Table of contents

Managing your research data: From collection to publication and beyond

Faculty Affairs and Mentoring Programs Present the Postdoctoral Seminar

Presenter: **Pranav K. Mishra, MD**
Post-Doctoral Research Fellow
Departments of Surgery and Orthopedic Surgery
Rush University

Event Details
May 29, 2024 (Wednesday)
3:00 pm – 5:00 pm
AAC 550A and Zoom

Learning Objectives

- What makes research data storage efficient and compliant with the new NIH guidelines
- Building a hierarchy to separate raw, analysis, and publication-level research data
- Automating steps between the "mundane tasks" and "complex analysis" to save time and improve reproducibility
- How to access and utilize Rush's Microsoft 365 cloud storage for 25 TB of storage, per project, for free
- 2nd-hour Interactive Session - bring your laptop if you'd like to showcase your setup or get help improving your project!

Repository structure

- Code for data processing and analysis is located in `code/`.
- Data structure:

- Raw data - unmodified / unanalyzed data: `/data/raw`
- Analyzed data - processed and analyzed data, generated from the raw data: `/data/analysis`
- Research output - figures, tables, text, etc. suitable or submitted for publication (abstract, presentation, journal article)

Datalad archival storage

The research data is being stored on the pminformatics server on the DATA-3 volume. An RIA storage has been created at this location:

```
datalad create-sibling-ria --new-store-ok -s datalad-ria --existing reconfigure ria+file:///
```

```
datalad siblings add -s origin --url git@github.com:pranavmishra90/courses-by-mishra-researc
```

Instructions

Creating the python environment with conda

```
conda env create -f code/python/environment.yml
```

Running the code

You can open each jupyter notebook file individually and see how the code runs. Alternatively, the entire analysis can be performed automatically using `papermill`. The research notebook is generated using `quarto`.

```
conda activate researchdata
```

```
datalad run --dry-run basic -i data/raw/cms -o data/analysis/bariatric/runs -o notebook/ --e
```

```
# Bariatric dataset (requires raw data not available on GitHub)
# datalad run --dry-run basic -i data/raw/mbsa-qip -o data/analysis/bariatric/runs -o notebo
```

About this dataset

General information

This is a DataLad dataset (id: ebf9a648-e396-4741-a38f-cc40cbf18823).

DataLad datasets and how to use them

This repository is a DataLad dataset. It provides fine-grained data access down to the level of individual files, and allows for tracking future updates. In order to use this repository for data retrieval, DataLad is required. It is a free and open source command line tool, available for all major operating systems, and builds up on Git and git-annex to allow sharing, synchronizing, and version controlling collections of large files.

More information on how to install DataLad and how to install it can be found in the DataLad Handbook.

Get the dataset

A DataLad dataset can be `cloned` by running

```
datalad clone https://github.com/pranavmishra90/courses-by-mishra-research-data-management
```

Once a dataset is cloned, it is a light-weight directory on your local machine. At this point, it contains only small metadata and information on the identity of the files in the dataset, but not actual *content* of the (sometimes large) data files.

Retrieve dataset content

After cloning a dataset, you can retrieve file contents by running

```
datalad get <path/to/directory/or/file>
```

This command will trigger a download of the files, directories, or subdatasets you have specified.

DataLad datasets can contain other datasets, so called *subdatasets*. If you clone the top-level dataset, subdatasets do not yet contain metadata and information on the identity of files, but appear to be empty directories. In order to retrieve file availability metadata in subdatasets, run

```
datalad get -n <path/to/subdataset>
```

Afterwards, you can browse the retrieved metadata to find out about subdataset contents, and retrieve individual files with `datalad get`. If you use `datalad get <path/to/subdataset>`, all contents of the subdataset will be downloaded at once.

Stay up-to-date

DataLad datasets can be updated. The command `datalad update` will *fetch* updates and store them on a different branch (by default `remotes/origin/master`). Running

```
datalad update --merge
```

will *pull* available updates and integrate them in one go.

Find out what has been done

DataLad datasets contain their history in the `git log`. By running `git log` (or a tool that displays Git history) in the dataset or on specific files, you can find out what has been done to the dataset or to individual files by whom, and when.