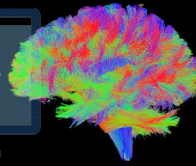
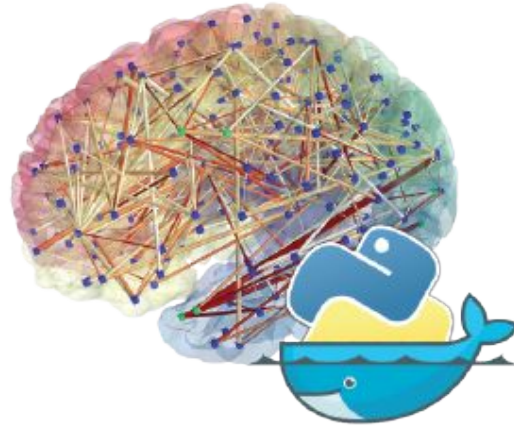


**Brain
Hack**



GLOBAL 20
21



Contributing to Connectome Mapper 3

TEAM



Jonathan Wirsich, Stefan Dvoretsekii, Anil Tuncel,
Jakub Jancovic, Mikkel Schöttner, Sébastien Tourbier

Python wrappers for the BIDS App (Anil and Jakub)

ENH: Python wrapper for docker run command #112

[Open](#) 4 tasks done · sebastientourbier opened this issue 8 hours ago · 0 comments

sebastientourbier commented 8 hours ago · edited · Member

- ✓ Creation of `connectomemapper3_docker.py` in `cmp.cli`.
- ✓ Edit `setup.py` and add `connectomemapper3_docker.py` to the list of scripts to be installed.
- ✓ Add example of usage in the docs
- ✓ Change one of the test in `.circleci/config.yml` to call `connectomemapper3_docker.py` instead of `docker run ...`

kuba-fidel mentioned this issue 23 minutes ago

ENH: add docker CLI to wrap calls to the Docker BIDS

ENH: Python wrapper for singularity run command. #113

[Open](#) 4 tasks done · sebastientourbier opened this issue 8 hours ago · 0 comments · May be fixed by #109

sebastientourbier commented 8 hours ago · edited · Member

- ✓ Creation of `connectomemapper3_singularity.py` in `cmp.cli`.
- ✓ Edit `setup.py` and add `connectomemapper3_singularity.py` to the list of scripts to be installed.
- ✓ Add example of usage in the docs
- ✓ Change one of the test in `.circleci/config.yml` to call `connectomemapper3_singularity` instead of `docker run ...`

Participant Level Analysis

With the wrappers

When you run `connectomemapper_docker`, it will generate a Docker command line for you, print it out for reporting purposes, and then execute it without further action needed, e.g.:

```
$ connectomemapper_docker \  
  /bids_dir /output_dir participant --participant_label 01 --session_label 01 \  
  --docker_image /tmp/cache/connectomemapper3.simg \  
  --anat_pipeline_config /config/ref_anatomical_config_2.ini \  
  --fs_license /bids_dir/code/license.txt
```

When you run `connectomemapper_singularity`, it will generate a Singularity command line for you, print it out for reporting purposes, and then execute it without further action needed, e.g.:

```
$ connectomemapper_singularity \  
  /home/localadmin/data/ds001 /media/localadmin/data/ds001/derivatives \  
  participant --participant_label 01 \  
  --singularity_image /tmp/cache/connectomemapper3.simg
```

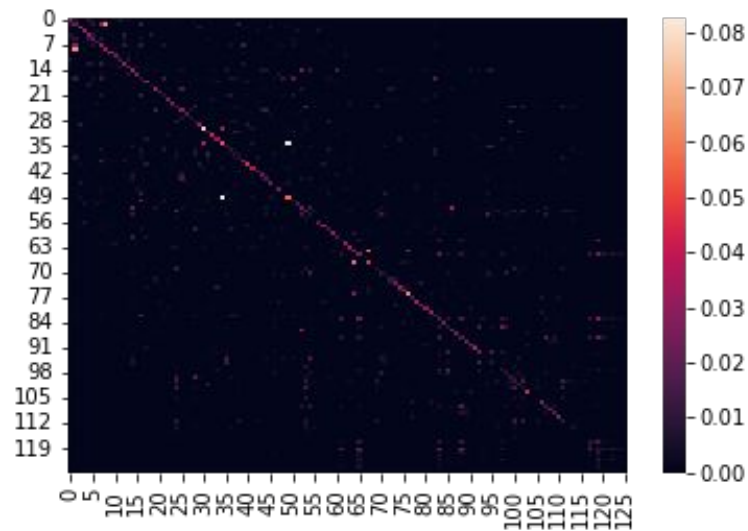
[Open](#)

6e6b803

Analysis Tutorial (Mikkel)

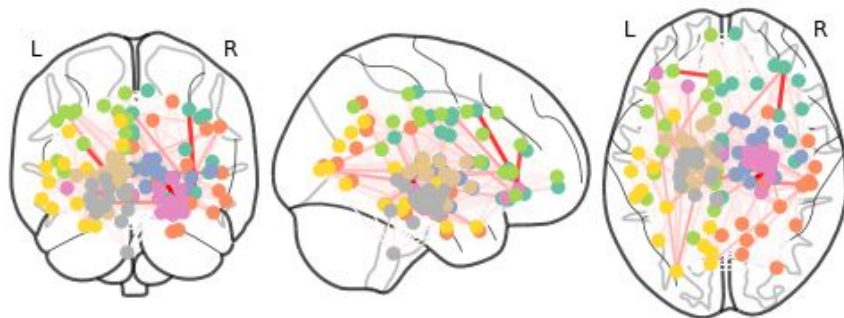
As the dataset is in BIDS, we can use `pybids` to help us load the data.

```
[24]: layout = BIDSLayout(vepcon_data.path)
# add derivative folder containing the connectivity matrices
layout.add_derivatives(os.path.join(vepcon_data.path, "derivatives", "cmp-v3.0.0-beta-RC1"))
```



[15]:

	source	target	number_of_fibers	fiber_length_mean	fiber_length_median	fiber_length_std	fiber_proportion
0	1	43	1275	18.864727	16.058386	11.975668	0.186114
1	1	1	3946	8.535786	6.804827	5.151622	0.576004
2	1	20	76	118.715965	114.774269	13.143189	0.011094
3	1	32	99	78.879990	98.564438	33.893044	0.014451
4	1	4	1073	11.444057	8.865350	6.503943	0.156627
...

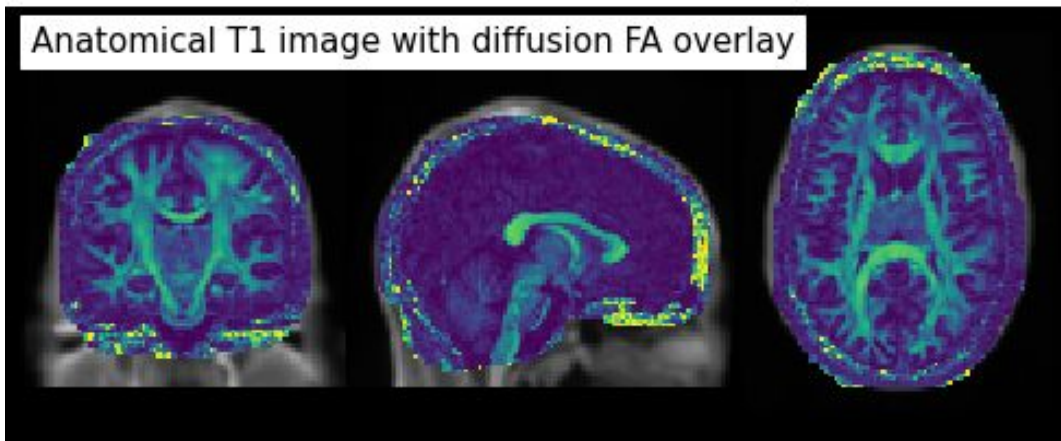


Towards a processing report - co-registration (Stefan)

```
In [57]: from nipype import Workflow, Node
plotAnatDWIQC = Node(overlayAnatDiffQC(anat_file = p.join(data_path, anat_file),
                                                         dwi_fa_file = p.join(data_path, diff_data),
                                                         out_anat_dwi_plot = p.join(data_path, "out_anat_dwi_plot.png")),
                    name = "overlayAnatDiffQC")
plotAnatDWIQC.run()
```

```
211204-14:48:19,192 nipype.workflow INFO:
[Node] Setting-up "overlayAnatDiffQC" in "/tmp/tmpb65tywr9/overlayAnatDiffQC".
211204-14:48:19,452 nipype.workflow INFO:
[Node] Running "overlayAnatDiffQC" ("__main__.overlayAnatDiffQC")
211204-14:48:26,289 nipype.workflow INFO:
[Node] Finished "overlayAnatDiffQC".
```

Anatomical T1 image with diffusion FA overlay



Towards a processing report - detect BOLD artifacts (Stefan)

```
In [134]: timeseries_npy = "sub-103818_ses-01_atlas-L2018_res-scale1_timeseries.npy"
carpetPlotter = Node(carpetPlot(timeseries_npy = p.join(data_path, timeseries_npy),
                             out_carpet_plot = p.join(data_path, "out_carpet_plot.png")),
                    name = "carpetPlot")
carpetPlotter.run()
```

211204-16:32:03,678 nipype.workflow INFO:

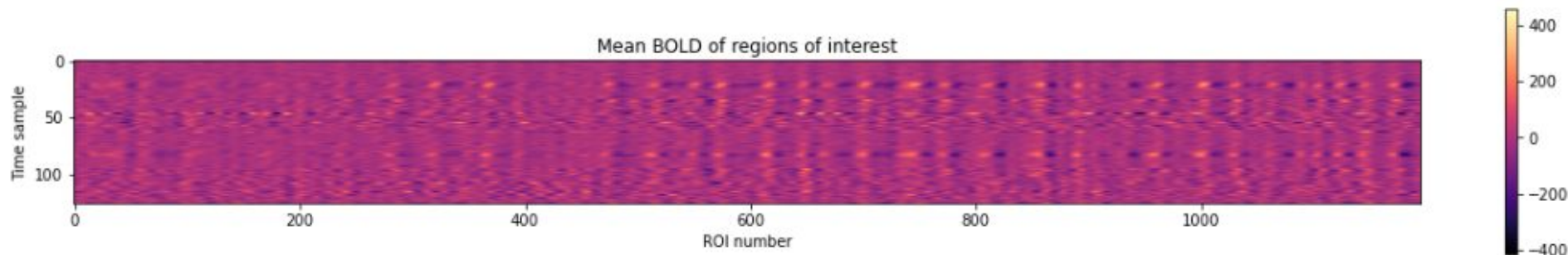
[Node] Setting-up "carpetPlot" in "/tmp/tmpztw8_2bf/carpetPlot".

211204-16:32:03,819 nipype.workflow INFO:

[Node] Running "carpetPlot" ("__main__.carpetPlot")

211204-16:32:09,239 nipype.workflow INFO:

[Node] Finished "carpetPlot".



Outcomes in brief


3 Pull Requests that will be integrated in the upcoming v3.0.0 release of Connectome Mapper3:

■  **ENH: add singularity CLI to wrap calls to the Singularity BIDS APP image** ●

#109 opened 19 hours ago by anilbey ⌚ updated 12 minutes ago

👁 1

💬 1

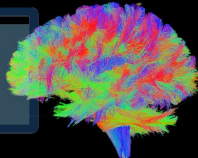
■  **ENH: add docker CLI to wrap calls to the Docker BIDS APP image** ●

#115 opened 33 minutes ago by kuba-fidel ⌚ updated 33 minutes ago

■  **Enh/release/v3.0.0/enhanced bidsappman messages** ✖

#114 opened 2 hours ago by jwirsich ⌚ updated 2 hours ago

Brain
Hack



GLOBAL 20
21

Emeline



Ale



Yasser



Joan



Katha



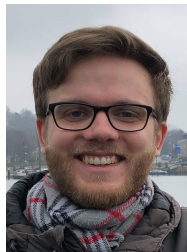
Jonathan



Anil



Mikkel



Stefan



Jakub



Sebastien



You?



Anyone is welcome to
join our community!

