bxh2bids: Goals, Structure, and Usage

John Graner
12/13/2019

`github.com/jlgraner/bxh2bids`

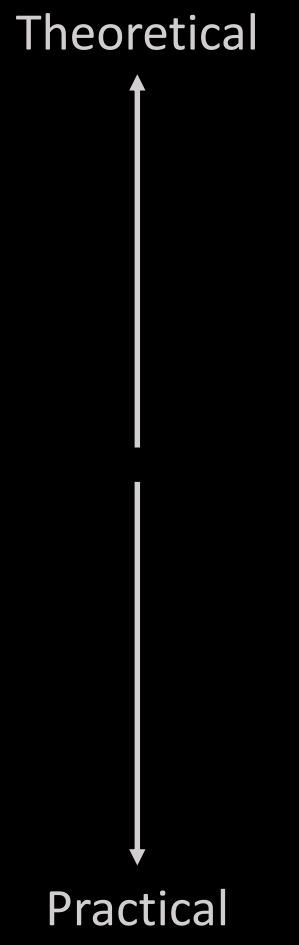
Outline

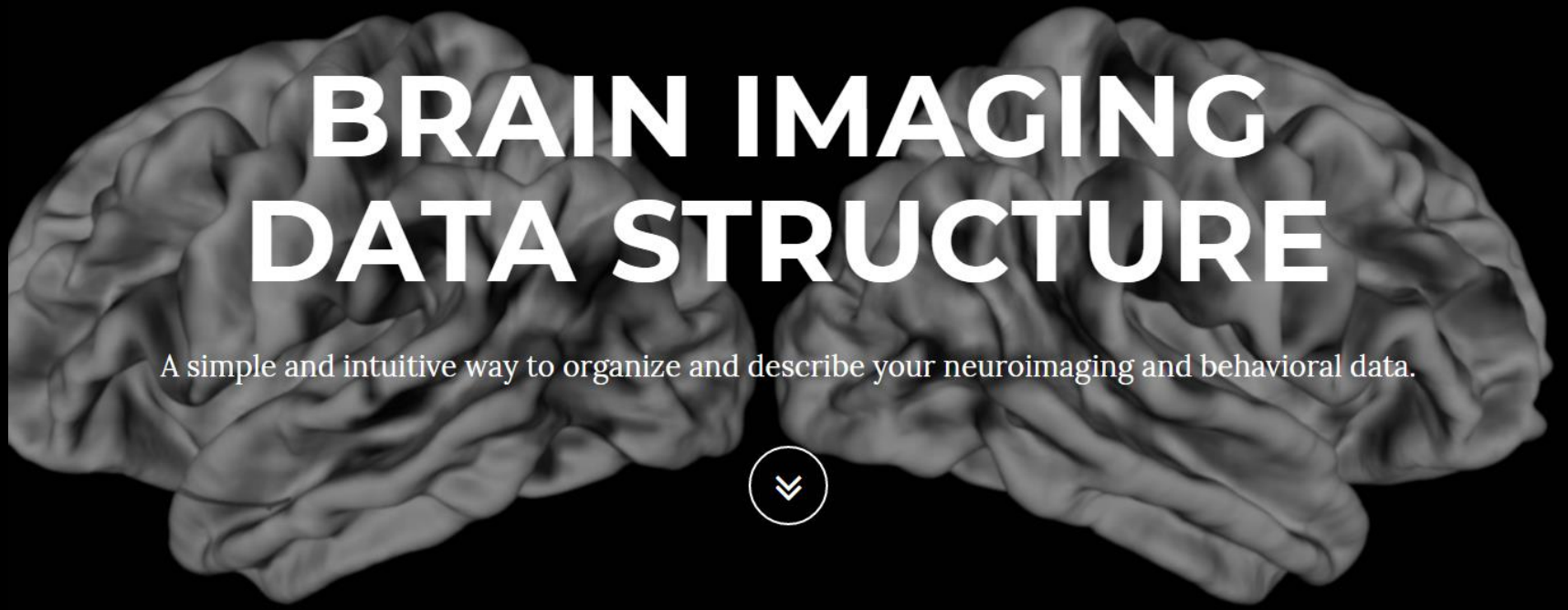
What is BIDS and why is it useful?

What does bxh2bids do?

How does bxh2bids work?

Short bxh2bids demo

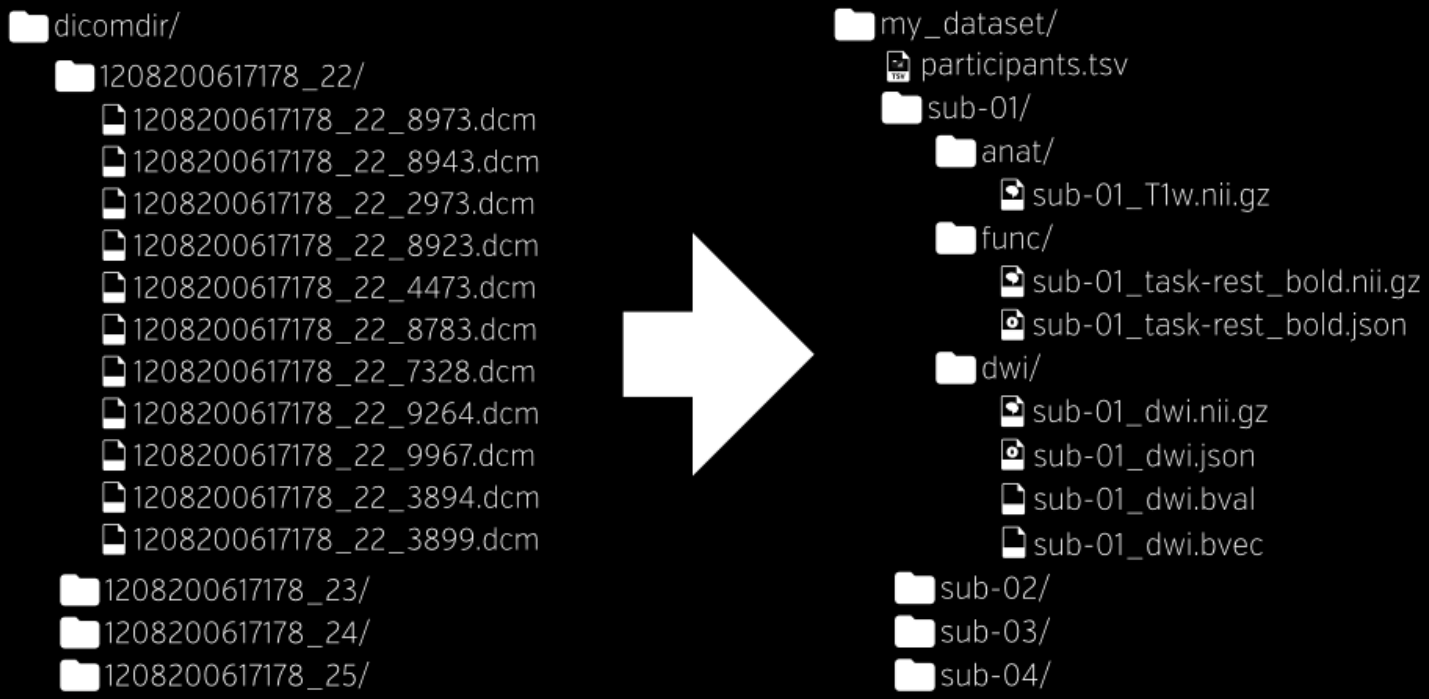




http://bids.neuroimaging.io/

John Graner, Ph.D., Duke University

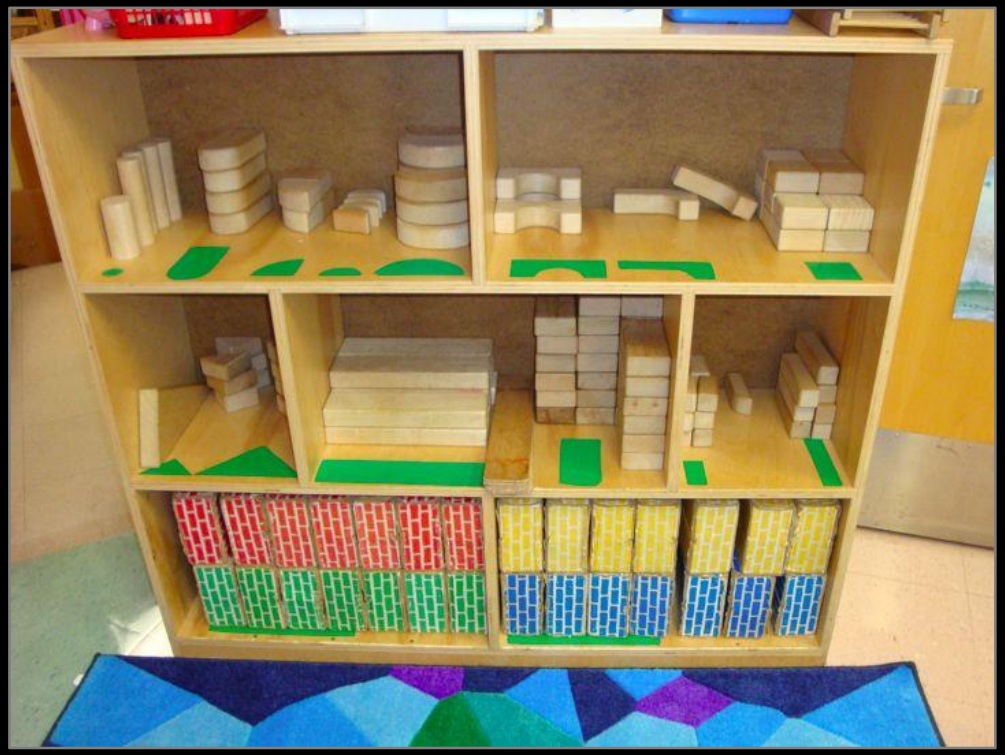
ABOUT DOWNLOAD EXAMPLES GET INVOLVED ACKNOWLEDGMENTS FEEDBACK



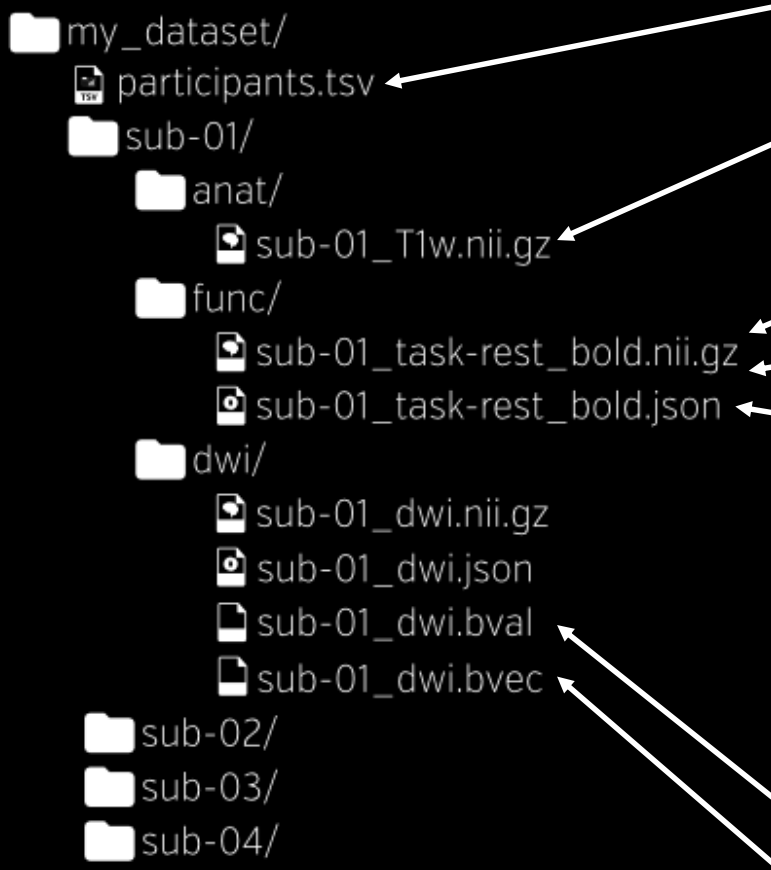
http://bids.neuroimaging.io/

Dissecting BIDS

- my_dataset/
  - participants.tsv
  - sub-01/
    - anat/
      - sub-01_T1w.nii.gz
    - func/
      - sub-01_task-rest_bold.nii.gz
      - sub-01_task-rest_bold.json
    - dwi/
      - sub-01_dwi.nii.gz
      - sub-01_dwi.json
      - sub-01_dwi.bval
      - sub-01_dwi.bvec
  - sub-02/
  - sub-03/
  - sub-04/



Dissecting BIDS



(optional) demographic info. of participants

T1-weighted anat. data

BOLD data (example: resting state)

Missing in this example:
.tsv file with functional task event timings!

json file containing scan parameters


```
{
    "EchoTime": 0.03,
    "RepetitionTime": 2.0,
    "MagneticFieldStrength": 3.0,
    "InversionTime": 0.0,
    "FlipAngle": 60,
    "NumberOfVolumesDiscardedByScanner": 0,
    "SliceTiming": [
        0.0,
        1.0270270270270272,
        0.05405405405405406,
```


b-values (row of numbers)

b-vectors (3 rows of numbers)

(".tsv" == tab-separated values)

How does this support open science and reproducibility?

- my_dataset/
  - participants.tsv
  - sub-01/
    - anat/
      - sub-01_T1w.nii.gz
    - func/
      - sub-01_task-rest_bold.nii.gz
      - sub-01_task-rest_bold.json
    - dwi/
      - sub-01_dwi.nii.gz
      - sub-01_dwi.json
      - sub-01_dwi.bval
      - sub-01_dwi.bvec
  - sub-02/
  - sub-03/
  - sub-04/

Transparency - Other people (*including future you*) can easily see what each file contains based on the file names

How does this support open science and reproducibility?

- my_dataset/
  - participants.tsv
  - sub-01/
    - anat/
      - sub-01_T1w.nii.gz
    - func/
      - sub-01_task-rest_bold.nii.gz
      - sub-01_task-rest_bold.json
    - dwi/
      - sub-01_dwi.nii.gz
      - sub-01_dwi.json
      - sub-01_dwi.bval
      - sub-01_dwi.bvec
  - sub-02/
  - sub-03/
  - sub-04/

Transparency - Other people (*including future you*) can easily see what each file contains based on the file names

Predictability - Standardization allows easy access with scripts, which facilitates standardization of processing/analysis

(mriqc) *(fmriprep)*

(BIDS allows you to use both of these very easily. See links on final slide!)

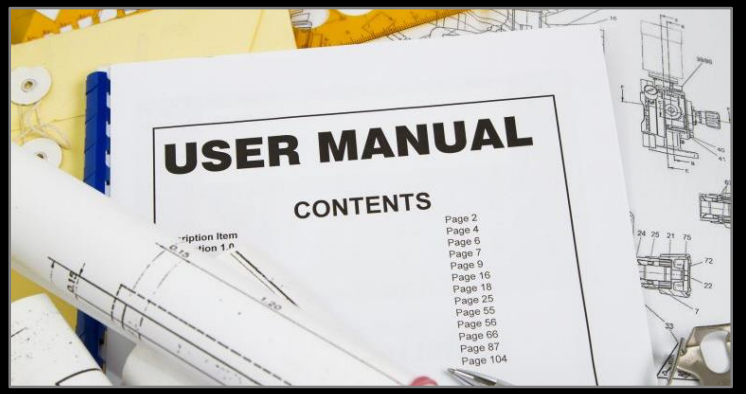
How does this support open science and reproducibility?

- my_dataset/
  - participants.tsv
  - sub-01/
    - anat/
      - sub-01_T1w.nii.gz
    - func/
      - sub-01_task-rest_bold.nii.gz
      - sub-01_task-rest_bold.json
    - dwi/
      - sub-01_dwi.nii.gz
      - sub-01_dwi.json
      - sub-01_dwi.bval
      - sub-01_dwi.bvec
  - sub-02/
  - sub-03/
  - sub-04/

Transparency - Other people (*including future you*) can easily see what each file contains based on the file names

Predictability - Standardization allows easy access with scripts, which facilitates standardization of processing/analysis

Provenance - Including task timing, acquisition parameters, etc. provides the context of the data alongside the data themselves



BIAC

```
MyStudy/
    Data/
        Anat/
            19540101_00001/
                bia6_00001_001.nii.gz
                bia6_00001_001.bxh
            19540102_00002/
                bia6_00002_001.nii.gz
                bia6_00002_001.bxh
                bia6_00002_100.nii.gz
                bia6_00002_100.bxh
        Func/
            19540101_00001/
                bia6_00001_003_01.nii.gz
                bia6_00001_003_01.bxh
                bia6_00001_003_02.nii.gz
                bia6_00001_003_02.bxh
            19540102_00002/
                bia6_00002_003_01.nii.gz
                bia6_00002_003_01.bxh
                bia6_00002_004_01.nii.gz
                bia6_00002_004_01.bxh
```

BIDS

```
my_dataset/
    participants.tsv
    sub-01/
        anat/
            sub-01_T1w.nii.gz
        func/
            sub-01_task-rest_bold.nii.gz
            sub-01_task-rest_bold.json
        dwi/
            sub-01_dwi.nii.gz
            sub-01_dwi.json
            sub-01_dwi.bval
            sub-01_dwi.bvec
    sub-02/
    sub-03/
    sub-04/
```

Goal

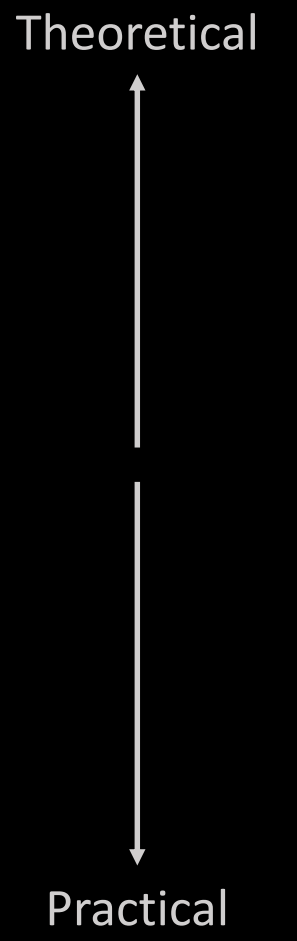
Make it as easy as possible to convert data from BIAC into BIDS

~~What is BIDS and why is it useful?~~

What does bxh2bids do?

How does bxh2bids work?

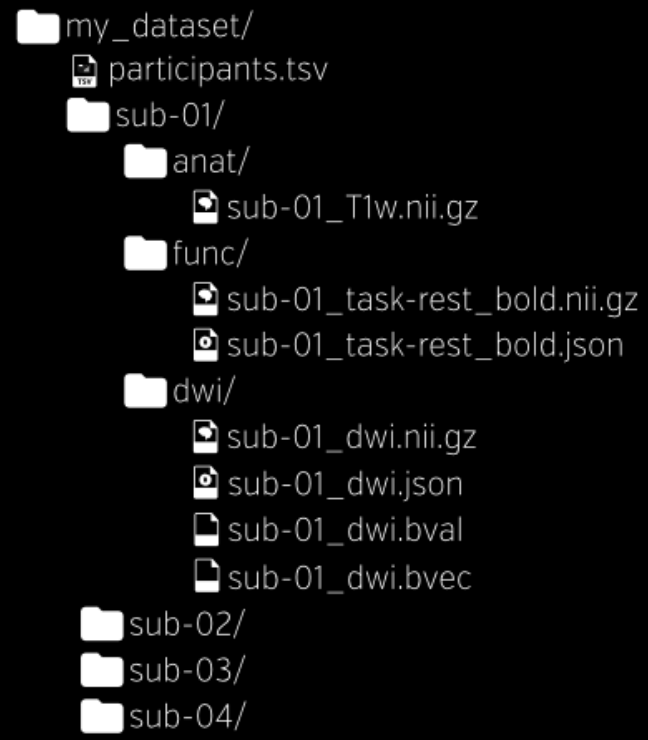
Short bxh2bids demo



What it does…

bxh2bids

Locate .bxh files



What it does…

Anat/
- 19540101_00001/
  - bia6_00001_001.nii.gz
  - bia6_00001_001.bxh
- 19540102_00002/
  - bia6_00002_001.nii.gz
  - bia6_00002_001.bxh
  - bia6_00002_100.nii.gz
  - bia6_00002_100.bxh

Func/
- 19540101_00001/
  - bia6_00001_003_01.nii.gz
  - bia6_00001_003_01.bxh
  - bia6_00001_004_01.nii.gz
  - bia6_00001_004_01.bxh
- 19540102_00002/
  - bia6_00002_003_01.nii.gz
  - bia6_00002_003_01.bxh
  - bia6_00002_004_01.nii.gz
  - bia6_00002_004_01.bxh

bxh2bids

For each .bxh file:

- Locate image data

- my_dataset/
  - participants.tsv
  - sub-01/
    - anat/
      - sub-01_T1w.nii.gz
    - func/
      - sub-01_task-rest_bold.nii.gz
      - sub-01_task-rest_bold.json
    - dwi/
      - sub-01_dwi.nii.gz
      - sub-01_dwi.json
      - sub-01_dwi.bval
      - sub-01_dwi.bvec
  - sub-02/
  - sub-03/
  - sub-04/

What it does…

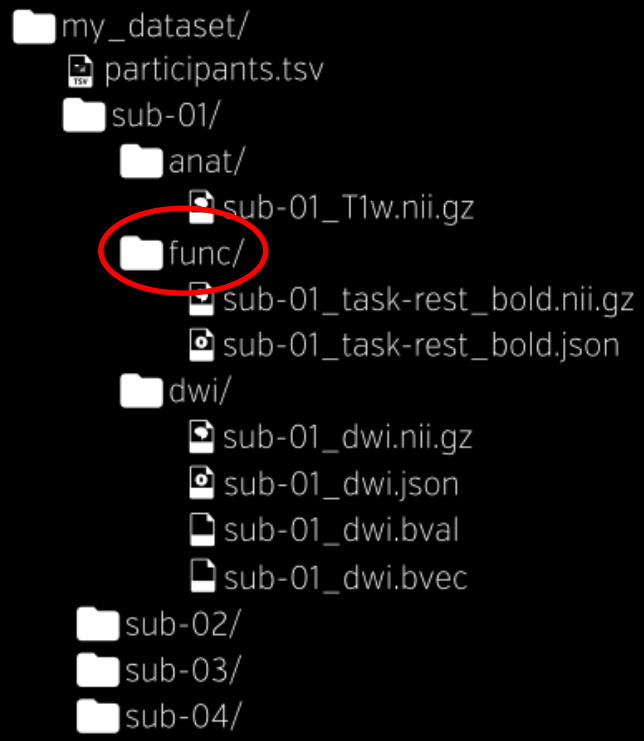
Anat/
- 19540101_00001/
  - bia6_00001_001.nii.gz
  - bia6_00001_001.bxh
- 19540102_00002/
  - bia6_00002_001.nii.gz
  - bia6_00002_001.bxh
  - bia6_00002_100.nii.gz
  - bia6_00002_100.bxh

Func/
- 19540101_00001/
  - bia6_00001_003_01.nii.gz
  - bia6_00001_003_01.bxh
  - bia6_00001_004_01.nii.gz
  - bia6_00001_004_01.bxh
- 19540102_00002/
  - bia6_00002_003_01.nii.gz
  - bia6_00002_003_01.bxh
  - bia6_00002_004_01.nii.gz
  - bia6_00002_004_01.bxh

bxh2bids

For each .bxh file:

- Locate image data
- Determine BIDS Type



What it does…

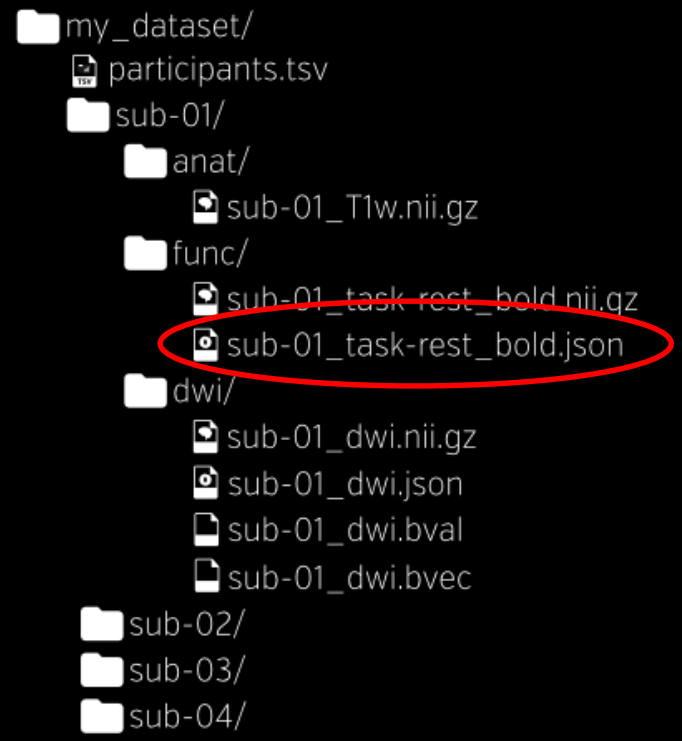
Anat/
- 19540101_00001/
  - bia6_00001_001.nii.gz
  - bia6_00001_001.bxh
- 19540102_00002/
  - bia6_00002_001.nii.gz
  - bia6_00002_001.bxh
  - bia6_00002_100.nii.gz
  - bia6_00002_100.bxh

Func/
- 19540101_00001/
  - bia6_00001_003_01.nii.gz
  - bia6_00001_003_01.bxh
  - bia6_00001_004_01.nii.gz
  - bia6_00001_004_01.bxh
- 19540102_00002/
  - bia6_00002_003_01.nii.gz
  - bia6_00002_003_01.bxh
  - bia6_00002_004_01.nii.gz
  - bia6_00002_004_01.bxh

bxh2bids

For each .bxh file:

- Locate image data
- Determine BIDS Type
- Create json file



What it does…

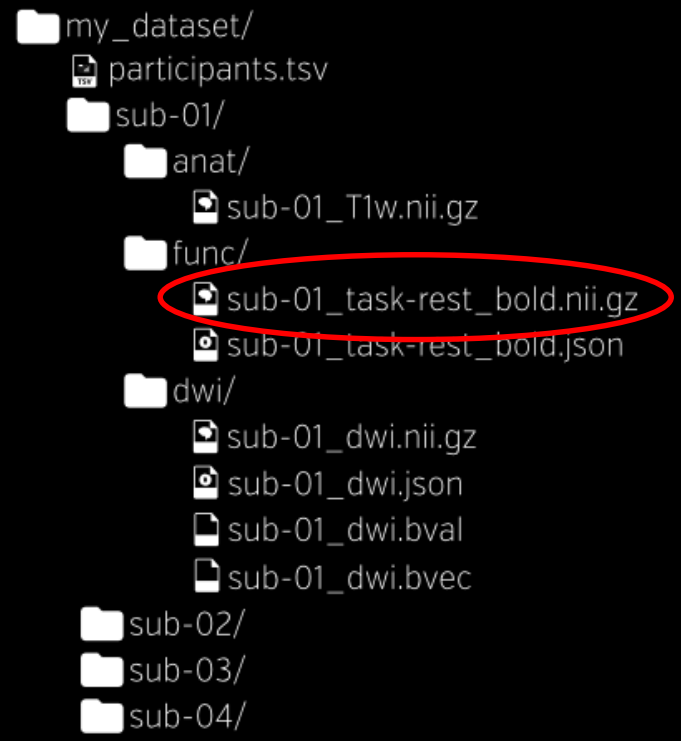
Anat/
- 19540101_00001/
  - bia6_00001_001.nii.gz
  - bia6_00001_001.bxh
- 19540102_00002/
  - bia6_00002_001.nii.gz
  - bia6_00002_001.bxh
  - bia6_00002_100.nii.gz
  - bia6_00002_100.bxh

Func/
- 19540101_00001/
  - bia6_00001_003_01.nii.gz
  - bia6_00001_003_01.bxh
  - bia6_00001_004_01.nii.gz
  - bia6_00001_004_01.bxh
- 19540102_00002/
  - bia6_00002_003_01.nii.gz
  - bia6_00002_003_01.bxh
  - bia6_00002_004_01.nii.gz
  - bia6_00002_004_01.bxh

bxh2bids

For each .bxh file:

- Locate image data
- Determine BIDS Type
- Create json file
- Rename/copy image



What it does…

Anat/
- 19540101_00001/
  - bia6_00001_001.nii.gz
  - bia6_00001_001.bxh
- 19540102_00002/
  - bia6_00002_001.nii.gz
  - bia6_00002_001.bxh
  - bia6_00002_100.nii.gz
  - bia6_00002_100.bxh

Func/
- 19540101_00001/
  - bia6_00001_003_01.nii.gz
  - bia6_00001_003_01.bxh
  - bia6_00001_004_01.nii.gz
  - bia6_00001_004_01.bxh
- 19540102_00002/
  - bia6_00002_003_01.nii.gz
  - bia6_00002_003_01.bxh
  - bia6_00002_004_01.nii.gz
  - bia6_00002_004_01.bxh

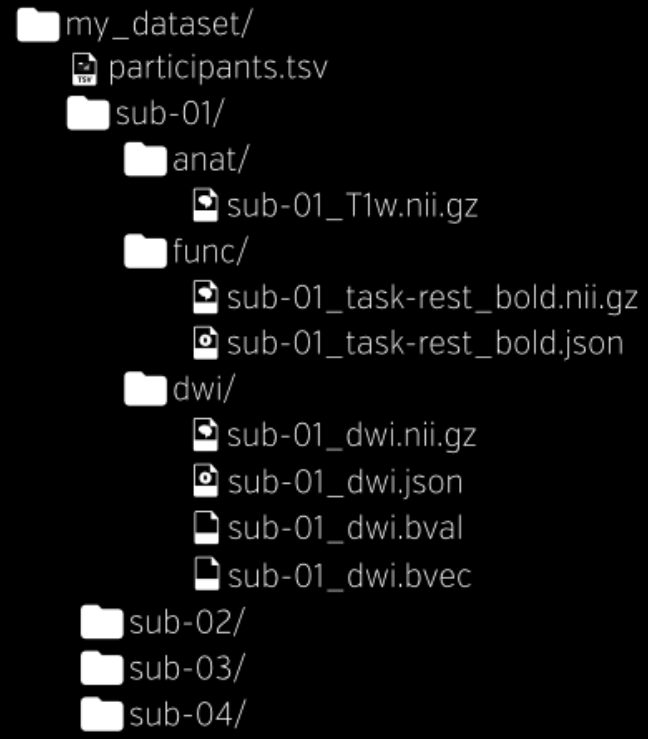
bxh2bids

For each .bxh file:

- Locate image data
- Determine BIDS Type
- Create json file
- Rename/copy image

If Functional:

- Rename/Copy event .tsv file



What it does…

Anat/
- 19540101_00001/
  - bia6_00001_001.nii.gz
  - bia6_00001_001.bxh
- 19540102_00002/
  - bia6_00002_001.nii.gz
  - bia6_00002_001.bxh
  - bia6_00002_100.nii.gz
  - bia6_00002_100.bxh

Func/
- 19540101_00001/
  - bia6_00001_003_01.nii.gz
  - bia6_00001_003_01.bxh
  - bia6_00001_004_01.nii.gz
  - bia6_00001_004_01.bxh
- 19540102_00002/
  - bia6_00002_003_01.nii.gz
  - bia6_00002_003_01.bxh
  - bia6_00002_004_01.nii.gz
  - bia6_00002_004_01.bxh

bxh2bids

For each .bxh file:

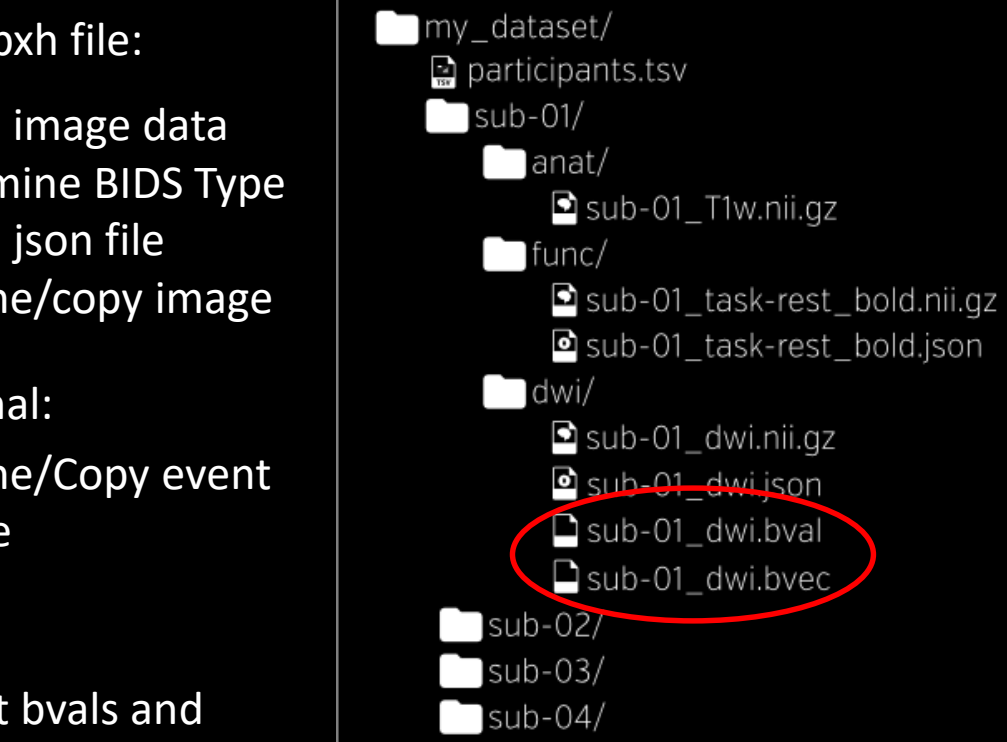
- Locate image data
- Determine BIDS Type
- Create json file
- Rename/copy image

If Functional:

- Rename/Copy event .tsv file

If DWI:

- Extract bvals and bvecs into files



Thanks to John Powers

What it does…

```
Anat/
    19540101_00001/
        bia6_00001_001.nii.gz
        bia6_00001_001.bxh
    19540102_00002/
        bia6_00002_001.nii.gz
        bia6_00002_001.bxh
        bia6_00002_100.nii.gz
        bia6_00002_100.bxh
Func/
    19540101_00001/
        bia6_00001_003_01.nii.gz
        bia6_00001_003_01.bxh
        bia6_00001_004_01.nii.gz
        bia6_00001_004_01.bxh
    19540102_00002/
        bia6_00002_003_01.nii.gz
        bia6_00002_003_01.bxh
        bia6_00002_004_01.nii.gz
        bia6_00002_004_01.bxh
```

…with a little help.

User

For each BIAC session:

- Fill out .json template with BIDS sub, ses, and run labels.
- If functional task, create a .tsv file.

For each study:

- Fill out a “run” python script with data input/output information

```
my_dataset/
    participants.tsv
    sub-01/
        anat/
            sub-01_T1w.nii.gz
        func/
            sub-01_task-rest_bold.nii.gz
            sub-01_task-rest_bold.json
        dwi/
            sub-01_dwi.nii.gz
            sub-01_dwi.json
            sub-01_dwi.bval
            sub-01_dwi.bvec
    sub-02/
    sub-03/
    sub-04/
```

What it does…

…with a little help.

User

For each BIAC session:

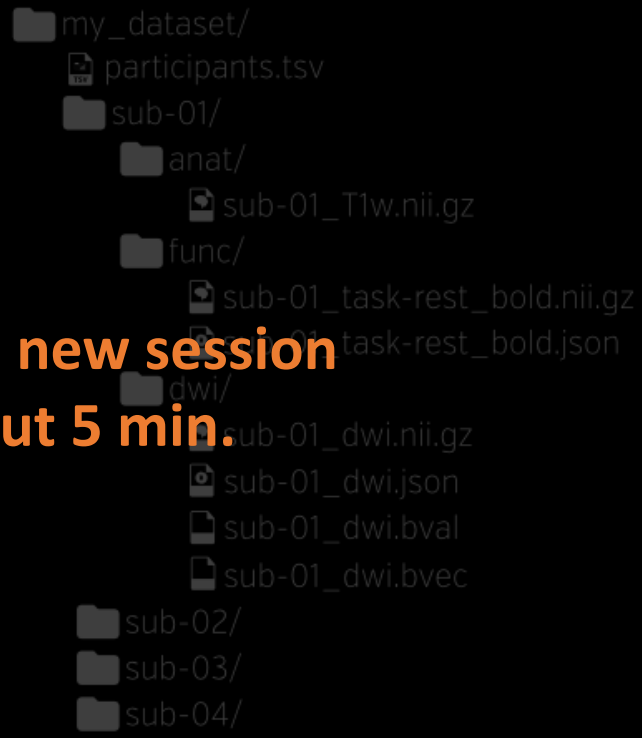
- Fill out .json template with BIDS sub, ses, and run labels.
- If functional/task, create a .tsv file.

For each study:

- Fill out .json template with basic directory info (input, output, etc.)

But fear not!

After initial set-up, getting a new session into BIDS takes only about 5 min.

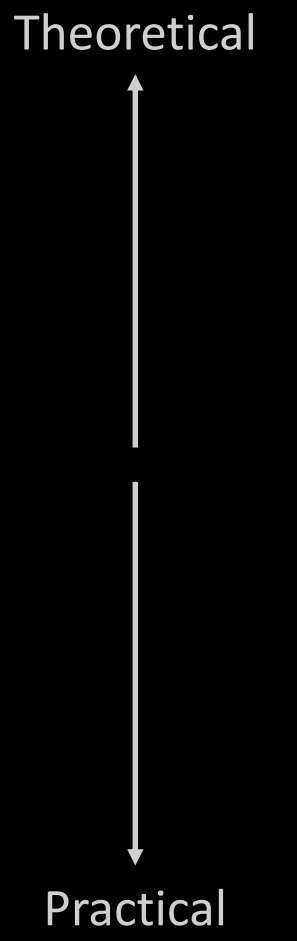


~~What is BIDS and why is it useful?~~

~~What does bxh2bids do?~~

How does bxh2bids work?

Short bxh2bids demo



Getting the Code

Available on Github: https://github.com/jlgraner/bxh2bids

- Click the “Clone or Download” button
- Save as a .zip, then unzip it

OR

Clone via git:

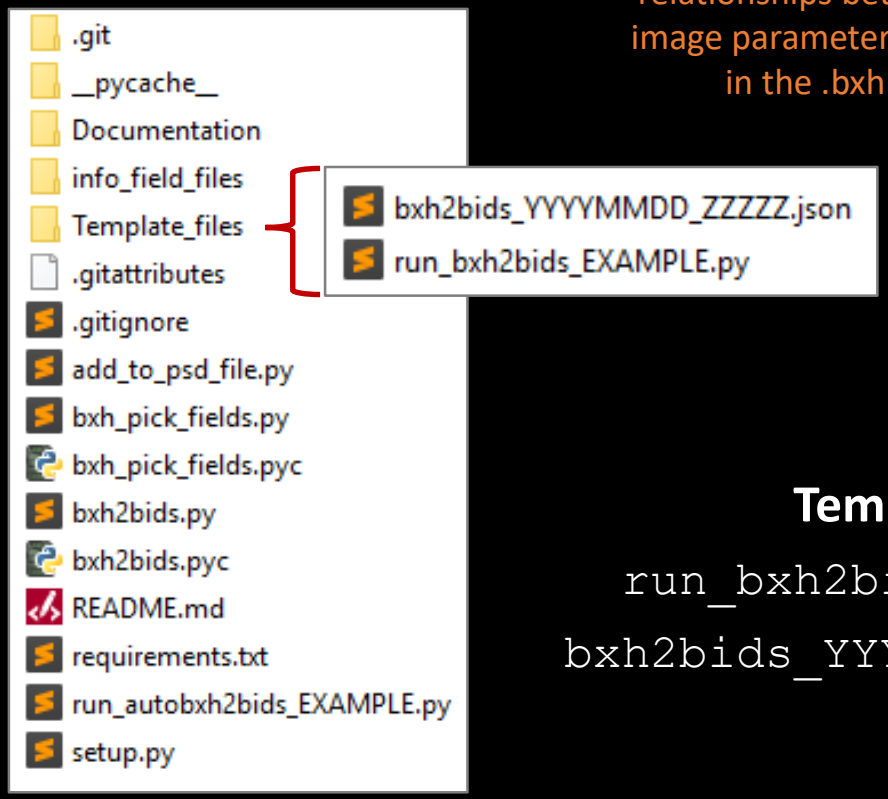
```
“git clone https://github.com/jlgraner/bxh2bids”
```

Things Inside the Zip

Python Scripts

`bxh2bids.py`: library containing the functions that actually do things

`bxh_pick_fields.py`: function that pulls BIDS-relevant info. from .bxh files



(files that store relationships between BIDS image parameters and fields in the .bxh files)

"Translation" Files

`anat_info_fields.json`

`func_info_fields.json`

`dwi_info_fields.json`

`psd_types.json`

Template Files

(files that store information BIDS needs to know but that isn't in the .bxh files)

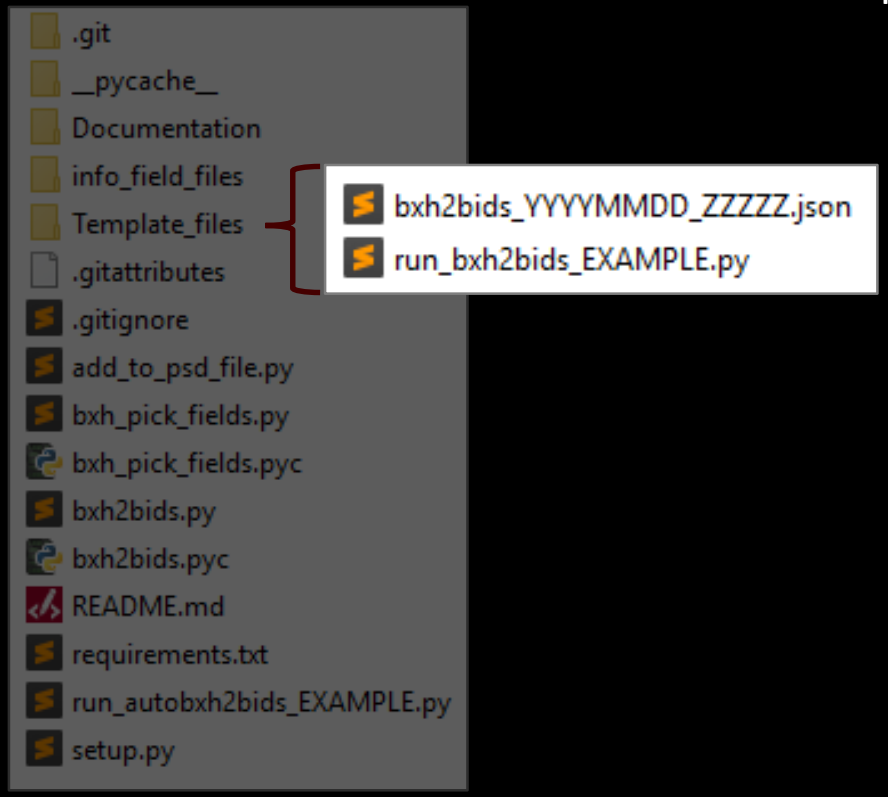
`run_bxh2bids_EXAMPLE.py`

`bxh2bids_YYYYMMDD_ZZZZ.json`

Things Inside the Zip
(That You'd Care About)

`bxh2bids_YYYYMMDD_ZZZZ.json`: Session information file template. You'll need one edited and renamed copy of this for each imaging session.

`run_bxh2bids_EXAMPLE.py`: Script that runs everything. You'll need one edited and renamed copy for each study.



run_bxh2bids.py

John Graner, Ph.D., Duke University

"Out of the box"

```
#COPY OF THE bxh2bids DIRECTORY AS A STRING.
bxh2bids_dir = "DIRECTORY CONTAINING YOUR bxh2bids.py FILE"

sys.path.append(bxh2bids_dir)
import bxh2bids as b2b

#The sessions you wish to put into BIDS format right now

to_run = [
          'YYYYMMDD_ZZZZZ'
          ]

#Set information about your study sessions
source_study_dir="PATH CONTAINING YOUR 'Data' DIRECTORY"
target_study_dir="WHERE YOU WANT THE BIDS DATA TO BE WRITTEN. THE LAST ELEMENT OF THE PATH WILL BE THE STUDY NAME."
log_dir="DIRECTORY WHERE YOU WANT THIS SCRIPT TO WRITE LOG FILES"
ses_info_dir="DIRECTORY WHERE YOU WILL STORE YOUR SESSION INFO FILES"
```

Ready to run

```
#COPY OF THE bxh2bids DIRECTORY AS A STRING.
bxh2bids_dir = "/mydir/code/bxh2bids"

sys.path.append(bxh2bids_dir)
import bxh2bids as b2b

#The sessions you wish to put into BIDS format right now

to_run = [
          '20191213_12345'
          ]

#Set information about your study sessions
source_study_dir="/mydir/studies/mystudy"
target_study_dir="/mydir/studies/mystudy/data_BIDS"
log_dir="/mydir/studies/mystudy/data_BIDS/bxh2bids_logs"
ses_info_dir="/mydir/studies/mystudy/data_BIDS/bxh2bids_ses_info"
```

John Graner, Ph.D., Duke University

bxh2bids_SESSION_INFO.json

Create a directory to house copies of this outside the repository.

You'll need one of these per session.

"Out of the box"

```
{
"sub": "[[SUBJECT ID 1]]",
"ses": "[[BIDS SES]]",
"funcs": {
          "###_##": {
                    "task": "[[YOUR TASK NAME]]",
                    "run": "##",
                    "tsv_file": "[[FULL PATH AND FILE NAME OF THE .TSV FILE ASSOCIATED WITH THIS RUN]].tsv"
                    },
          "###_##": {
                    "task": "[[YOUR TASK NAME]]",
                    "run": "##",
                    "tsv_file": "[[FULL PATH AND FILE NAME OF THE .TSV FILE ASSOCIATED WITH THIS RUN]].tsv"
                    }
          }
}
```

bxh2bids_19540101_00001.json

Ready to run

```
{
"sub": "01",
"ses": "day1",
"funcs": {
          "003_01": {
                    "task": "MyTask",
                    "run": "01",
                    "tsv_file": "/where/my/event/files/are/sub-01_ses-day1_task-MyTask_run-run1_events.tsv"
                    },
          "004_01": {
                    "task": "MyOtherTask",
                    "run": "01",
                    "tsv_file": "/where/my/event/files/are/sub-01_ses-day1_task-MyOtherTask_run-run1_events.tsv"
                    }
          }
}
```

Example Functional Event .tsv File

It is up to the user* to create these to match the BIDS specification.

The first 2 columns (onset, duration) are required; other optional columns are specified in the BIDS spec.

onset	duration	trial_type	stim_info	response_time
8.27	2.90	arrow	correct	0.47282
11.30	2.90	arrow	correct	0.33608
14.33	2.90	arrow	correct	0.34302
17.52	5.01	negMemCue	CENSORED	n/a
22.53	10.01	negStratCue	DISTRACT	n/a
32.55	5.02	negVRate	4	2.9466
37.78	5.00	negARate	4	1.2226
52.83	2.90	arrow	correct	0.60232
55.86	2.90	arrow	correct	0.37413
58.90	2.90	arrow	correct	0.32297
62.00	5.01	negMemCue	CENSORED	n/a
67.01	10.01	negStratCue	DISTRACT	n/a
77.03	5.02	negVRate	4	2.1837
82.26	5.00	negARate	4	1.0129
94.31	2.90	arrow	correct	0.39248
97.34	2.90	arrow	None	None
100.38	2.90	arrow	correct	0.44316
103.48	5.01	negMemCue	CENSORED	n/a
108.49	10.01	negStratCue	DISTRACT	n/a
118.51	5.02	negVRate	4	2.2781
123.74	5.00	negARate	4	0.74089
137.81	2.90	arrow	correct	0.46613
140.84	2.90	arrow	correct	0.45588
143.87	2.90	arrow	correct	0.3679
146.97	5.01	neuMemCue	CENSORED	n/a
151.99	10.01	neuStratCue	FLOW	n/a
162.01	5.02	neuVRate	4	1.2277
167.24	5.00	neuARate	4	0.74964
182.29	2.90	arrow	correct	0.47444

*John G. has some code to convert FSL condition files to a BIDS .tsv.

Thanks to Leonard Faul

Logistical Summary

Initial Setup (~30 min)

https://github.com/jlgraner/bxh2bids — Download

run_bxh2bids_EXAMPLE.py

bxh2bids_YYYYMMDD_ZZZZZ.json

} Copy and edit

Things to do for each BIAC session (each unique YYYYMMDD_##### directory) (~5 min)

Event .tsv files

bxh2bids_YYYYMMDD_ZZZZZ.json file

} Create

When you want to run it

`run_bxh2bids_MYSTUDY.py` - Edit session list

"`python -m run_bxh2bids_MYSTUDY`"

Useful Links

bxh2bids: https://github.com/jlgraner/bxh2bids

BIDS main site: http://bids.neuroimaging.io

BIDS validator: http://incf.github.io/bids-validator/

BIDS spec: http://bids.neuroimaging.io/bids_spec1.0.2.pdf

mriqc: http://mriqc.readthedocs.io/en/latest/

fmriprep: http://fmriprep.readthedocs.io/en/latest/