NiBetaSeries

Release 0.2.3

Jan 29, 2019

Contents

NiBetaSeries

docs	
tests	
package	

NiBetaSeries is BIDS compatible application that calculates betaseries correlations. In brief, a beta coefficient is calculated for each trial (or event) resulting in a series of betas that can be used to correlate regions of interest with each other.

NiBetaSeries takes preprocessed data as input that satisfy the BIDS deriviatives specification. In practical terms, NiBetaSeries uses the output of fmriprep, a great BIDS compatible preprocessing tool. NiBetaSeries requires the input and the atlas to already be in MNI space since currently no transformations are applied to the data. You can use any arbitrary atlas as long as it is in MNI space (the same space as the preprocessed data).

With NiBetaSeries you can receive:

- betaseries images (TODO)
- correlation matrices

This is a very young project that still needs some tender loving care to grow. That's where you fit in! If you would like to contribute, please read our code of conduct and contributing page.

This project heavily leverages nipype, nilearn, pybids, and nistats for development. Please check out their pages and support the developers.

- Free software: MIT license

1.1 Installation

```
pip install nibetaseries
```

1.2 Documentation

https://nibetaseries.readthedocs.io

If you're interested in contributing to this project, here are some guidelines for contributing. Another good place to start is by checking out the open issues.

1.3 Development

To run the all tests run:

```
tox
```

Note, to combine the coverage data from all the tox environments run:

Windows	
	`set PYTEST_ADDOPTS=--cov-append` `tox`
Other	
	`PYTEST_ADDOPTS=--cov-append tox`

Betaseries

2.1 Introduction

Betaseries track the trial-to-trial modelled hemodynamic fluctuations that occur within task functional magnetic resonance imaging (task fMRI). This fills an important analytical gap between measuring hemodynamic fluctuations in resting state fMRI and measuring regional activations via cognitive subtraction in task fMRI.

2.2 Relationship to Resting State

Betaseries is similar to resting state because the same analytical strategies applied to resting state data can ostensibly be applied to betaseries. At the core of both resting state and betaseries we are working with a list of numbers at each voxel. We can correlate, estimate regional homogeneity, perform independent components analysis, or conduct a number of different analyses with these voxels. However, betaseries deviates from resting state in two important ways. First, you can do cognitive subtraction using betaseries. Second, interpretation of the lists of numbers are different for resting state and betaseries. Resting state measures the unmodelled hemodynamic fluctuations that occur without explicit stimuli. Betaseries, on the other hand, measures the modelled hemodynamic fluctuations that occur in response to an explicit stimulus. Both resting state and betaseries may measure intrinsic connectivity, but betaseries may also measure the task evoked connectivity (e.g. connectivity between regions that is increased during some cognitive process).

2.3 Relationship to Traditional Task Analysis

Betaseries is also similar to traditional task analysis because cognitive subtraction can be used in both. As with resting state, betaseries deviates from traditional task analysis in several important ways. Say we are interested in observing how the brain responds to faces versus houses. The experimenter has a timestamp of exactly when and how long a face or house is presented. That timestamp information is typically convolved with a hemodynamic response function (HRF) to represent how the brain stereotypically responds to any stimulus resulting in a model of how we expect the brain to respond to places and/or faces. This is where traditional task analysis and betaseries diverge. In traditional task analysis all the face trials are estimated at once, giving one summary measure for how strongly each voxel was

activated (same for house trials). The experimenter can subtract the summary measure of faces from houses to see which voxels are more responsive to houses relative to faces (i.e. cognitive subtraction). In betaseries each trial is estimated separately each voxel has as many estimates at there are trials (which can be labelled as either face or house trials). The experimenter can now reduce the series of estimates (a betaseries) for each voxel into a summary measure such as a correlation with region(s) of interest. The correlation map for faces can be subtracted from houses, giving voxels that are more correlated with the region of interest for houses relative to faces. Whereas traditional task analysis treats the variance of brain responses between trials of the same type (e.g. face or house) as noise, betaseries leverages this variance to make conclusions about which brain regions may communicate with each other during a particular trial type (e.g. faces or houses).

2.4 Summary

Betaseries is not in opposition to resting state or traditional task analysis, the methods are complementary. For example, network parcelations derived from resting state data can be used on betaseries data to ascertain if the networks observed in resting state follow a similar pattern with betaseries. Additionally, regions determined from traditional task analysis can be used as regions of interest for betaseries analysis. Betaseries straddles the line between traditional task analysis and resting state, observing task data through a network lens.

2.5 Conceptual Background

Jesse Rissman *[Rissman2004]* was the first to publish on betaseries correlations describing their usage in the context of a working memory task. In this task, participants saw a cue, a delay, and a probe, all occurring close in time. A cue was presented for one second, a delay occurred for seven seconds, and a probe was presented for one second. Given the HRF takes approximately six seconds to reach its peak, and generally takes over 20 seconds to completely resolve, we can begin to see a problem. The events within the trials occur too close to each other to discern what brain responses are related to encoding the cue, the delay, or the probe. To discern how the activated brain regions form networks, Rissman conceptualized betaseries correlations. Instead of having a single regressor to describe all the cue events, a single regressor for all the delay events, and a single regressor for all the probe events (as is done in traditional task analysis), there is an individual regressor for every event in the experiment. For example, if your experiment has 40 trials, each with a cue, delay, and probe event, the model will have a total of 120 regressors, fitting a beta estimate for each trial. Complete this process for each trial of a given event type (e.g. cue), at the end you will have `4D` volume where each volume represents the beta estimates for a particular trial, and each voxel represents a specific beta estimate.

Having one regressor per trial in a single model is known as least squares all. This method, however, has limitations in the context of fast event related designs. Since each trial has its own regressor, trials that occur very close in time are colinear (e.g. are very overlapping). Jeanette Mumford et al. *[Mumford2012]* investigated this issue in the context of image classification. In this article, Mumford introduces another modelling strategy known as least squares separate. In this modelling strategy, instead of having one GLM with a regressor per trial, least squares separate implements a GLM per trial with two regressors: 1) one for the trial of interest, and 2) one for every other trial in the experiment. This process reduces the colinearity of the regressors and creates a more valid estimate of how each regressor fits the data.

2.6 Math Background

$Y = X\beta + \epsilon$ (2.1)

The above equation is the General Linear Model (GLM) presented using matrix notation. Y represents the time-series we are attempting to explain. The β assumes any value that minimizes the squared error between the modeled data and the actual data, Y. Finally, ϵ (epsilon) refers to the error that is not captured by the model. Within a GLM, trial-to-trial betas may be averaged for a given trial type and the variance is treated as noise. However, those trial-to-trial fluctuations may also contain important information the typical GLM will ignore/penalize. With a couple modifications to the above equation, we arrive at calculating a betaseries.

$$Y = X_1\beta_1 + X_2\beta_2 + ... + X_n\beta_n + \epsilon (2.2)$$

With the betaseries equation, a beta is estimated for every trial, instead of for each trial type (or whatever logical grouping). This gives us the ability to align all the trial betas from a single trial type into a list (i.e. series). This operation is completed for all voxels, giving us as many lists of betas as there are voxels in the data. Essentially, we are given a `4-D` dataset where the fourth dimension represents trial number instead of time (as the fourth dimension is represented in resting state). And analogous to resting state data, we can perform correlations between the voxels to discern which voxels (or which aggregation of voxels) synchronize best with other voxels.

There is one final concept to cover in order to understand how the betas are estimated in `NiBetaSeries`. You can model individual betas using a couple different strategies; with a least squares all (LSA) approximation which the equation above represents, and a least squares separate (LSS) approximation in which each trial is given it's own GLM. The advantage of LSS comes from reducing the colinearity between closely spaced trials. In LSA, if trials occurred close in time then it would be difficult to model whether the fluctuations should be attributed to one trial or the other. LSS reduces this ambiguity by only having two regressors: one for the trial of interest and another for every other trial. This reduces the colinearity between regressors and makes each beta estimate more reliable.

```python
for trial, beta in zip(trials, betas):
    data = X[trial] * beta + error
```

This python psuedocode demonstrates LSS where each trial is given it's own model.

2.7 Relevent Software

BASCO (BetA Series COrrelations) is a matlab program that also performs betaseries correlations

2.8 Other Relevant Readings

- *[Cisler2014]*
- *[Gottlich2015]*
- *[Abdulrahman2016]*

2.9 References

Installation

At the command line:

```
pip install nibetaseries
```

Usage

4.1 Command-Line Arguments

NiBetaSeries BIDS arguments

```
usage: nibs [-h] [-v] [-sm SMOOTHING_KERNEL] [-lp LOW_PASS]
            [-c CONFOUNDS [CONFOUNDS ...]] [-w WORK_DIR]
            [--participant_label PARTICIPANT_LABEL [PARTICIPANT_LABEL ...]]
            [--session_label SESSION_LABEL] [-t TASK_LABEL]
            [--run_label RUN_LABEL] [-sp {MNI152NLin2009cAsym}]
            [--variant_label VARIANT_LABEL] [--exclude_variant_label]
            [--hrf_model {glover,spm,fir,glover + derivative,glover + derivative +␣
→dispersion,spm + derivative,spm + derivative + dispersion}]
            [-a ATLAS_IMG] -l ATLAS_LUT [--nthreads NTHREADS]
            [--use-plugin USE_PLUGIN] [--graph]
            bids_dir derivatives_pipeline output_dir {participant,group}
```

4.1.1 Positional Arguments

bids_dir   The directory with the input dataset formatted according to the BIDS standard.

derivatives_pipeline   The pipeline that contains minimally preprocessed img, brainmask, and confounds.tsv

output_dir   The directory where the output directory and files should be stored. If you are running group level analysis this folder should be prepopulated with the results of theparticipant level analysis.

analysis_level   Possible choices: participant, group

                           Level of the analysis that will be performed Multiple participant level analyses can be run independently (in parallel) using the same output_dir

4.1.2 Named Arguments

-v, --version   show program's version number and exit

--participant_label   The label(s) of the participant(s) that should be analyzed. The label corresponds to sub-<participant_label> from the BIDS spec (so it does not include "sub-"). If this parameter is not provided all subjects should be analyzed. Multiple participants can be specified with a space separated list.

4.1.3 Options for preprocessing

-sm, --smoothing_kernel   select a smoothing kernel (mm)

-lp, --low_pass   low pass filter (Hz)

-c, --confounds   The confound column names that are to be included in nuisance regression. write the confounds you wish to include separated by a space

-w, --work_dir   directory where temporary files are stored

4.1.4 Options for selecting images

--session_label   select a session to analyze

-t, --task_label   select a specific task to be processed

--run_label   select a run to analyze

-sp, --space_label   Possible choices: MNI152NLin2009cAsym

select a bold derivative in a specific space to be used

--variant_label   select a variant bold to process

--exclude_variant_label   exclude the variant from FMRIPREP

4.1.5 Options for processing beta_series

--hrf_model   Possible choices: glover, spm, fir, glover + derivative, glover + derivative + dispersion, spm + derivative, spm + derivative + dispersion

convolve your regressors with one of the following hemodynamic response functions

-a, --atlas-img   input atlas nifti where each voxel within a "region" is labeled with the same integer and there is a unique integer associated with each region of interest

-l, --atlas-lut   atlas look up table (tsv) formatted with the columns: index, regions which correspond to the regions in the nifti file specified by –atlas-img

4.1.6 Options to handle performance

--nthreads, --n_cpus, -n-cpus   maximum number of threads across all processes

--use-plugin   nipype plugin configuration file

4.1.7 misc options

--graph   generates a graph png of the workflow

How to run NiBetaSeries

Below is an example run through of NiBetaSeries

Note: Click *here* to download the full example code

5.1 Running NiBetaSeries using ds000164 (Stroop Task)

This example runs through a basic call of NiBetaSeries using the commandline entry point `nibs`. While this example is using python, typically `nibs` will be called directly on the commandline.

5.2 Import all the necessary packages

```python
import tempfile  # make a temporary directory for files
import os  # interact with the filesystem
import urllib.request  # grad data from internet
import tarfile  # extract files from tar
from subprocess import Popen, PIPE, STDOUT  # enable calling commandline

import matplotlib.pyplot as plt  # manipulate figures
import seaborn as sns  # display results
import pandas as pd   # manipulate tabular data
```

5.3 Download relevant data from ds000164 (and Atlas Files)

The subject data came from openneuro. The atlas data came from a recently published parcellation in a publically accessible github repository.

```
# atlas github repo for reference:
"""https://github.com/ThomasYeoLab/CBIG/raw/master/stable_projects/\
brain_parcellation/Schaefer2018_LocalGlobal/Parcellations/MNI/"""
data_dir = tempfile.mkdtemp()
print('Our working directory: {}'.format(data_dir))

# download the tar data
url = "https://www.dropbox.com/s/qoqbiya1ou7vi78/ds000164-test_v1.tar.gz?dl=1"
tar_file = os.path.join(data_dir, "ds000164.tar.gz")
u = urllib.request.urlopen(url)
data = u.read()
u.close()

# write tar data to file
with open(tar_file, "wb") as f:
    f.write(data)

# extract the data
tar = tarfile.open(tar_file, mode='r|gz')
tar.extractall(path=data_dir)

os.remove(tar_file)
```

Out:

```
Our working directory: /tmp/tmpuu2igxzh
```

5.4 Display the minimal dataset necessary to run nibs

```
# https://stackoverflow.com/questions/9727673/list-directory-tree-structure-in-python
def list_files(startpath):
    for root, dirs, files in os.walk(startpath):
        level = root.replace(startpath, '').count(os.sep)
        indent = ' ' * 4 * (level)
        print('{}{}/'.format(indent, os.path.basename(root)))
        subindent = ' ' * 4 * (level + 1)
        for f in files:
            print('{}{}'.format(subindent, f))


list_files(data_dir)
```

Out:

```
tmpuu2igxzh/
    ds000164/
        T1w.json
        README
        task-stroop_bold.json
        dataset_description.json
        task-stroop_events.json
        CHANGES
        derivatives/
            data/
```

(continues on next page)

```
                    Schaefer2018_100Parcels_7Networks_order.txt
                    Schaefer2018_100Parcels_7Networks_order_FSLMNI152_2mm.nii.gz
            fmriprep/
                sub-001/
                    func/
                        sub-001_task-stroop_bold_space-MNI152NLin2009cAsym_brainmask.
→nii.gz
                        sub-001_task-stroop_bold_confounds.tsv
                        sub-001_task-stroop_bold_space-MNI152NLin2009cAsym_preproc.
→nii.gz
        sub-001/
            anat/
                sub-001_T1w.nii.gz
            func/
                sub-001_task-stroop_bold.nii.gz
                sub-001_task-stroop_events.tsv
```

5.5 Manipulate events file so it satifies assumptions

1. the correct column has 1's and 0's corresponding to correct and incorrect, respectively. 2. the condition column is renamed to trial_type nibs currently depends on the "correct" column being binary and the "trial_type" column to contain the trial types of interest.

5.5.1 read the file

```
events_file = os.path.join(data_dir,
                        "ds000164",
                        "sub-001",
                        "func",
                        "sub-001_task-stroop_events.tsv")
events_df = pd.read_csv(events_file, sep='\t', na_values="n/a")
print(events_df.head())
```

Out:

```
onset   duration correct   condition   response_time
0   0.342          1       Y    neutral         1.186
1   3.345          1       Y  congruent         0.667
2  12.346          1       Y  congruent         0.614
3  15.349          1       Y    neutral         0.696
4  18.350          1       Y    neutral         0.752
```

5.5.2 change the Y/N to 1/0

```
events_df['correct'].replace({"Y": 1, "N": 0}, inplace=True)
print(events_df.head())
```

Out:

```
onset   duration   correct   condition   response_time
0   0.342        1         1     neutral         1.186
1   3.345        1         1   congruent         0.667
2  12.346        1         1   congruent         0.614
3  15.349        1         1     neutral         0.696
4  18.350        1         1     neutral         0.752
```

5.5.3 replace condition with trial_type

```
events_df.rename({"condition": "trial_type"}, axis='columns', inplace=True)
print(events_df.head())
```

Out:

```
onset   duration   correct trial_type   response_time
0   0.342        1         1     neutral         1.186
1   3.345        1         1   congruent         0.667
2  12.346        1         1   congruent         0.614
3  15.349        1         1     neutral         0.696
4  18.350        1         1     neutral         0.752
```

5.5.4 save the file

```
events_df.to_csv(events_file, sep="\t", na_rep="n/a", index=False)
```

5.6 Manipulate the region order file

There are several adjustments to the atlas file that need to be completed before we can pass it into nibs. Importantly, the relevant column names **MUST** be named "index" and "regions". "index" refers to which integer within the file corresponds to which region in the atlas nifti file. "regions" refers the name of each region in the atlas nifti file.

5.6.1 read the atlas file

```
atlas_txt = os.path.join(data_dir,
                         "ds000164",
                         "derivatives",
                         "data",
                         "Schaefer2018_100Parcels_7Networks_order.txt")
atlas_df = pd.read_csv(atlas_txt, sep="\t", header=None)
print(atlas_df.head())
```

Out:

```
0                   1    2    3    4  5
0  1  7Networks_LH_Vis_1  120  18  131  0
1  2  7Networks_LH_Vis_2  120  18  132  0
2  3  7Networks_LH_Vis_3  120  18  133  0
3  4  7Networks_LH_Vis_4  120  18  135  0
4  5  7Networks_LH_Vis_5  120  18  136  0
```

5.6.2 drop coordinate columns

```
atlas_df.drop([2, 3, 4, 5], axis='columns', inplace=True)
print(atlas_df.head())
```

Out:

```
0                 1
0  1  7Networks_LH_Vis_1
1  2  7Networks_LH_Vis_2
2  3  7Networks_LH_Vis_3
3  4  7Networks_LH_Vis_4
4  5  7Networks_LH_Vis_5
```

5.6.3 rename columns with the approved headings: "index" and "regions"

```
atlas_df.rename({0: 'index', 1: 'regions'}, axis='columns', inplace=True)
print(atlas_df.head())
```

Out:

```
index            regions
0      1  7Networks_LH_Vis_1
1      2  7Networks_LH_Vis_2
2      3  7Networks_LH_Vis_3
3      4  7Networks_LH_Vis_4
4      5  7Networks_LH_Vis_5
```

5.6.4 remove prefix "7Networks"

```
atlas_df.replace(regex={'7Networks_(.*)': '\\1'}, inplace=True)
print(atlas_df.head())
```

Out:

```
index    regions
0      1  LH_Vis_1
1      2  LH_Vis_2
2      3  LH_Vis_3
3      4  LH_Vis_4
4      5  LH_Vis_5
```

5.6.5 write out the file as .tsv

```
atlas_tsv = atlas_txt.replace(".txt", ".tsv")
atlas_df.to_csv(atlas_tsv, sep="\t", index=False)
```

5.7 Run nibs

```python
out_dir = os.path.join(data_dir, "ds000164", "derivatives")
work_dir = os.path.join(out_dir, "work")
atlas_mni_file = os.path.join(data_dir,
                              "ds000164",
                              "derivatives",
                              "data",
                              "Schaefer2018_100Parcels_7Networks_order_FSLMNI152_2mm.
↪nii.gz")
cmd = """\
nibs -c WhiteMatter CSF \
--participant_label 001 \
-w {work_dir} \
-a {atlas_mni_file} \
-l {atlas_tsv} \
{bids_dir} \
fmriprep \
{out_dir} \
participant
""".format(atlas_mni_file=atlas_mni_file,
           atlas_tsv=atlas_tsv,
           bids_dir=os.path.join(data_dir, "ds000164"),
           out_dir=out_dir,
           work_dir=work_dir)
# call nibs
p = Popen(cmd, shell=True, stdout=PIPE, stderr=STDOUT)

while True:
    line = p.stdout.readline()
    if not line:
        break
    print(line)
```

Out:

```
b'190129-14:37:59,784 nipype.workflow INFO:\n'
b"\t Workflow nibetaseries_participant_wf settings: ['check', 'execution', 'logging',
↪'monitoring']\n"
b'190129-14:37:59,795 nipype.workflow INFO:\n'
b'\t Running in parallel.\n'
b'190129-14:37:59,803 nipype.workflow INFO:\n'
b'\t [MultiProc] Running 0 tasks, and 1 jobs ready. Free memory (GB): 7.01/7.01, Free
↪processors: 4/4.\n'
b"/home/docs/checkouts/readthedocs.org/user_builds/nibetaseries/envs/v0.2.3/lib/
↪python3.5/site-packages/grabbit/core.py:449: UserWarning: Domain with name 'bids'
↪already exists; returning existing Domain configuration.\n"
b'  warnings.warn(msg)\n'
b'190129-14:37:59,853 nipype.workflow INFO:\n'
b'\t [Node] Setting-up "nibetaseries_participant_wf.single_subject001_wf.betaseries_
↪wf.betaseries_node" in "/tmp/tmpuu2igxzh/ds000164/derivatives/work/NiBetaSeries_
↪work/nibetaseries_participant_wf/single_subject001_wf/betaseries_wf/
↪a7dc49b58761b73b7aeeeee9d35703cc8ab43895/betaseries_node".\n'
b'190129-14:37:59,857 nipype.workflow INFO:\n'
b'\t [Node] Running "betaseries_node" ("nibetaseries.interfaces.nistats.BetaSeries")\n
↪'
b'190129-14:38:01,808 nipype.workflow INFO:\n'
```

(continues on next page)

```
b'\t [MultiProc] Running 1 tasks, and 0 jobs ready. Free memory (GB): 6.81/7.01, Free␣
↪processors: 3/4.\n'
b'                     Currently running:\n'
b'                       * nibetaseries_participant_wf.single_subject001_wf.
↪betaseries_wf.betaseries_node\n'
b'/home/docs/checkouts/readthedocs.org/user_builds/nibetaseries/envs/v0.2.3/lib/
↪python3.5/importlib/_bootstrap.py:222: RuntimeWarning: numpy.dtype size changed,␣
↪may indicate binary incompatibility. Expected 96, got 88\n'
b'  return f(*args, **kwds)\n'
b'/home/docs/checkouts/readthedocs.org/user_builds/nibetaseries/envs/v0.2.3/lib/
↪python3.5/site-packages/nibabel/nifti1.py:582: DeprecationWarning: The binary mode␣
↪of fromstring is deprecated, as it behaves surprisingly on unicode inputs. Use␣
↪frombuffer instead\n'
b'  ext_def = np.fromstring(ext_def, dtype=np.int32)\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b"/home/docs/checkouts/readthedocs.org/user_builds/nibetaseries/envs/v0.2.3/lib/
↪python3.5/site-packages/nistats/hemodynamic_models.py:268: DeprecationWarning:␣
↪object of type <class 'numpy.float64'> cannot be safely interpreted as an integer.\n
↪"
b'  frame_times.max() * (1 + 1. / (n - 1)), n_hr)\n'
b"/home/docs/checkouts/readthedocs.org/user_builds/nibetaseries/envs/v0.2.3/lib/
↪python3.5/site-packages/nistats/hemodynamic_models.py:55: DeprecationWarning:␣
↪object of type <class 'float'> cannot be safely interpreted as an integer.\n"
b'  time_stamps = np.linspace(0, time_length, float(time_length) / dt)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
```

```
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
```

```
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
```

```
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
```

```
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
```

```
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'190129-14:40:06,940 nipype.workflow INFO:\n'
b'\t [Node] Finished "nibetaseries_participant_wf.single_subject001_wf.betaseries_wf.
↪betaseries_node".\n'
b'190129-14:40:07,946 nipype.workflow INFO:\n'
b'\t [Job 0] Completed (nibetaseries_participant_wf.single_subject001_wf.betaseries_
↪wf.betaseries_node).\n'
b'190129-14:40:07,949 nipype.workflow INFO:\n'
b'\t [MultiProc] Running 0 tasks, and 1 jobs ready. Free memory (GB): 7.01/7.01, Free
↪processors: 4/4.\n'
b'190129-14:40:09,949 nipype.workflow INFO:\n'
b'\t [MultiProc] Running 0 tasks, and 3 jobs ready. Free memory (GB): 7.01/7.01, Free
↪processors: 4/4.\n'
b'190129-14:40:09,989 nipype.workflow INFO:\n'
b'\t [Node] Setting-up "_atlas_corr_node0" in "/tmp/tmpuu2igxzh/ds000164/derivatives/
↪work/NiBetaSeries_work/nibetaseries_participant_wf/single_subject001_wf/correlation_
↪wf/a7dc49b58761b73b7aeeeee9d35703cc8ab43895/atlas_corr_node/mapflow/_atlas_corr_
↪node0".\n'
b'190129-14:40:09,990 nipype.workflow INFO:\n'
b'\t [Node] Setting-up "_atlas_corr_node1" in "/tmp/tmpuu2igxzh/ds000164/derivatives/
↪work/NiBetaSeries_work/nibetaseries_participant_wf/single_subject001_wf/correlation_
↪wf/a7dc49b58761b73b7aeeeee9d35703cc8ab43895/atlas_corr_node/mapflow/_atlas_corr_
↪node1".\n'
b'190129-14:40:09,993 nipype.workflow INFO:\n'
b'\t [Node] Running "_atlas_corr_node0" ("nibetaseries.interfaces.nilearn.
↪AtlasConnectivity")\n'
b'190129-14:40:09,994 nipype.workflow INFO:\n'
b'\t [Node] Running "_atlas_corr_node1" ("nibetaseries.interfaces.nilearn.
↪AtlasConnectivity")\n'
b'190129-14:40:09,997 nipype.workflow INFO:\n'
b'\t [Node] Setting-up "_atlas_corr_node2" in "/tmp/tmpuu2igxzh/ds000164/derivatives/
↪work/NiBetaSeries_work/nibetaseries_participant_wf/single_subject001_wf/correlation_
↪wf/a7dc49b58761b73b7aeeeee9d35703cc8ab43895/atlas_corr_node/mapflow/_atlas_corr_
↪node2".\n'
b'190129-14:40:10,0 nipype.workflow INFO:\n'
b'\t [Node] Running "_atlas_corr_node2" ("nibetaseries.interfaces.nilearn.
↪AtlasConnectivity")\n'
b'[NiftiLabelsMasker.fit_transform] loading data from /tmp/tmpuu2igxzh/ds000164/
↪derivatives/data/Schaefer2018_100Parcels_7Networks_order_FSLMNI152_2mm.nii.gz\n'
b'Resampling labels\n'
b'[NiftiLabelsMasker.transform_single_imgs] Loading data from /tmp/tmpuu2igxzh/
↪ds000164/derivatives/work/NiBetaSeries_work/nibetaseries_participant_wf/single_
↪subject001_wf/betaseries_wf/a7dc49b58761b73b7aeeeee9d35703cc8ab43895/betaseries_
↪node/betaseries_trialtyp'\n'
```

```
b'[NiftiLabelsMasker.transform_single_imgs] Extracting region signals\n'
b'[NiftiLabelsMasker.transform_single_imgs] Cleaning extracted signals\n'
b'190129-14:40:10,909 nipype.workflow INFO:\n'
b'\t [Node] Finished "_atlas_corr_node1".\n'
b'[NiftiLabelsMasker.fit_transform] loading data from /tmp/tmpuu2igxzh/ds000164/
↪derivatives/data/Schaefer2018_100Parcels_7Networks_order_FSLMNI152_2mm.nii.gz\n'
b'Resampling labels\n'
b'[NiftiLabelsMasker.transform_single_imgs] Loading data from /tmp/tmpuu2igxzh/
↪ds000164/derivatives/work/NiBetaSeries_work/nibetaseries_participant_wf/single_
↪subject001_wf/betaseries_wf/a7dc49b58761b73b7aeeeee9d35703cc8ab43895/betaseries_
↪node/betaseries_trialtyp\n'
b'[NiftiLabelsMasker.transform_single_imgs] Extracting region signals\n'
b'[NiftiLabelsMasker.transform_single_imgs] Cleaning extracted signals\n'
b'190129-14:40:10,917 nipype.workflow INFO:\n'
b'\t [Node] Finished "_atlas_corr_node0".\n'
b'[NiftiLabelsMasker.fit_transform] loading data from /tmp/tmpuu2igxzh/ds000164/
↪derivatives/data/Schaefer2018_100Parcels_7Networks_order_FSLMNI152_2mm.nii.gz\n'
b'Resampling labels\n'
b'[NiftiLabelsMasker.transform_single_imgs] Loading data from /tmp/tmpuu2igxzh/
↪ds000164/derivatives/work/NiBetaSeries_work/nibetaseries_participant_wf/single_
↪subject001_wf/betaseries_wf/a7dc49b58761b73b7aeeeee9d35703cc8ab43895/betaseries_
↪node/betaseries_trialtyp\n'
b'[NiftiLabelsMasker.transform_single_imgs] Extracting region signals\n'
b'[NiftiLabelsMasker.transform_single_imgs] Cleaning extracted signals\n'
b'190129-14:40:10,919 nipype.workflow INFO:\n'
b'\t [Node] Finished "_atlas_corr_node2".\n'
b'190129-14:40:11,951 nipype.workflow INFO:\n'
b'\t [Job 4] Completed (_atlas_corr_node0).\n'
b'190129-14:40:11,952 nipype.workflow INFO:\n'
b'\t [Job 5] Completed (_atlas_corr_node1).\n'
b'190129-14:40:11,952 nipype.workflow INFO:\n'
b'\t [Job 6] Completed (_atlas_corr_node2).\n'
b'190129-14:40:11,954 nipype.workflow INFO:\n'
b'\t [MultiProc] Running 0 tasks, and 1 jobs ready. Free memory (GB): 7.01/7.01, Free
↪processors: 4/4.\n'
b'190129-14:40:12,0 nipype.workflow INFO:\n'
b'\t [Node] Setting-up "nibetaseries_participant_wf.single_subject001_wf.correlation_
↪wf.atlas_corr_node" in "/tmp/tmpuu2igxzh/ds000164/derivatives/work/NiBetaSeries_
↪work/nibetaseries_participant_wf/single_subject001_wf/correlation_wf/
↪a7dc49b58761b73b7aeeeee9d35703cc8ab43895/atlas_corr_node".\n'
b'190129-14:40:12,3 nipype.workflow INFO:\n'
b'\t [Node] Setting-up "_atlas_corr_node0" in "/tmp/tmpuu2igxzh/ds000164/derivatives/
↪work/NiBetaSeries_work/nibetaseries_participant_wf/single_subject001_wf/correlation_
↪wf/a7dc49b58761b73b7aeeeee9d35703cc8ab43895/atlas_corr_node/mapflow/_atlas_corr_
↪node0".\n'
b'190129-14:40:12,4 nipype.workflow INFO:\n'
b'\t [Node] Cached "_atlas_corr_node0" - collecting precomputed outputs\n'
b'190129-14:40:12,4 nipype.workflow INFO:\n'
b'\t [Node] "_atlas_corr_node0" found cached.\n'
b'190129-14:40:12,5 nipype.workflow INFO:\n'
b'\t [Node] Setting-up "_atlas_corr_node1" in "/tmp/tmpuu2igxzh/ds000164/derivatives/
↪work/NiBetaSeries_work/nibetaseries_participant_wf/single_subject001_wf/correlation_
↪wf/a7dc49b58761b73b7aeeeee9d35703cc8ab43895/atlas_corr_node/mapflow/_atlas_corr_
↪node1".\n'
b'190129-14:40:12,6 nipype.workflow INFO:\n'
b'\t [Node] Cached "_atlas_corr_node1" - collecting precomputed outputs\n'
b'190129-14:40:12,6 nipype.workflow INFO:\n'
```

```
b'\t [Node] "_atlas_corr_node1" found cached.\n'
b'190129-14:40:12,7 nipype.workflow INFO:\n'
b'\t [Node] Setting-up "_atlas_corr_node2" in "/tmp/tmpuu2igxzh/ds000164/derivatives/
↪work/NiBetaSeries_work/nibetaseries_participant_wf/single_subject001_wf/correlation_
↪wf/a7dc49b58761b73b7aeeeee9d35703cc8ab43895/atlas_corr_node/mapflow/_atlas_corr_
↪node2".\n'
b'190129-14:40:12,8 nipype.workflow INFO:\n'
b'\t [Node] Cached "_atlas_corr_node2" - collecting precomputed outputs\n'
b'190129-14:40:12,8 nipype.workflow INFO:\n'
b'\t [Node] "_atlas_corr_node2" found cached.\n'
b'190129-14:40:12,10 nipype.workflow INFO:\n'
b'\t [Node] Finished "nibetaseries_participant_wf.single_subject001_wf.correlation_wf.
↪atlas_corr_node".\n'
b'190129-14:40:13,952 nipype.workflow INFO:\n'
b'\t [Job 1] Completed (nibetaseries_participant_wf.single_subject001_wf.correlation_
↪wf.atlas_corr_node).\n'
b'190129-14:40:13,954 nipype.workflow INFO:\n'
b'\t [MultiProc] Running 0 tasks, and 1 jobs ready. Free memory (GB): 7.01/7.01, Free␣
↪processors: 4/4.\n'
b'190129-14:40:15,955 nipype.workflow INFO:\n'
b'\t [MultiProc] Running 0 tasks, and 3 jobs ready. Free memory (GB): 7.01/7.01, Free␣
↪processors: 4/4.\n'
b'190129-14:40:15,995 nipype.workflow INFO:\n'
b'\t [Node] Setting-up "_rename_matrix_node0" in "/tmp/tmpuu2igxzh/ds000164/
↪derivatives/work/NiBetaSeries_work/nibetaseries_participant_wf/single_subject001_wf/
↪correlation_wf/a7dc49b58761b73b7aeeeee9d35703cc8ab43895/rename_matrix_node/mapflow/_
↪rename_matrix_node0".\n'
b'190129-14:40:15,996 nipype.workflow INFO:\n'
b'\t [Node] Setting-up "_rename_matrix_node1" in "/tmp/tmpuu2igxzh/ds000164/
↪derivatives/work/NiBetaSeries_work/nibetaseries_participant_wf/single_subject001_wf/
↪correlation_wf/a7dc49b58761b73b7aeeeee9d35703cc8ab43895/rename_matrix_node/mapflow/_
↪rename_matrix_node1".\n'
b'190129-14:40:15,997 nipype.workflow INFO:\n'
b'\t [Node] Running "_rename_matrix_node0" ("nipype.interfaces.utility.wrappers.
↪Function")\n'
b'190129-14:40:15,998 nipype.workflow INFO:\n'
b'\t [Node] Setting-up "_rename_matrix_node2" in "/tmp/tmpuu2igxzh/ds000164/
↪derivatives/work/NiBetaSeries_work/nibetaseries_participant_wf/single_subject001_wf/
↪correlation_wf/a7dc49b58761b73b7aeeeee9d35703cc8ab43895/rename_matrix_node/mapflow/_
↪rename_matrix_node2".\n'
b'190129-14:40:15,999 nipype.workflow INFO:\n'
b'\t [Node] Running "_rename_matrix_node1" ("nipype.interfaces.utility.wrappers.
↪Function")\n'
b'190129-14:40:16,0 nipype.workflow INFO:\n'
b'\t [Node] Running "_rename_matrix_node2" ("nipype.interfaces.utility.wrappers.
↪Function")\n'
b'190129-14:40:16,2 nipype.workflow INFO:\n'
b'\t [Node] Finished "_rename_matrix_node0".\n'
b'190129-14:40:16,3 nipype.workflow INFO:\n'
b'\t [Node] Finished "_rename_matrix_node1".\n'
b'190129-14:40:16,5 nipype.workflow INFO:\n'
b'\t [Node] Finished "_rename_matrix_node2".\n'
b'190129-14:40:17,956 nipype.workflow INFO:\n'
b'\t [Job 7] Completed (_rename_matrix_node0).\n'
b'190129-14:40:17,957 nipype.workflow INFO:\n'
b'\t [Job 8] Completed (_rename_matrix_node1).\n'
b'190129-14:40:17,957 nipype.workflow INFO:\n'
```

```
b'\t [Job 9] Completed (_rename_matrix_node2).\n'
b'190129-14:40:17,959 nipype.workflow INFO:\n'
b'\t [MultiProc] Running 0 tasks, and 1 jobs ready. Free memory (GB): 7.01/7.01, Free␣
→processors: 4/4.\n'
b'190129-14:40:17,999 nipype.workflow INFO:\n'
b'\t [Node] Setting-up "nibetaseries_participant_wf.single_subject001_wf.correlation_␣
→wf.rename_matrix_node" in "/tmp/tmpuu2igxzh/ds000164/derivatives/work/NiBetaSeries_␣
→work/nibetaseries_participant_wf/single_subject001_wf/correlation_wf/
→a7dc49b58761b73b7aeeeee9d35703cc8ab43895/rename_matrix_node".\n'
b'190129-14:40:18,2 nipype.workflow INFO:\n'
b'\t [Node] Setting-up "_rename_matrix_node0" in "/tmp/tmpuu2igxzh/ds000164/
→derivatives/work/NiBetaSeries_work/nibetaseries_participant_wf/single_subject001_wf/
→correlation_wf/a7dc49b58761b73b7aeeeee9d35703cc8ab43895/rename_matrix_node/mapflow/_
→rename_matrix_node0".\n'
b'190129-14:40:18,3 nipype.workflow INFO:\n'
b'\t [Node] Cached "_rename_matrix_node0" - collecting precomputed outputs\n'
b'190129-14:40:18,3 nipype.workflow INFO:\n'
b'\t [Node] "_rename_matrix_node0" found cached.\n'
b'190129-14:40:18,4 nipype.workflow INFO:\n'
b'\t [Node] Setting-up "_rename_matrix_node1" in "/tmp/tmpuu2igxzh/ds000164/
→derivatives/work/NiBetaSeries_work/nibetaseries_participant_wf/single_subject001_wf/
→correlation_wf/a7dc49b58761b73b7aeeeee9d35703cc8ab43895/rename_matrix_node/mapflow/_
→rename_matrix_node1".\n'
b'190129-14:40:18,5 nipype.workflow INFO:\n'
b'\t [Node] Cached "_rename_matrix_node1" - collecting precomputed outputs\n'
b'190129-14:40:18,5 nipype.workflow INFO:\n'
b'\t [Node] "_rename_matrix_node1" found cached.\n'
b'190129-14:40:18,6 nipype.workflow INFO:\n'
b'\t [Node] Setting-up "_rename_matrix_node2" in "/tmp/tmpuu2igxzh/ds000164/
→derivatives/work/NiBetaSeries_work/nibetaseries_participant_wf/single_subject001_wf/
→correlation_wf/a7dc49b58761b73b7aeeeee9d35703cc8ab43895/rename_matrix_node/mapflow/_
→rename_matrix_node2".\n'
b'190129-14:40:18,7 nipype.workflow INFO:\n'
b'\t [Node] Cached "_rename_matrix_node2" - collecting precomputed outputs\n'
b'190129-14:40:18,7 nipype.workflow INFO:\n'
b'\t [Node] "_rename_matrix_node2" found cached.\n'
b'190129-14:40:18,9 nipype.workflow INFO:\n'
b'\t [Node] Finished "nibetaseries_participant_wf.single_subject001_wf.correlation_wf.
→rename_matrix_node".\n'
b'190129-14:40:19,958 nipype.workflow INFO:\n'
b'\t [Job 2] Completed (nibetaseries_participant_wf.single_subject001_wf.correlation_
→wf.rename_matrix_node).\n'
b'190129-14:40:19,960 nipype.workflow INFO:\n'
b'\t [MultiProc] Running 0 tasks, and 1 jobs ready. Free memory (GB): 7.01/7.01, Free␣
→processors: 4/4.\n'
b'190129-14:40:21,961 nipype.workflow INFO:\n'
b'\t [MultiProc] Running 0 tasks, and 3 jobs ready. Free memory (GB): 7.01/7.01, Free␣
→processors: 4/4.\n'
b'190129-14:40:22,6 nipype.workflow INFO:\n'
b'\t [Node] Setting-up "_ds_correlation_matrix0" in "/tmp/tmpuu2igxzh/ds000164/
→derivatives/work/NiBetaSeries_work/nibetaseries_participant_wf/single_subject001_wf/
→a7dc49b58761b73b7aeeeee9d35703cc8ab43895/ds_correlation_matrix/mapflow/_ds_
→correlation_matrix0".\n'
b'190129-14:40:22,7 nipype.workflow INFO:\n'
b'\t [Node] Setting-up "_ds_correlation_matrix1" in "/tmp/tmpuu2igxzh/ds000164/
→derivatives/work/NiBetaSeries_work/nibetaseries_participant_wf/single_subject001_wf/
→a7dc49b58761b73b7aeeeee9d35703cc8ab43895/ds_correlation_matrix/mapflow/_ds_
→correlation_matrix1".\n'
```

```
b'190129-14:40:22,8 nipype.workflow INFO:\n'
b'\t [Node] Running "_ds_correlation_matrix0" ("nibetaseries.interfaces.bids.
↪DerivativesDataSink")\n'
b'190129-14:40:22,9 nipype.workflow INFO:\n'
b'\t [Node] Running "_ds_correlation_matrix1" ("nibetaseries.interfaces.bids.
↪DerivativesDataSink")\n'
b'190129-14:40:22,12 nipype.workflow INFO:\n'
b'\t [Node] Finished "_ds_correlation_matrix0".\n'
b'190129-14:40:22,15 nipype.workflow INFO:\n'
b'\t [Node] Finished "_ds_correlation_matrix1".\n'
b'190129-14:40:22,16 nipype.workflow INFO:\n'
b'\t [Node] Setting-up "_ds_correlation_matrix2" in "/tmp/tmpuu2igxzh/ds000164/
↪derivatives/work/NiBetaSeries_work/nibetaseries_participant_wf/single_subject001_wf/
↪a7dc49b58761b73b7aeeeee9d35703cc8ab43895/ds_correlation_matrix/mapflow/_ds_
↪correlation_matrix2".\n'
b'190129-14:40:22,17 nipype.workflow INFO:\n'
b'\t [Node] Running "_ds_correlation_matrix2" ("nibetaseries.interfaces.bids.
↪DerivativesDataSink")\n'
b'190129-14:40:22,21 nipype.workflow INFO:\n'
b'\t [Node] Finished "_ds_correlation_matrix2".\n'
b'190129-14:40:23,962 nipype.workflow INFO:\n'
b'\t [Job 10] Completed (_ds_correlation_matrix0).\n'
b'190129-14:40:23,966 nipype.workflow INFO:\n'
b'\t [Job 11] Completed (_ds_correlation_matrix1).\n'
b'190129-14:40:23,968 nipype.workflow INFO:\n'
b'\t [Job 12] Completed (_ds_correlation_matrix2).\n'
b'190129-14:40:23,970 nipype.workflow INFO:\n'
b'\t [MultiProc] Running 0 tasks, and 1 jobs ready. Free memory (GB): 7.01/7.01, Free
↪processors: 4/4.\n'
b'190129-14:40:24,8 nipype.workflow INFO:\n'
b'\t [Node] Setting-up "nibetaseries_participant_wf.single_subject001_wf.ds_
↪correlation_matrix" in "/tmp/tmpuu2igxzh/ds000164/derivatives/work/NiBetaSeries_
↪work/nibetaseries_participant_wf/single_subject001_wf/
↪a7dc49b58761b73b7aeeeee9d35703cc8ab43895/ds_correlation_matrix".\n'
b'190129-14:40:24,12 nipype.workflow INFO:\n'
b'\t [Node] Setting-up "_ds_correlation_matrix0" in "/tmp/tmpuu2igxzh/ds000164/
↪derivatives/work/NiBetaSeries_work/nibetaseries_participant_wf/single_subject001_wf/
↪a7dc49b58761b73b7aeeeee9d35703cc8ab43895/ds_correlation_matrix/mapflow/_ds_
↪correlation_matrix0".\n'
b'190129-14:40:24,19 nipype.workflow INFO:\n'
b'\t [Node] Running "_ds_correlation_matrix0" ("nibetaseries.interfaces.bids.
↪DerivativesDataSink")\n'
b'190129-14:40:24,22 nipype.workflow INFO:\n'
b'\t [Node] Finished "_ds_correlation_matrix0".\n'
b'190129-14:40:24,23 nipype.workflow INFO:\n'
b'\t [Node] Setting-up "_ds_correlation_matrix1" in "/tmp/tmpuu2igxzh/ds000164/
↪derivatives/work/NiBetaSeries_work/nibetaseries_participant_wf/single_subject001_wf/
↪a7dc49b58761b73b7aeeeee9d35703cc8ab43895/ds_correlation_matrix/mapflow/_ds_
↪correlation_matrix1".\n'
b'190129-14:40:24,26 nipype.workflow INFO:\n'
b'\t [Node] Running "_ds_correlation_matrix1" ("nibetaseries.interfaces.bids.
↪DerivativesDataSink")\n'
b'190129-14:40:24,30 nipype.workflow INFO:\n'
b'\t [Node] Finished "_ds_correlation_matrix1".\n'
b'190129-14:40:24,31 nipype.workflow INFO:\n'
b'\t [Node] Setting-up "_ds_correlation_matrix2" in "/tmp/tmpuu2igxzh/ds000164/
↪derivatives/work/NiBetaSeries_work/nibetaseries_participant_wf/single_subject001_wf/
↪a7dc49b58761b73b7aeeeee9d35703cc8ab43895/ds_correlation_matrix/mapflow/_ds_
↪correlation_matrix2".\n'
```

```
b'190129-14:40:24,34 nipype.workflow INFO:\n'
b'\t [Node] Running "_ds_correlation_matrix2" ("nibetaseries.interfaces.bids.
↪DerivativesDataSink")\n'
b'190129-14:40:24,38 nipype.workflow INFO:\n'
b'\t [Node] Finished "_ds_correlation_matrix2".\n'
b'190129-14:40:24,40 nipype.workflow INFO:\n'
b'\t [Node] Finished "nibetaseries_participant_wf.single_subject001_wf.ds_correlation_
↪matrix".\n'
b'190129-14:40:25,964 nipype.workflow INFO:\n'
b'\t [Job 3] Completed (nibetaseries_participant_wf.single_subject001_wf.ds_
↪correlation_matrix).\n'
b'190129-14:40:25,966 nipype.workflow INFO:\n'
b'\t [MultiProc] Running 0 tasks, and 0 jobs ready. Free memory (GB): 7.01/7.01, Free␣
↪processors: 4/4.\n'
b'/home/docs/checkouts/readthedocs.org/user_builds/nibetaseries/envs/v0.2.3/lib/
↪python3.5/importlib/_bootstrap.py:222: RuntimeWarning: numpy.dtype size changed,␣
↪may indicate binary incompatibility. Expected 96, got 88\n'
b'  return f(*args, **kwds)\n'
b'/home/docs/checkouts/readthedocs.org/user_builds/nibetaseries/envs/v0.2.3/lib/
↪python3.5/importlib/_bootstrap.py:222: RuntimeWarning: numpy.dtype size changed,␣
↪may indicate binary incompatibility. Expected 96, got 88\n'
b'  return f(*args, **kwds)\n'
b'/home/docs/checkouts/readthedocs.org/user_builds/nibetaseries/envs/v0.2.3/lib/
↪python3.5/importlib/_bootstrap.py:222: RuntimeWarning: numpy.dtype size changed,␣
↪may indicate binary incompatibility. Expected 96, got 88\n'
b'  return f(*args, **kwds)\n'
b'/home/docs/checkouts/readthedocs.org/user_builds/nibetaseries/envs/v0.2.3/lib/
↪python3.5/site-packages/nipype/pipeline/engine/utils.py:307: DeprecationWarning:␣
↪use "HasTraits.trait_set" instead\n'
b'  result.outputs.set(**modify_paths(tosave, relative=True, basedir=cwd))\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
```

```
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 2 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 2 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
```

```
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
```

```
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 1 seconds\n'
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
```

```
b'\n'
b'Computing run 1 out of 1 runs (go take a coffee, a big one)\n'
b'\n'
b'Computation of 1 runs done in 0 seconds\n'
b'\n'
b'/home/docs/checkouts/readthedocs.org/user_builds/nibetaseries/envs/v0.2.3/lib/
↪python3.5/site-packages/nipype/pipeline/engine/utils.py:307: DeprecationWarning:
↪use "HasTraits.trait_set" instead\n'
b'  result.outputs.set(**modify_paths(tosave, relative=True, basedir=cwd))\n'
b'/home/docs/checkouts/readthedocs.org/user_builds/nibetaseries/envs/v0.2.3/lib/
↪python3.5/importlib/_bootstrap.py:222: RuntimeWarning: numpy.dtype size changed,
↪may indicate binary incompatibility. Expected 96, got 88\n'
b'  return f(*args, **kwds)\n'
b'/home/docs/checkouts/readthedocs.org/user_builds/nibetaseries/envs/v0.2.3/lib/
↪python3.5/importlib/_bootstrap.py:222: RuntimeWarning: numpy.dtype size changed,
↪may indicate binary incompatibility. Expected 96, got 88\n'
b'  return f(*args, **kwds)\n'
b'/home/docs/checkouts/readthedocs.org/user_builds/nibetaseries/envs/v0.2.3/lib/
↪python3.5/importlib/_bootstrap.py:222: RuntimeWarning: numpy.dtype size changed,
↪may indicate binary incompatibility. Expected 96, got 88\n'
b'  return f(*args, **kwds)\n'
b'/home/docs/checkouts/readthedocs.org/user_builds/nibetaseries/envs/v0.2.3/lib/
↪python3.5/site-packages/nipype/pipeline/engine/utils.py:307: DeprecationWarning:
↪use "HasTraits.trait_set" instead\n'
b'  result.outputs.set(**modify_paths(tosave, relative=True, basedir=cwd))\n'
b'/home/docs/checkouts/readthedocs.org/user_builds/nibetaseries/envs/v0.2.3/lib/
↪python3.5/importlib/_bootstrap.py:222: RuntimeWarning: numpy.dtype size changed,
↪may indicate binary incompatibility. Expected 96, got 88\n'
b'  return f(*args, **kwds)\n'
b'/home/docs/checkouts/readthedocs.org/user_builds/nibetaseries/envs/v0.2.3/lib/
↪python3.5/importlib/_bootstrap.py:222: RuntimeWarning: numpy.dtype size changed,
↪may indicate binary incompatibility. Expected 96, got 88\n'
b'  return f(*args, **kwds)\n'
b'/home/docs/checkouts/readthedocs.org/user_builds/nibetaseries/envs/v0.2.3/lib/
↪python3.5/importlib/_bootstrap.py:222: RuntimeWarning: numpy.dtype size changed,
↪may indicate binary incompatibility. Expected 96, got 88\n'
b'  return f(*args, **kwds)\n'
b'/home/docs/checkouts/readthedocs.org/user_builds/nibetaseries/envs/v0.2.3/lib/
↪python3.5/site-packages/nipype/pipeline/engine/utils.py:307: DeprecationWarning:
↪use "HasTraits.trait_set" instead\n'
b'  result.outputs.set(**modify_paths(tosave, relative=True, basedir=cwd))\n'
```

5.8 Observe generated outputs

```
list_files(data_dir)
```

Out:

```
tmpuu2igxzh/
    ds000164/
        T1w.json
        README
        task-stroop_bold.json
        dataset_description.json
```

```
        task-stroop_events.json
        CHANGES
        derivatives/
            NiBetaSeries/
                nibetaseries/
                    sub-001/
                        func/
                            sub-001_task-stroop_bold_space-MNI152NLin2009cAsym_
→preproc_trialtype-congruent_matrix.tsv
                            sub-001_task-stroop_bold_space-MNI152NLin2009cAsym_
→preproc_trialtype-incongruent_matrix.tsv
                            sub-001_task-stroop_bold_space-MNI152NLin2009cAsym_
→preproc_trialtype-neutral_matrix.tsv
                logs/
        data/
            Schaefer2018_100Parcels_7Networks_order.tsv
            Schaefer2018_100Parcels_7Networks_order.txt
            Schaefer2018_100Parcels_7Networks_order_FSLMNI152_2mm.nii.gz
        work/
            NiBetaSeries_work/
                nibetaseries_participant_wf/
                    graph1.json
                    d3.js
                    index.html
                    graph.json
                    single_subject001_wf/
                        a7dc49b58761b73b7aeeeee9d35703cc8ab43895/
                            ds_correlation_matrix/
                                _0x71f4d002e2c91ce513522bea20d623bd.json
                                _node.pklz
                                result_ds_correlation_matrix.pklz
                                _inputs.pklz
                                _report/
                                    report.rst
                                mapflow/
                                    _ds_correlation_matrix0/
                                        _node.pklz
                                        result__ds_correlation_matrix0.pklz
                                        _inputs.pklz
                                        _0x88f8d315f589eeaac7d70cb68d8a014e.json
                                        _report/
                                            report.rst
                                    _ds_correlation_matrix1/
                                        _node.pklz
                                        _inputs.pklz
                                        _0x3f0a8f030067a681159da302b0d8678b.json
                                        result__ds_correlation_matrix1.pklz
                                        _report/
                                            report.rst
                                    _ds_correlation_matrix2/
                                        _node.pklz
                                        _inputs.pklz
                                        _0xccb9f070c4554dd0dd86356895be3dff.json
                                        result__ds_correlation_matrix2.pklz
                                        _report/
                                            report.rst
                            correlation_wf/
```

(continued from previous page)

```
                                    a7dc49b58761b73b7aeeeee9d35703cc8ab43895/
                                        atlas_corr_node/
                                            _node.pklz
                                            result_atlas_corr_node.pklz
                                            _inputs.pklz
                                            _0xe429a02dadec120752df8309cfec5434.json
                                            _report/
                                                report.rst
                                            mapflow/
                                                _atlas_corr_node2/
                                                    fisher_z_correlation.tsv
                                                    _node.pklz
                                                    _inputs.pklz
                                                    _0xf833ae8eaec6073d5d1f2f9f3d78038b.
→json
                                                    result__atlas_corr_node2.pklz
                                                    _report/
                                                        report.rst
                                                _atlas_corr_node0/
                                                    fisher_z_correlation.tsv
                                                    _node.pklz
                                                    _inputs.pklz
                                                    result__atlas_corr_node0.pklz
                                                    _0x0027e94ede3403f7800288af1d5de310.
→json
                                                    _report/
                                                        report.rst
                                                _atlas_corr_node1/
                                                    fisher_z_correlation.tsv
                                                    _node.pklz
                                                    _inputs.pklz
                                                    _0x9e7d17fcd6da69a6071a10aa34bfd767.
→json
                                                    result__atlas_corr_node1.pklz
                                                    _report/
                                                        report.rst
                                        rename_matrix_node/
                                            _node.pklz
                                            result_rename_matrix_node.pklz
                                            _inputs.pklz
                                            _0x5e36fb501fbee11912d6c173ac4f0573.json
                                            _report/
                                                report.rst
                                            mapflow/
                                                _rename_matrix_node1/
                                                    _node.pklz
                                                    _inputs.pklz
                                                    result__rename_matrix_node1.pklz
                                                    correlation-matrix_trialtype-
→incongruent.tsv
                                                    _0x09f938ebbd08bd01b0fdb9735da975cf.
→json
                                                    _report/
                                                        report.rst
                                                _rename_matrix_node2/
                                                    _node.pklz
                                                    _inputs.pklz
```

(continues on next page)

```
                                                result__rename_matrix_node2.pklz
                                                correlation-matrix_trialtype-neutral.
↪tsv
                                                _0x09b2b427be3a30edca3d550c457949d7.
↪json
                                                _report/
                                                    report.rst
                                        _rename_matrix_node0/
                                                _node.pklz
                                                _inputs.pklz
                                                result__rename_matrix_node0.pklz
                                                _0xab741b49debbe7844f004f5dabf45fd8.
↪json
                                                correlation-matrix_trialtype-
↪congruent.tsv
                                                _report/
                                                    report.rst
                            betaseries_wf/
                                a7dc49b58761b73b7aeeeee9d35703cc8ab43895/
                                    betaseries_node/
                                        _node.pklz
                                        _inputs.pklz
                                        betaseries_trialtype-neutral.nii.gz
                                        betaseries_trialtype-congruent.nii.gz
                                        betaseries_trialtype-incongruent.nii.gz
                                        result_betaseries_node.pklz
                                        _0xaeb6e1e256ee0af78f95368119ba2faa.json
                                        _report/
                                            report.rst
            fmriprep/
                sub-001/
                    func/
                        sub-001_task-stroop_bold_space-MNI152NLin2009cAsym_brainmask.
↪nii.gz
                        sub-001_task-stroop_bold_confounds.tsv
                        sub-001_task-stroop_bold_space-MNI152NLin2009cAsym_preproc.
↪nii.gz
        sub-001/
            anat/
                sub-001_T1w.nii.gz
            func/
                sub-001_task-stroop_bold.nii.gz
                sub-001_task-stroop_events.tsv
```

5.9 Collect results

```
corr_mat_path = os.path.join(out_dir, "NiBetaSeries", "nibetaseries", "sub-001", "func
↪")
trial_types = ['congruent', 'incongruent', 'neutral']
filename_template = "sub-001_task-stroop_bold_space-MNI152NLin2009cAsym_preproc_
↪trialtype-{trial_type}_matrix.tsv"
pd_dict = {}
for trial_type in trial_types:
    file_path = os.path.join(corr_mat_path, filename_template.format(trial_type=trial_
↪type))
```

```
    pd_dict[trial_type] = pd.read_csv(file_path, sep='\t', na_values="n/a", index_
↪col=0)
# display example matrix
print(pd_dict[trial_type].head())
```

Out:

```
LH_Vis_1   LH_Vis_2  ...   RH_Default_PCC_1   RH_Default_PCC_2
LH_Vis_1       NaN  0.092135  ...           0.095624           0.016799
LH_Vis_2  0.092135       NaN  ...          -0.119613          -0.007679
LH_Vis_3 -0.003990  0.216346  ...           0.202673           0.177828
LH_Vis_4  0.075498 -0.088788  ...          -0.019256          -0.034034
LH_Vis_5  0.314494  0.354525  ...          -0.235334           0.032317

[5 rows x 100 columns]
```

5.10 Graph the results

```python
fig, axes = plt.subplots(nrows=3, ncols=1, sharex=True, sharey=True, figsize=(10, 30),
                         gridspec_kw={'wspace': 0.025, 'hspace': 0.075})

cbar_ax = fig.add_axes([.91, .3, .03, .4])
r = 0
for trial_type, df in pd_dict.items():
    g = sns.heatmap(df, ax=axes[r], vmin=-.5, vmax=1., square=True,
                    cbar=True, cbar_ax=cbar_ax)
    axes[r].set_title(trial_type)
    # iterate over rows
    r += 1
plt.tight_layout()
```
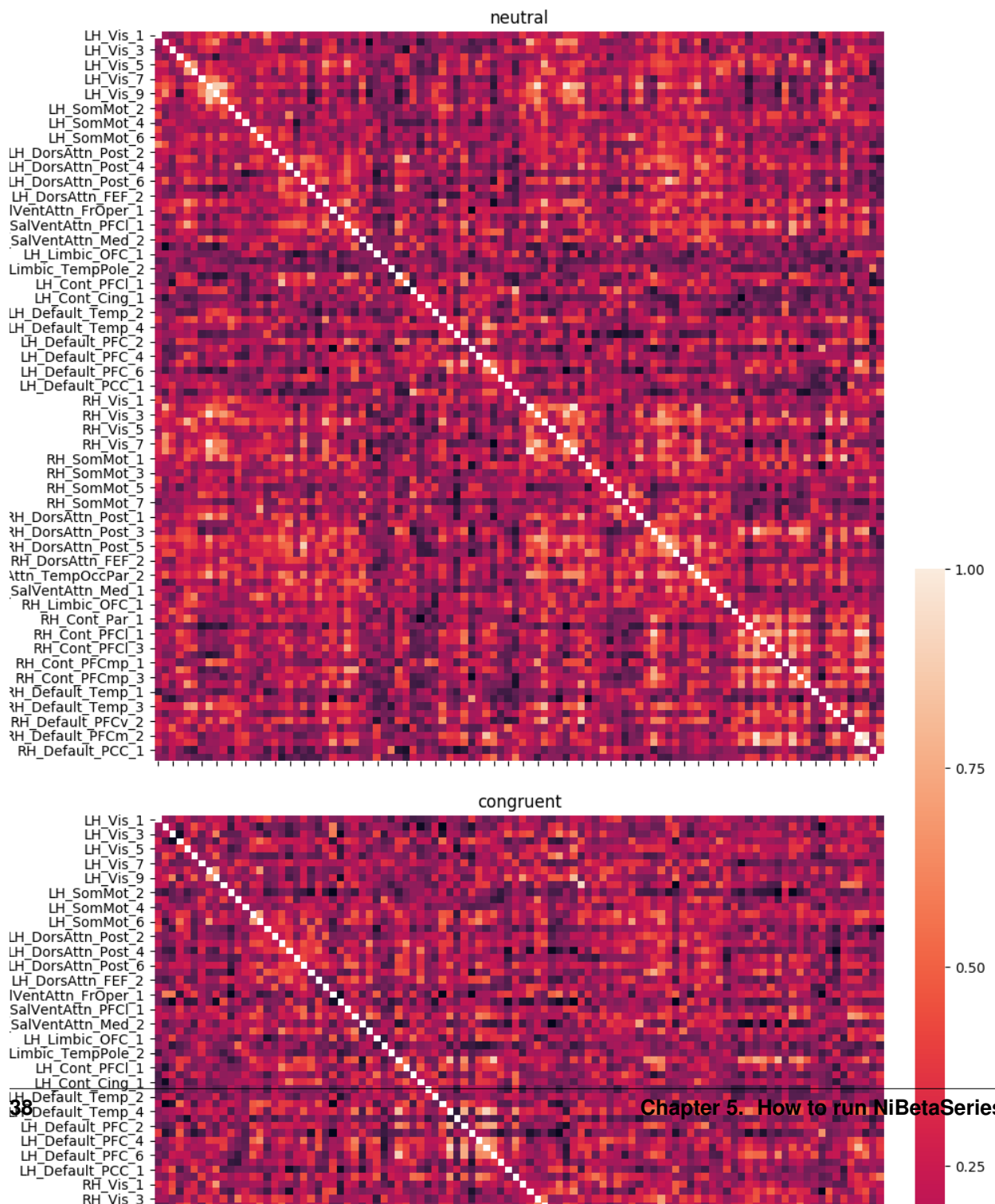
Total running time of the script: (2 minutes 34.309 seconds)

Workflows

6.1 Participant Workflow

The general workflow for a participant models the betaseries for each trial type for each bold file associated with the participant. Then betas within a region of interest are based off a parcellation are averaged together. This occurs as many times as there are trials for that particular trial type, resulting in a psuedo-timeseries (e.g. each point in "time" represents an occurrence of that trial). All the psuedo time-series within a trial type are correlated with each other, resulting in a final correlation (adjacency) matrix.

6.2 BetaSeries Workflow

The bold file is temporally filtered by nilearn (high pass and/or low pass) before being passed into nistats for modelling by least squares separate.

6.3 Correlation Workflow

The betaseries file has signal averaged across trials within a region defined by an atlas parcellation. After signal extraction has occurred for all regions, the signals are all correlated with each other to generate a correlation matrix.

input_node (utility)

betaseries_node (nistats)

output_node (utility)

input_node (utility)

atlas_corr_node (nilearn)

rename_matrix_node (utility)

output_node (utility)
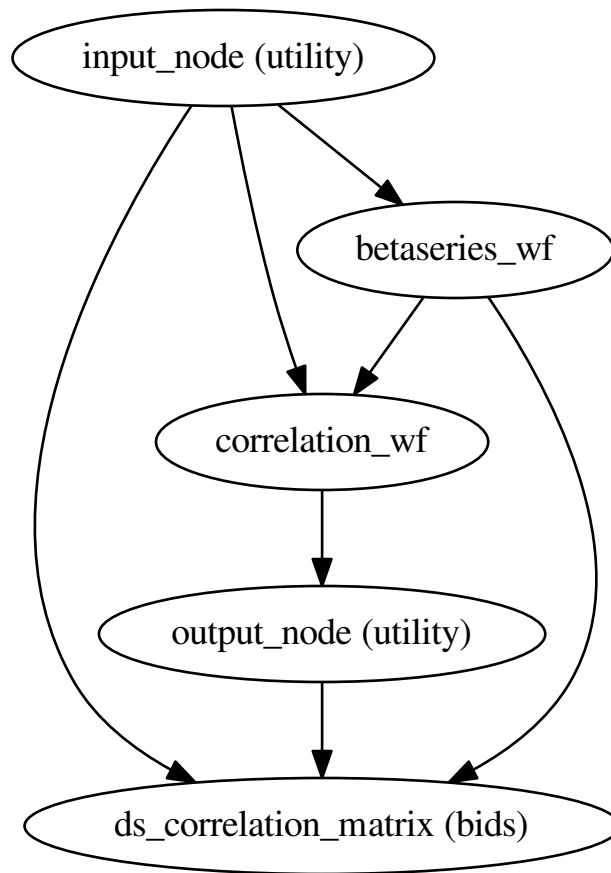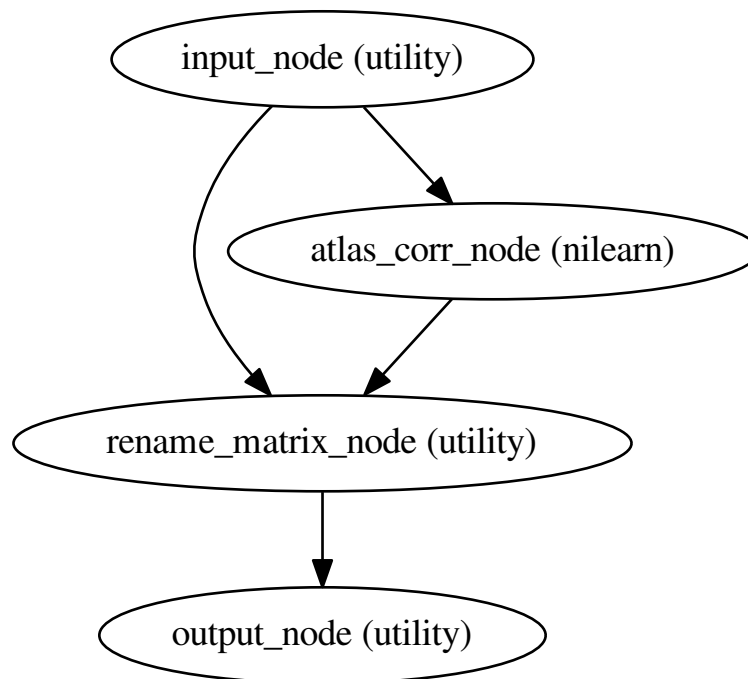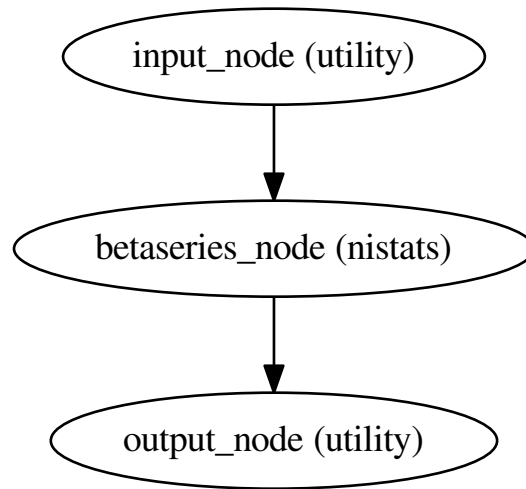
Reference

7.1 nibetaseries

Contributing to NiBetaSeries

Welcome to the NiBetaSeries repository!

We're so excited you're here and want to contribute.

NiBetaSeries calculates betaseries correlations using python! We hope that these guidelines are designed to make it as easy as possible to contribute to NiBetaSeries and the broader Brain Imaging Data Structure (BIDS) community. If you have any questions that aren't discussed below, please let us know through one of the many ways to *get in touch*.

8.1 Table of contents

Been here before? Already know what you're looking for in this guide? Jump to the following sections:

- *Joining the BIDS community*
- *Check out the BIDS Starter Kit*
- *Get in touch*
- *Contributing through GitHub*
- *Where to start: wiki, code and templates*
- *Where to start: issue labels*
- *Make a change with a pull request*
- *Example pull request*
- *Recognizing contributions*

8.2 Joining the BIDS community

BIDS - the Brain Imaging Data Structure - is a growing community of neuroimaging enthusiasts, and we want to make our resources accessible to and engaging for as many researchers as possible. NiBetaSeries will hopefully play a small

part towards introducing people to BIDS and help the broader community.

We therefore require that all contributions **adhere to our** Code of Conduct.

How do you know that you're a member of the BIDS community? You're here! You know that BIDS exists! You're officially a member of the community. It's THAT easy! Welcome!

8.3 Check out the BIDS Starter Kit

If you're new to BIDS make sure to check out the amazing BIDS Starter Kit.

8.4 Get in touch

There are lots of ways to get in touch with the team maintaining NiBetaSeries if you have general questions about the BIDS ecosystem. For specific questions about NiBetaSeries, please see the practical guide (Currently the main contact is James Kent)

- Message jdkent on the brainhack slack
- Click here for an invite to the slack workspace.
- The BIDS mailing list
- **Via the** Neurostars forum**.**
    - **This is our preferred way to answer questions so that others who have similar questions can benefit too!** Even if your question is not well-defined, just post what you have so far and we will be able to point you in the right direction!

8.5 Contributing through GitHub

git is a really useful tool for version control. GitHub sits on top of git and supports collaborative and distributed working.

We know that it can be daunting to start using git and GitHub if you haven't worked with them in the past, but the NibetaSeries maintainer(s) are here to help you figure out any of the jargon or confusing instructions you encounter!

In order to contribute via GitHub you'll need to set up a free account and sign in. Here are some instructions to help you get going. Remember that you can ask us any questions you need to along the way.

8.6 Writing in markdown

GitHub has a helpful page on getting started with writing and formatting on GitHub.

Most of the writing that you'll do in github will be in Markdown. You can think of Markdown as a few little symbols around your text that will allow GitHub to render the text with a little bit of formatting. For example you could write words as bold (`**bold**`), or in italics (`*italics*`), or as a link to another webpage.

8.7 Writing in ReStructuredText

This file and the rest of the documentation files in this project are written using ReStructuredText. This is another markup language that interfaces with sphinx, a documentation generator. Sphinx is used on ReadTheDocs, a documentation hosting service. Putting it all together, ReadTheDocs is an online documentation hosting service that uses sphinx, and sphinx is a documentation generation service that uses ReStructuredText to format the content. What a mouthfull!

8.8 Where to start: issue labels

The list of labels for current issues can be found here and includes:

- If you feel that you can contribute to one of these issues, we especially encourage you to do so!

- There are *lots of ways to ask questions* but opening an issue is a great way to start a conversation and get your answer. Ideally, we'll close it out by adding the answer to the docs!

- *These issues are particularly appropriate if it is your first contribution to NiBetaSeries, or to GitHub overall.*

  If you're not sure about how to go about contributing, these are good places to start. You'll be mentored through the process by the maintainers team. If you're a seasoned contributor, please select a different issue to work from and keep these available for the newer and potentially more anxious team members.

- If you want to ask for something new, please try to make sure that your request is distinct from any others that are already in the queue (or part of NiBetaSeries!). If you find one that's similar but there are subtle differences please reference the other enhancement in your issue.

- These are usually really great issues to help out with: our goal is to make it easy to understand BIDS without having to ask anyone any questions! Documentation is the ultimate solution

- *These issues are reporting a problem or a mistake in the project.*

  The more details you can provide the better! If you know how to fix the bug, please open an issue first and then *submit a pull request*

  We like to model best practice, so NiBetaSeries itself is managed through these issues. We may occasionally have some to coordinate some logistics.

8.9 Making a change with a pull request

We appreciate all contributions to NiBetaSeries. **THANK YOU** for helping us build this useful resource.

Remember that if you're adding information to the *wiki* you **don't need to submit a pull request**. You can just log into GitHub, navigate to the wiki and click the **edit** button.

If you're updating the *code*, the following steps are a guide to help you contribute in a way that will be easy for everyone to review and accept with ease.

8.9.1 1. Comment on an existing issue or open a new issue referencing your addition

This allows other members of the NiBetaSeries team to confirm that you aren't overlapping with work that's currently underway and that everyone is on the same page with the goal of the work you're going to carry out.

This blog is a nice explanation of why putting this work in up front is so useful to everyone involved.

8.9.2 2. Fork the NiBetaSeries repository to your profile

This is now your own unique copy of NiBetaSeries. Changes here won't affect anyone else's work, so it's a safe space to explore edits to the code!

Make sure to keep your fork up to date with the master repository, otherwise you can end up with lots of dreaded merge conflicts.

8.9.3 3. Clone your forked NiBetaSeries to your work machine

Now that you have your own repository to explore you should clone it to your work machine so you can easily edit the files:

```
# clone the repository
git clone https://github.com/YOUR-USERNAME/NiBetaSeries
# change directories into NiBetaSeries
cd NiBetaSeries
# add the upstream repository (i.e. https://github.com/HBClab/NiBetaSeries)
git remote add upstream https://github.com/HBClab/NiBetaSeries
```

8.9.4 4. Make the changes you've discussed

Try to keep the changes focused. If you submit a large amount of work in all in one go it will be much more work for whomever is reviewing your pull request. Help them help you

This project requires you to "branch out" and make new branch and a new issue to go with it if the issue doesn't already exist.

Example:

```
# create the branch on which you will make your issues
git checkout -b your_issue_branch
```

8.9.5 5. Run the tests

When you're done making changes, run all the checks, doc builder and spell checker with tox. First you will install all the development requirements for the project with pip:

```
pip install requirements-dev.txt
```

Then you can run tox by simply typing:

```
tox
```

If the checks fail and you know what went wrong, make the change and run tox again. If you are not sure what the error is, go ahead to step 6.

Note: tox doesn't work on everyone's machine, so don't worry about getting the tests working on your machine.

8.9.6 6. Add/Commit/Push the changes to the NiBetaSeries repository

Once you've made the changes on your branch you are ready to 1) add the files to be tracked by git 2) commit the files to take a snapshot of the branch, and 3) push the changes to your forked repository. You can do complete the add/commit/push process following this github help page.

8.9.7 7. Submit a pull request

A member of the NiBetaSeries team will review your changes to confirm that they can be merged into the main codebase.

A review will probably consist of a few questions to help clarify the work you've done. Keep an eye on your github notifications and be prepared to join in that conversation.

You can update your fork of the NiBetaSeries repository and the pull request will automatically update with those changes. You don't need to submit a new pull request when you make a change in response to a review.

GitHub has a nice introduction to the pull request workflow, but please *get in touch* if you have any questions.

8.9.8 NiBetaSeries coding style guide

Whenever possible, instances of Nodes and Workflows should use the same names as the variables they are assigned to. This makes it easier to relate the content of the working directory to the code that generated it when debugging.

Workflow variables should end in *_wf* to indicate that they refer to Workflows and not Nodes. For instance, a workflow whose basename is *myworkflow* might be defined as follows:

```python
from nipype.pipeline import engine as pe

myworkflow_wf = pe.Workflow(name='myworkflow_wf')
```

If a workflow is generated by a function, the name of the function should take the form *init_<basename>_wf*:

```python
def init_myworkflow_wf(name='myworkflow_wf'):
    workflow = pe.Workflow(name=name)
    ...
    return workflow

myworkflow_wf = init_workflow_wf(name='myworkflow_wf')
```

If multiple instances of the same workflow might be instantiated in the same namespace, the workflow names and variables should include either a numeric identifier or a one-word description, such as:

```python
myworkflow0_wf = init_workflow_wf(name='myworkflow0_wf')
myworkflow1_wf = init_workflow_wf(name='myworkflow1_wf')

# or

myworkflow_lh_wf = init_workflow_wf(name='myworkflow_lh_wf')
myworkflow_rh_wf = init_workflow_wf(name='myworkflow_rh_wf')
```

8.10 Recognizing contributions

BIDS follows the all-contributors specification, so we welcome and recognize all contributions from documentation to testing to code development. You can see a list of current contributors in the BIDS specification.

8.11 Thank you!

You're awesome.

— *Based on contributing guidelines from the* STEMMRoleModels *project.*

Authors

- James Kent - https://github.com/jdkent

0.2.3 (January 29, 2019)

Various documentation and testing changes. We will be using readthedocs going forward and not doctr.

- [FIX] Remove high_pass references from documentation (#90) @RaginSagan
- [FIX] Update betaseries.rst (#91) @ilkayisik
- [ENH] autogenerate test data (#93) @jdkent
- [FIX] add codecov back into testing (#94) @jdkent
- [FIX] refactor dependencies (#95) @jdkent
- [ENH] add example (#99) @jdkent
- [FIX] first pass at configuring doctr (#100) @jdkent
- [FIX] configure doctr (#101) @jdkent
- [FIX] track version with docs (#102) @jdkent
- [ENH] add sphinx versioning (#104) @jdkent
- [FIX] first pass at simplifying example (#106) @jdkent
- [FIX] add master back in to docs (#107) @jdkent
- [MAINT] use readthedocs (#109) @jdkent
- [DOC] add explicit download instruction (#112) @jdkent
- [FIX] add graphviz as dependency for building docs (#115) @jdkent
- [FIX] remove redundant/irrelevant doc building options (#116) @jdkent
- [DOC] fix links in docs (#114) @PeerHerholz
- [FIX,MAINT] rm 3.4 and test add 3.7 (#121) @jdkent
- [FIX] pybids link (#120) @PeerHerholz
- [FIX] syntax links (#119) @PeerHerholz

CHAPTER 11

0.2.2 (November 15, 2018)

Quick bug fixes, one related to updating the nipype dependency to a newer version (1.1.5)

- [ENH] add nthreads option and make multiproc the default (#81) @jdkent
- [FIX] add missing comma in hrf_models (#83) @jdkent

0.2.1 (November 13, 2018)

Large thanks to everyone at neurohackademy that helped make this a reality. This release is still a bit premature because I'm testing out my workflow for making releases.

- [ENH] Add link to Zenodo DOI (#57) @kdestasio
- [ENH] run versioneer install (#60) @jdkent
- [FIX] connect derivative outputs (#61) @jdkent
- [FIX] add CODEOWNERS file (#63) @jdkent
- [FIX] Fix pull request template (#65) @kristianeschenburg
- [ENH]Update CONTRIBUTING.rst (#66) @PeerHerholz
- [FIX] ignore sourcedata and derivatives directories in layout (#69) @jdkent
- [DOC] Added zenodo file (#70) @ctoroserey
- [FIX] file logic (#71) @jdkent
- [FIX] confound removal (#72) @jdkent
- [FIX] Find metadata (#74) @jdkent
- [FIX] various fixes for a real dataset (#75) @jdkent
- [ENH] allow confounds to be none (#76) @jdkent
- [ENH] Reword docs (#77) @jdkent
- [TST] Add more tests (#78) @jdkent
- [MGT] simplify and create deployment (#79) @jdkent

0.2.0 (November 13, 2018)

- [MGT] simplify and create deployment (#79)

- [TST] Add more tests (#78)

- [ENH] Reword docs (#77)

- [ENH]: allow confounds to be none (#76)

- various fixes for a real dataset (#75)

- [FIX]: Find metadata (#74)

- [FIX] confound removal (#72)

- [WIP, FIX]: file logic (#71)

- [DOC] Added zenodo file (#70)

- [FIX]: ignore sourcedata and derivatives directories in layout (#69)

- Update CONTRIBUTING.rst (#66)

- Fix pull request template (#65)

- FIX: add CODEOWNERS file (#63)

- FIX: connect derivative outputs (#61)

- run versioneer install (#60)

- Fix issue #29: Add link to Zenodo DOI (#57)

- Fix issue #45: conform colors of labels (#56)

- fix links in readme.rst (#55)

- Added code of conduct (#53)

- Add link to contributing in README (#52)

- removed acknowledgments section of pull request template (#50)

- [TST]: Add functional test (#49)

- [FIX]: remove references to bootstrap (#48)

- FIX: test remove base .travis.yml (#47)

- removed data directory (#40)

- Add pull request template (#41)

- Update issue templates (#44)

- Update contributing (#43)

- README (where's the beef?) (#37)

- change jdkent to HBClab (#38)

- [FIX]: pass tests (#14)

- [ENH]: improve docs (#13)

- add documentation (#11)

- FIX: add graph (#10)

- Refactor NiBetaSeries (#9)

- Refactor (#2)

0.1.0 (2018-06-08)

- First release on PyPI.

CHAPTER 15

Indices and tables

- genindex
- modindex
- search

Bibliography

[Rissman2004] https://www.ncbi.nlm.nih.gov/pubmed/15488425

[Mumford2012] https://www.ncbi.nlm.nih.gov/pubmed/21924359

[Cisler2014] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4019671/

[Gottlich2015] https://www.frontiersin.org/articles/10.3389/fnsys.2015.00126/full

[Abdulrahman2016] https://www.ncbi.nlm.nih.gov/pubmed/26549299

Python Module Index

n

nibetaseries, 45

Index

N
nibetaseries (module), 45