Data Basics, Management and Sharing

Steve Slevinski
Agenda

Why Data Management?
Brain Imaging Data Structure (BIDS)
NIMH Data Archive (NDA)
Globus data transfers
Flywheel
Data Basics
Why Data Management?

- Make data accessible and understandable
- Reduce data redundancy
- Produce more meaningful results
- Improve decision-making
- Increase reproducibility and transparency
- Facilitate scientific discovery and enquiry
- Expand research recognition and attribution
Brain Imaging Data Structure (BIDS)

https://bids.neuroimaging.io/
<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1.7.0</td>
<td>2022-02-15</td>
</tr>
<tr>
<td>v1.6.0</td>
<td>2021-04-22</td>
</tr>
<tr>
<td>v1.5.0</td>
<td>2021-02-23</td>
</tr>
<tr>
<td>v1.4.1</td>
<td>2020-10-13</td>
</tr>
<tr>
<td>v1.4.0</td>
<td>2020-06-11</td>
</tr>
<tr>
<td>v1.3.0</td>
<td>2020-04-14</td>
</tr>
<tr>
<td>v1.2.2</td>
<td>2020-02-12</td>
</tr>
<tr>
<td>v1.2.1</td>
<td>2019-08-14</td>
</tr>
<tr>
<td>v1.2.0</td>
<td>2019-03-04</td>
</tr>
<tr>
<td>v1.1.2</td>
<td>2019-01-10</td>
</tr>
</tbody>
</table>
Scanner to BIDS Organization

DICOM (.dcm) Digital Imaging and Communications in Medicine
NIFTI (.nii) Neuroimaging Informatics Technology Initiative
Benefits of BIDS

- Easier to use
- Reduces error
- Optimized analysis
- Automated tools

https://xkcd.com/2054
Features of BIDS

- Modular data
- Standard folder hierarchy
- Meaningful file names for humans and code
Modular Data

Source Data
Beginning input

Raw Data
Standard data formats

Derivatives
Output by pipeline
Standard Folder Hierarchy

```
dataset/
  ├── sourcedata
  │    ├── input_stream_1
  │    └── input_stream_2
  └── rawdata
      ├── dataset_description.json
      │    └── participants.tsv
      │    └── participants.json
      │    └── phenotype
      │        └── <measurement_tool_name>.tsv
      │        └── <measurement_tool_name>.json
      └── sub-01
          ├── anat
          │    └── dwi
          │    └── fmap
          │    └── func
          └── sub-02
              └── anat
                  └── dwi
                  └── fmap
                  └── func
```

Subject Only

Subject and Session
Meaningful File Names for Humans and Code

- **Template**: sub-<label>[_ses-<label>][_acq-<label>][_run-<index>]<suffix>.nii.gz
- **File Name**: sub-10622_ses-20201020133843_acq-SL50_T1rho.nii.gz

A file name is a combination of key-value pairs, a suffix and an extension
- Dashes - separate key and value
- Underscores _ separate key-value pairs and separates the suffix
- Square brackets [ ] indicate that a key-value pair is optional
- <label> is an alphanumeric string
- <index> is a positive integer
- <suffix> is an alphanumeric string
BIDS Validator

Validate Dataset           Validate File Names

Chrome Browser    Node JS    Python

https://github.com/bids-standard/bids-validator
BIDS Validator in the Browser

https://bids-standard.github.io/bids-validator/
BIDS Validator with Node JS

```shell
# install latest node and npm versions
curl -o https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
nvm install --lts
nvm install-latest-npm

# install bids-validator
npm install -g bids-validator

# validate dataset
bids-validator rawdata/
```
BIDS Validator with Python

Install with “pip install --user bids_validator”

```python
from bids_validator import BIDSValidator
d= BIDSValidator()
filepaths = ["/sub-01/anat/sub-01_rec-CSD_T1w.nii.gz", "/sub-01/anat/sub-01_acq-23_rec-CSD_T1w.exe"]
for filepath in filepaths:
    print(d.is_bids(filepath))  # will print True, and then False
```

Welcome to the NIMH Data Archive

The National Institute of Mental Health Data Archive (NDA) makes available human subjects data collected from hundreds of research projects across many scientific domains. NDA provides infrastructure for sharing research data, tools, methods, and analyses enabling collaborative science and discovery. De-identified human subjects data, harmonized to a common standard, are available to qualified researchers. Summary data are available to all.

The NDA mission is to accelerate scientific research and discovery through data sharing, data harmonization, and the reporting of research results.

https://nda.nih.gov/
NDA Webinars And Tutorials

New Grantee Orientation

Data Harmonization

Data Validation & Submission

Data Access

https://nda.nih.gov/webinars-and-tutorials
Data Sharing Process Overview
## NDA Collection

**Growth and development of Striatal-Cerebellum circuitry in subjects at risk for Huntington’s Disease #3409**

**State:** Shared

<table>
<thead>
<tr>
<th>General</th>
<th>Experiments (1)</th>
<th>Shared Data</th>
<th>Publications (9)</th>
<th>Data Expected (14)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Submissions</td>
<td>Associated Studies (0)</td>
<td>Permissions</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Experiments for fMRI
- Data Expected for data to submit
- Permissions for users: query, submission, and admin
Global Unique Identifier

Example GUID - NDARAB533LMW

- Download GUID Tool from NDA
- Enter information from birth certificate
- Generate GUID
- Batch mode available

https://nda.nih.gov/s/guid/nda-guid.html
NDA Data Dictionary

https://nda.nih.gov/general-query.html

<table>
<thead>
<tr>
<th>SELECT</th>
<th>TITLE</th>
<th>SHORT NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Axis 1 Diagnoses</strong></td>
<td>a1d01</td>
</tr>
<tr>
<td></td>
<td><strong>Alternative Aspects of Personality Disorders and Social Functioning</strong></td>
<td>aa01</td>
</tr>
<tr>
<td></td>
<td><strong>Adapted ADOS Module 1</strong></td>
<td>aados_m101</td>
</tr>
<tr>
<td></td>
<td><strong>Adapted ADOS Module 2</strong></td>
<td>aados_m201</td>
</tr>
<tr>
<td></td>
<td><strong>Attitudes and Expectations Form</strong></td>
<td>aae01</td>
</tr>
</tbody>
</table>
## NDA Data Structure

<table>
<thead>
<tr>
<th>ELEMENT NAME</th>
<th>DATA TYPE</th>
<th>SIZE</th>
<th>REQUIRED</th>
<th>CONDITION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>subjectkey</td>
<td>GUID</td>
<td></td>
<td>Required</td>
<td></td>
<td>The NDAR Global Unique Identifier (GUID) for research subject</td>
</tr>
<tr>
<td>src_subject_id</td>
<td>String</td>
<td>20</td>
<td>Required</td>
<td></td>
<td>Subject ID how it's defined in lab/project</td>
</tr>
<tr>
<td>interview_date</td>
<td>Date</td>
<td></td>
<td>Required</td>
<td></td>
<td>Date on which the interview/genetic test/sampling/imaging/biospecimen was completed. MM/DD/YYYY</td>
</tr>
<tr>
<td>interview_age</td>
<td>Integer</td>
<td></td>
<td>Required</td>
<td></td>
<td>Age in months at the time of the interview/test/sampling/imaging.</td>
</tr>
<tr>
<td>sex</td>
<td>String</td>
<td>20</td>
<td>Required</td>
<td></td>
<td>Sex of subject at birth</td>
</tr>
</tbody>
</table>

Additional columns of value range, notes, and aliases

Required column can be Required, Recommended, or Conditional

Recommended rows must be submitted if available

Conditional rows specify a Boolean test using another element value
# NDA Data Expected

<table>
<thead>
<tr>
<th>Data Expected</th>
<th>Targeted Enrollment</th>
<th>Initial Submission</th>
<th>Subjects Submitted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Child Behavior Checklist (CBCL)</td>
<td>400</td>
<td>07/15/2021</td>
<td>41</td>
</tr>
<tr>
<td>Adult Behavior Checklist (ABCL)</td>
<td>400</td>
<td>07/15/2021</td>
<td>37</td>
</tr>
<tr>
<td>Research Subject and Pedigree</td>
<td>400</td>
<td>07/15/2021</td>
<td>95</td>
</tr>
<tr>
<td>Picture Sequence Memory</td>
<td>400</td>
<td>07/15/2021</td>
<td>67</td>
</tr>
</tbody>
</table>
Format Image Data with bids2nda

- Download main.py and rename to bids2nda.py
- Requires code modifications for errors and issues
- Creates image03 data structure and zipped metadata files
- Ready to validate and upload

Validation and Upload Tool

Chrome Browser

Python

https://nda.nih.gov/tools/nda-tools.html#vt
Validation and Upload Tool

https://nda.nih.gov/vt/
Validation and Upload Tool

# install validation and upload tool
pip install --user nda-tools

# validate datasheet
vtcmd psm.csv

# upload datasheet after passing validation
vtcmd psm.csv -l /dir/associated/files -b -c 3409 -u username -d "Description" -t "Title"

https://nda.nih.gov/tools/nda-tools.html#vt
NDA Data Access

Request Access  Search Data  Download Data
NDA Request Access

- Make an account with NDA
- Go to Data Permissions
- Complete Data Use Agreement
- Send to DSP workflow for signature
- Upload signed Data Use Agreement to NDA
- Once approved, acknowledge permissions on NDA site
General Search

- NIMH Data Archive
- Adolescent Brain Cognitive Development Study
- CCF
- Osteoarthritis Initiative
- NIAAA
- Accelerating Medicines Partnership - Schizophrenia (AMP SCZ) Data Repository

Filter Data Types

- CONTENT (15)
  - NDA STUDIES (37)
  - PUBLICATIONS (1166)
  - COLLECTIONS (601)
  - DATA STRUCTURES (118)
  - DATA ELEMENTS (80)
  - EXPERIMENTS (133)

2150 search results for

https://nda.nih.gov/search
Data Search

Anyone
• Summary
• General information

Approved Access
• Detailed subject level data
• Build data packages

Search by
- Featured Datasets
- Data from Labs
- Data from Paper
- Data Dictionary
- Experiments
- Global Unique Identifier (GUID)

Download Data

- Create data package
- Create miNDAR from package
- Oracle database
  - Requires Oracle SQL Developer
- Amazon Web Services
- Granular download
- Download Manager (Electron app with 200GB limit)
- nda-tools (Python with security token)
Globus

https://www.globus.org

Large Scale Data Transfer
Collection:
uiowadata#data

Access University of Iowa data

https://www.youtube.com/channel/UCeSfxCh8iohP-9Bg0TOhDJA
globus connect software

globus connect personal
For researchers and other end users

globus connect

globus connect server
For network and system administrators
Transfer

- Move data from endpoint to endpoint
- Endpoints on Globus connect server
- Endpoints on Globus connect personal
Sharing

- Authenticated University Login
- Share authenticated security rights
  - Globus interacts as you
- Share read and/or write access
  - Let others interact as you
- Share with users and groups
  - Authenticated with Globus

Know your data classification

Public, Internal, Restricted, Critical

https://its.uiowa.edu/support/article/110901
globus connect personal
Flywheel

https://flywheel.io

Platform for imaging research
Data Basics

JSON

ASCII and Unicode

Bash variables, scripts, and arguments

Regular Expressions for text

Globbing for files
JSON

A lightweight data-interchange format

https://jsonbeautify.com

https://www.json.org
JSON

A lightweight data-interchange format

https://jsonbeautify.com

string = "hello world"
number = 0.2
array = [1, 2, 3]

object = {
    "string": "hello",
    "number": 7,
    "array": [1, 2, 3]
}

https://www.json.org
ASCII

The universal 7-bit character set

- 7-bits have 128 different combinations
- Int: 0 to 127
- Hex: 00 to 7F

- 8-bits have 256 different combinations
- Int: 0 to 255
- Hex: 00 to FF
Unicode

The international 32-bit character set

- Divided into multiple 16-bit planes
- 16-bits have 65,536 different combinations
- Hex: 0000 to FFFF
- Each character can be referenced with a plane followed by a 16-bit value.
- U+0007F is plane zero and character 007F.
  - This is ASCII character DEL.
- U+1D800 is plane one with character D800.
  - This character is not available in ASCII.
UTF-8 encoding form: ASCII superset

<table>
<thead>
<tr>
<th>First code point</th>
<th>Last code point</th>
<th>Byte 1</th>
<th>Byte 2</th>
<th>Byte 3</th>
<th>Byte 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>U+0000</td>
<td>U+007F</td>
<td>0xxxxx</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U+0080</td>
<td>U+07FF</td>
<td>110xxxx</td>
<td>10xxxxx</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U+0800</td>
<td>U+FFFF</td>
<td>1110xxxx</td>
<td>10xxxxx</td>
<td>10xxxxx</td>
<td></td>
</tr>
<tr>
<td>U+10000</td>
<td>U+10FFF</td>
<td>11110xxx</td>
<td>10xxxxx</td>
<td>10xxxxx</td>
<td>10xxxxx</td>
</tr>
</tbody>
</table>

https://www.urlencoder.io/

"ASCII-Double-Quotes"
“Unicode-Fancy-Quotes”
UTF-16 encoding: an unfortunate reality

<table>
<thead>
<tr>
<th>High</th>
<th>Low</th>
<th>DC00</th>
<th>DC01</th>
<th>…</th>
<th>DFFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>D800</td>
<td></td>
<td>010000</td>
<td>010001</td>
<td>…</td>
<td>0103FF</td>
</tr>
<tr>
<td>D801</td>
<td></td>
<td>010400</td>
<td>010401</td>
<td>…</td>
<td>0107FF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>\</td>
<td></td>
</tr>
<tr>
<td>DBFF</td>
<td></td>
<td>10FC00</td>
<td>10FC01</td>
<td>…</td>
<td>10FFFF</td>
</tr>
</tbody>
</table>

https://nddapp.com/json-encoder.html
"ASCII-Double-Quotes"
“Unicode-Fancy-Quotes"
Bash

Bash cheat sheet
https://devhints.io/bash

Bash quick start
https://learnxinyminutes.com/docs/bash/

Bash reference
https://www.gnu.org/software/bash/manual/
## Variables

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>test1=hello</td>
<td>Variable test1 = “hello”</td>
</tr>
<tr>
<td>test2=“hello world”</td>
<td>Variable test2 = “hello world”</td>
</tr>
<tr>
<td>readonly test3=“value”</td>
<td>Variable test3 = “value” and can not be change</td>
</tr>
<tr>
<td>echo $test2</td>
<td>Prints “hello world” with $1 = hello, $2 = world</td>
</tr>
<tr>
<td>echo “$test2”</td>
<td>Prints “hello world” with $1 = “hello world”</td>
</tr>
<tr>
<td>echo ${test1}</td>
<td>Prints “hello”</td>
</tr>
<tr>
<td>echo $test1_world</td>
<td>Prints nothing</td>
</tr>
<tr>
<td>echo ${test1}_world</td>
<td>Prints “hello_world”</td>
</tr>
</tbody>
</table>
Variable parameter expansion

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>echo ${test:-alternate}</td>
<td>Prints “alternate” if test is not set</td>
</tr>
<tr>
<td>echo ${test:=alternate}</td>
<td>Updates test and prints if test is not set</td>
</tr>
<tr>
<td>echo ${test:+alternate}</td>
<td>Prints “alternate” if test is set, otherwise prints nothing</td>
</tr>
<tr>
<td>echo ${test:1}</td>
<td>Prints test starting after first character</td>
</tr>
<tr>
<td>echo ${test:3:2}</td>
<td>Prints two characters starting after the third character</td>
</tr>
<tr>
<td>echo ${#test}</td>
<td>Prints length of test</td>
</tr>
<tr>
<td>echo ${test/pattern/string}</td>
<td>Prints test after substituting first matching pattern with string</td>
</tr>
<tr>
<td>echo ${test//pattern/string}</td>
<td>Prints test after substituting all matching patterns with string</td>
</tr>
<tr>
<td>echo ${!test}</td>
<td>Prints values of variable referenced by value of test</td>
</tr>
</tbody>
</table>

[https://wiki.bash-hackers.org/syntax/pe](https://wiki.bash-hackers.org/syntax/pe)
### Array Variables

<table>
<thead>
<tr>
<th>Command</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Declare –a ARRAYNAME</td>
<td>Explicitly create an empty array variable</td>
</tr>
<tr>
<td>ARRAYNAME=(A B “C D”)</td>
<td>Create array variable with three values</td>
</tr>
<tr>
<td>ARRAYNAME[3]=“E F G”</td>
<td>Set value of array variable at index</td>
</tr>
<tr>
<td>ARRAYNAME+=(H)</td>
<td>Add an element to end of array variable</td>
</tr>
<tr>
<td>ARRAYNAME+=(I “J K L”)</td>
<td>Add two elements to end of array variable</td>
</tr>
<tr>
<td>echo $ARRAYNAME</td>
<td>Prints first element of array variable</td>
</tr>
<tr>
<td>echo ${ARRAYNAME[0]}</td>
<td>Prints first element of array variable</td>
</tr>
<tr>
<td>echo ${ARRAYNAME[*]}</td>
<td>Prints all elements of array variable</td>
</tr>
<tr>
<td>echo ${#ARRAYNAME[@]}</td>
<td>Prints length of array</td>
</tr>
<tr>
<td>for ELEMENT in &quot;${ARRAYNAME[@]}&quot;; do echo $ELEMENT done</td>
<td>Prints all elements of array variable</td>
</tr>
</tbody>
</table>
# Scripts and input arguments

```bash
#!/bin/bash

echo "Script is $0"

echo "First arg is ${1:-unset}"

echo "Second arg is ${2:-unset}"

../script.sh

../script.sh hello world

../script.sh "hello world"
```
### Positional Arguments versus flags

<table>
<thead>
<tr>
<th>Example</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>echo hello world</td>
<td>$1 is “hello”, $2 is “world”</td>
</tr>
<tr>
<td>echo “hello world”</td>
<td>$1 is “hello world”</td>
</tr>
<tr>
<td>cp current.txt new.txt</td>
<td>$1 is “current.txt”, $2 is “new.txt”</td>
</tr>
<tr>
<td>cp “old file” “new file”</td>
<td>$1 is “old file”, $2 is “new file”</td>
</tr>
<tr>
<td>ls -l -t -r</td>
<td>Turn on flags “-l” for long, “-t” for time order, and “-r” for reverse</td>
</tr>
<tr>
<td>ls --ignore README.txt</td>
<td>Flag “--ignore” will list files other than README.txt</td>
</tr>
</tbody>
</table>

For flag arguments use getopts command.
## Regular Expression for text

<table>
<thead>
<tr>
<th>Characters</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>Match any character</td>
<td>. matches A, B, C, or any single character</td>
</tr>
<tr>
<td>*</td>
<td>Match preceding literal 0 or more times</td>
<td>ABC* matches AB, ABC, ABCC, ...</td>
</tr>
<tr>
<td>+</td>
<td>Match preceding literal 1 or more times</td>
<td>ABC+ matches ABC, ABCC, ABCCC, ...</td>
</tr>
<tr>
<td>?</td>
<td>Match preceding literal 0 or 1 times</td>
<td>ABC? matches AB or ABC</td>
</tr>
<tr>
<td>{#}</td>
<td>Match preceding literal “#” times</td>
<td>AB{2} matches ABB</td>
</tr>
<tr>
<td>[ ]</td>
<td>Match any single literal from a list</td>
<td>[ABC] matches A, B or C</td>
</tr>
<tr>
<td>[ - ]</td>
<td>Match any single literal from a range</td>
<td>[A-C] matches A, B, or C</td>
</tr>
<tr>
<td>( )</td>
<td>Creates a group for matching</td>
<td>A(BC)+ matches ABC, ABCBC, ABCBCBC, ...</td>
</tr>
<tr>
<td>(</td>
<td>)</td>
<td>Matches one of several alternatives</td>
</tr>
<tr>
<td>(?: )</td>
<td>Creates a non-capturing group</td>
<td>A(?:BC) will match ABC as one group</td>
</tr>
<tr>
<td>^</td>
<td>Match beginning of line</td>
<td>^A.* will match any line starting with A</td>
</tr>
<tr>
<td>$</td>
<td>Match ending of line</td>
<td>.*A$ will match any line ending with A</td>
</tr>
</tbody>
</table>
# Globbing for files

<table>
<thead>
<tr>
<th>Characters</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>Match any single character</td>
<td><code>ls test.??</code></td>
</tr>
<tr>
<td>*</td>
<td>Match any string of characters, except /</td>
<td><code>ls test.*</code></td>
</tr>
<tr>
<td>**</td>
<td>Match any string of characters, including /</td>
<td><code>ls **/test.txt</code></td>
</tr>
<tr>
<td>[ ]</td>
<td>Match any characters from a list</td>
<td><code>ls [ac-z]*</code></td>
</tr>
<tr>
<td>[! ]</td>
<td>Match any characters not in a list</td>
<td><code>ls ![ac-z]*</code></td>
</tr>
<tr>
<td>{ }</td>
<td>Use multiple matching patterns</td>
<td><code>ls {*.txt,*.zip}</code></td>
</tr>
<tr>
<td>+(</td>
<td>)</td>
<td>Match one of several alternatives</td>
</tr>
</tbody>
</table>

Use “`set –f`” to turn off globbing and “`set +f`” to globbing back on.

Use “`shopt -s globstar`” to turn on ** if not enabled.
# Globs versus Regular Expressions

<table>
<thead>
<tr>
<th>Characters</th>
<th>Meaning in Globs</th>
<th>Meaning in Regular Expressions</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Match zero or more characters</td>
<td>Match zero or more of the preceding literal</td>
</tr>
<tr>
<td>?</td>
<td>Match any single character</td>
<td>Match zero or one of the preceding literal</td>
</tr>
<tr>
<td>.</td>
<td>Literal “.” character</td>
<td>Match any single character</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Globs for files</th>
<th>Regular Expressions for text</th>
</tr>
</thead>
<tbody>
<tr>
<td>names* matches names.txt and names.zip</td>
<td>names* matches name, names, and namess</td>
</tr>
</tbody>
</table>
Thank You

Steve Slevinski
Data Manager

steve-slevinski@uiowa.edu

https://neuroimaging.uiowa.edu