata convention. There are more nice features and options in datalad but they are not required to start working in a productive manner in your team.

... basic instructions for starting your own datalad repository. See [datalad handbook](#) for more details.

- Create dataset == repository with `-c text2git`, then add a text file

```
$> datalad create -c text2git datalad-demo  
$> cd casus_rdm_demo.datalad  
$> vim README.md  
$> datalad status  
$> datalad save -m "Add README"
```

- Where `-c text2git` will make all text files become normal git managed files and all binary files become git-annex files. This can be fine-tuned in `.gitattributes` in the root of your repo.

- Configure Helmholtz gitlab to be known in your python-gitlab settings (once)

```
$> cat ~/.python-gitlab.cfg

[global]
default = hzdr-gitlab
ssl_verify = true
timeout = 5

[hzdr-gitlab]
url = https://codebase.helmholtz.cloud
private_token = <gitlab API token here>
api_version = 4
```

- Add central gitlab repository with datalad convenience command

```
$> datalad create-sibling-gitlab --site hzdr-gitlab \  
--layout collection --access ssh \  
--project casus-data/datalad-demo
```

- Where `hzdr-gitlab` is the gitlab site from `.python-gitlab.cfg`
- `casus-data` is an *existing* group / organization at the gitlab site
- `datalad-demo` is an *newly created* subgroup in this gitlab group
- "project" is the *actual project* in this subgroup
... it is the top-level repository in case of more sub-datasets / submodules
- Now the new remote sibling is still empty, need to push to there:

```
$> datalad push --to hzdr-gitlab
```

Tutorial Appendix: Add remote annex storage

- Configure rclone access to S3 storage (once)

```
$> rclone config  
$> cat ~/.config/rclone/rclone.conf  
$> rclone lsd $REMOTE:
```

- ... or any kind of remote storage with rclone access, i.e. almost anything
- Add remote annex storage with `git annex` directly

```
$> git annex initremote minio type=external \  
externaltype=rclone chunk=100MiB encryption=none \  
target=minio prefix=datalad-demo
```

... DNS-style names only, no underscores!

- Then add some binary files and push to the annex storage:

```
$> datalad push --to minio . # or selectively per subdir
```

Next steps: Integration with HPC

Does it work with HPC? Mostly yes.

- HPC relies on batch jobs (Slurm)
- Job script pretty much defines how to run everything
 - Software packages and version through modules
 - Which types of parallelism and how much
 - One or multiple job steps

Datalad outside:

```
datalad run "sbatch slurm.sh"...
```

- **Fails**, because no output available when Datalad wants to save (commit)

Datalad inside:

Use `datalad run` in job steps

- Works, but does not reference the `slurm.sh` batch script
- ... which you would want to re-run
- **need a fresh clone for every job : (**
- **git clone|commit|push in a job ???**

Goal: Added value of RDM for data collection owners

- Dataset inventory provides the overview of data collection ... current and past
- Allows convenient searching and browsing
- Sharing of structured metadata with others

Cost:

- git + gitlab style workflow with some additions
- Basic metadata conventions within the work group / project
- NocoDB instance is easy to setup and maintain

Freedom:

- No money per storage capacity / per file / per access / per month / ...
- Complete ownership of data, infrastructure, and processing steps

1) Datalad repositories for research data

- Dataset owner: create repository
 - Dataset owner: invite others
 - All: create datasets like normally
 - All: commit and push datasets
 - All: work with branches and tags ...
 - Some: combine with HPC, AI, workflow engines, ...
- ▶ You are one important step ahead w.r.t. data management, data safety, and knowledge management
- ▶ Work with it and make it common practise in your groups

2) DATASET.json files

- Create robust metadata extraction scripts `metadata_extract.py`
- ... SCC will help you

3) Metadata inventory

- Create generic inventory (NOCODB)
- ... SCC will help you
- or create community-standard inventory (e.g. with NOMAD)
- ... SCC will help you

▶ F.A.I.R. data management achieved



CASUS

CENTER FOR ADVANCED
SYSTEMS UNDERSTANDING

www.casus.science

