bids_apps_slides

April 27, 2018

1 BIDS Apps

Franz Liem (franziskus.liem@uzh.ch)

This tutorial aims to introduce BIDS Apps. After briefly introducing some background, we will run the mriqc BIDS App on our laptops to get quality reports for MRI data.

2 Before we start

- Laptop with Docker installed (see guides for mac, windows, linux)
- if possible, already download (pull) the docker container we will be using:
  - open a terminal window/command line
  - paste the following command: `docker pull poldracklab/mriqc:0.10.4`
  - press enter and wait for the download to be finished
- download the this BIDS-formatted example data set (430 MB)

3 Background

3.1 Neuroimaging software

- Installation of neuroimaging software can be painful
- Complex workflows might require to install multiple software packages
- Needs to be repeated for new system (e.g., cloud system)
- Different software version might give different results

Makes it more difficult to reproduce analyses

3.2 What is BIDS

- Brain Imaging Data Structure
- A standardized way to represent data and metadata from neuroimaging studies
- Gorgolewski et al., 2016

bids-apps.neuroimaging.io/apps/

3.3 What are BIDS Apps

- Portable neuroimaging pipelines shipped as **software containers**
- Understand BIDS
- Developed by different labs all over the world
- `http://bids-apps.neuroimaging.io`, Gorgolewski et al., 2017

3.4 BIDS Apps examples

3.4.1 Data quality

- mriqc
- qap

3.4.2 Functional MRI

- cpac

- fmriprep
- niak

3.4.3 Structural MRI

- antsCorticalThickness
- baracus
- freesurfer
- mindboggle
- tracula

3.5 What are software containers

- A box that has software in it
- You don't need to install single software packages
- You just need to download/install
  - the container
  - a software that runs the container

3.6 What is Docker

A software that executes software containers.
Getting started with Docker

4 Running BIDS Apps

4.1 BIDS Apps are plug-and-play

To process your data, you only need to specify

- BIDS App
- Input folder (with BIDS-formatted data)
- Output folder that stores the results

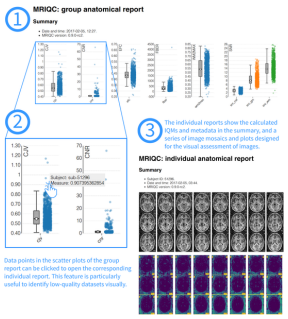
4.2 mriqc

- MRI quality control tool
- Developed by the Poldrack Lab
- Structural and functional MRI data

4.3 mriqc

Two analysis levels

- participant
- group



4.4 mriqc

Results

- visual reports
- IQMs (Image Quality Metrics; see Esteban et al., 2017)

Fig 5, Esteban et al., 2017

4.5 BIDS Apps in the cloud

OpenNeuro

5 Hands-on

We will now run mriqc on example data from the ABIDE study.
See `00Info.txt` for furhter details.

Data location Download the example data and unpack it into `~/data` (or adapt the paths in the examples accordingly).

There should now be a folder **`~/databids_apps_data`** with the data inside.

```
|-- 00Info.txt
|-- derivates
|-- sourcedata
```

5.0.1 BIDS sourcedata

```
|-- sourcedata
    |-- T1w.json
    |-- sub-0051160
        |-- anat
            |-- sub-0051160_T1w.nii.gz
```

5.0.2 Precomputed mriqc data

```
|-- derivates
|   |-- mriqc_0.10.4_precomputed
|       |-- 00INFO.txt
|       |-- derivatives
|       |   |-- sub-0051160_T1w.json
|       |   |-- ....
|       |-- logs
|       |-- reports
|           |-- sub-0051160_T1w.html
|           |-- ...
...
```

5.1 Download image with `docker pull`

- BIDS Apps provide images on Docker Hub, e.g., mriqc
- Docker Hub images can be downloaded with the `docker pull` command

`docker pull poldracklab/mriqc:0.10.4`
downloads tag (version) `0.10.4` of image `poldracklab/mriqc`

5.2 List of available mriqc tags

hub.docker.com/r/poldracklab/mriqc/tags/

5.3 Get a list of locally available images

`docker images`
gives you a list of all images that are downloaded to your computer

5.4 Running an analysis

5.4.1 mriqc help

To print help text for mriqc run

```
docker run --rm -ti poldracklab/mriqc:0.10.4 -h
```

5.4.2 Architecture of a command

```
docker run --rm -it \
   -v [...] \
   image_name bids_dir output_dir analysis_level
```

5.4.3 Participant level

```
docker run --rm -it \
-v ~/data/bids_apps_data/sourcedata:/d/in:ro \
-v ~/data/bids_apps_data/derivates/mriqc_0.10.4:/d/out \
poldracklab/mriqc:0.10.4 /d/in /d/out participant
```

5.4.4 Participant level command: line 1

```
docker run --rm -it \
   -v ~/data/bids_apps_data/sourcedata:/d/in:ro \
   -v ~/data/bids_apps_data/derivates/mriqc_0.10.4:/d/out \
   poldracklab/mriqc:0.10.4 /d/in /d/out participant
```

- Run a docker container
- Clean up after the container exits
- Run it in interactive mode

5.4.5 Participant level command: line 2

```
docker run --rm -it \
   -v ~/data/bids_apps_data/sourcedata:/d/in:ro \
   -v ~/data/bids_apps_data/derivates/mriqc_0.10.4:/d/out \
   poldracklab/mriqc:0.10.4 /d/in /d/out participant
```

- By default, docker does not have access to data on the HD
- `-v` (or `--volume`) makes a folder on your HD available inside the docker container
- `-v {folder_name_on_HD}:{folder_name_inside_container}:[{mode, e.g., ro}]`

- `~/data/bids_apps_data/sourcedata` is a folder on my HD, it contains the input data
- the docker container will see this folder as `/d/in`
- it will not be able to write into this folder (`ro`: read only)

5.4.6 Participant level command: line 3

```
docker run --rm -it \
   -v ~/data/bids_apps_data/sourcedata:/d/in:ro \
   -v ~/data/bids_apps_data/derivates/mriqc_0.10.4:/d/out \
   poldracklab/mriqc:0.10.4 /d/in /d/out participant
```

- `~/data/bids_apps_data/derivates/mriqc_0.10.4:` is a folder on my HD, it will be populated with the output data
- the docker container will see this folder as `/d/out`
- no other option is given: docker will be able to write into this folder

5.4.7 Participant level command: line 4

```
docker run --rm -it \
   -v ~/data/bids_apps_data/sourcedata:/d/in:ro \
   -v ~/data/bids_apps_data/derivates/mriqc_0.10.4:/d/out \
   poldracklab/mriqc:0.10.4 /d/in /d/out participant
```

- `poldracklab/mriqc:0.10.4`: software image to use
- `/d/in`: bids_dir, folder with input data (has to be visible inside container)
- `/d/out`: output_dir, folder for output data (has to be visible inside container)
- `participant`: analysis level (options are: `participant`, `group`)

5.4.8 Running the participant level analysis on your laptop

```
docker run --rm -it \
-v ~/data/bids_apps_data/sourcedata:/d/in:ro \
-v ~/data/bids_apps_data/derivates/mriqc_0.10.4:/d/out \
poldracklab/mriqc:0.10.4 /d/in /d/out participant
```

This might take 15 min

5.4.9 Adding options

Take a look at mriqc's help for a list of options

```
docker run --rm -it \
-v ~/data/bids_apps_data/sourcedata:/d/in:ro \
-v ~/data/bids_apps_data/derivates/mriqc_0.10.4:/d/out \
poldracklab/mriqc:0.10.4 /d/in /d/out participant \
--participant_label 0051160 --n_procs 2
```

5.4.10 Participant level outputs

Outputs in derivates/mriqc_0.10.4

- derivatives/sub-{subject}_T1w.json
- reports/sub-{subject}_T1w.html

Open one of the precomputed outputs in `bids_apps_data/derivates/mriqc_0.10.4_precomputed/reports`

5.4.11 Running the group level analysis on your laptop

Requires participant level analysis. To speed things up, the example data has precomputed participant level data in `derivates/mriqc_0.10.4_precomputed`.

To run the group analysis, just replace **`participant`** with **`group`**.

```
docker run --rm -it \
-v ~/data/bids_apps_data/sourcedata:/d/in:ro \
-v ~/data/bids_apps_data/derivates/mriqc_0.10.4_precomputed:/d/out \
poldracklab/mriqc:0.10.4 /d/in /d/out group
```

This will take a couple of seconds

5.4.12 Group level outputs

Outputs in derivates/mriqc_0.10.4_precomputed

- T1w.csv
- reports/T1w_group.html

5.4.13 Group level outputs

Let's look at `reports/T1w_group.html`

Click on the outlier points