

RESEARCH DATA MANAGEMENT



WITH DATALAD

Adina Wagner

 @AdinaKrik

Michael Hanke

 @eknahm

Psychoinformatics lab,
Institute of Neuroscience and Medicine (INM-7)
Research Center Jülich

Slides: <https://github.com/datalad-handbook/course/>

WELCOME!

Approximate workshop schedule

Session 1 (now, 13.30-15.00)

Logistics & Intro ,
Hands-on Terminal Basics ,
Demo of core functionality  

Session 2 (today, 16.00-18.00)

Hands-on DataLad Basics & Exercises 

Session 3 (tomorrow, 11.00-12.30)

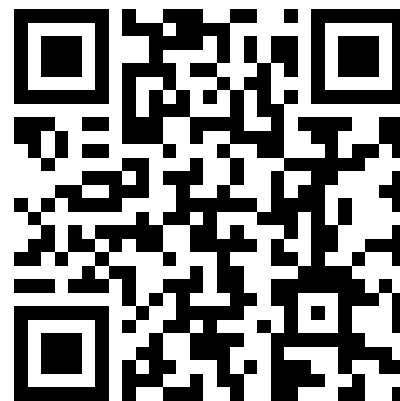
Sharing and Collaboration ,
Hands-on Data publication 

Session 4 (tomorrow, 13.30-15.00)

Computational reproducibility  ,
Outro ,
Final QA ?

LOGISTICS AND LINKS

- You can download these slides at <https://doi.org/10.5281/zenodo.6827086> (scan the QR code), and you can find their sources at github.com/datalad-handbook/datalad-course
- Some of today's code-along workshop contents are at psychoinformatics-de.github.io/rdm-course
- The workshop will be interactive. If you do not have the software installed on your own system, you can access a Jupyterhub from your browser at datalad-hub.inm7.de
- You can log in to the Jupyterhub with a pre-set username (take one out of the jar) and a self-set password. Remember the password for tomorrow!
- A [requirements.txt](#) file on Zenodo details the software environment we setup on the Jupyterhub



INTERACTIVITY

- The workshop centers around **DataLad** (version 0.16 and up) for real-world **research data management** use cases
 - There are no stupid questions; ask anything any time
 - Something doesn't look right on your system? Stick a post-it to your screen. We'll take a look together
 - We're available outside of sessions, too. Chat about your use cases or questions over a coffee or meal
-
- 4 sessions = time for more than a standard introduction.
 - Materials are available online & persistent, we can be flexible & spontaneous if specific topics interest you

AFTER THE WORKSHOP

If you have a question after the workshop, you can reach out for help:

Reach out to to the DataLad team via

- [Matrix](#) (free, decentralized communication app, no app needed). We run a weekly Zoom office hour (Thursday, 4pm Berlin time) from this room as well.
- [the development repository on GitHub](#)

Reach out to the user community with

- A question on [neurostars.org](#) with a `data lad` tag

Find more user tutorials or workshop recordings

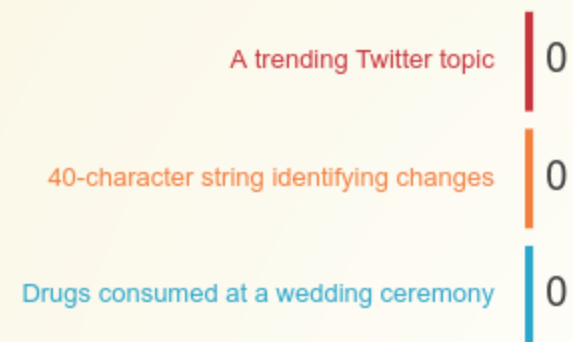
- On [DataLad's YouTube channel](#)
- In the [DataLad Handbook](#)
- In the [DataLad RDM course](#)
- In the [Official API documentation](#)

AUDIENCE RESPONSE SYSTEM

Use your phone to scan the QR code, or open the link in a new browser window

What is a commit hash?

<http://etc.ch/7YEK>



0 votes - 0 participants

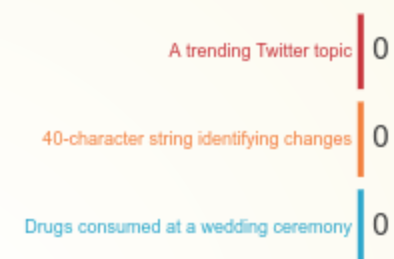
Direct
Poll

ON A SCALE OF RUBBER DUCKS...



What is a commit hash?

<http://etc.ch/7YEk>



0 votes - 0 participants

Direct
Poll

RESEARCH DATA MANAGEMENT

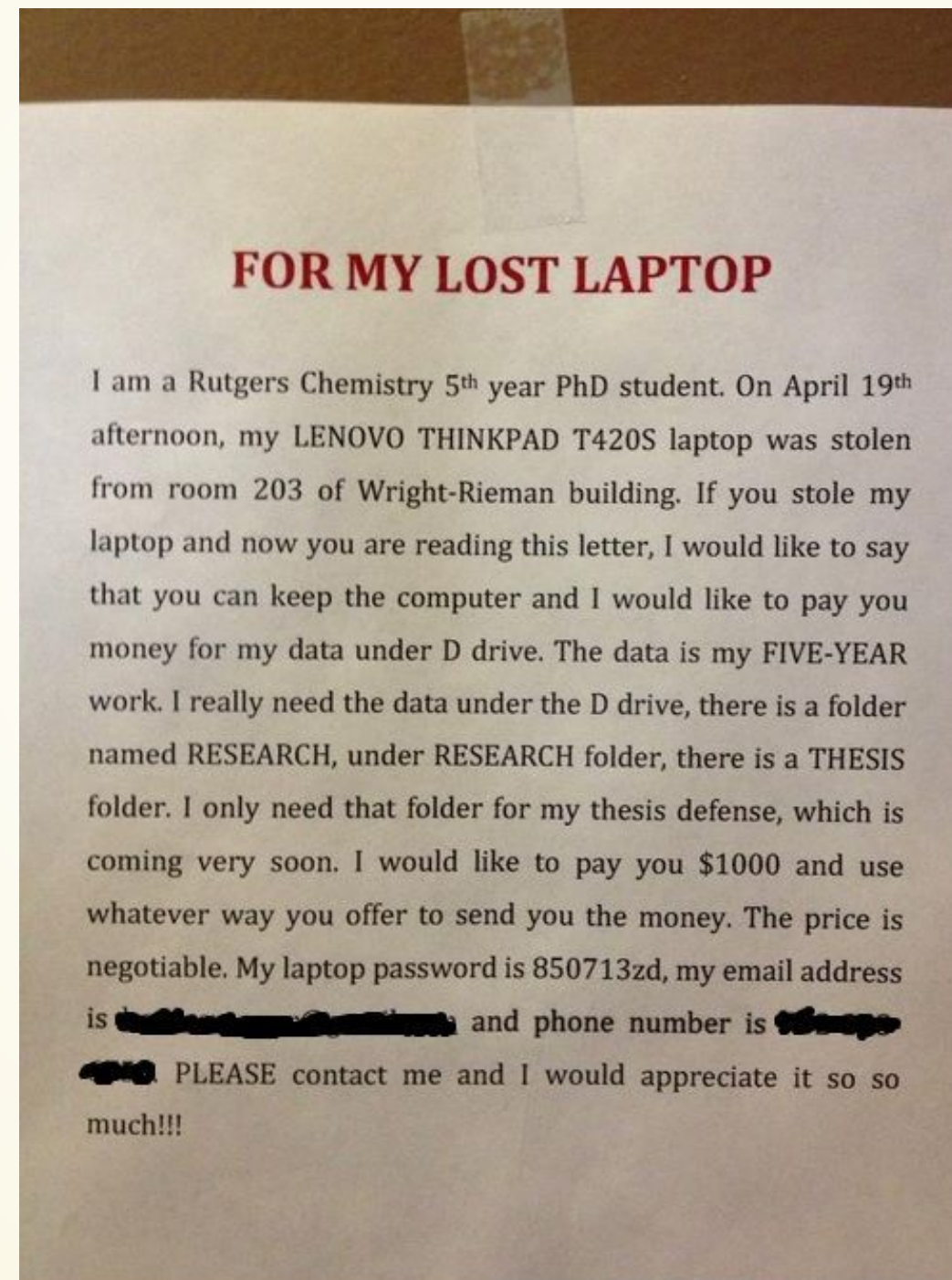
COMMON PROBLEMS IN SCIENCE

You write a paper & stay up late to generate good-looking figures, but you have to tweak many parameters and display options. The next morning, you have no idea which parameters produced which figures, and which of the figures fit to what you report in the paper.



COMMON PROBLEMS IN SCIENCE

Your research project produces phenomenal results, but your laptop, the only place that stores the source code for the results, is stolen or breaks



COMMON PROBLEMS IN SCIENCE

A graduate student complains that a research idea does not work. Their supervisor can't figure out what the student did and how, and the student can't sufficiently explain their approach (data, algorithms, software). Weeks of discussion and mis-communication ensues because the supervisor can't first-hand explore or use the students project.

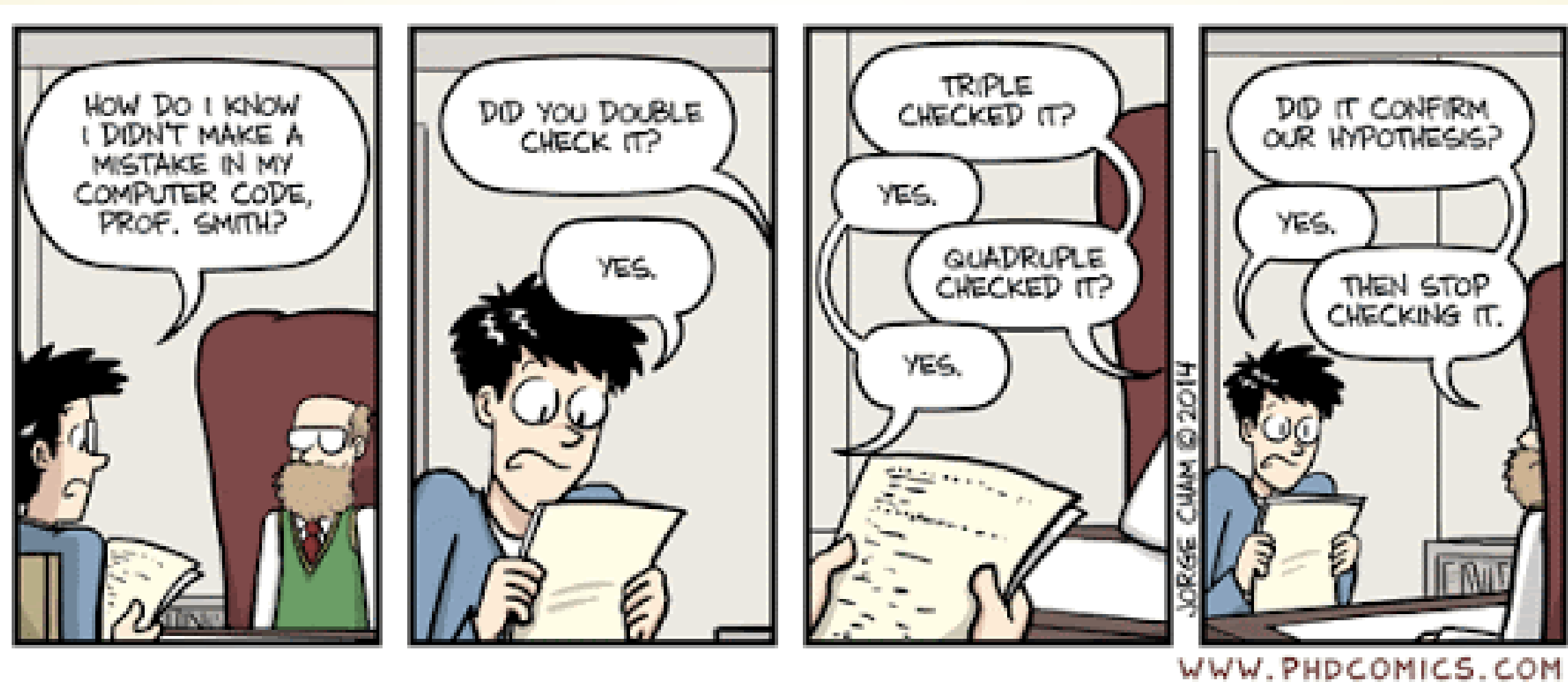


Image credit: <http://phdcomics.com/comics.php?f=1693>

COMMON PROBLEMS IN SCIENCE

You wrote a script during your PhD that applied a specific method to a dataset. Now, with new data and a new project, you try to reuse the script, but forgot how it worked.



Image credit: <http://phdcomics.com/comics.php?f=1693>

COMMON PROBLEMS IN SCIENCE

You try to recreate results from another lab's published paper. You base your re-implementation on everything reported in their paper, but the results you obtain look nowhere like the original.



~~COMMON~~ OLD PROBLEMS IN SCIENCE

All these problems were paraphrased from **Buckheit & Donoho, 1995**
Let's do better!



DataLad can help
with small or large-scale
data management

Free,
open source,
command line tool & Python API



- A command-line tool, available for all major operating systems (Linux, macOS/OSX, Windows), MIT-licensed
- Build on top of **Git** and **Git-annex**
- **Allows...**
 - ... **version-controlling arbitrarily large content**
version control data and software alongside to code!
 - ... **transport mechanisms for sharing and obtaining data**
consume and collaborate on data (analyses) like software
 - ... **(computationally) reproducible data analysis**
Track and share provenance of all digital objects
 - ... **and much more**
- Completely domain-agnostic

ACKNOWLEDGEMENTS

Software

- Joey Hess (git-annex)
- The DataLad team & contributors

Illustrations

- The Turing Way project & Scriberia



Funders



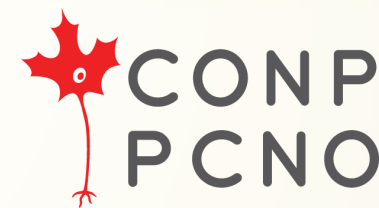
EUROPEAN UNION
European Regional Development Fund



Collaborators



Human Brain Project



VirtualBrainCloud



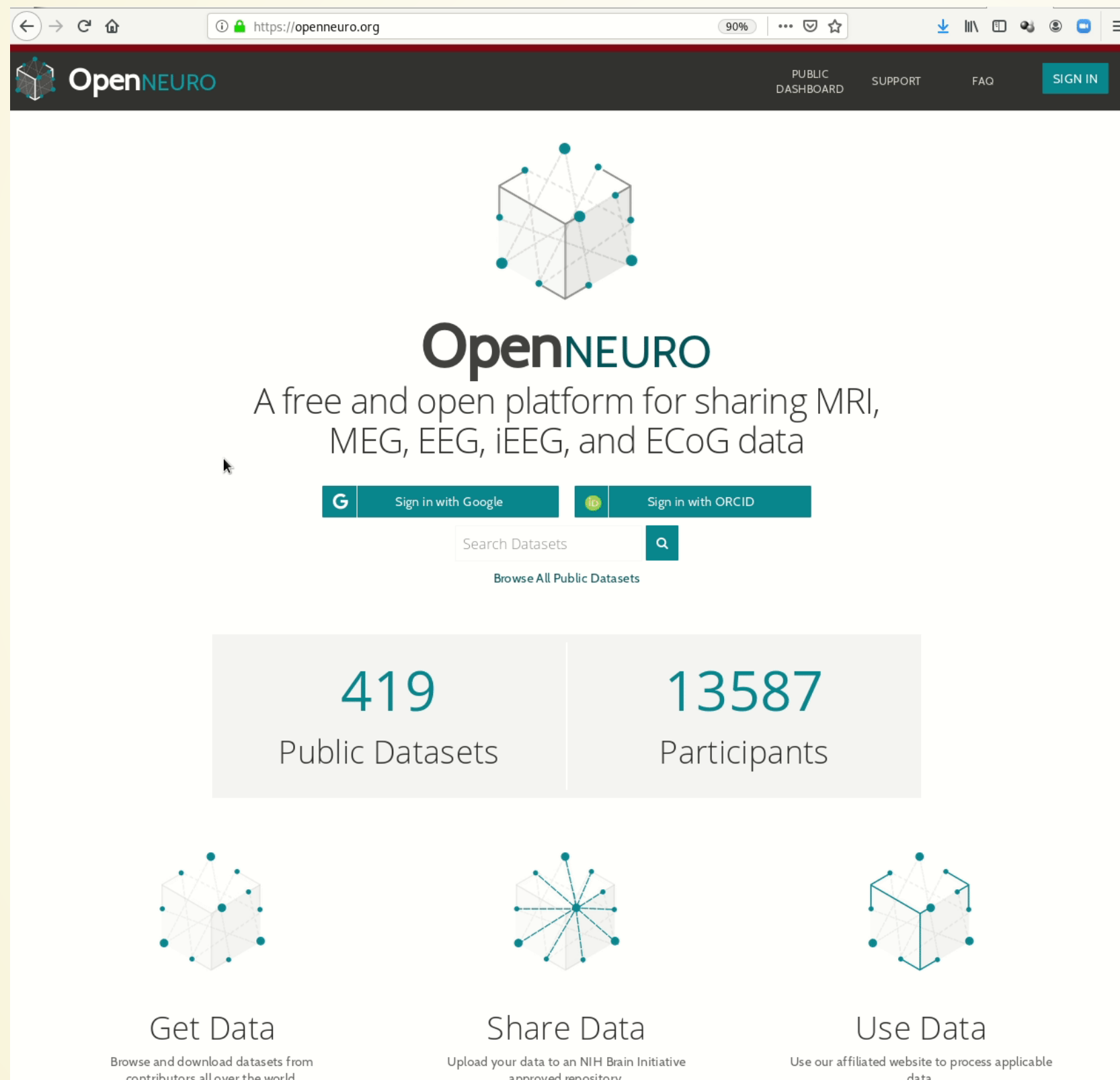
OpenNEURO



brainlife.io

EXAMPLES OF WHAT DATALAD CAN BE USED FOR:

- Behind-the-scenes infrastructure component for data transport and **versioning** (e.g., used by OpenNeuro, brainlife.io, the Canadian Open Neuroscience Platform (CONP), CBRAIN)



EXAMPLES OF WHAT DATALAD CAN BE USED FOR:

- Creating and sharing reproducible, open science: Sharing data, software, code, and provenance

The screenshot shows a GitHub repository page for 'psychoinformatics-de / paper-remodnav'. The repository has 8 Unwatch, 2 Stars, and 2 Forks. The main content area displays a list of files and folders with their commit history. The files include 'code', 'data', 'img', 'remodnav @ d289118', '.gitignore', '.gitmodules', 'COPYING', 'Makefile', 'README.md', 'main.tex', 'references.bib', 'results_def.tex', 'spbasic.bst', 'svglov3.clo', and 'svjour3.cls'. The 'README.md' file is highlighted, showing the title 'REMoDNaV: Robust Eye Movement'. The right sidebar contains sections for 'About', 'Releases', 'Packages', 'Contributors', and 'Languages'.

Repository: psychoinformatics-de / paper-remodnav

Unwatch 8 Stars 2 Fork 2

Code Issues 1 Pull requests 1 Actions Projects Wiki Security Insights

master 7 branches 2 tags Go to file Add file Code

Files:

File	Description	Commit Date
code	Merge pull request #12 from psychoinformatics-de/finalr...	4 months ago
data	Point to latest label dataset	17 months ago
img	Label panels in flowchart	6 months ago
remodnav @ d289118	Update remodnav with latest test dataset	17 months ago
.gitignore	Prevent permanent rebuilds of the figures	17 months ago
.gitmodules	Add and use studyforrest raw eyegaze dataset as direct...	17 months ago
COPYING	Declare CC-BY license	17 months ago
Makefile	Simplify reference management	7 months ago
README.md	Fix installation instructions	4 months ago
main.tex	Minor edits as suggested by reviewer 2	5 months ago
references.bib	velocity versus dispersion-based: cite a few algorithms ...	7 months ago
results_def.tex	Put generated results back into Git	8 months ago
spbasic.bst	Basic switch to new layout	2 years ago
svglov3.clo	Basic switch to new layout	2 years ago
svjour3.cls	Basic switch to new layout	2 years ago

README.md

REMoDNaV: Robust Eye Movement

About

Code, data and manuscript for <https://doi.org/10.1101/619254>

Readme

CC-BY-4.0 License

Releases

2 tags

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Contributors 3

- adswa Adina Wagner
- mih Michael Hanke
- ElectronicTeaCup Asim H ...

Languages

- TeX 79.4%
- Python 20.0%
- Makefile 0.6%

EXAMPLES OF WHAT DATALAD CAN BE USED FOR:

- Creating and sharing reproducible, open science: Sharing data, software, code, and provenance

The screenshot shows a Twitter thread on a mobile web interface. The main tweet is from Lennart Wittkuhn (@lnnrtwttkhn) dated 19. März. It discusses a new fMRI analysis method and includes a link to a paper on Nature. Below the main tweet is a reply from the same user, stating that all code and data are shared via @gnode and #GitHub, and that the data is version-controlled with @datalad (ca. 1.5 TB). The thread also includes a link to the project website wittkuhn.mpib.berlin. The interface shows engagement metrics (4 replies, 93 retweets, 270 likes) and a sidebar with 'Relevante Personen' (Lennart Wittkuhn, INCF G-Node, DataLad) and 'Trends für dich' (Regierung · Trends, #Merkel, Trend in Deutschland, #Generalstreik, Regierung · Trends, #Laschet, Trend in Deutschland, #AliAkbar, Trend in Deutschland, #Tanzverbot).

Thread

Lennart Wittkuhn @lnnrtwttkhn · 19. März
Excited to share work w/ @nico_schuck out now in @NatureComms! 🌟

We introduce a new fMRI analysis method to decode fast neural event sequences and report replay in visual cortex following a non-mnemonic task! 🧠

- 📄 Paper: [nature.com/articles/s41466...](https://www.nature.com/articles/s41466-021-00846-1)
- 📖 Thread below! 🗨️ [1/n]

Dynamics of fMRI patterns reflect sub-second activation sequences. Non-invasive measurement of fast neural activity with spatial precision in humans is difficult. Here, t...
🔗 [nature.com](https://www.nature.com/articles/s41466-021-00846-1)

4 93 270

Lennart Wittkuhn @lnnrtwttkhn
Antwort an @lnnrtwttkhn

We share all code + data via @gnode + #GitHub, version-controlled with @datalad (ca. 1.5 TB): MRI in @BIDSstandard, #fMRIPrep data, #MRIQC metrics, GLMs + anatomical masks, task code, decoding pipeline, statistical analyses: [wittkuhn.mpib.berlin /highspeed/](https://wittkuhn.mpib.berlin/highspeed/) #OpenScience 🇩🇪 [2/n]

[Tweet übersetzen](#)

Dynamics of fMRI patterns reflect sub-second activation se...
This is the project website of the accompanying the paper 'Faster than thought: Detecting sub-second activation ...'
🔗 wittkuhn.mpib.berlin

11:35 vorm. · 19. März 2021 · Twitter Web App

7 Retweets 2 Zitierte Tweets 43 „Gefällt mir“-Angaben

Lennart Wittkuhn @lnnrtwttkhn · 19. März
Antwort an @lnnrtwttkhn

Relevante Personen

Lennart Wit... @l... Folgt Dir Folge ich
PhD candidate @mpib_berlin and @MPC_CompPsych interested in hippocampal replay, decision-making and open science tools

INCF G-Node @gnode Folge ich
The German Neuroinformatics Node

DataLad @datalad Folge ich
There was Debian. git came, followed by git-annex. DataLad was born to be a data distribution, but grew into a distributed Research Data Management solution.

Trends für dich

Regierung · Trends
#Merkel
47.000 Tweets

Trend in Deutschland
#Generalstreik
2.678 Tweets

Regierung · Trends
#Laschet
4.584 Tweets

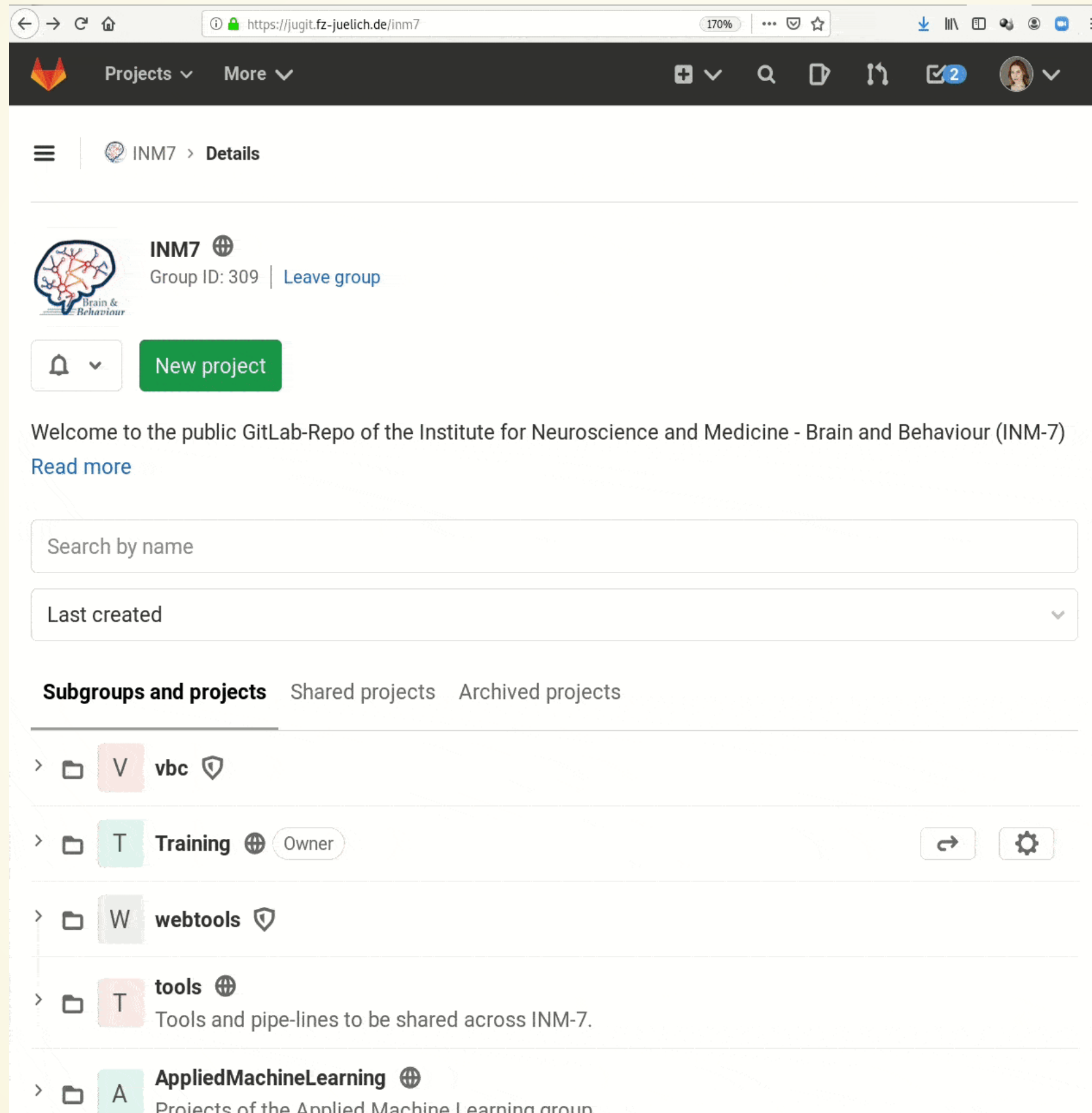
Trend in Deutschland
#AliAkbar
27.500 Tweets

Trend in Deutschland
#Tanzverbot

[Mehr anzeigen](#)

EXAMPLES OF WHAT DATALAD CAN BE USED FOR:

- Central data management and archival system



EXAMPLES OF WHAT DATALAD CAN BE USED FOR:

- Scalable computing framework for reproducible science

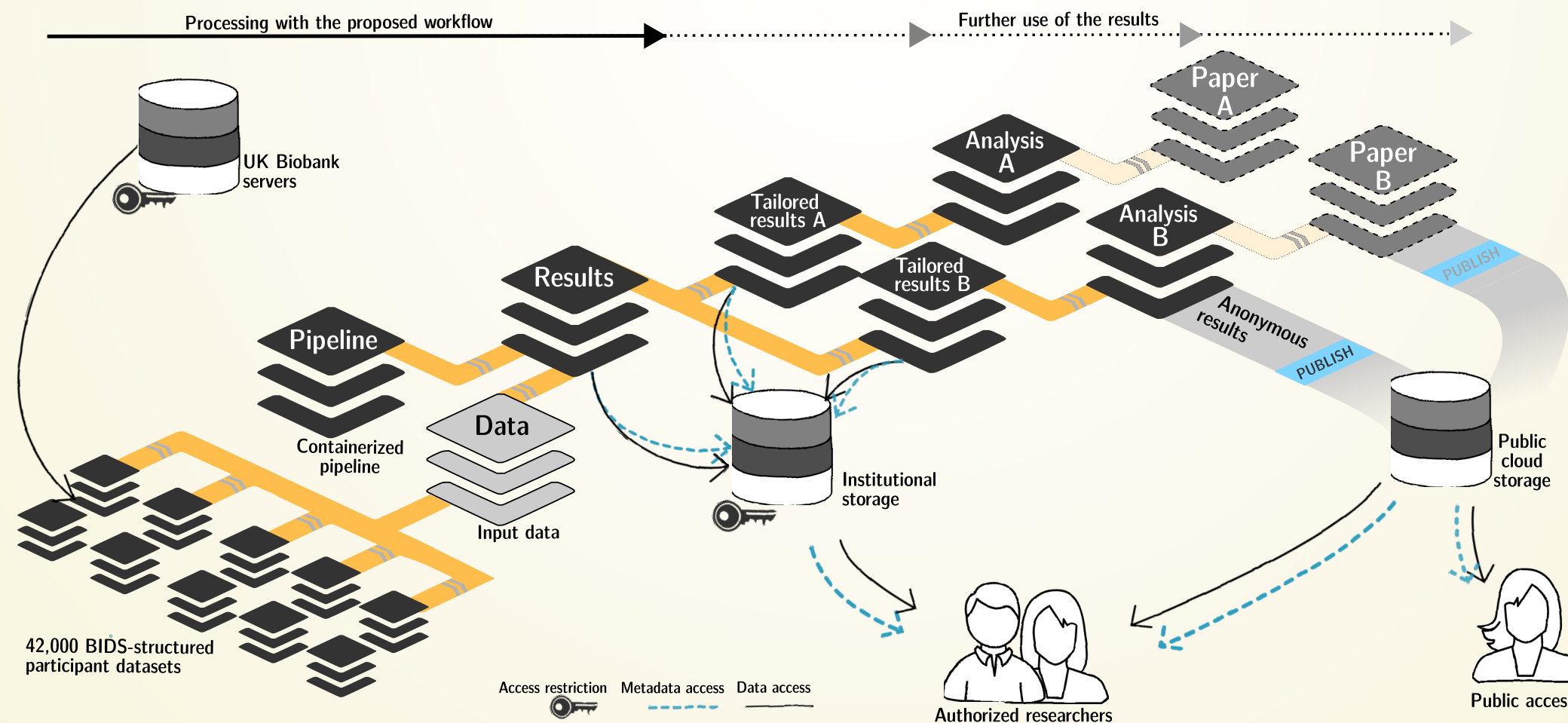
Article | [Open Access](#) | [Published: 11 March 2022](#)

FAIRly big: A framework for computationally reproducible processing of large-scale data

[Adina S. Wagner](#) , [Laura K. Waite](#), [Małgorzata Wierzbą](#), [Felix Hoffstaedter](#), [Alexander Q. Waite](#), [Benjamin Poldrack](#), [Simon B. Eickhoff](#) & [Michael Hanke](#)

[Scientific Data](#) **9**, Article number: 80 (2022) | [Cite this article](#)

813 Accesses | **23** Altmetric | [Metrics](#)



PREREQUISITES: TERMINAL

- DataLad can be used from the command line

```
datalad create mydataset
```

- ... or with its Python API

```
import datalad.api as dl  
dl.create(path="mydataset")
```

- ... and other programming languages can use it via system call

```
# in R  
> system("datalad create mydataset")
```

PREREQUISITES: TERMINAL



datalad-hub.inm7.de

Unix terminal cheatsheet (incl. Windows equivalents)

PREREQUISITES: INSTALLATION AND CONFIGURATION

- Your installed version of DataLad should be at least 0.16

```
datalad --version  
0.17.2
```

- DataLad relies on Git to create a revision history with detailed information on what was changes, when, and how. Therefore, you should tell Git who you are and configure a Git identity (name and email). Find out if an identity is set by running either of:

```
$ git config --get user.name  
Adina Wagner  
$ git config --get user.email  
adina.wagner@t-online.de
```

```
$ datalad configuration get user.name user.email  
Adina Wagner  
adina.wagner@t-online.de
```

- Set a Git identity using either of

```
$ git config set --global \  
user.name "Adina Wagner"  
$ git config set --global \  
user.email "adina.wagner@t-online.de"
```

```
$ datalad configuration --scope global \  
set user.name="Adina Wagner"  
$ datalad configuration --scope global \  
set user.email="adina.wagner@t-online.de"
```

- Allow brand-new DataLad functionality:

```
datalad configuration --scope global set datalad.extensions.load=next
```

Find installation and configuration instructions at handbook.datalad.org

PREREQUISITES: USING DATALAD

- Every DataLad command consists of a main command followed by a sub-command. The main and the sub-command can have options.

API	datalad [--GLOBAL-OPTION <opt. flag spec.>] COMMAND [ARGUMENTS] [--OPTION <opt. flag spec.>]		Each datalad invocation can have two sets of options: general options are given first, command-specific ones go after the subcommand.
GLOBAL OPTIONS	-c KEY=VALUE Set config variables (overrides configurations in files) -f/--output-format default json json_pp tailored Specify the format for command result rendering -l/--log-level critical error warning info debug Set logging verbosity level	COMMAND OPTIONS	
		-d/--dataset Dataset location: path to root, or ^ for superdataset -D/--description A location description (e.g., "my backup server") -f/--force Force execution of a command (Dangerzone!) -m/--message A description about a change made to the dataset -r/--recursive Perform an operation recursively across subdatasets -R/--recursion-limit <n> Limit recursion to n subdataset levels	

- Example (main command, subcommand, several subcommand options):

```
$ datalad save -m "Saving changes" --recursive
```

- Use `--help` to find out more about any (sub)command and its options, including detailed description and examples (`q` to close). Use `-h` to get a short overview of all options

```
$ datalad save -h
Usage: datalad save [-h] [-m MESSAGE] [-d DATASET] [-t ID] [-r] [-R LEVELS]
                  [-u] [-F MESSAGE_FILE] [--to-git] [-J NJOBS] [--amend]
                  [--version]
                  [PATH ...]

Use '--help' to get more comprehensive information.
```

If everything is important...
...track everything!



<http://datalad.org>

EXHAUSTIVE TRACKING OF RESEARCH COMPONENTS



- text document
- source code
- binary file
- regular file
- file reference:
identity and
availability

Well-structured datasets (using community standards), and portable computational environments — and their evolution — are the precondition for reproducibility

```
# turn any directory into a dataset  
# with version control
```

```
% datalad create <directory>
```

```
# save a new state of a dataset with  
# file content of any size
```

```
% datalad save
```

CAPTURE COMPUTATIONAL PROVENANCE

- text document
- source code
- binary file
- regular file
- file reference:
identity and
availability



Which data was needed at which version, as input into which code, running with what parameterization in which computational environment, to generate an outcome?

```
# execute any command and capture its output
# while recording all input versions too

% datalad run --input ... --output ... <command>
```

EXHAUSTIVE CAPTURE ENABLES PORTABILITY



Precise identification of data and computational environments, combined for provenance records form a comprehensive and portable data structure, capturing all aspects of an investigation.

```
# transfer data and metadata to other sites and services
# with fine-grained access control for dataset components

% datalad push --to <site-or-service>
```

REPRODUCIBILITY STRENGTHENS TRUST



Outcomes of computational transformations can be validated by authorized 3rd-parties. This enables audits, promotes accountability, and streamlines automated "upgrades" of outputs

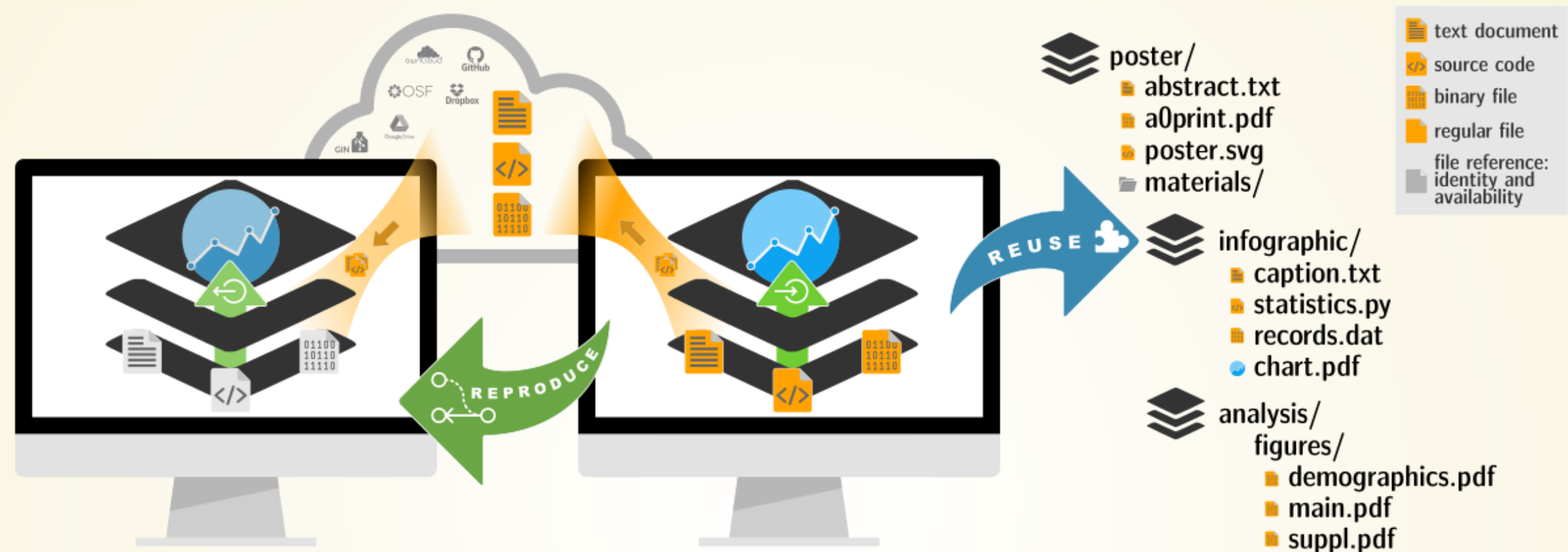
```
# obtain dataset (initially only identity,  
# availability, and provenance metadata)
```

```
% datalad clone <url>
```

```
# immediately actionable provenance records  
# full abstraction of input data retrieval
```

```
% datalad rerun <commit|tag|range>
```

ULTIMATE GOAL: (RE-)USABILITY

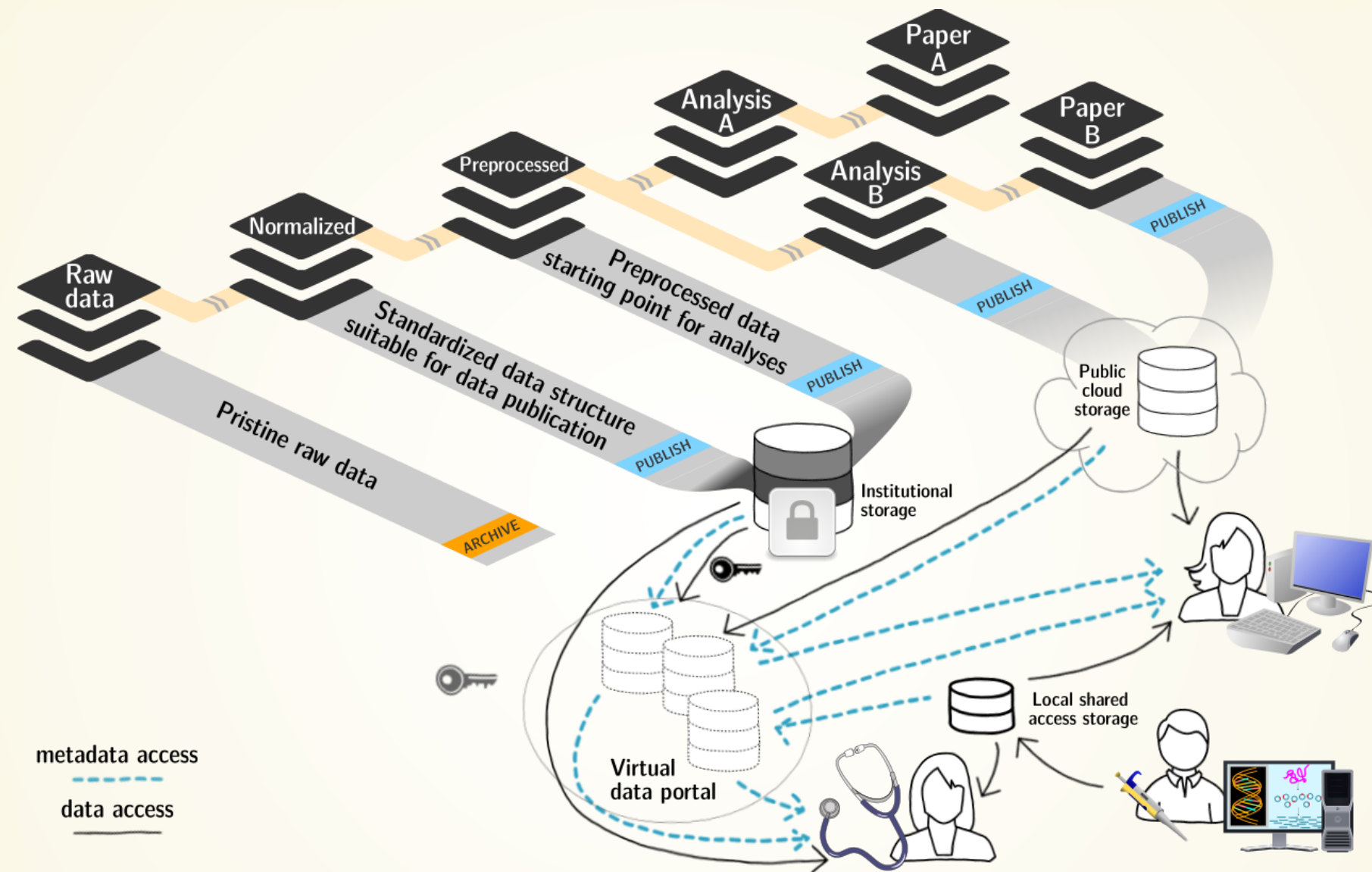


Verifiable, portable, self-contained data structures that track all aspects of an investigation exhaustively can be (re-)used as modular components in larger contexts — propagating their traits

```
# declare a dependency on another dataset and
# re-use it a particular state in a new context

% datalad clone -d <superdataset> <url> <path-in-dataset>
```

DATALAD: MANAGE (CO-)EVOLUTION OF DIGITAL OBJECTS



Consume, create, curate, analyze, publish, and query data with full provenance capture and "universal" metadata support.

DataLad is free and open source (MIT-licensed). <http://datalad.org>

Halchenko, Meyer, Poldrack, ... & Hanke, M. (2021). DataLad: distributed system for joint management of code, data, and their relationship. *Journal of Open Source Software*, 6(63), 3262.

LET'S TRY...