

---

# **dashQC\_fmri Documentation**

*Release alpha*

**Sebastian Urchs**

**Jul 21, 2020**



---

## Contents:

---

|          |                             |          |
|----------|-----------------------------|----------|
| <b>1</b> | <b>About</b>                | <b>3</b> |
| <b>2</b> | <b>Roadmap</b>              | <b>5</b> |
| 2.1      | short term goal . . . . .   | 5        |
| 2.2      | intermediate goal . . . . . | 5        |
| 2.3      | long term goal . . . . .    | 5        |



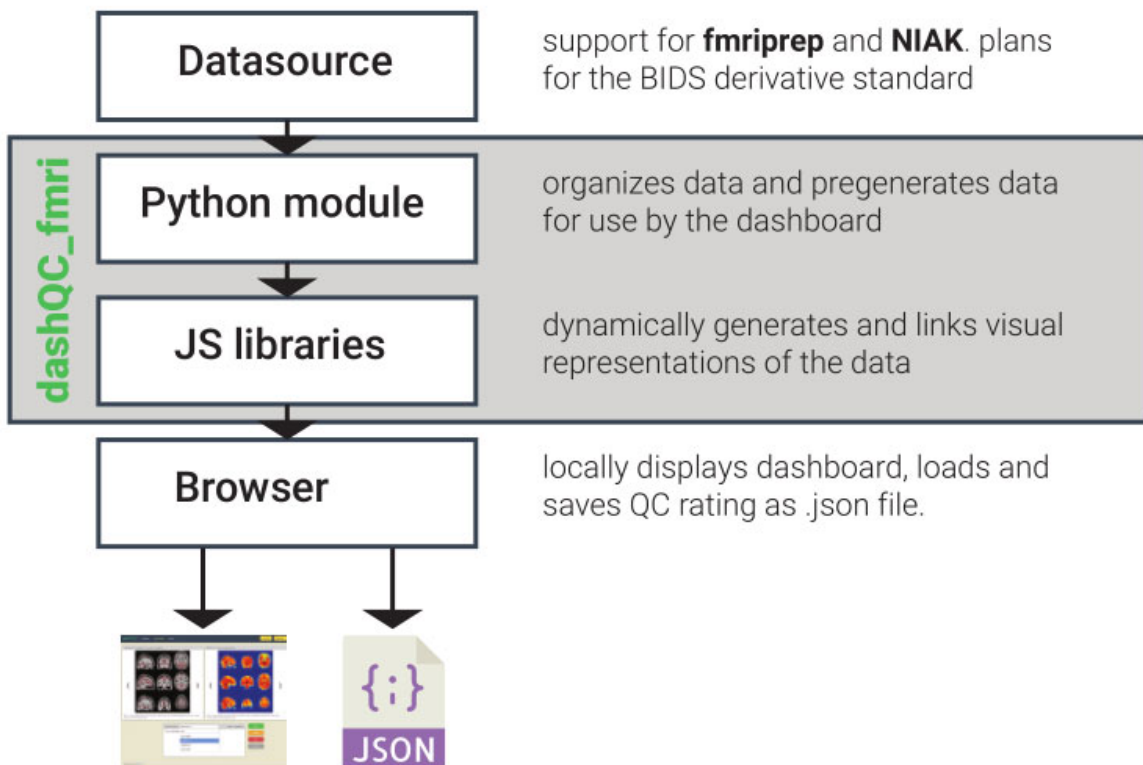
This dashboard is developed by the [SIMEXP lab](#) at the [Centre de recherche de l'Institut universitaire de gériatrie de Montréal](#). We use the dashboard daily in our lab internal quality control procedure to process large scale datasets. A demo of the dashboard can be found [here](#).

---

**Note:** `dashQC_fmri` is at an early stage of development and while we use it for our own quality control and are committed to developing it further, you may encounter problems. Please let us know about these by opening an [issue](#). Thanks a lot and have fun.

---





dashQC is a modular, javascript based dashboard built to facilitate the manual quality control process of functional MRI data. We currently provide support for the outputs generated by [fmriprep](#) and [NIAK](#). We have plans to extend this to a prospective [BIDS derivative](#) standard.

If you encounter a bug or would like to contribute to this project, please open an issue on our [github repository](#). If you have questions about how to use dashQC\_fmri please head over to [neurostars.org](#) and open a topic.





We are currently focused on implementing feedback from users of `dashQC_fmri` to facilitate the QC workflow. Our medium term goals are to create similar functionality for quality control with a focus on structural MRI.

### 2.1 short term goal

- Separate QC rating for T1-template and EPI-T1 coregistraion (#29)
- Change the internal datastructure to allow for faster searching and richer linkage of subject level and run level information (#17)
- Changing QC rating layout

### 2.2 intermediate goal

- Implement tagging
- fork project to create `dashQC_structural`

### 2.3 long term goal

- Performance optimization
- Subject and run data linkage

#### 2.3.1 Contents

### Installation

We distribute `dashQC_fmri` as a python package that includes both the python modules used to re-organize your pipeline output data into the format required by the dashboard and the javascript libraries and html templates that are used to render these data interactively,

You can obtain `dashQC_fmri` either from our [github repository](#), which has the most recent development version and is generally a good place to report and discuss problems and extensions. Alternatively, `dashQC_fmri` can also be installed via the Python Package Index.

### Dependencies

`dashQC_fmri` depends on the following python packages

- `numpy`
- `pandas`
- `nibabel`
- `pillow`
- `matplotlib`
- `nilearn`
- `scipy`
- `scikit-learn`

If you install `dashQC_fmri` in your python environment using the methods described below, we will take care of these dependencies for you.

If you want to run the python modules of `dashQC_fmri` directly without installation, please make sure that you install these dependencies manually.

### Install from source

To install directly from our development version, run the following command:

```
git clone git@github.com:SIMEXP/dashQC_fmri.git
cd dashQC_fmri
```

Then you can install `dashQC_fmri` using:

```
pip install .
```

If you do not have sudo permission on your system, you can also run:

```
python setup.py install --user
```

To install `dashQC_fmri` in user space.

### Install from PyPI

`dashQC_fmri` will be available soon from the Python Package Index.

## Running dashQC\_fmri without installation

The python modules of “dashQC\_fmri” can be run directly from the command line without installation. Please make sure to first install the *Dependencies*.

## Preparing data for dashQC\_fmri

To run dashQC\_fmri on your own preprocessed data from *fmrip* or *NIAK* you need to first run the corresponding data converter module. The converter modules are python scripts that reorganize the output data from *fmrip* or *NIAK* into the folder structure expected by dashQC\_fmri. They also pregenerate some of the larger visual representations so you can later run the dashboard without any lag. This process can take some time but only has to be performed once per preprocessed dataset.

---

**Note:** Make sure that you have completed the *installation* of dashQC\_fmri before you try to follow the steps in this section.

---

## Preparing data from a fmrip pipeline

---

**Note:** To generate the dashQC\_fmri data for a *fmrip* pipeline we currently need access to the unprocessed raw data folder on which *fmrip* was run. Please make sure that this location is reachable before you run the converter.

---

Let’s assume the raw data folder that *fmrip* was run on is located at `/path/to/my/raw/data` and the output directory of your successfully completed *fmrip* pipeline is located at `/path/to/my/fmrip/output/`. To generate your *dashQC\_fmri* report at `/path/to/my/dashQC_fmri/report` you would then run

```
python -m dashQC_fmri.fmrip_report /path/to/my/fmrip/output/ /path/to/my/dashQC_fmri/report
/path/to/my/raw/data
```

Once the process has completed, navigate to `/path/to/my/dashQC_fmri/report` e.g.:

```
cd /path/to/my/dashQC_fmri/report
```

find the `index.html` file and open it in a browser (e.g. on Unix):

```
sensible-browser index.html
```

This will open the dashQC\_fmri dashboard you have just generated.

## Preparing data from a niak pipeline

Let’s assume the output directory of your successfully completed *NIAK* pipeline is located at `/path/to/my/niak/output/`. To generate your dashQC\_fmri report at `/path/to/my/dashQC_fmri/report` you would then run

```
python -m dashQC_fmri.niak_report /path/to/my/niak/output/ /path/to/my/dashQC_fmri/report
```

Once the process has completed, navigate to `/path/to/my/dashQC_fmri/report` e.g.:

```
cd /path/to/my/dashQC_fmri/report
```

find the `index.html` file and open it in a browser (e.g. on Unix):

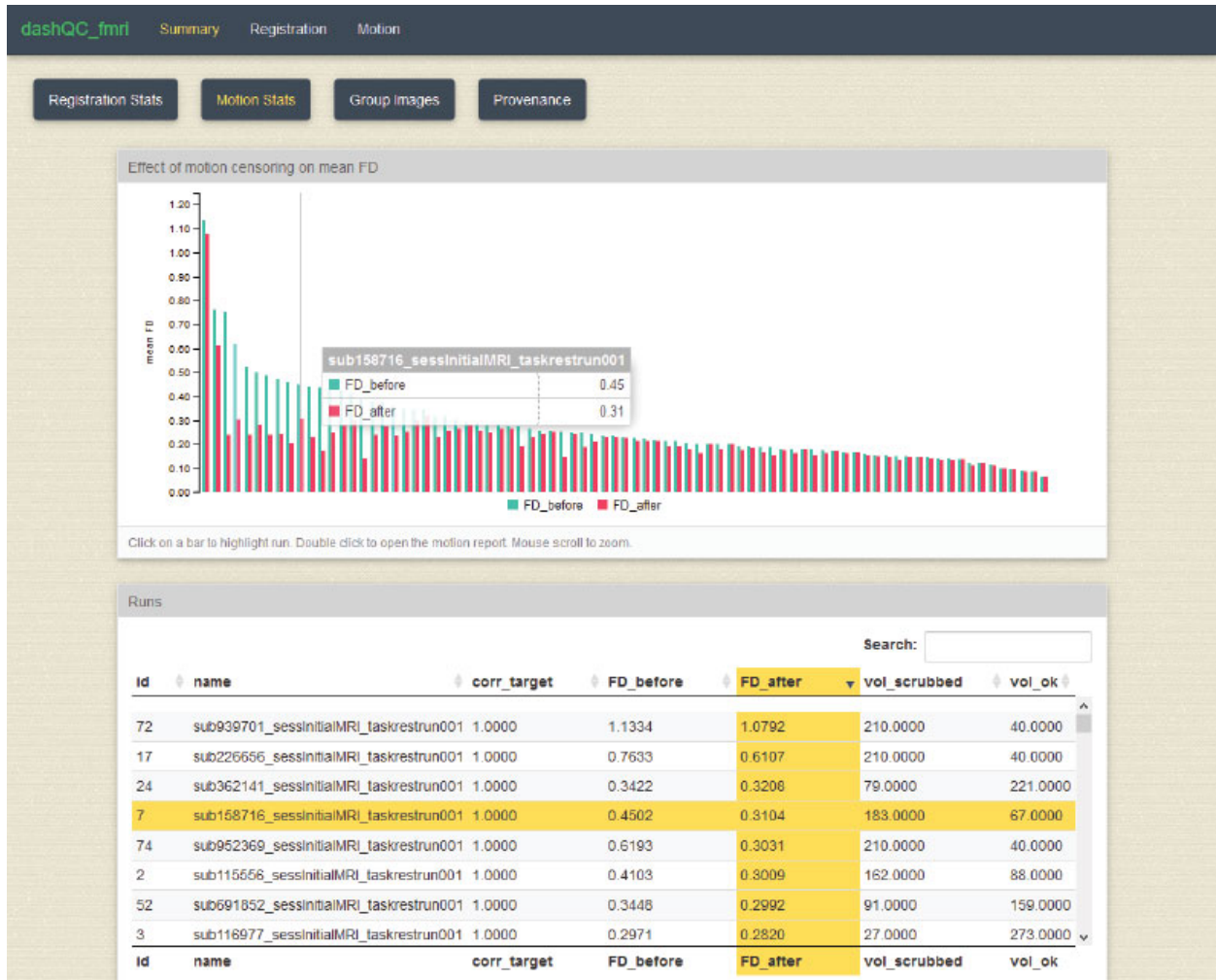
sensible-browser index.html

This will open the dashQC\_fmri dashboard you have just generated.

## Using dashQC\_fmri

dashQC\_fmri presents your data in 3 thematically separate but interlinked views:

### Summary Page



Information on the dashboard’s summary page is divided into several subsections accessible via the ‘Registration Stats’, ‘Motion Stats’, ‘Group Images’, and ‘Provenance’ pills.

### Registration Stats

The Registration Stats subsection consists of subject registration chart and table views. Data on the table is displayed on the chart above. The table itself is sortable by clicking the table column headers and searchable via the search box in the upper right of the table. Each row of the table represents a subject. Each column in the table (‘T1 corr\_target’, ‘T1 overlap\_brain’, ‘BOLD corr\_target’, and ‘BOLD overlap\_brain’) corresponds to its own chart. Clicking on column

headers highlights and sorts that entire column and displays the appropriate chart in the view above. Clicking on single rows highlight that subject in the chart view above as well. Similarly, single clicks on bars in the chart will highlight that subject's value for the chart. Double clicking a row or a bar in a chart will take you to the registration page for that subject.

### Motion Stats

The Motion Stats subsection consists of run motion chart and table views. The table works similarly to the subject table in the Registration Stats subsection, but here each row represents a run. The columns in this table, however, do not correspond to a single chart. Clicking 'corr\_target' reveals a chart view of the column data, but the pairs 'FD\_Before' and 'FD\_After' as well as 'vol\_scrubbed' and "vol\_ok" are both represented on one chart each. Single clicks in the chart and table function like single clicks in Registration Stats, but double clicking a row or a bar in a Motion Stats chart will instead take you to the motion page for that run.

### Group Images

Moving the mouse over the four images in this subsection allows comparison between group averaged images to templates.

### Provenance

The Provenance subsection gives information on the overall pipeline process and options, as well as per subject information. Click the "Input files" dropdown to switch between subjects. The "Input files for subject" and "Pipeline options" views feature collapsible JSON for easier reading.

## Registration Page

This page is the area where users can actively perform quality control on their subject images. Two main image views available are “Individual vs template T1 scan for <the current subject>” and “BOLD vs T1 scan for <the current subject>”. These images make use of comparison via mouse over similar to the Group Images subsection on the Summary page. Subject images can be moved between by either clicking the arrow buttons on either side of the two images or using the keyboard’s left and right arrow keys. Below these two images is an area where the user can mark the quality level of the scans via buttons (pass, maybe, fail, or unmarked), and can also enter comments for each subject. These ratings are also enterable via hotkeys as well (p for pass, m for maybe, f for fail, and u for unmarked). Subjects can be moved between more directly via the “Current subject” dropdown in this area. Comments and ratings for the scans of each subject are stored in the browser’s web session storage. Unless the browser cache is cleared or the data set in the underlying folder is swapped out, these comments and ratings will remain stored even when the browser is closed. However, should users want to save a local JSON copy of this data, they can click the “Save QC” button in the top right of the page on the navbar. Users can also load previous ratings and comments (from a previously saved QC JSON file) on the current scans via the “Load QC” button, also on the top right of the page.

## Motion Page



This page shows several views of subject motion during runs of scans. Different runs are accessible via the ‘Select a run’ dropdown on the top right of the page. The three charts on the left feature frame displacement as well as translation and rotation parameters frame by frame. The two top images on the right display an animated view of fMRI frames in both native and stereotaxic space over time. The bottom image on the right displays the overall run versus reference volume. Frames on the top two images can be advanced via the arrow buttons as well as the left and right arrows on the keyboard. Moving between frames in this way also highlights the appropriate data point for the frame in the charts on the left. Hovering over points on the charts on the left reveal their data values and clicking on them advances the animation to the appropriate frame in the top two images.

### MIT License

Copyright (c) 2018 The dashQC\_fmri developers

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the

Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

### 2.3.2 Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)