DEPENd Lab Data Management

Nate Hall, Michael Hallquist, UNC-CH Developmental Personality L

Nov 15, 2022

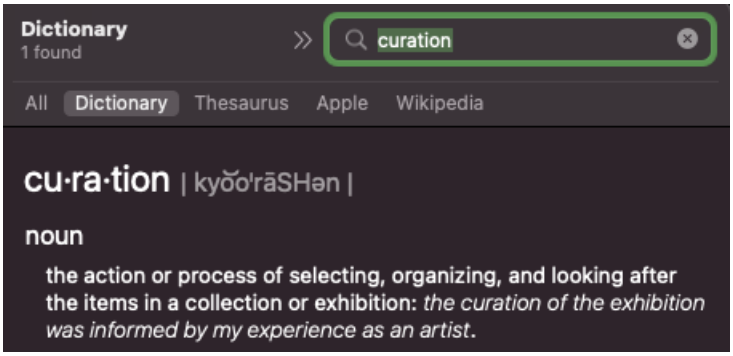
CONTENTS:

This “Data Curation Hub” provides a complete live set of documents for how the UNC DEPENd Lab (PI Dr. Michael N. Hallquist, Ph.D) handles and organizes data and processes relevant to the curation of data for the good of the lab!

CHAPTER

ONE

MISSION STATEMENT



The goal of the data curation team is to set up and implement a centralized set of procedures for the management of incoming data in the DEPENd Lab. At its inception, our goal was to implement a data curation system that provide ongoing checks on the quality of incoming data for the NeuroMAP and Momentum studies. However, we have since decided to try and generalize this data curation system into a more universal set of guidelines that should apply to *any* mid-to-large-scale humans subjects research study.

Ultimately, the Data Curation Hub should serve as a reference for:

1. How data ought to be processed at every stage of its "life"
    1. Data Collection
    2. Data Transfer
    3. Data Validation
    4. Data Preprocessing
    5. Data Quality Control
    6. Data Storage and Backup
2. How to edit this documentation to reflect the most up-to-date version of the data curation system.
3. Where to find data at different stages of processing

This data curation team will also oversee implementing these guidelines and organizes relevant information for reporting data quality in regular study meetings.

CHAPTER

TWO

ONGOING DATA CURATION TASKS

Currently, the data curation guidelines outlined in this hub are under development. If you are interested in a list of tasks that are currently active, check out this Slite document.

CHAPTER

THREE

CONTENTS OF THE DATA CURATION HUB

- Editing this Document
  - If you are new to this documentation and would like to learn more about how to edit this document, please read through the materials in the “Editing this Document” Tab.
- Directory Structure Guidelines
- Storage Plan
- Codebooks
- NDA Submission Guidelines

CHAPTER

FOUR

QUESTIONS

Please direct any questions regarding this Data Curation Hub to Nate Hall (primary), Nidhi Desai, or Michael Hallquist.

4.1 Editing this Document

4.1.1 Getting Started

First and foremost: Please consult the readthedocs documentation (written in readthedocs!) for detailed and up-to-date information on using RTD.

If you are planning on making changes to this documentation, I highly recommend working through the first 20 videos in this youtube series. Some highlights:

- 04 Install Read the Docs Prerequisites shows you how to sign up for RTD, and install important dependencies through the python pip command.
- 21 Goes into git and github if you want to learn

A cheatsheet for restructured text and sphinx syntax can be found here but you can also look at source code for other pages as a guide.

4.1.2 How-to Guide

Editing this document requires a bit of one-time setup but after reading this page my hope is that you can edit this documentation hub with relative ease. These are the only steps required to make changes:

```
## 1. Open up your .rst file in any text editor
open <Tabs/tab_name.rst> # I like to use VScode so I run [code <Tabs/tab_name.rst>] but␣
↪you can use anything.

## 2. Make changes to this file! Save it and make sure if it's a new tab that it is␣
↪listed in index.rst
## when you are done making changes, run:

## 3. Render the html, I created a bash alias <build> that functions as an easier␣
↪shortcut
make html

## 4. Now you can open up the .html file in your browser!
open _build/html/index.html
```

At this point, your local copy is up-to-date and now you'll just need to get your changes on github. I'd highly recommend running step 4 and inspecting the resulting page to make sure your changes render correctly before pushing to github.

```
## 5. Compare local copy to remote (optional, but useful if you have made multiple␣
↪changes)
git status

## 6. Add changes to git "staging area"
git add . # add all changes to staging area
git add <Tabs/tab_name.rst> # add specific files to staging area, to add multiple you␣
↪can separate them with a space

## 7. Commit changes in your staging area with an informative commit message
git commit -m "Updates to tab_name [be specific but concise]"

## 8. Push your changes to remote. You should be able to see your changes on the DEPENd␣
↪lab github!
git push

## 9. That's it, you should now be able to view the most recent build.
```

And some of these "steps" aren't even "steps".. pretty easy right?

One-Time Setup and Dependencies

Additional Resources and helpful tidbits

Here are some additional resources

hashtags need to be the same length as the title

4.2 Study-General Directory Structure

4.3 Data and Documentation Storage Plan

4.4 Codebooks

here is codebook 1

4.5 NDA Guidelines