

DataLad – decentralized data distribution for consumption and sharing of scientific datasets

Yaroslav O. Halchenko¹, Benjamin Poldrack², Michael Hanke²

¹ Dartmouth College, Hanover, NH, USA

² Otto-von-Guericke University, Magdeburg, Germany

OHBM 2016, Geneva, Switzerland 2016



<http://datalad.org>



<http://www.pympva.org>



<http://NeuroDebian.net>



<http://duecredit.org>

Visit our DataLad/NeuroDebian exhibit table and posters #1855, #1870

Acknowledgements

Centroids

Yaroslav O. Halchenko
James V. Haxby
Matteo Visconti di Oleggio
Castello
Samuel Nastase

Collaborators

Michael Hanke
Nikolaas N. Oosterhof
Matthew Brett
Joey Hess
Benjamin Poldrack



debian.org



1429999

Collaborating projects



Partners



Houston, we've got a problem...

Data is a 2nd-class citizen within software platforms

Why?

- tarballs are **inefficient** distribution format
- **absent versioning** of data

derived and/or curated data does change!

Why?

A STORY TOLD IN FILE NAMES:

Location: C:\user\research\data

Filename	Date Modified	Size	Type
data_2010.05.28_test.dat	3:37 PM 5/28/2010	420 KB	DAT file
data_2010.05.28_re-test.dat	4:29 PM 5/28/2010	421 KB	DAT file
data_2010.05.28_re-re-test.dat	5:43 PM 5/28/2010	420 KB	DAT file
data_2010.05.28_calibrate.dat	7:17 PM 5/28/2010	1,256 KB	DAT file
data_2010.05.28_huh??.dat	7:20 PM 5/28/2010	30 KB	DAT file
data_2010.05.28_WTF.dat	9:58 PM 5/28/2010	30 KB	DAT file
data_2010.05.29_aaarrgh.dat	12:37 AM 5/29/2010	30 KB	DAT file
data_2010.05.29_#&*!&!.dat	2:40 AM 5/29/2010	0 KB	DAT file
data_2010.05.29_crap.dat	3:22 AM 5/29/2010	437 KB	DAT file
data_2010.05.29_notbad.dat	4:16 AM 5/29/2010	670 KB	DAT file
data_2010.05.29_woohoo!.dat	4:47 AM 5/29/2010	1,349 KB	DAT file
data_2010.05.29_USETHISONE.dat	5:08 AM 5/29/2010	2,894 KB	DAT file
analysis_graphs.xls	7:13 AM 5/29/2010	455 KB	XLS file
ThesisOutline!.doc	7:26 AM 5/29/2010	38 KB	DOC file
Notes_Meeting_with_ProfSmith.txt	11:38 AM 5/29/2010	1,673 KB	TXT file
JUNK...	2:45 PM 5/29/2010		Folder
data_2010.05.30_startingover.dat	8:37 AM 5/30/2010	420 KB	DAT file

ye!

Why?

- tarballs are **inefficient** distribution format

- **absent versioning** of data

derived and/or curated data does change!

- code version control systems are **inadequate** for data

duplication, monolithic storage, etc.

- **absent data distributions**

no efficient ways to install and upgrade

- **cacophony** of authorization schemes and interfaces

- **absent data testing**

*data can and **does** have bugs (see e.g. Halchenko, 2012;
Rohlfing, 2013)*

- **difficulty to share** derivative data

shareable? where to host? how to “link” back?

Welcome [datalad.org](https://data-lad.org)

[DataLad](#)[About](#)[Development](#)[Articles](#)[Archives](#)[←](#) [→](#) [↺](#) [🏠](#) [📄](#) [git-scm.com](#)

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Using Git ...

Git is **easy to learn** and has **many features with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.

[Visit git website »](#)

Learn Git in your browser for free with [Try Git](#).

DataLad aims to provide access to scientific data available from various sources (e.g. lab or consortium web-sites such as Human connectome; data sharing portals such as OpenFMRI and CRCNS) through a single convenient interface and integrated with your software package managers (such as APT in Debian). Although initially targeting neuroimaging and neuroscience data in general, it will not be limited by the domain and we would welcome a wide range of contributions.

DataLad's goal

is to develop a data distribution platform

- with **unambiguous versioning**
- **without data duplication** - data and authorization stays with original data providers
- **distributed** data storage and management model
- providing **uniform access** to a wide range of data sources
- **scalable** to manage terabytes of data
- **integrated** with existing software distributions
- **trustworthy** to rely upon in critical applications
- **available** across all major operating systems

DataLad's goal

is to develop a data distribution platform

- with **unambiguous versioning**
- **without data duplication** - data and authorization stays with original data providers
- **distributed** data storage and management model
- providing **uniform access** to a wide range of data sources
- **scalable** to manage terabytes of data
- **integrated** with existing software distributions
- **trustworthy** to rely upon in critical applications
- **available** across all major operating systems

Managing data should be as easy as managing code and software

How: Foundation #1 – Git

DataLad is

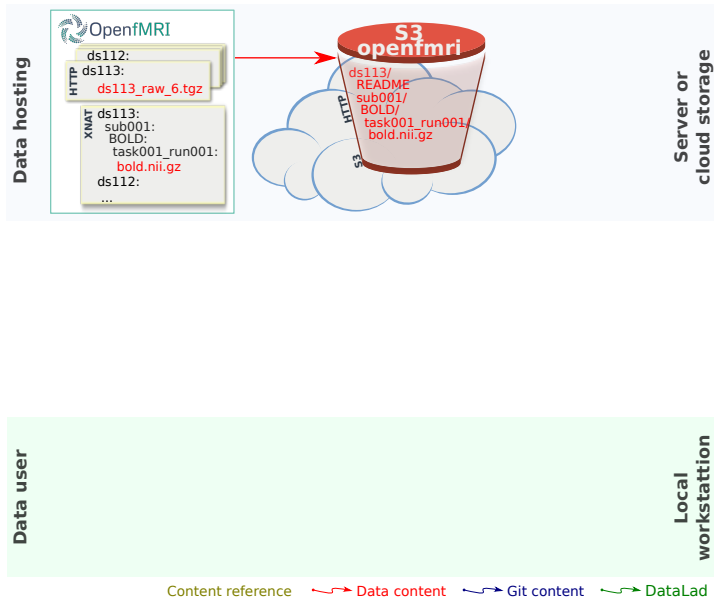
- built on and compatible with git
- all version-control and (distributed) workflow features are supported
- a datalad “distribution” is a plain git repository with sub-modules filled with meta data
- use GitHub or any other git server for collaboration, make data available from elsewhere (institutional website, cloud, *etc.*)

How: Foundation #2 – Git-annex

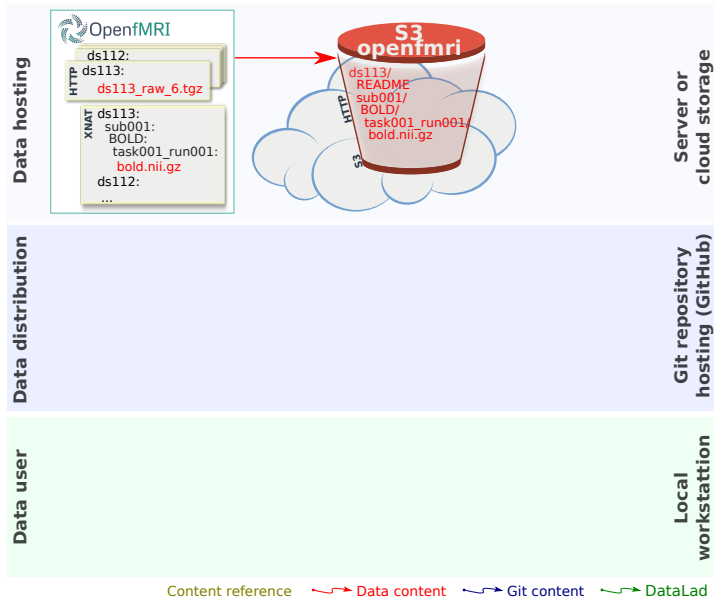
Git-annex

- provides access to data load from variety of sources: HTTP, FTP, RSYNC, Amazon S3, *etc.*
- allows for custom extensions to get access to the data.
DataLad uses that facility to provide access to data from tarballs, XNAT, COINS (let's talk!), ...*etc.*
- features optional Dropbox-like synchronization facility via *git-annex assistant*.

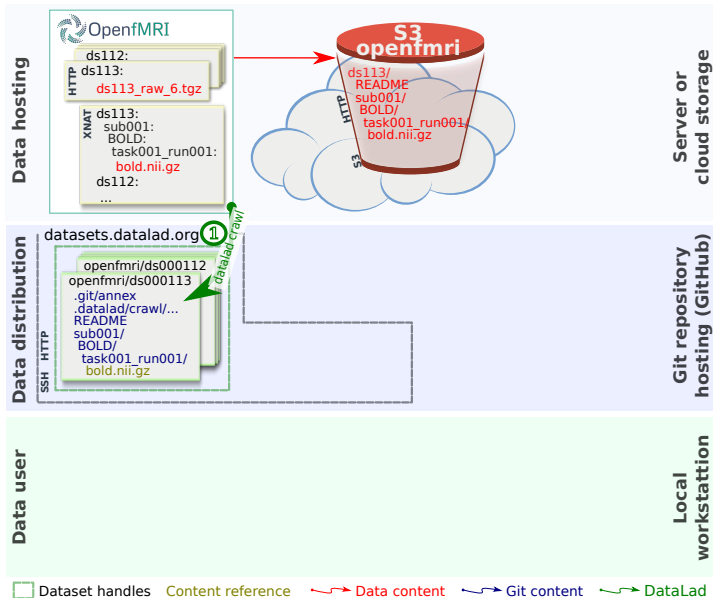
DataLad data distribution: Data life cycle



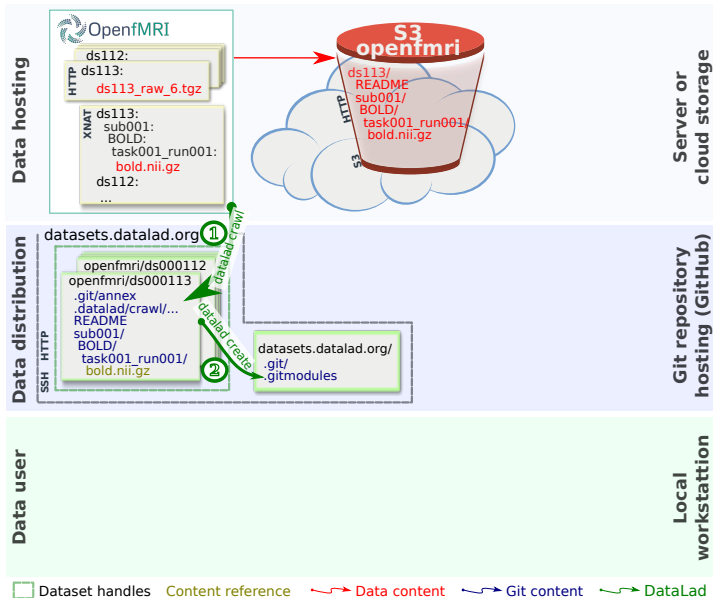
DataLad data distribution: Data life cycle



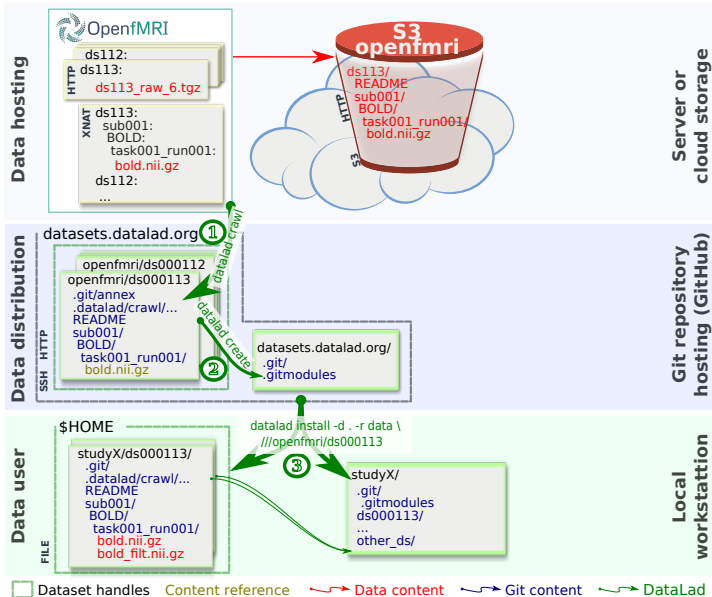
DataLad data distribution: Data life cycle



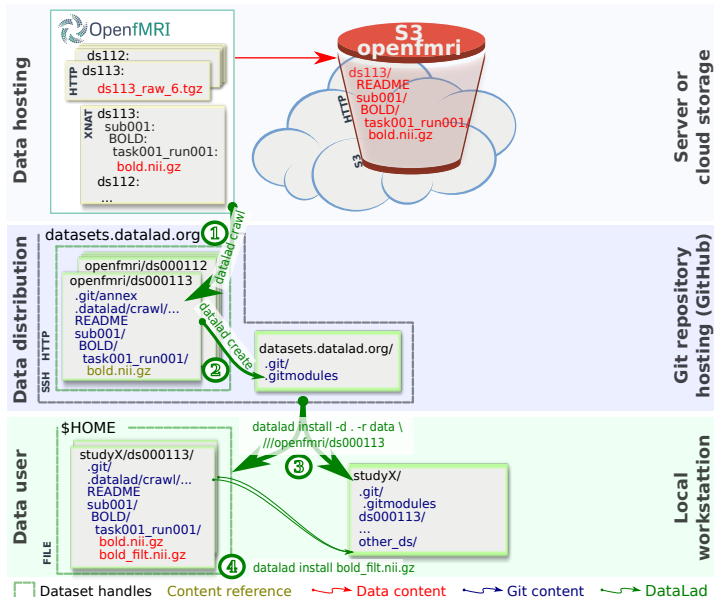
DataLad data distribution: Data life cycle



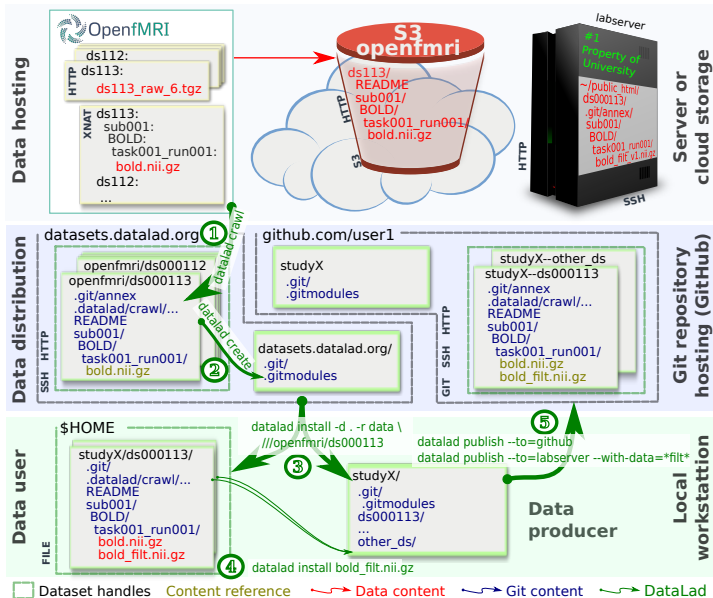
DataLad data distribution: Data life cycle



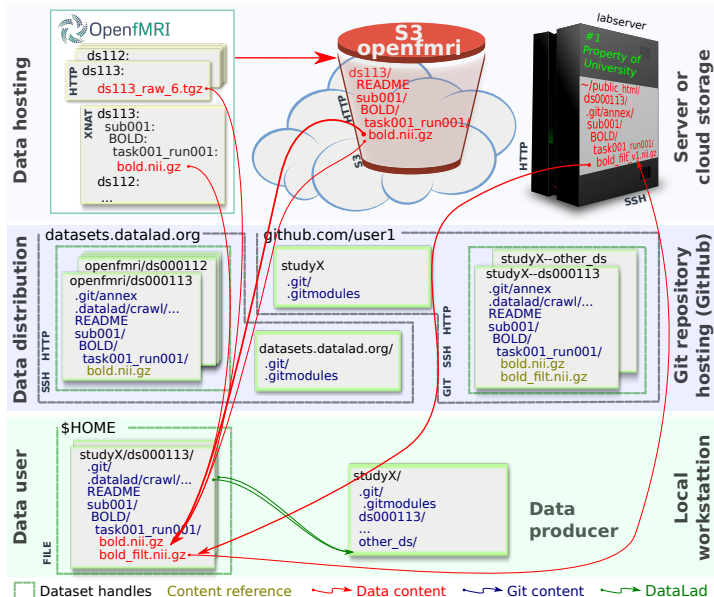
DataLad data distribution: Data life cycle



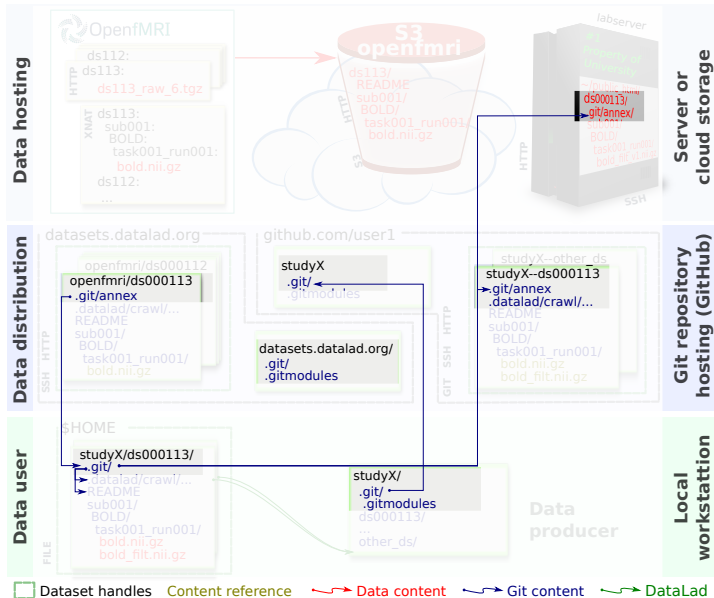
DataLad data distribution: Data life cycle




DataLad data distribution: Data life cycle



DataLad data distribution: Data life cycle



OpenfMRI ds000001: [website](https://openfmri.org/dataset/ds000001/)

 <https://openfmri.org/dataset/ds000001/>

Revision: 2.0.0 Date Set: May 24, 2016, 7:26 p.m.

Notes:

- Converted to BIDS standard.

Data Associated with Revision:

- [Raw data on AWS](#)

Revision: 1.1.0 Date Set: Feb. 18, 2016, 8:28 p.m.

Notes:

Updated orientation information in NIFTI headers for better left-right determination.

Data Associated with Revision:

- [Raw data checksums](#)
- [Raw data on AWS](#)

Revision: 1.0.0 Date Set: July 10, 2012, 8:28 p.m.

Data Associated with Revision:

OpenfMRI ds000001: gitk

File Edit View Help

2.0.0+2 master remotes/origin/master Merge branch 'incoming-processed'

remotes/origin/incoming-processed Added files from extracted archives

remotes/origin/incoming Updated git/annex from a remote location

2.0.0+1 Merge branch 'incoming-processed'

Added files from extracted archives

Updated git/annex from a remote location

2.0.0 Merge branch 'incoming-processed'

Added files from extracted archives

Updated git/annex from a remote location

1.1.0+1 Merge branch 'incoming-processed'

Added files from extracted archives

Updated git/annex from a remote location

Adjusted crawler configuration: crawl:pipeline section and _dataset

1.1.0 Merge branch 'incoming-processed'

Added files from extracted archives

Updated git/annex from a remote location (Multi-version commit #2/2: 1.1.0. Remainin

1.0.0 Merge branch 'incoming-processed'

Added files from extracted archives

SHA1 ID: 6a6549410895bce39a9fb56da36fd915faa49dd8 Row 19/ 31

Find commit containing: Exact All fields

Search

◆ Diff ◆ Old version ◆ New version Lines of context: 3 Ignore space change Line dif

Follows:
Precedes: 2.0.0

Added files from extracted archives

Files processed: 134
renamed: 133
+git: 5
+annex: 128

◆ Patch ◆ Tree

Comments

CHANGES

dataset_description.json

participants.tsv

sub-01/anat/sub-01_T1w.nii.gz

sub-01/anat/sub-01_inplaneT2.nii.gz

sub-01/func/sub-01_task-balloonanalogrisktask

run-01_bold.nii.gz

Yaroslav Halchenko <debian@c 2016-06-08 18:01:53

Yaroslav Halchenko <debian@c 2016-06-08 18:01:52

Yaroslav Halchenko <debian@c 2016-06-08 17:59:07

Yaroslav Halchenko <debian@c 2016-05-25 11:00:47

Yaroslav Halchenko <debian@c 2016-05-25 11:00:46

Yaroslav Halchenko <debian@c 2016-05-25 10:58:15

Yaroslav Halchenko <debian@c 2016-05-24 16:03:33

Yaroslav Halchenko <debian@c 2016-05-24 16:03:33

Yaroslav Halchenko <debian@c 2016-05-24 16:00:55

Yaroslav Halchenko <debian@c 2016-05-23 15:29:24

Yaroslav Halchenko <debian@c 2016-05-23 15:29:23

Yaroslav Halchenko <debian@c 2016-05-23 15:27:13

Yaroslav Halchenko <debian@c 2016-05-23 14:53:27

Yaroslav Halchenko <debian@c 2016-03-31 00:28:17

Yaroslav Halchenko <debian@c 2016-03-31 00:28:17

Yaroslav Halchenko <debian@c 2016-03-31 00:26:39

Yaroslav Halchenko <debian@c 2016-03-31 00:27:13

Yaroslav Halchenko <debian@c 2016-03-31 00:27:13

Our growing “distribution” :

- <http://datasets.datalad.org>

Covered :

- <http://openfmri.org> (S3)
- <http://crcns.org>
- <http://studyforrest.org>

Coming :

- <http://humanconnectome.org> (S3, XNAT)
- <http://nitrc.org/ir> (INDI, FCP, *etc.*)
- <http://coins.mrn.org> (COINS)

Meta-data to facilitate search, custom views etc

Integration NeuroDebian

```
apt-get install openfmri-ds000113  
apt-get install openfmri
```



AutomagicIO: automatically fetch necessary files

Given Python code which accesses files within annex repository (example from PyMVPA):

www.pympva.org/examples/hyperalignment.html

using a 12 dof linear transformation.

```
verbose(1, "Loading data...")
filepath = os.path.join(cfg.get('location', 'tutorial data'),
                        'hyperalignment_tutorial_data.hdf5.gz')
ds_all = h5load(filepath)
# zscore all datasets individually
_ = [zscore(ds) for ds in ds_all]
# inject the subject ID into all datasets
for i, sd in enumerate(ds_all):
    sd.sa['subject'] = np.repeat(i, len(sd))
# number of subjects
nsubjs = len(ds_all)
# number of categories
ncats = len(ds_all[0].UT)
# number of run
nruns = len(ds_all[0].UC)
verbose(2, "%d subjects" % len(ds_all))
verbose(2, "Per-subject dataset: %i samples with %i features" % ds_all[0].shape)
verbose(2, "Stimulus categories: %s" % ', '.join(ds_all[0].UT))
```


AutomagicIO: automatically fetch necessary files

DataLad can automatically fetch necessary load whenever specific file is requested:

```
2 5329.....:Thu 23 Jun 2016 12:39:11 PM CEST:.  
(git)hopa:/tmp/PyMVPA[master]  
$> datalad install -s http://data.pympa.org/datasets/tutorial_data /tmp/tutorial_data  
2016-06-23 12:39:13,771 [INFO ] Installing /tmp/tutorial_data (install.py:353)  
1 installed item is available at  
<Dataset path=/tmp/tutorial_data>  
2 5329.....:Thu 23 Jun 2016 12:39:13 PM CEST:.  
(git)hopa:/tmp/PyMVPA[master]  
$> MVPA_LOCATION_TUTORIAL_DATA=/tmp/tutorial_data python -m datalad doc/examples/hyperalignment.  
py  
Loading data...  
2016-06-23 12:39:19,746 [INFO ] File /tmp/tutorial_data/hyperalignment_tutorial_data.hdf5.gz h  
as no content -- retrieving (auto.py:164)  
/tmp/tutorial_data/.git 100%[=====>] 15.04M --.-KB/s in 0.02s  
10 subjects  
Per-subject dataset: 56 samples with 3509 features  
Stimulus categories: Chair, DogFace, FemaleFace, House, MaleFace, MonkeyFace, Shoe  
Performing classification analyses...  
within-subject... done in 1.2 seconds  
between-subject (anatomically aligned)...done in 0.6 seconds  
between-subject (hyperaligned)...done in 3.3 seconds  
Average classification accuracies:  
within-subject: 0.57 +/-0.063  
between-subject (anatomically aligned): 0.42 +/-0.035  
between-subject (hyperaligned): 0.62 +/-0.050
```

DataLad's summary

DataLad ...

- provides simplified interface
- uses pure git/git-annex repositories under – power users can stay in power
- helps with authentication, crawling of available resources, and accessing data from archives
- is ready for you to start using it, documentation is growing:
datalad.readthedocs.org

DataLad's summary

DataLad ...

- provides simplified interface
- uses pure git/git-annex repositories under – power users can stay in power
- helps with authentication, crawling of available resources, and accessing data from archives
- is ready for you to start using it, documentation is growing:
datalad.readthedocs.org

Managing data is similar to managing code and software

Brain Download:



iz compltes.

Thank you!

For more information visit

DataLad poster: #1855 12:45 - 02:45 PM (today)

DataLad exhibit table (thank you OHBM)

Website: datalad.org

Github: github.com/datalad

Twitter: [@datalad](https://twitter.com/datalad) (I am [@yarikoptic](https://twitter.com/yarikoptic))

References

- Halchenko, Y. O. (2012). Incorrect probabilities in Harvard-Oxford-sub left hemisphere. [Retrived 11-Mar-2013].
- Rohlfing, T. (2013). Incorrect icbm-dti-81 atlas orientation and white matter labels. *Frontiers in Neuroscience*, 7(4).

DataLad's testing



✓ All is well — 9 successful checks

[Hide all checks](#)

✓ **datalad-pr-virtualbox-dl-win7-64** — DEV build done.

[Details](#)

✓ **datalad-pr-docker-dl-nd80** — DEV build done.

[Details](#)

✓ **datalad-pr-docker-dl-nd14_10** — DEV build done.

[Details](#)

✓ **datalad-pr-docker-dl-nd70** — DEV build done.

[Details](#)

✓ **datalad-pr-docker-dl-nd14_04** — DEV build done.

[Details](#)

✓ **datalad-pr-docker-dl-nd90** — DEV build done.

[Details](#)

✓ **continuous-integration/travis-ci/pr** — The Travis CI build passed

[Details](#)

✓ **coverage/coveralls** — Coverage increased (+0.18%) to 83.88%

[Details](#)

✓ **datalad-pr-dl-osx-64** — DEV build done.

[Details](#)

This pull request can be automatically merged.

You can also merge branches on the [command line](#).

 **Merge pull request**