MRS Hackathon 2023

Collaborating with yourself: Best practices for managing data, code, and results for MRS analysis projects

Chris Davies-Jenkins[1,2],

1 The Russell H. Morgan Department of Radiology and Radiological Science, Johns Hopkins University School of Medicine, Baltimore, MD, USA
2 F.M. Kirby Research Center for Functional Brain Imaging, Kennedy Krieger Institute, Baltimore, MD, USA

Collaborating with yourself: Best practices for managing data, coils, and results for MRS analysis projects

- Partially inspired by the Brianhack school lecture given by Elizabeth Dupre (QLSC-612, 2020).

- I wanted to present something that was MRS-specific, hacky, and (hopefully) also relevant to all levels.

- Best practices for getting your data from the scanner to publication, without the messy basement…
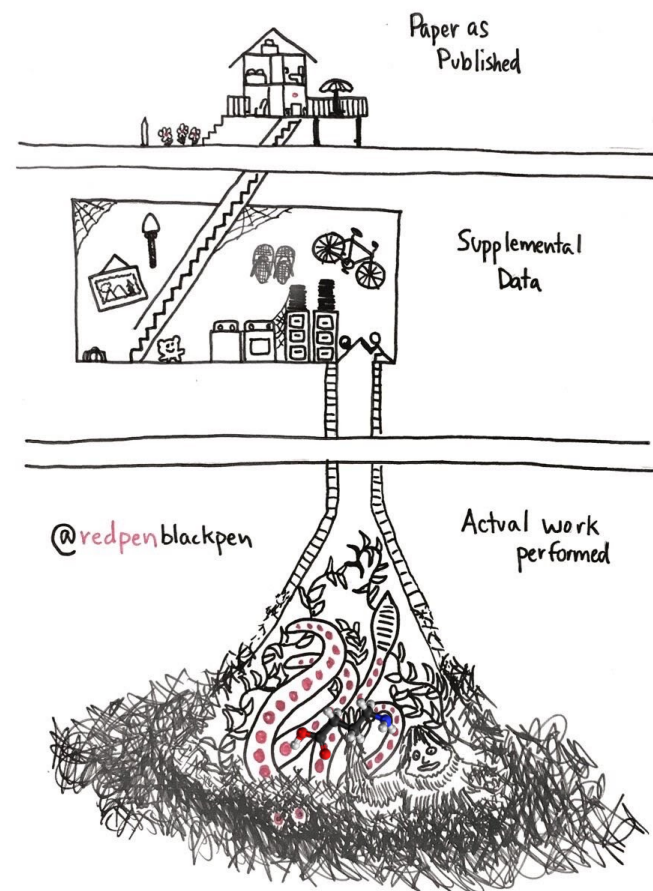
Collaborating with yourself: Best practices for managing data, coils, and results for MRS analysis projects

1. **Data formatting**: File formats, file names, and folder structures for MRS data.

2. **Project formatting**: How to store and manage several projects.

3. **Example workflow:** Applying principles above

4. **Group suggestions**: Any suggestions
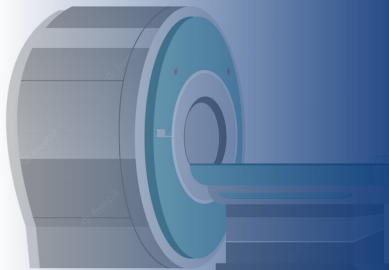
- MR Vendors provide a variety of formats for data export (dicom, raw: "k-space" formats)

- Raw data: Saving individual transients and coil elements is preferable

- Several secondary data formats also exist e.g. LCModel RAW and JMRUI .txt

- MRS fitting software does not necessarily handle every format and interpreting new sequences/scanners/software is a challenge

- NIfTI (Neuroimaging Informatics Technology Initiative) is a standardized data format used across neuroimaging.

- NIfTI MRS is a recent effort to extend NIfTI to MRS

Received: 21 June 2022 | Revised: 26 July 2022 | Accepted: 29 July 2022 | Published on: 11 September 2022

DOI: 10.1002/mrm.29418

RESEARCH ARTICLE

Magnetic Resonance in Medicine

NIfTI-MRS: A standard data format for magnetic resonance spectroscopy

William T. Clarke[1] | Tiffany K. Bell[2,3,4] | Uzay E. Emir[5,6] | Mark Mikkelsen[7] |
Georg Oeltzschner[8,9] | Amirmohammad Shamaei[10,11] | Brian J. Soher[12] |
Martin Wilson[13]

- Will Clarke's spec2nii tool can convert most vendor-specific data formats to NIfTI.

Collaborating with yourself: Best practices for managing data, coils, and results for MRS analysis projects

- BIDS: "Brain Imaging Data Structure"
  - Standardized guidelines for data filenames, directory structures, and outputs

- Benefits of using BIDS:
  - Improved data sharing
  - Easier reproduction of results
  - Easier software development
  - Online repositories require BIDS (e.g. OpenNeuro)

www.nature.com/scientificdata

SCIENTIFIC DATA

OPEN

SUBJECT CATEGORIES
» Data publication and archiving
» Research data

The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments

Received: 18 December 2015
Accepted: 19 May 2016
Published: 21 June 2016

Krzysztof J. Gorgolewski[1], Tibor Auer[2], Vince D. Calhoun[3,4], R. Cameron Craddock[5,6], Samir Das[7], Eugene P. Duff[8], Guillaume Flandin[9], Satrajit S. Ghosh[10,11], Tristan Glatard[7,12], Yaroslav O. Halchenko[13], Daniel A. Handwerker[14], Michael Hanke[15,16], David Keator[17], Xiangrui Li[18], Zachary Michael[19], Camille Maumet[20], B. Nolan Nichols[21,22], Thomas E. Nichols[20,23], John Pellman[6], Jean-Baptiste Poline[24], Ariel Rokem[25], Gunnar Schaefer[3,26], Vanessa Sochat[27], William Triplett[1], Jessica A. Turner[3,28], Gaël Varoquaux[29] & Russell A. Poldrack[1]

- Directory structure hierarchy
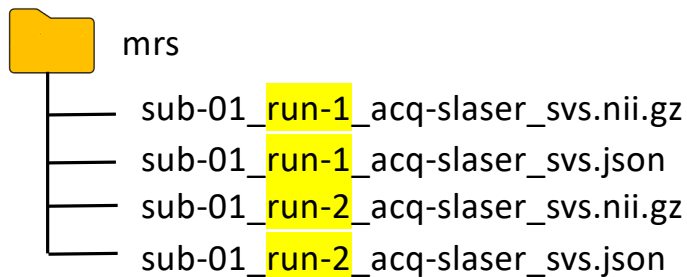 - Subject
  - Session (optional)
  - Modality

- Filenames are underscore-separated sets of:
  - N Key-value pairs
  - A prefix
  - A suffix

- Data formats standardized where possible (NIfTI)

- Data provenance maintained in JSON "sidecars"

- Top-level directory
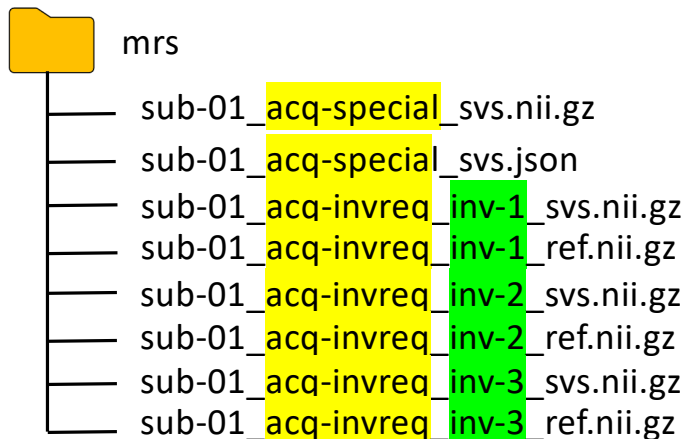 - sourcedata->rawdata->derivatives
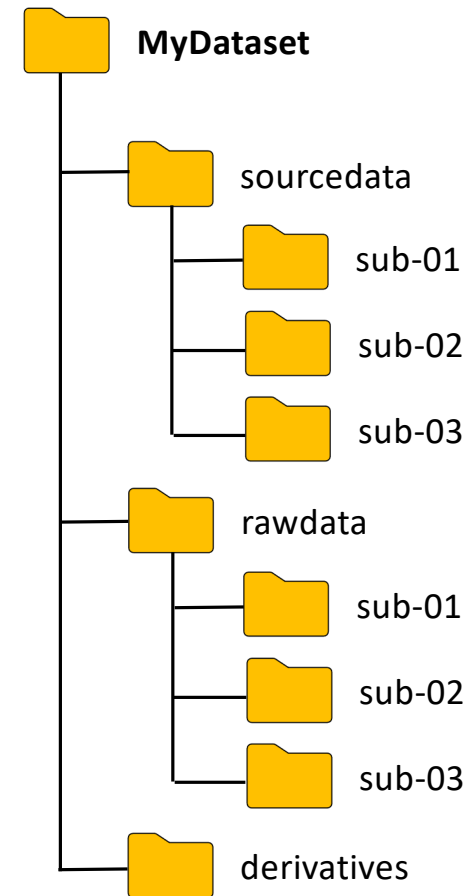
MyDataset
- sourcedata
  - sub-01
  - sub-02
  - sub-03
- rawdata
  - sub-01
  - sub-02
  - sub-03
- derivatives

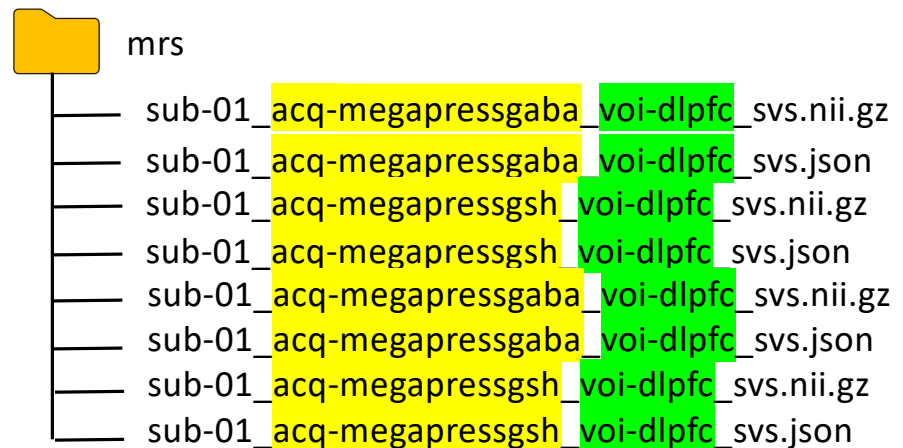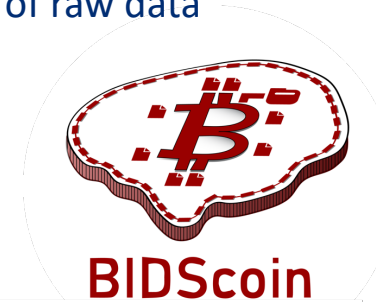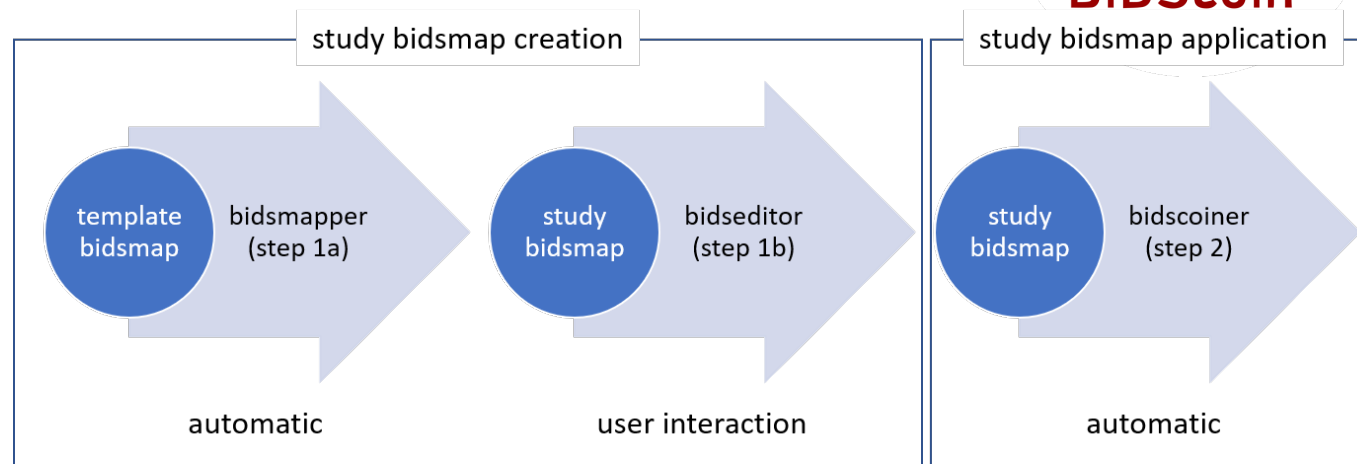**Collaborating with yourself: Best practices for managing data, coils, and results for MRS analysis projects**

sLASER reproducibility:

📁 mrs

— sub-01_run-1_acq-slaser_svs.nii.gz
— sub-01_run-1_acq-slaser_svs.json
— sub-01_run-2_acq-slaser_svs.nii.gz
— sub-01_run-2_acq-slaser_svs.json

SPECIAL with inversion recovery for MM:

📁 mrs

— sub-01_acq-special_svs.nii.gz
— sub-01_acq-special_svs.json
— sub-01_acq-invreq_inv-1_svs.nii.gz
— sub-01_acq-invreq_inv-1_ref.nii.gz
— sub-01_acq-invreq_inv-2_svs.nii.gz
— sub-01_acq-invreq_inv-2_ref.nii.gz
— sub-01_acq-invreq_inv-3_svs.nii.gz
— sub-01_acq-invreq_inv-3_ref.nii.gz

MEGA-PRESS with two VOIs:

📁 mrs

— sub-01_acq-megapressgaba_voi-dlpfc_svs.nii.gz
— sub-01_acq-megapressgaba_voi-dlpfc_svs.json
— sub-01_acq-megapressgsh_voi-dlpfc_svs.nii.gz
— sub-01_acq-megapressgsh_voi-dlpfc_svs.json
— sub-01_acq-megapressgaba_voi-dlpfc_svs.nii.gz
— sub-01_acq-megapressgaba_voi-dlpfc_svs.json
— sub-01_acq-megapressgsh_voi-dlpfc_svs.nii.gz
— sub-01_acq-megapressgsh_voi-dlpfc_svs.json

BIDS-ification toolboxes can be used to automate the conversion of raw data to BIDS-compliant structures:

- dcm2bids
- heudiconv
- bidskit
- bidscoin

BIDScoin is particularly useful due to spec2nii integration

Automated BIDS converters typically need some effort to set up but pay dividends

There are also tools to validate BIDS structures to ensure your data are BIDS compliant (e.g. BIDS-validator)
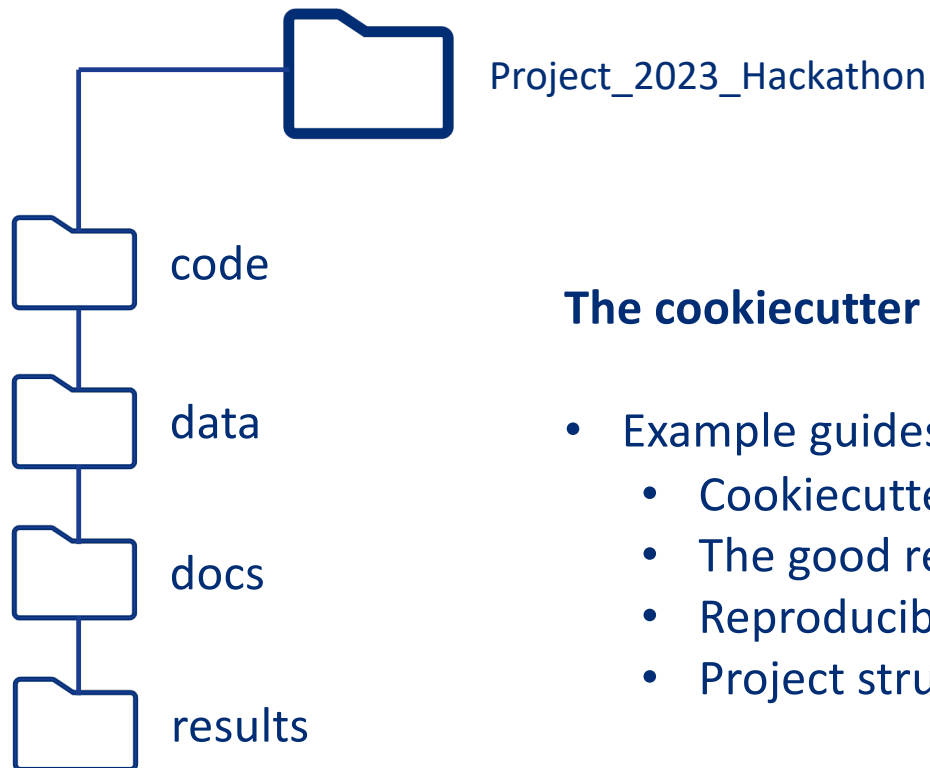
The data are organized but now comes the clutter:
- Multiple data iterations
- Odd bits of Code
- Analysis: tables, plots, and secondary data
- Machine learning model iterations
- Presentations
- Manuscript(s)

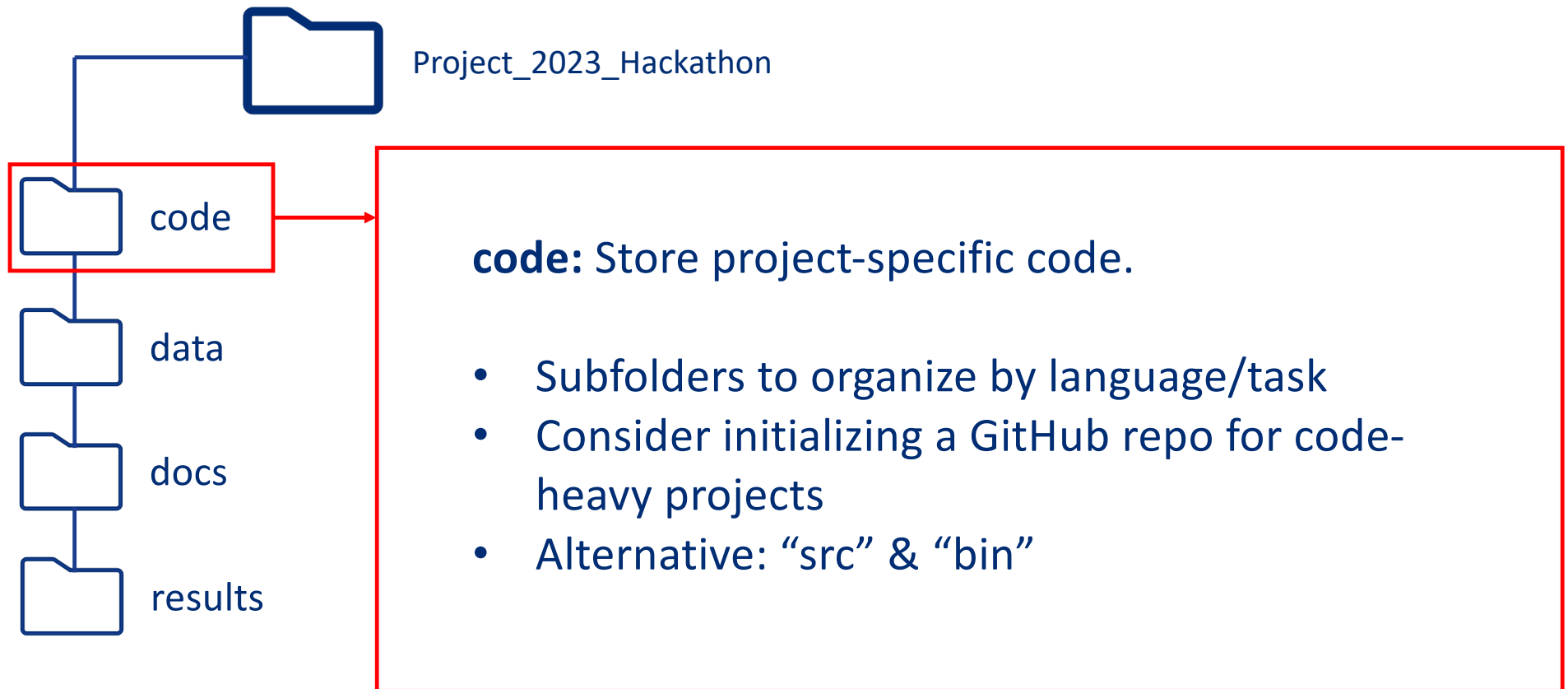**Need a consistent organizational structure to cope with this!**

Neat & tidy BIDS data

This one does not spark joy.

Project_2023_Hackathon

code

data

docs

results

The cookiecutter project folder

- Example guides:
  - Cookiecutter-data-science
  - The good research handbook, *Patrick J. Mineault*
  - Reproducible Data Science, *Simona Picardi*
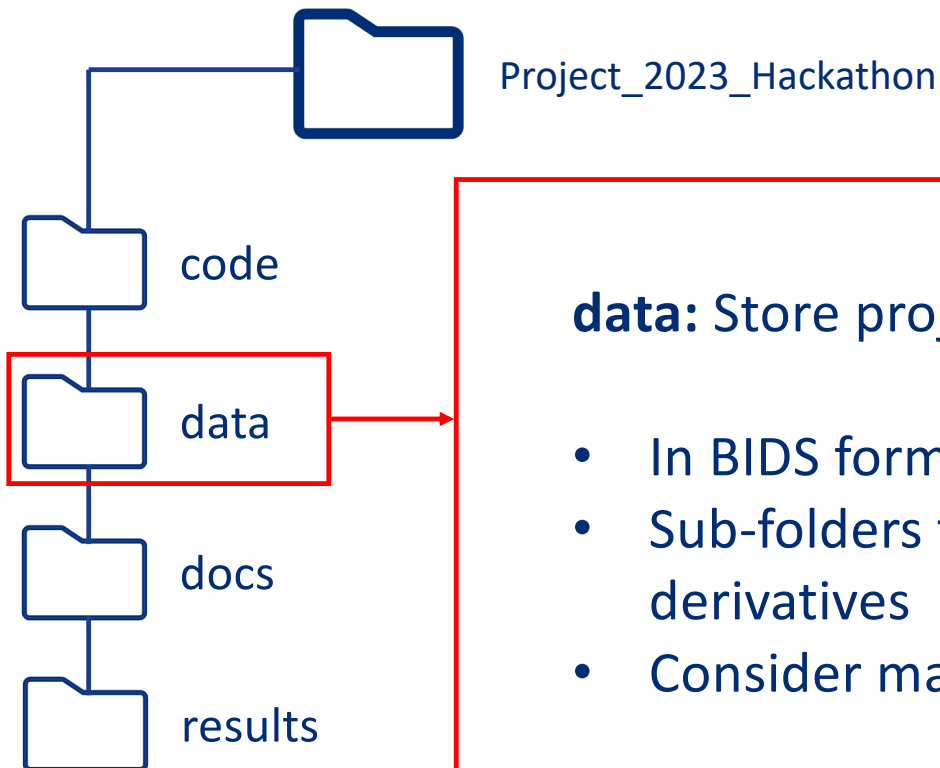 - Project structure, *Danielle Navarro*

- "Flexible but consistent!"

Project_2023_Hackathon

code

data

docs

results

code: Store project-specific code.

- Subfolders to organize by language/task
- Consider initializing a GitHub repo for code-heavy projects
- Alternative: "src" & "bin"

Project_2023_Hackathon
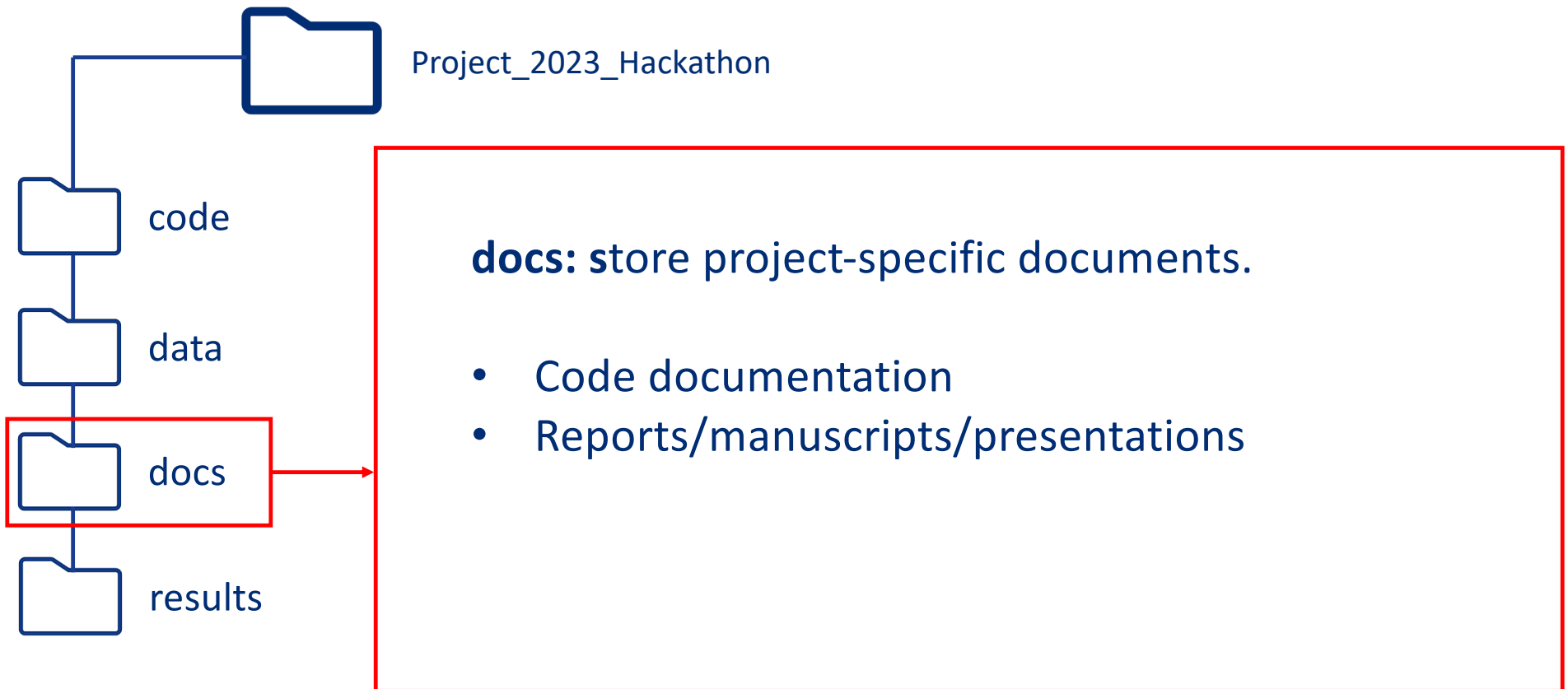
code

data

docs

results

data: Store project-specific data

- In BIDS format!!
- Sub-folders for sourcedata, rawdata, and derivatives
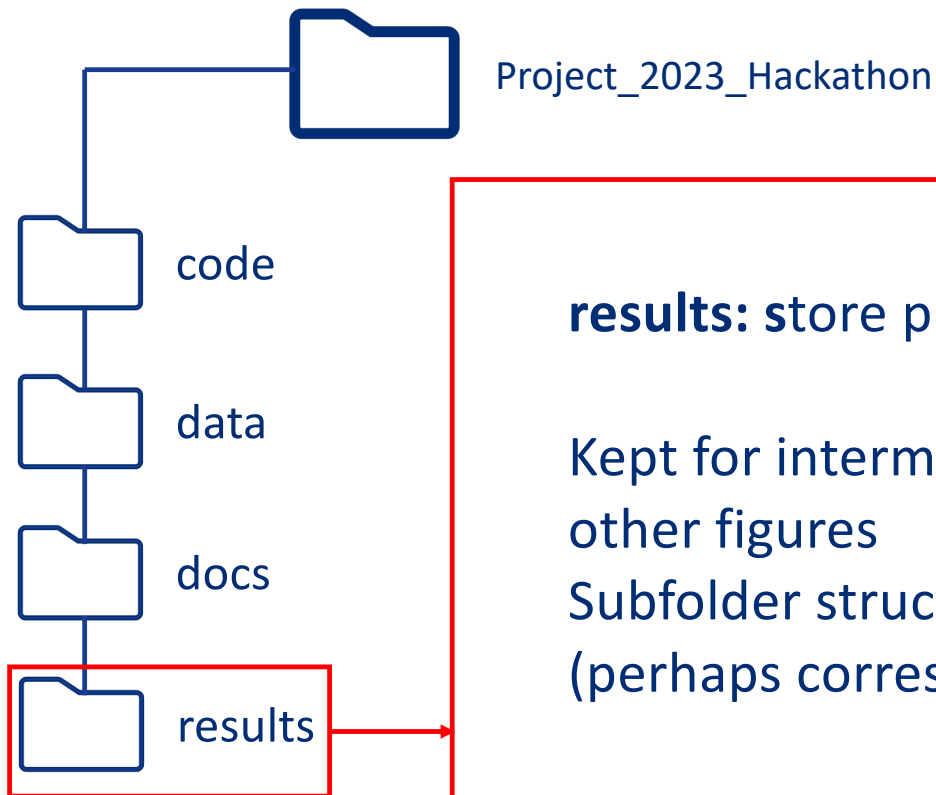- Consider making source/raw read-only

Project_2023_Hackathon

code

data

docs

results

docs: store project-specific documents.

- Code documentation
- Reports/manuscripts/presentations

Project_2023_Hackathon

code

data

docs

results

results: store project-specific results.

Kept for intermediate results e.g. Tables, plots, other figures
Subfolder structure should be designated early (perhaps corresponding to code)
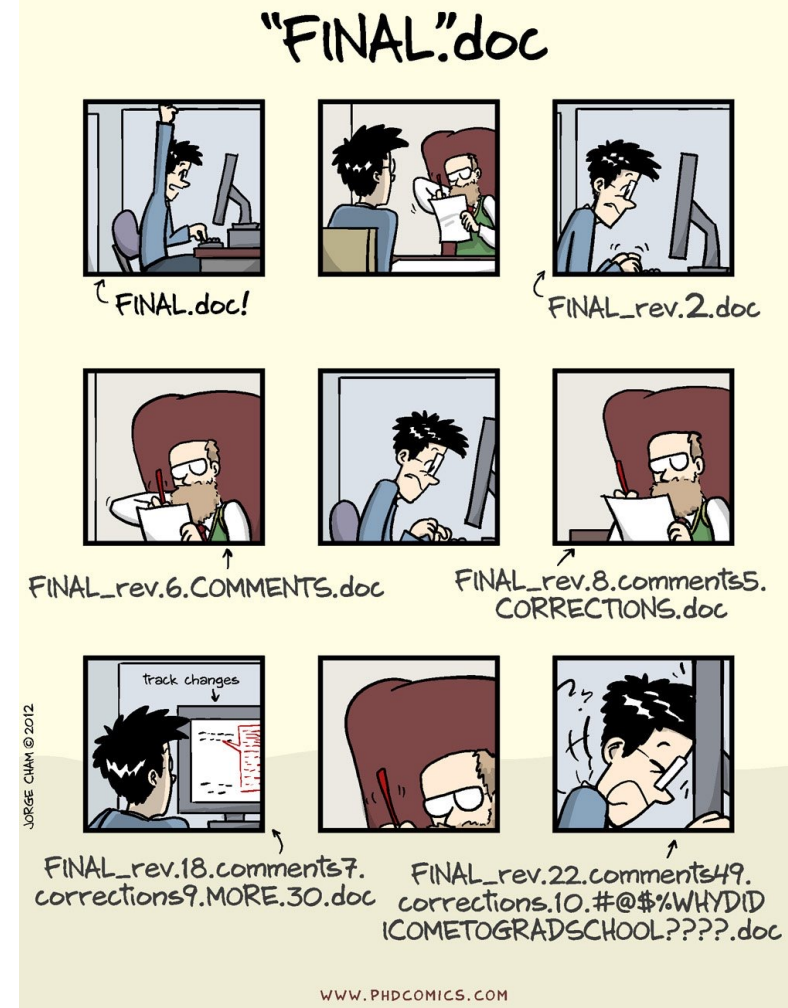
Guidelines to remember:

1) Machine-readable
- **Avoid:** empty spaces & special characters (`` ` ``, @, ^, *, #)
- **Use:** Case sensitivity, '-', and '_'

2) Human-readable
- **Avoid:** Long and arbitrary names
- **Use:** CamelCase or snake_case for readability

3) Utilize default ordering
- **Avoid:** Numerical suffixes, jumbled naming conventions
- **Use:** Numerical prefix, ISO 8601: YYYY-MM-DD

"FINAL".doc

FINAL.doc!

FINAL_rev.2.doc

FINAL_rev.6.COMMENTS.doc

FINAL_rev.8.comments5.CORRECTIONS.doc

track changes

FINAL_rev.18.comments7.corrections9.MORE.30.doc

FINAL_rev.22.comments49.corrections.10.#@$%WHYDIDICOMETOGRADSCHOOL????.doc

JORGE CHAM © 2012

WWW.PHDCOMICS.COM

Collaborating with yourself: Best practices for managing data, coils, and results for MRS analysis projects

Project_2023_Hackathon

code

data

docs
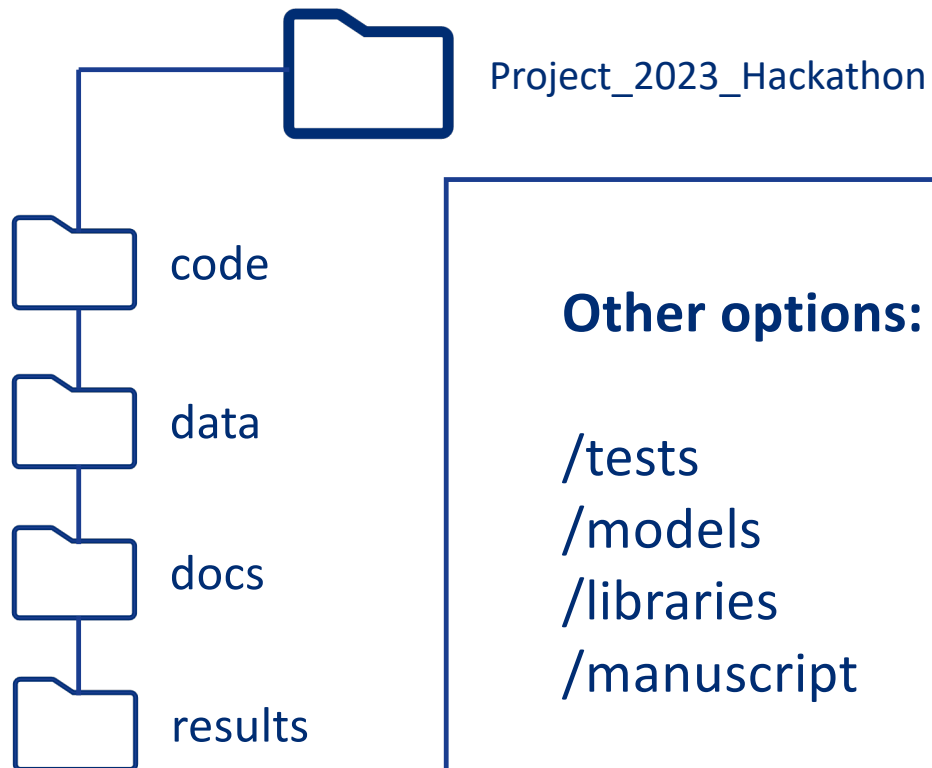
results

Other options:

/tests
/models
/libraries
/manuscript

Collaborating with yourself: Best practices for managing data, coils, and results for MRS analysis projects

Project_2023_Hackathon

code

data

docs

results

README.txt!!!!
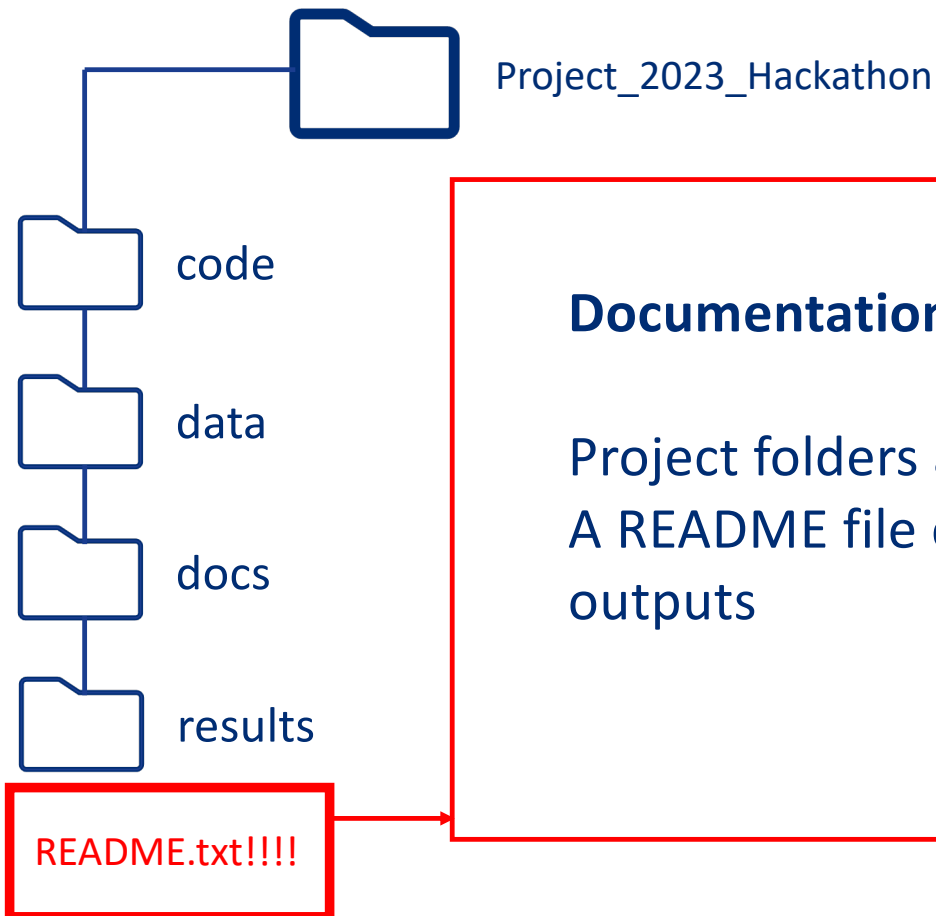
Documentation

Project folders are not self-explanatory
A README file can document the logic of data and outputs

Lab books:

- Parallel to README
- For documenting experiments, analyses etc.
- Track goals and intermediate results
- Should mirror your workflow

Electronic lab books:

- Easier to maintain in tandem with project folders
- Searchable—easier to find specific details
- Backups avoid disasters

Examples

- OneNote
- Evernote
- Confluence
- Benchling
- eLabNext
- LabArchives

Obsidian
Markdown, custom Wiki

Electronic lab books have some advantages

Individual notes:
- Customized templates/plugins
- Links between notes

Project-specific notes:
- Links to relevant locations (data/code)
- Track tasks/goals per project
- Javascript cookiecutter generation
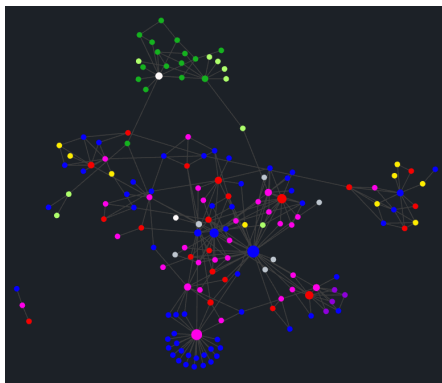
< Mon 22-05-2023 |Tue 23-05-2023 | Wed 24-05-2023 >

📝 Planning:

✅ Tasks:

Scheduled today:

Due today:
- ☐ Create hackathon slides 📅🔗✏️

Due this week:
- ☐ Analyze some MRS data 📅🔗✏️

💼 Happenings:
- worked on Proj-23x-Test

Parent project: Proj-23x-Hackathon

Summary of this phase:
Running the dummy analysis for my slides.

Diary:

2023-05-22 Mon

- Created a dummy dataset located here ↗.
- BIDS-ified the data using this ↗ BIDSmap.
- Ran this ↗ Osprey job file.

2023-05-23 Tue

- Ran the stats in R using this ↗ script

Collaborating with yourself: Best practices for managing data, coils, and results for MRS analysis projects

Applying it in a workflow:

- BIDS conversion — BIDScoin
- Project Folder — Cookiecutter
- Lab book — Obsidian template
- MRS analysis — Osprey

Example study:

- 2 subjects, scanned multiple times
- Short-TE PRESS and MPRAGE

Step 1: Initiate project

- Create new entry in lab book:
  - Give it a title (template)
  - Give it a short description

Obsidian template then:

- Generates a project note
- Creates cookiecutter folder structure
- Creates link to the folder structure
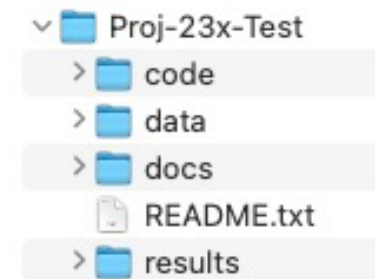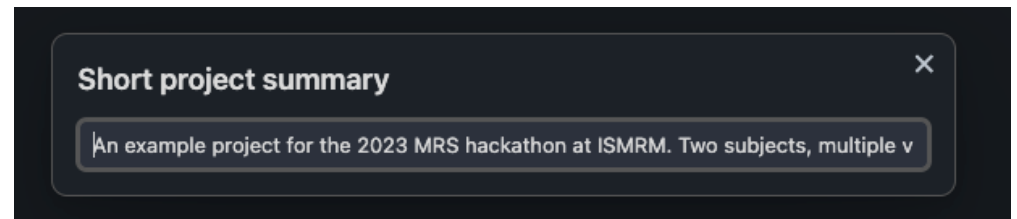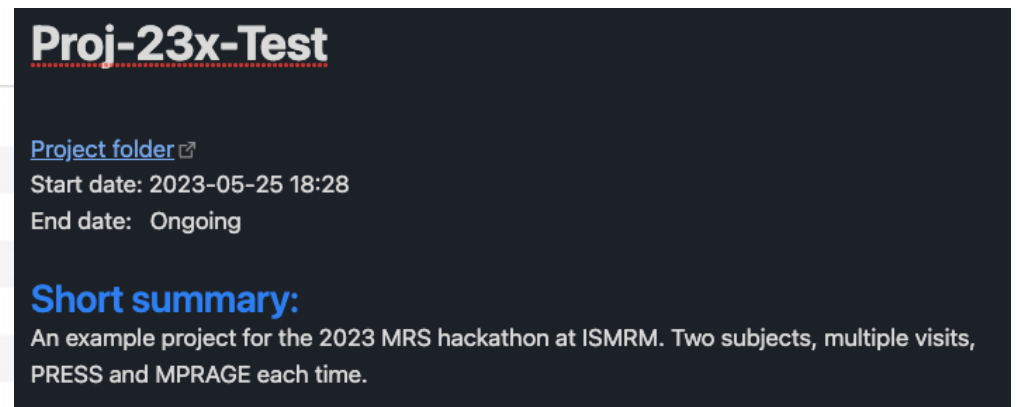- Populates readme file

Collaborating with yourself: Best practices for managing data, coils, and results for MRS analysis projects

Step 2: Arrange the data:

- Pull the data into standardized input format*
- Generate study BIDSmap
- Run BIDScoiner as/when needed
- [make read-only]

Step 3: Validate and document

- Validate the structure by inspection or BIDS validation tool
- Update README, participant log, and lab book

```
>>
Bidsmapper  sourcedata  sourcedata
Bidseditor  rawdata

bidscoiner  rawdata

chmod a=rX rawdata
```

> code
> data
> > code
> > rawdata
>   > code
>   dataset_description.json
>   participants.json
>   participants.tsv
>   README
>   > sub-01
>   > sub-02
> > sourcedata
>   > sub-01
>   > sub-02
> docs
  README.txt
> results

2023-05-23 Tue
- 2 subjects and 2 sessions were converted to BIDS (2x PRESS, 1x mprage)
  - Had to regenerate BIDSmap here
  - Validated
- Updated participants log here

*https://bidscoin.readthedocs.io/en/stable/preparation.html

Step 4: set up the analysis
Ensure reproducibility
- Download a specific release version of Osprey
- Osprey job file in the BIDS code directory
  - Delineate files programmatically
  - Define output directory as BIDS derivative
- Map out analysis steps in wrapper function

Aside:
- Relative paths:
  data/rawdata/
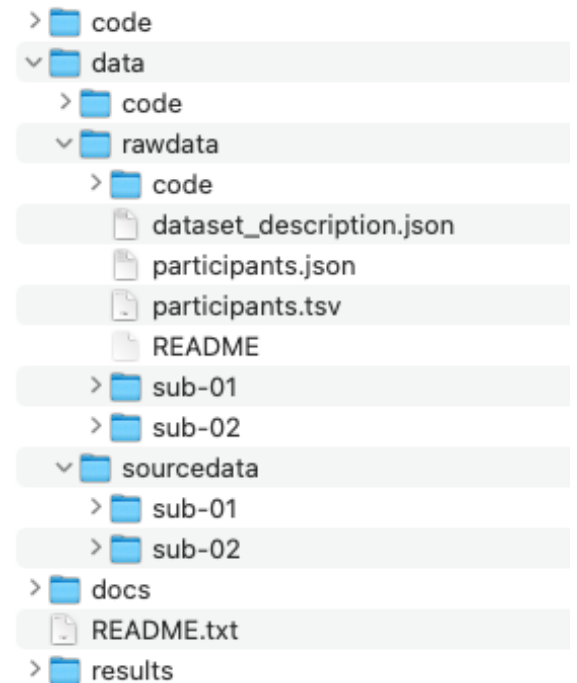- Absolute paths:
  /Users/Local/Chris/BoringWorkStuff/ThatDumbProject4Hackathon/data/rawdata

```matlab
% Script for analysis of Hackathon example data

%% Load & process the data
MRSCont = OspreyJob('data/code/MRSAnalysis_Osprey-jobfile');
MRSCont = OspreyLoad(MRSCont);
MRSCont = OspreyProcess(MRSCont);
%% Coregister and segment T1
MRSCont = OspreyCoreg(MRSCont);
MRSCont = OspreySeg(MRSCont);
%% Fit and quantify
MRSCont = OspreyFit(MRSCont);
MRSCont = OspreyQuantify(MRSCont);
%% Overview
MRSCont = OspreyOverview(MRSCont);
```

Step 5: Document
- Update README/lab book (Osprey version, jobfile, and outputs)

2023-05-24 Wed
- Pulled Osprey v2.5.0 here ⤤
- Jobfile is here ⤤. Baseline parameters and basis set from here ⤤
- Ran the analysis wrapper script here ⤤

Osprey

- Creates BIDS-compliant derivatives
  - Voxel masks
  - Segmentation results
  - Metabolite-measure tables
  - Full Matlab struct
  - MRSinMRS

- Rawdata and sourcedata left untouched

- Osprey derivative files used for secondary analysis
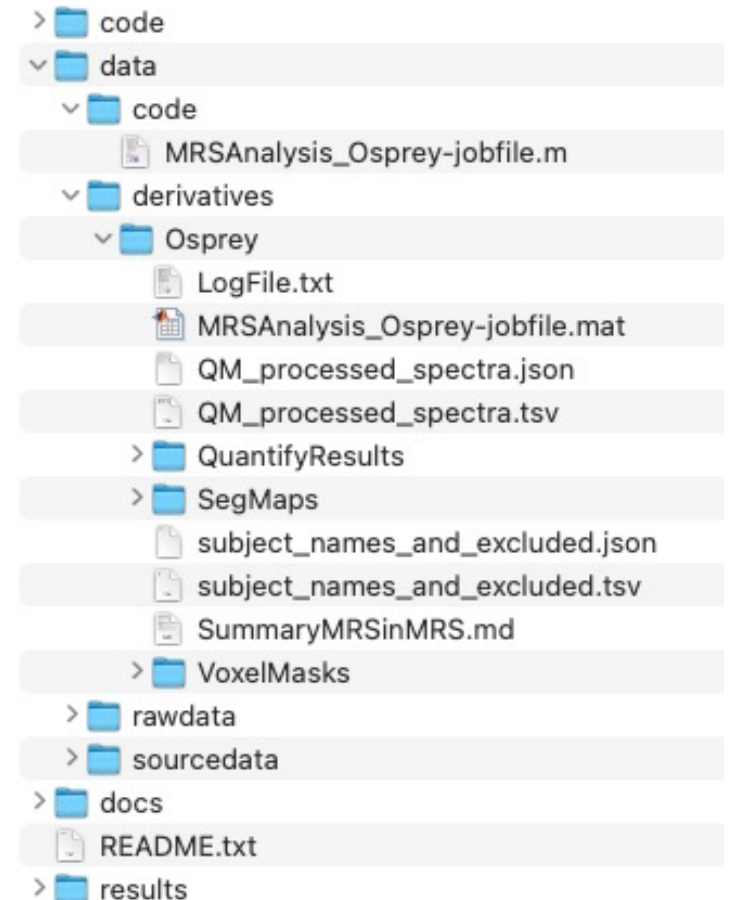
```
> 📁 code
∨ 📁 data
   ∨ 📁 code
      📄 MRSAnalysis_Osprey-jobfile.m
   ∨ 📁 derivatives
      ∨ 📁 Osprey
         📄 LogFile.txt
         📄 MRSAnalysis_Osprey-jobfile.mat
         📄 QM_processed_spectra.json
         📄 QM_processed_spectra.tsv
         > 📁 QuantifyResults
         > 📁 SegMaps
         📄 subject_names_and_excluded.json
         📄 subject_names_and_excluded.tsv
         📄 SummaryMRSinMRS.md
         > 📁 VoxelMasks
   > 📁 rawdata
   > 📁 sourcedata
> 📁 docs
📄 README.txt
> 📁 results
```

Step 7: Generate some figures
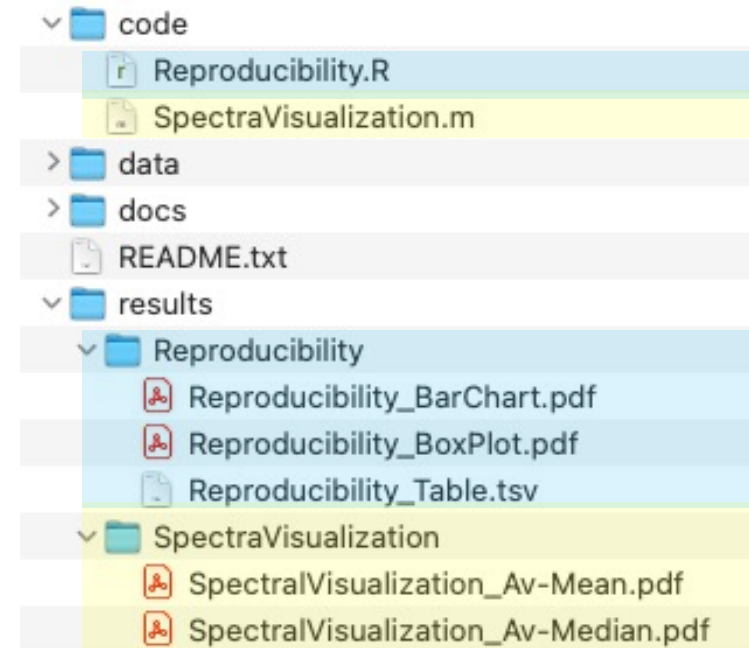- Want to visualize the spectra and analyze reproducibility
- Structure:
  - Allign code and results
  - CammelCase Filenames
  - Sub-directories for multiple files
- Consider a GitHub repository

Sourcedata -> rawdata -> derivatives -> results -> docs

Step 8: Document… Again
- Update README/lab book
- If the order of analysis is important, document it


```
∨ 📁 code
    r  Reproducibility.R
    .  SpectraVisualization.m
> 📁 data
> 📁 docs
   📄 README.txt
∨ 📁 results
   ∨ 📁 Reproducibility
         Reproducibility_BarChart.pdf
         Reproducibility_BoxPlot.pdf
      📄 Reproducibility_Table.tsv
   ∨ 📁 SpectraVisualization
         SpectralVisualization_Av-Mean.pdf
         SpectralVisualization_Av-Median.pdf
```

2023-05-25 Thu
- Script to visualize the spectra is here ⬀
  - plotted both the mean and median. Results here ⬀
- The reproducibility script generates ICC and plots group differences here ⬀
  - Bar and box plots generated with full stats table here ⬀

Step 9: Report results
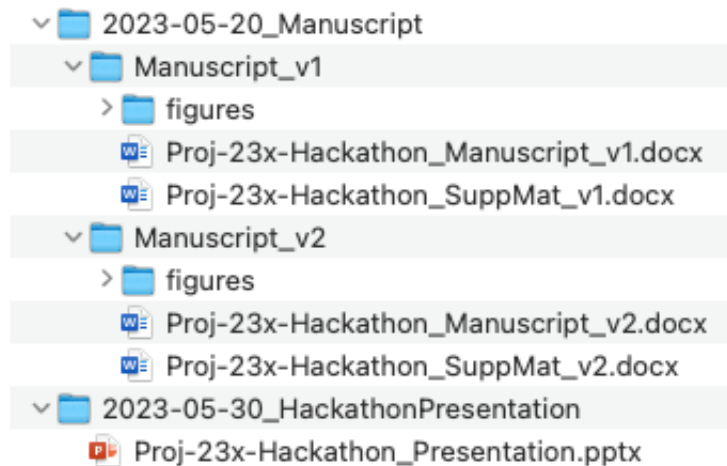- Subdirectories of docs use ISO 8601 subdirectories
- Document filenames defined with the intent to share
- Version numbers in directories and documents to track distinct versions

Step 8: Document… Some more
- Update README/lab book

Collaborating with yourself: Best practices for managing data, coils, and results for MRS analysis projects

Summary:
- Standardize data with NIfTI & BIDS
- Standardize projects with a cookie cutter template
- Choose filenames & sub-directories with forethought
- Document everything with a lab book

Benefits:
- The project note aids in retracing steps
- Data/project sharing requires less work
- Reproducible

Drawback:
- Time investment?

Diary:

2023-05-23 Tue
- 2 subjects and 2 sessions were converted to BIDS (2x PRESS, 1x mprage)
  - Had to regenerate BIDSmap here 🗗
  - Validated
- Updated participants log here 🗗

2023-05-24 Wed
- Pulled Osprey v2.5.0 here 🗗
- Jobfile is here 🗗. Baseline parameters and basis set from here 🗗
- Ran the analysis wrapper script here 🗗

2023-05-25 Thu
- Script to visualize the spectra is here 🗗
  - plotted both the mean and median. Results here 🗗
- The reproducibility script generates ICC and plots group differences here 🗗
  - Bar and box plots generated with full stats table here 🗗

2023-05-26 Fri
- Began the manuscript draft here 🗗

2023-05-27 Sat
- Compiled slides for Hackathon presentation here 🗗
- New manuscript version following feedback here 🗗

Collaborating with yourself: Best practices for managing data, coils, and results for MRS analysis projects

Brain hack talk
Elizabeth Dupre 2020: https://school.brainhackmtl.org/modules/project_management/

BIDS
Spec2nii: https://github.com/wtclarke/spec2nii
BIDS specification: https://bids-specification.readthedocs.io/
Online BIDS validator: http://bids-standard.github.io/bids-validator/
BIDScoin: https://github.com/Donders-Institute/bidscoin

Cookiecutter project folder specification
Cookiecutter: https://cookiecutter.readthedocs.io/en/stable/README.html
Good research handbook: https://goodresearch.dev/
Reproducible data science: https://ecorepsci.github.io/reproducible-science/
Danielle Navarro talk: https://djnavarro.net/slides-project-structure/

Lab book
Obsidian download: https://obsidian.md/download
Lab book template: https://github.com/CWDAVIESJENKINS/ObsidianLabBookTemplate

MRS analysis
Osprey: https://github.com/schorschinho/osprey