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

Yaroslav O. Halchenko\textsuperscript{1}, Benjamin Poldrack\textsuperscript{2}, Michael Hanke\textsuperscript{2}

\textsuperscript{1} Dartmouth College, Hanover, NH, USA
\textsuperscript{2} Otto-von-Guericke University, Magdeburg, Germany

OHBM 2016, Geneva, Switzerland 2016

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

- 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?
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

Using Git ... the most popular state-of-the-art distributed version control system

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.

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

http://datalad.org
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

http://datalad.org
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.)
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

**Local workstation**

**Data user**

**Content reference** → **Data content** → **Git content** → **DataLad**

**Data hosting**

**Server or cloud storage**

OpenfMRI

| ds112: |
| ds113: |
| ds113: |

XNAT

| sub001: |
| BOLD: |
| task001_run001: |
| bold.nii.gz |

S3openfmri

| ds113/ |
| README |
| sub001/ |
| BOLD/ |
| task001_run001/ |
| bold.nii.gz |

| ds113_raw_6.tgz |

| ds112: |
| ... |
DataLad data distribution: Data life cycle
DataLad data distribution: Data life cycle

- **Data user**
  - Local workstation
  - Dataset handles

- **Data hosting**
  - HTTP
  - XNAT
  - SSH
  - datasets.datalad.org

- **Data distribution**
  - Git content
  - Content reference
  - DataLad
  - datalad crawl
  - .git/annex
  - .datalad/crawl/
  - README
  - sub001/
  - BOLD/
  - task001_run001/
  - bold.nii.gz

- **Git repository hosting (GitHub)**
  - openfmri/ds000112
  - openfmri/ds000113

- **Server or cloud storage**
  - S3
  - bold.nii.gz
  - ds112:
  - ds113:
  - ds113_raw_6.tgz

- **OpenfMRI**
  - ds112:
  - ds113:
  - ds113:
  - sub001:
  - BOLD:
  - task001_run001:
  - bold.nii.gz
  - ds112:

- **XNAT**
  - ds113:
  - ds112:
DataLad data distribution: Data life cycle

Datasets can be accessed from a variety of locations and formats, including:

- **Data hosting**
  - OpenfMRI
    - ds112:
    - ds113:
      - ds113_raw_6.tgz
  - XNAT
    - ds113:
      - sub001:
        - BOLD:
          - task001_run001:
            - bold.nii.gz
      - ...

- **Server or cloud storage**
  - S3
    - openfmri
    - ds000112
    - ...

- **Data distribution**
  - datasets.datalad.org
    - openfmri/ds000112
      - .git/annex
      - .datalad/crawl/
      - README
      - sub001/
        - BOLD/
          - task001_run001/
            - bold.nii.gz

- **Git repository hosting (GitHub)**
  - SSH
    - HTTP
  - datasets.datalad.org
    - openfmri/ds000112
      - .git/annex
      - .datalad/crawl/
      - README
      - sub001/
        - BOLD/
          - task001_run001/
            - bold.nii.gz

- **Content reference**
  - DataLad
  - datalad create
  - datalad crawl

- **Data user**
  - Local workstation

Dataset handles

Content reference

Data content

Git content

DataLad

Diagram keys:
- Dataset handles
- Content reference
- Data content
- Git content
- DataLad
DataLad data distribution: Data life cycle

Data producer

Data user

Local workstation

Server or cloud storage

Property of University

Dataset handles

Content reference

Data content

Git content

DataLad

Dataset handles

Content reference

DataLad

Data producer

Local workstation

Server or cloud storage

Property of University

Dataset handles

Content reference

Data content

Git content

DataLad

Data producer

Local workstation

Server or cloud storage

Property of University

Dataset handles

Content reference

Data content

Git content

DataLad

Data producer

Local workstation

Server or cloud storage

Property of University

Dataset handles

Content reference

Data content

Git content

DataLad

Data producer

Local workstation

Server or cloud storage

Property of University

Dataset handles

Content reference

Data content

Git content

DataLad

Data producer

Local workstation

Server or cloud storage

Property of University

Dataset handles

Content reference

Data content

Git content

DataLad

Data producer

Local workstation

Server or cloud storage

Property of University

Dataset handles

Content reference

Data content

Git content

DataLad

Data producer

Local workstation

Server or cloud storage

Property of University

Dataset handles

Content reference

Data content

Git content

DataLad

Data producer

Local workstation

Server or cloud storage

Property of University

Dataset handles

Content reference

Data content

Git content

DataLad

Data producer

Local workstation

Server or cloud storage

Property of University

Dataset handles

Content reference

Data content

Git content

DataLad

Data producer

Local workstation

Server or cloud storage

Property of University

Dataset handles

Content reference

Data content

Git content

DataLad
DataLad data distribution: Data life cycle

[Diagram showing data distribution and handling processes]
DataLad data distribution: Data life cycle

- **Data producer**
- **Data user**
- **Server or cloud storage**
- **Git repository hosting (GitHub)**

**Data hosting**
- SSH
- HTTP
- **datasets.datalad.org**
- **github.com/user1**

**Data distribution**
- **S3openfmri**
- **datasets.datalad.org**
- **openfmri/ds000112**
- **studyX**

**Data content**
- **openfmri/ds000113**
- **ds112**
- **ds113**
- **ds113_raw_6.tgz**
- **sub001**
- **BOLD**
- **task001_run001**
- **bold.nii.gz**
- **bold_filt.nii.gz**

**Git content**
- **.git/annex**
- **.gitmodules**
- **README**

**Content reference**
- **$HOME**
- **localworkstation**
- **labserver**
- **XNAT**
- **ds113_raw_6.tgz**
- **ds113**
- **ds112**

**DataLad**
- Dataset handles
- Content reference
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

Follows: 2.0.0

Added files from extracted archives

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

dataset_description.json
participants.tsv
sub-01/anat/sub-01_T1w.nii.gz
sub-01/anat/sub-01_inplaneT2.nii.gz
sub-01/fun/sub-01_task-balloonanalogisktask_run-01_bold.nii.gz
DataLad WiP

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):

```python
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 runs
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:

```
$ hopa:/tmp/PyMVPA[master]
> datalad install -s http://data.pymvpa.org/datasets/tutorial_data /tmp/tutorial_data
Thu 23 Jun 2016 12:39:13 PM CEST:
[INFO   ] Installing /tmp/tutorial_data (install.py:353)
1 installed item is available at
<Dataset path=/tmp/tutorial_data>
```

```
25329.................................: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/hyeralignment.py
Loading data...
/tmp/tutorial_data/.git 100%[========================================] 15.04M --.KB/s in 0.02s
 10 subjects
Per-subject dataset: 55 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 +/-0.053
  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 (I am @yarikoptic)
DataLad's testing

Add more commits by pushing to the `nf-repo-slimming-down` branch on yarikoptic/datalad.

- All is well — 9 successful checks
  - datalad-pr-virtualbox-dl-win7-64 — DEV build done.
  - datalad-pr-docker-dl-nd80 — DEV build done.
  - datalad-pr-docker-dl-nd14_10 — DEV build done.
  - datalad-pr-docker-dl-nd70 — DEV build done.
  - datalad-pr-docker-dl-nd14_04 — DEV build done.
  - datalad-pr-docker-dl-nd90 — DEV build done.
  - continuous-integration/travis-ci/pr — The Travis CI build passed
  - coverage/coveralls — Coverage increased (+0.18%) to 83.88%
  - datalad-pr-dl-osx-64 — DEV build done.

This pull request can be automatically merged.
You can also merge branches on the command line.