Open Workflows:
Reproducible Research Objects with DataLad

OHBM Open Science Room, June 24th, 2020

Adina Wagner , INM-7 Juelich Research Center

Mattermost: @adina

Find this talk at handbook.datalad.org/r.html?OHBM2020

Versatile data management and data publication tool

Command-line tool + Python API

Built on Git and git-annex

Major features:

- size-independent version control
- tools and principles for reproducible and FAIR analyses
- 3rd-party integrations and publication routines

DataLad

Hanke, Halchenko et al.
www.datalad.org

Learn about what DataLad can do for you in handbook.datalad.org/r.html?about

Fully reproducible from start to finish

Data consumption

- Install data like a software package
- Retrieve data on demand
- Link versioned data to analysis code

Data analysis

- Link code, data, software, & code execution
- Yield machine-readable, re-executable run-records

Publication of results

- Share results with complete provenance records

Find the results at gin.g-node.org/adswa/OSRdemo

Setup:

- DataLad **0.13** or higher

```
❯ datalad --version
datalad 0.13.0rc2.dev7
```

- DataLad extension “datalad-containers”

```
❯ pip show datalad-container
Name: datalad-container
Version: 0.5.2
```

Get installations instructions from handbook.datalad.org/intro/installation.html

Everything happens in datasets:

Dataset =

- a directory on your computer managed by DataLad
- a Git/git-annex repository
- datasets can version control their contents, be shared and installed, and they can be nested (linked)

Create an analysis dataset with `datalad create`:

```
(master) adina@cpu6 in /data/group/psyinf
❯
```

Learn about datasets and dataset nesting in handbook.datalad.org/basics/basics-datasets.html

Projects are modular, linked analysis components:

- **Analysis output datasets** to hold code and results
- Utility-datasets with containerized pipelines as “toolboxes”
- Link **input data** and **toolboxes** in the analysis datasets

```
my_analysis
      ├── code/ …            # subdirectory with scripts
      ├── outputs/ …         # subdirectory or dataset to collect results
      ├── .source            # subdataset with input data
      │     └── sub-001/ ...
      └── .toolbox           # subdataset with container
```

Read about these principles in handbook.datalad.org/r.html?yoda

Analysis components as modular units, installed on demand

Install data from GitHub, or from a RIA-store in precise versions (tags, branches):

```
(master) adina@juseless in /data/group/psyinf/HCP_structural on git:master
❯ datalad clone -d . 'ria+http://store.datalad.org#~hcp-structural-preprocessed@bids' .source
```

New in 0.13! BIDS-formatted subset of HCP data, hosted in a public RIA store.

Read about RIA stores: handbook.datalad.org/r.html?RIA

Large-scale analysis? No problem.

- Datasets scale: a few 100k files per dataset are fine
- If analyses produce more files, nested dataset ensure scalable performance

In anticipation of ~500k fmriprep outputs, I install two (precreated, empty) subdatasets)

```
(master) adina@juseless in /data/group/psyinf/OSRdemo on git:master
❯ datalad clone -d . git@gin.g-node.org:/adswa/OSRfmriprep.git fmriprep
```

```
OSRdemo
      ├── fmriprep/ …      # subdataset to collect results
      ├── freesurfer/ …    # subdataset to collect results
      ├── .source          # subdataset with input data
      └── .toolbox         # subdataset with containerized pipeline
```

Learn about large scale datasets in handbook.datalad.org/r.html?gobig

“Toolboxes”: container image, call specification, relevant auxiliary files

Custom toolbox creation:

```
(master) adina@juseless in ~
❯ datalad create -c text2git fmriprep_toolbox
```

Example: **fmriprep**

Consume existing containers from github.com/ReproNim/containers

Analysis components as modular units, installed on demand

Link the toolbox to the analysis as a subdataset with `datalad clone -d . <...>`:

```
adina@juseless in /data/group/psyinf/HCP_structural on git:master
❯ datalad clone -d . ~/fmriprep_toolbox .tools
```

Link data, results, code, and software

Use `datalad containers-run` to run a linked containerized pipeline & get a re-executable run-record:

```
(master) adina@juseless in /data/group/psyinf/OSRdemo on git:master
❯ datalad containers-run -n .tools/fmriprep \
   -m "preprocess examplary subject with fmriprep" \
   --input .source/sub-170631 \
   --output fmriprep \
   --output freesurfer \
   ".source . participant --participant-label 170631 --skip-bids-validation --anat-only -w /tmp --fs-license-file
.tools/license.txt"
```

Read about containers-run in handbook.datalad.org/r.html?containers

Result publication

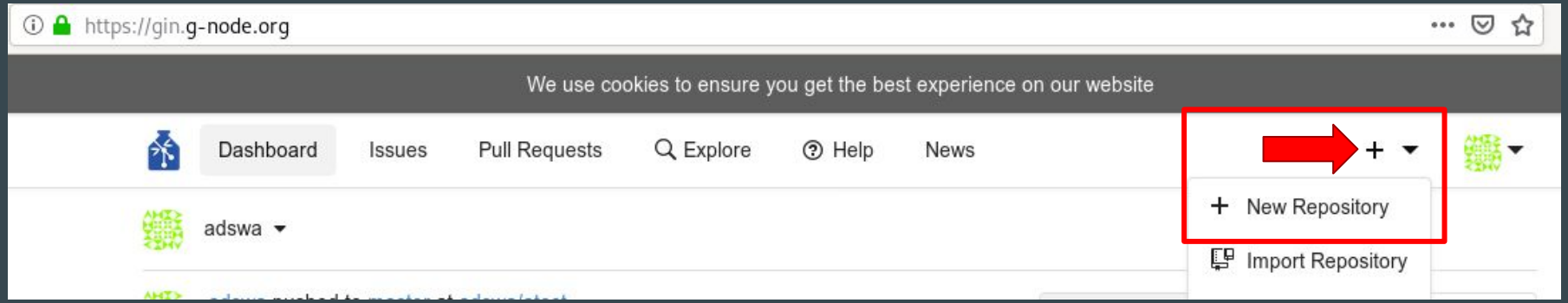
- Select publication targets depending on desired audience: Public, private, …
  - **Easy public access**: Publish data to a public repository on GIN (https://gin.g-node.org/), a free DataLad-dataset hosting service with support for controlled and anonymous access

Read about publishing datasets to GIN in handbook.datalad.org/r.html?GIN

Result publication

Step 1: Create a repository

Create it empty and make it public



Result publication

Step 2: Add the repository as a sibling

```
❯ datalad siblings add --name origin --url git@gin.g-node.org:/adswa/OSRdemo.git                130 !
```

Read about publishing datasets in handbook.datalad.org/basics/101-141-push.html

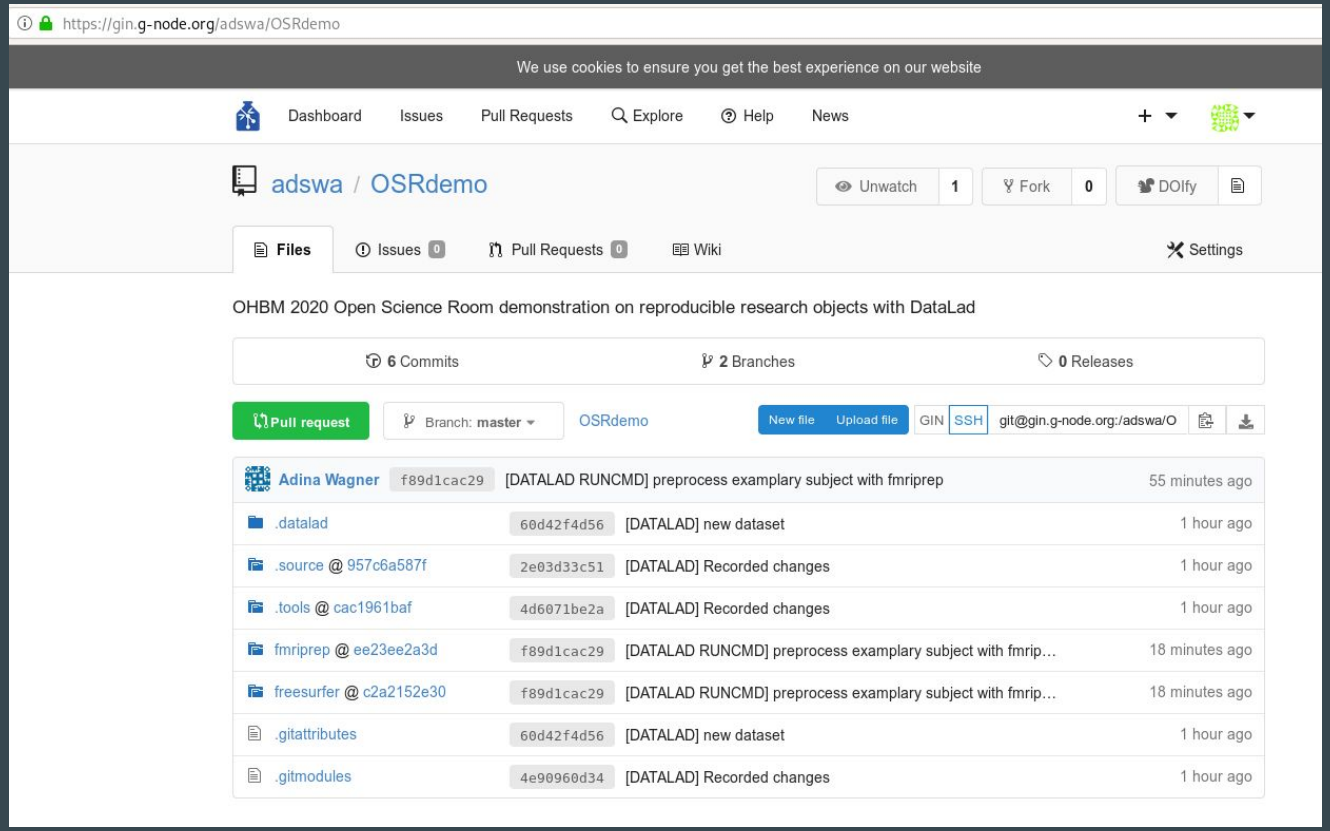
Result publication

Step 3: Publish the data

```
❯ datalad push --to origin -r
```

Result publication

Result



Result retrieval

Public Gin repositories support anonymous HTTP access:

```
adina@juseless in ~
❯ datalad clone https://gin.g-node.org/adswa/OSRdemo
```

Result recomputation

Automatic recomputation with `datalad rerun`:

```
❯ datalad rerun                                                                 1
[INFO   ] Making sure inputs are available (this may take some time)
get(ok): .tools/.datalad/environments/fmriprep/image (file) [from origin...]
unlock(ok): fmriprep/dataset_description.json (file)
unlock(ok): fmriprep/logs/CITATION.bib (file)
unlock(ok): fmriprep/logs/CITATION.md (file)
unlock(ok): fmriprep/logs/CITATION.tex (file)
unlock(ok): fmriprep/sub-179952.html (file)
unlock(ok): fmriprep/sub-179952/anat/sub-179952_desc-preproc_T1w.json (file)
unlock(ok): fmriprep/sub-179952/anat/sub-179952_desc-preproc_T1w.nii.gz (file)
unlock(ok): fmriprep/sub-179952/anat/sub-179952_from-T1w_to-fsnative_mode-image_xfm.txt (file)
unlock(ok): fmriprep/sub-179952/anat/sub-179952_from-fsnative_to-T1w_mode-image_xfm.txt (file)
unlock(ok): fmriprep/sub-179952/anat/sub-179952_from-orig_to-T1w_mode-image_xfm.txt (file)
[INFO   ] == Command start (output follows) =====
200606-07:51:31,155 nipype.workflow WARNING:
         Previous output generated by version 20.1.0 found.
200606-07:51:40,327 nipype.workflow IMPORTANT:

    Running fMRIPREP version 20.1.1:
      * BIDS dataset path: /data/project/prepwar/data/OSR/.source.
      * Participant list: ['179952'].
      * Run identifier: 20200606-074752_3f32048f-e2fa-467a-b497-19a3730d3c7f.
      * Output spaces: MNI152NLin2009cAsym:res-native.
      * Pre-run FreeSurfer's SUBJECTS_DIR: /data/project/prepwar/data/OSR/freesurfer.
200606-07:53:08,431 nipype.workflow INFO:
         fMRIPrep workflow graph with 155 nodes built successfully.
```

Acknowledgements

DataLad

Michael Hanke
Yaroslav Halchenko
Benjamin Poldrack
Kyle Meyer
Joey Hess (git-annex)
20+ contributors

The DataLad Handbook

Adina Wagner
Michael Hanke
Laura Waite
26+ contributors

SPONSORED BY THE

Federal Ministry
of Education
and Research

01GQ1112
01GQ1411

1129855
1429999

Human Brain Project

EUROPEAN UNION

VirtualBrainCloud

cbbs
center for behavioral
brain sciences