MRIQC: automatic prediction of quality and visual reporting of MRI scans

Using MRIQC

An step-by-step introduction, examples, and discussion on MRI quality control.

fMRI meeting 11/7/2018

University of Denver

Alexander.Dufford@du.edu

What is mriqc?

- Developed by Poldrack Lab at Stanford and Center for Reproducible Neuroscience (Oscar Esteban, Krzysztof Gorgolewski, Russ Poldrack)
- Written in python (nipype)

- A quality control pipeline that extracts multiple Image Quality Metrics (IQMs) that can be used for automated prediction of quality and generate group and individual visual reports

- What goals of an MRI researcher does mriqc help with?
- 1. Find systematic errors in acquisition
- 2. Maximize quality of ongoing studies
- 3. Find subpar images
- 4. Prevent biasing analysis

What is mriqc?

- Open source and built upon the following principles:

- 1. modularity and integrability: uses nipype under the hood to call several softwares (FSL, ANTs, AFNI)
- 2. minimal preprocessing: should be as much as possible on the original data (off the scanner)
- 3. Interoperability and standards: based on brain imaging data structure (BIDS)
- 4. reliability and robustness: robustness tested on OpenfMRI data

Step 1. BIDS your data

- What is BIDS?
- Stands for brain imaging data structure
- http://bids.neuroimaging.io/
- A standard file structure and naming for MRI researchers to share data
- “your primary collaborator is yourself 6 months from now, and your past self doesn’t answer emails”

- Tools to automate BIDS-ifying your data:

https://github.com/nipy/heudiconv **

https://github.com/cbedetti/Dcm2Bids

- **sub-control01**
  - **anat**
    - sub-control01_T1w.nii.gz
    - sub-control01_T1w.json
    - sub-control01_T2w.nii.gz
    - sub-control01_T2w.json
  - **func**
    - sub-control01_task-nback_bold.nii.gz
    - sub-control01_task-nback_bold.json
    - sub-control01_task-nback_events.tsv
    - sub-control01_task-nback_physio.tsv.gz
    - sub-control01_task-nback_physio.json
    - sub-control01_task-nback_sbref.nii.gz
  - **dwi**
    - sub-control01_dwi.nii.gz
    - sub-control01_dwi.bval
    - sub-control01_dwi.bvec
  - **fmap**
    - sub-control01_phasediff.nii.gz
    - sub-control01_phasediff.json

Step 2: Run mriqc! (3 options possibly in order of recommendation)

- 1. Compiled locally: available on DU's HPC, use this module: tools/mriqc-20170630-without-ImageMath

- 2. Run in docker container

- Run in OpenNeuro (requires data sharing)

Step 2: Run mriqc!

How to mriqc on HPC:

After loading the module type:

mriqc bids-root/ output-folder participant

This will run on T1w and bold images found in bids-root to extract first level (participant IQMs)

Change ‘participant’ to ‘group’ to generate your group reports and group IQMs

Step 2: Run mriqc! Functional workflow

- Sanitize (revise data types and xforms) input data, read associated metadata and discard non-steady state frames.
- HMC based on 3dvolreg from AFNI – hmc_afni().
- Skull-stripping of the time-series (AFNI) – fmri_bmsk_workflow().
- Calculate mean time-series, and tSNR.
- Spatial Normalization to MNI (ANTs) – epi_mni_align()
- Extraction of IQMs – compute_iqms().
- Individual-reports generation – individual_reports().

Step 2: Run mriqc! Anatomical workflow

- Conform (reorientations, revise data types) input data and read associated metadata.
- Skull-stripping (AFNI).
- Calculate head mask – headmsk_wf().
- Spatial Normalization to MNI (ANTs)
- Calculate air mask above the nasial-cerebellum plane – airmsk_wf().
- Brain tissue segmentation (FAST).
- Extraction of IQMs – compute_iqms().
- Individual-reports generation – individual_reports().

Step 3: Look at your visual reports

- Functional IQMS: read more in mriqc documentation

- DVARS: temporal derivative of timecourses, VARS referring to TMS variance over voxels, rate of change of BOLD signal across the entire brain at each frame of data

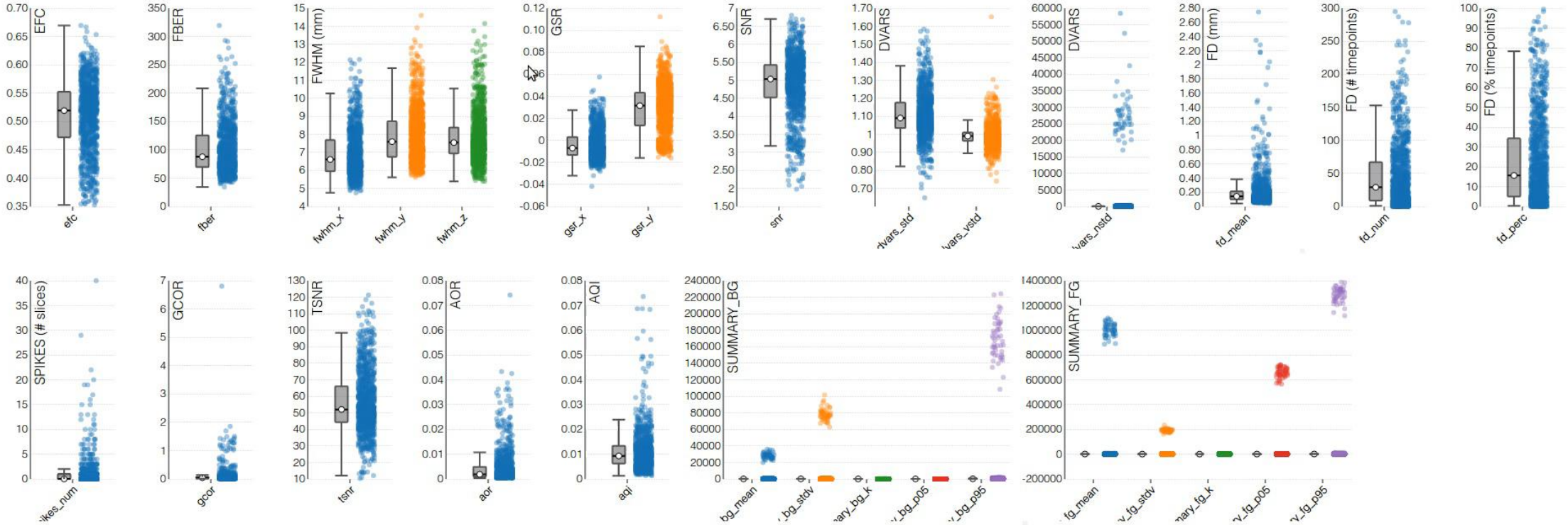
- Framewise Displacement: instantaneous head motion
- Gives you fd_mean, fd_num (number above FD thresh) and fd_perc: percentage of FDs above threshold
- The threshold is set at 0.2 mm
- Caveat: am I despiking, censoring, or scrubbing? *potential topic for another meeting

Step 3: Look at your visual reports (group)

MRIQC: group functional report

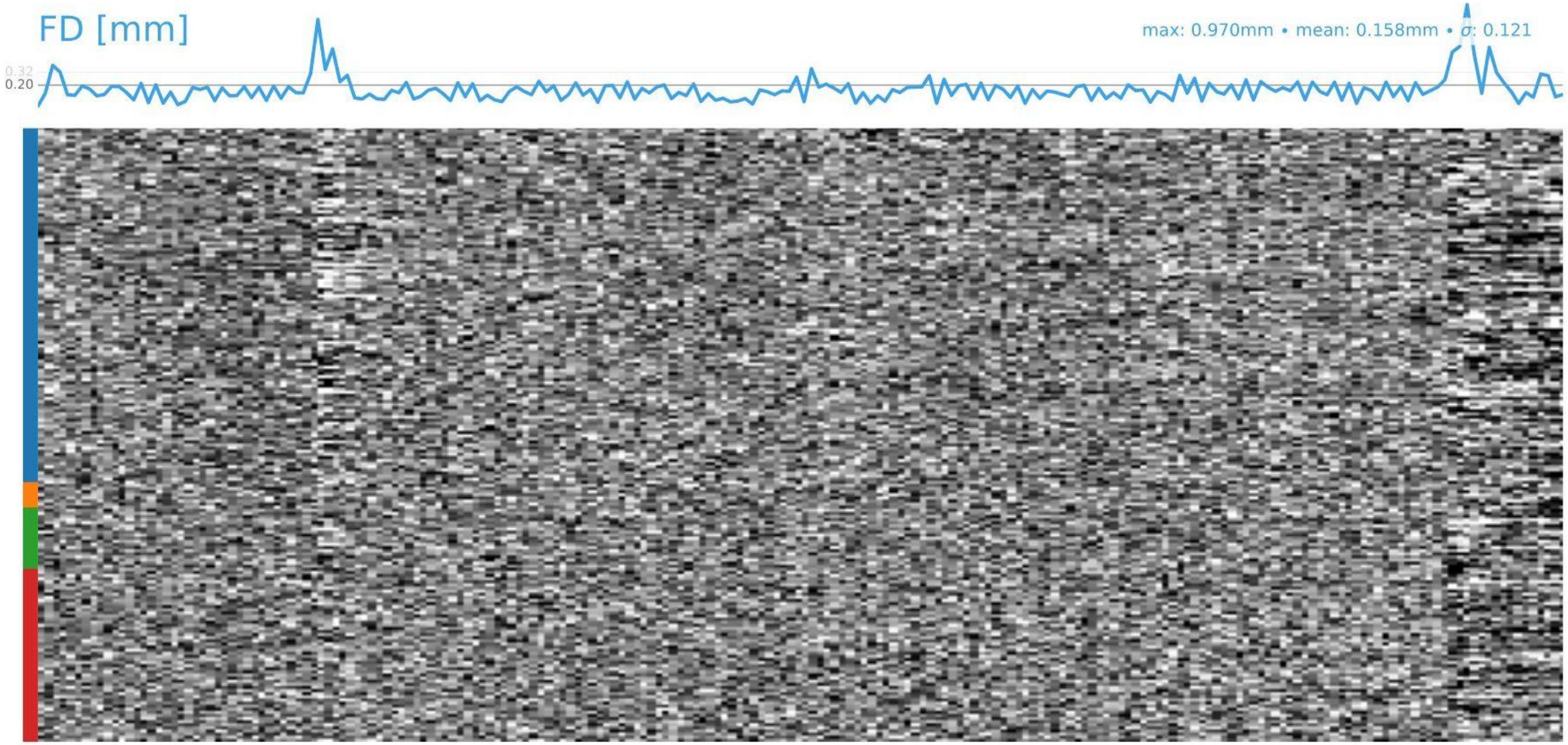
Summary

- Date and time: 2017-02-05, 12:27.
- MRIQC version: 0.9.0-rc2.



Step 3: Look at your visual reports (individual)

Step 3: Look at your visual reports (individual)

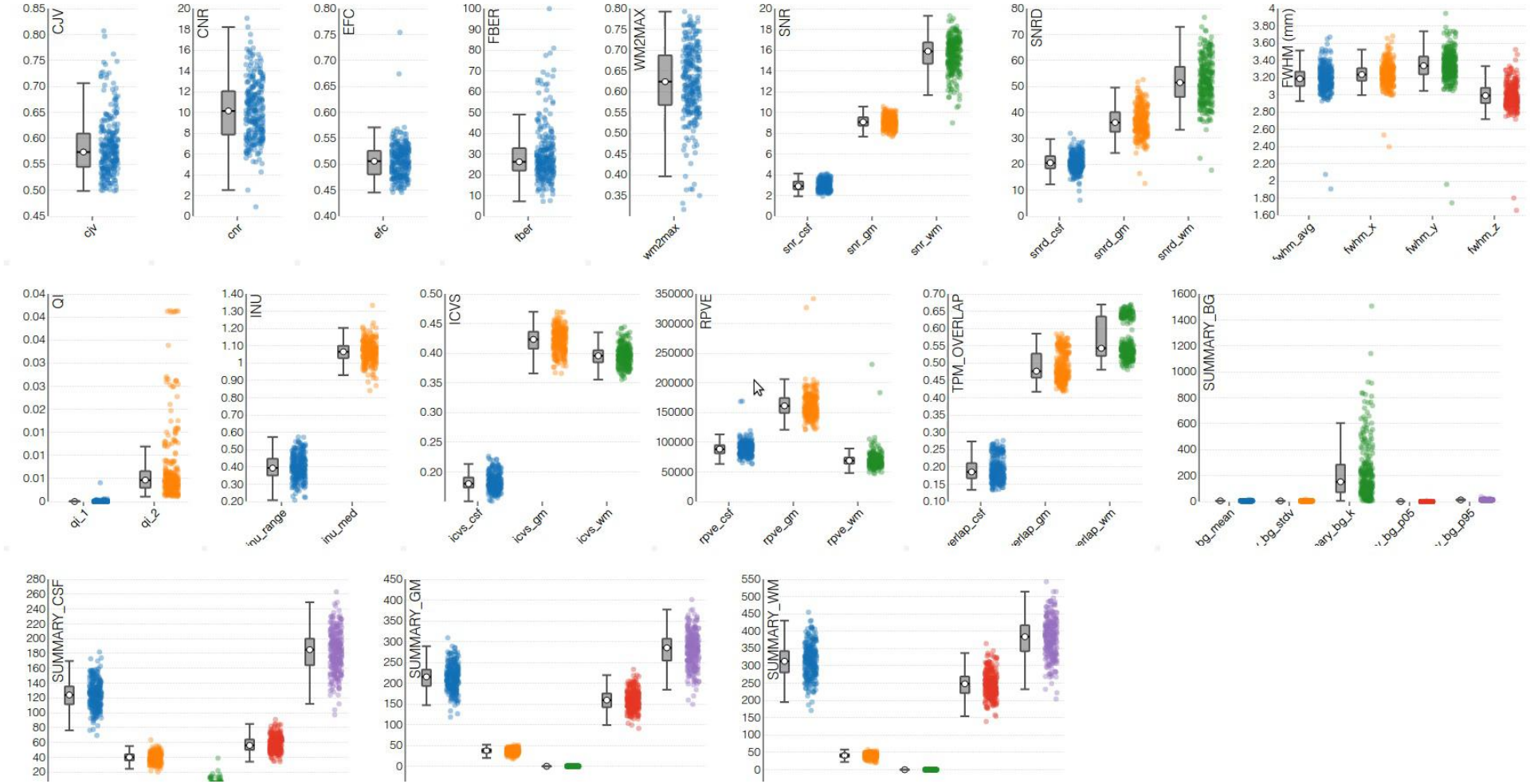


Step 3. Look at your visual reports

MRIQC: group anatomical report

Summary

- Date and time: 2017-02-06, 08:56.
- MRIQC version: 0.9.0-rc3.



Step 4: Make decisions about quality

- Plot your group data! Find out who the outliers are, mark them for later?
- Set a threshold before collecting data about your IQMs

- Anatomical data: can use random forest classifier to use your IQMs and compare to a multi-site data set (N=1102), you can also train on your scanner's data (a Boulder classifer, Anschutz classifer)
- Returns a recommendation for pass/fail
- Functional data: see how different participants fall on several IQMs

- Can have an idea of quality before preprocessing, feed into afniproc or fMRIprep

Advantages vs. Disadvantages

Disadvantages:

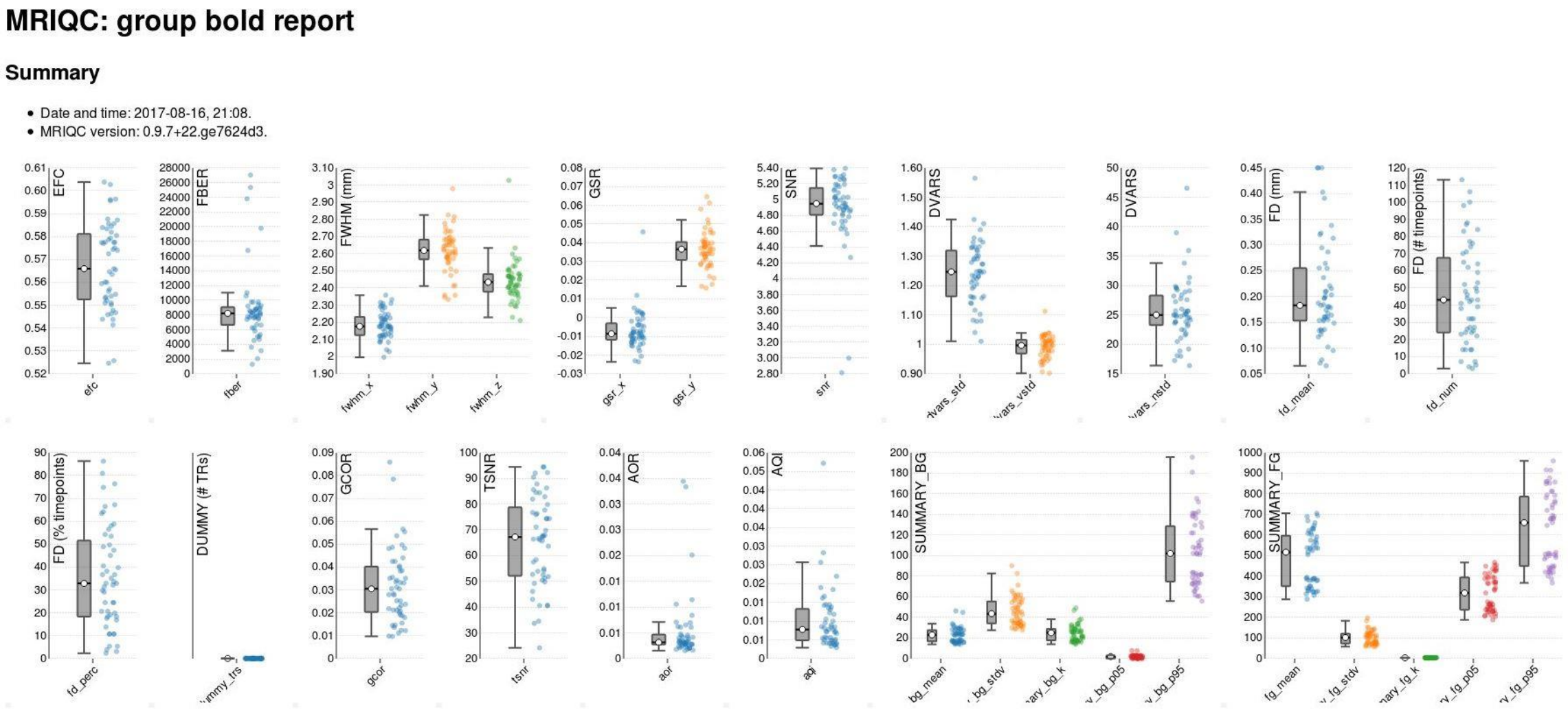
- 1. My preprocessing already does all of that!
- 2. Data must be BIDS-compliant
- 3. I have to install all of those imaging packages I don’t normally use (perhaps docker is the way to go)
- Doesn’t provide you information on task
- 4. Cases not yet support: animal MRI, infant MRI, certain types of pathology

Advantages vs. Disadvantages

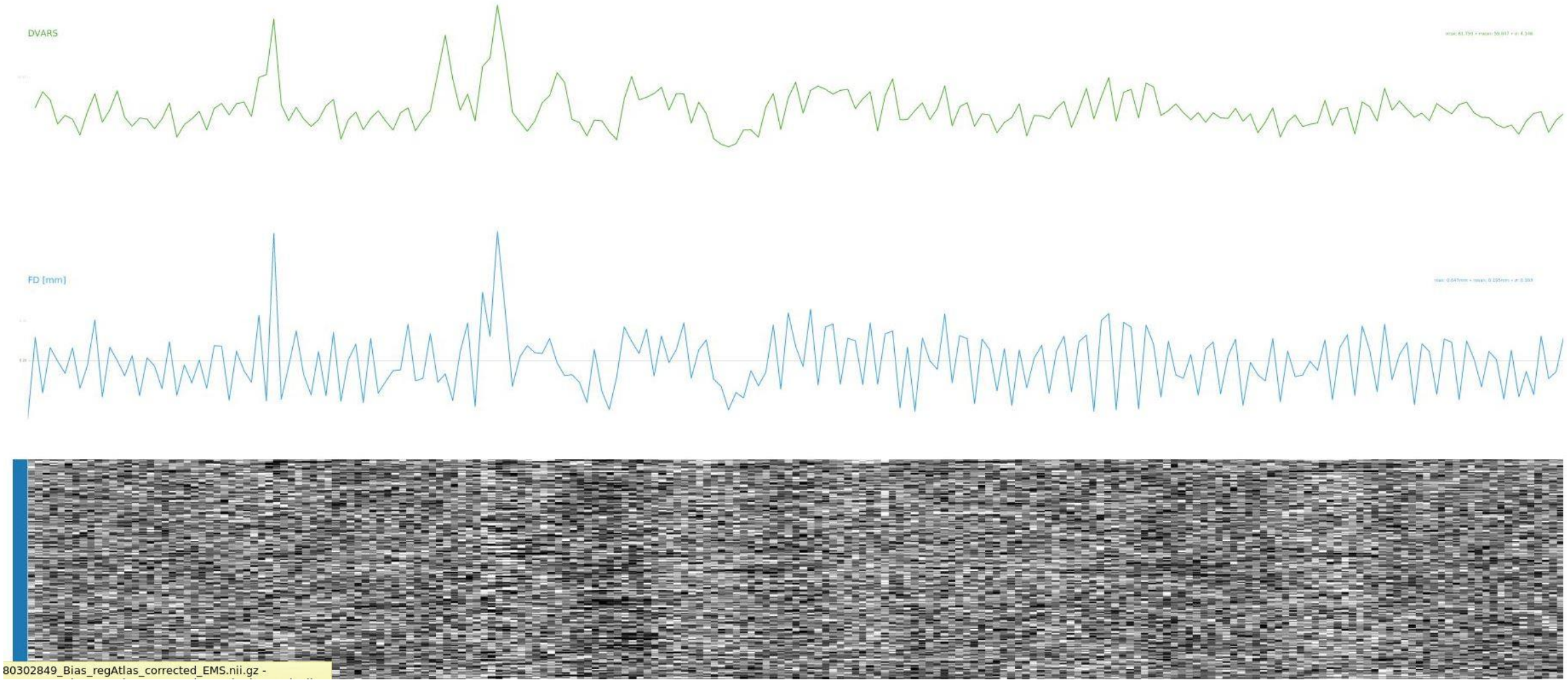
Advantages:

- 1. Can give you a rough estimate of the quality of a scan (especially while collecting data), you can run it quickly the next day
- This can help catch acquisition errors, hardware failures, participant artifacts (don’t wait until all your data is collected to find out)
- 2. Gives you multiple IQMs and then summarizes them in the group plot
- 3. May introduce less bias into your QC, especially for anatomical images (raters can be unreliable)
- 4. Automated, can run several steps with one command
- 5. Great documentation and responsive developers: https://mriqc.readthedocs.io/en/stable/index.html

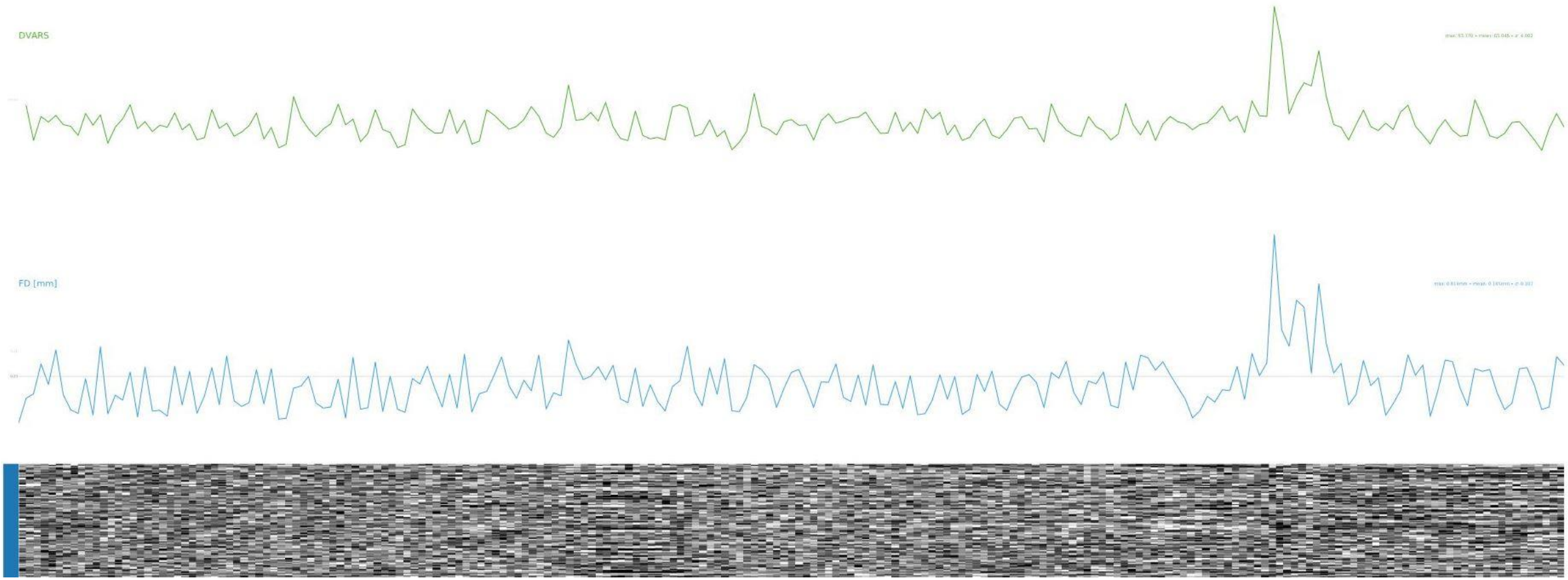
Let’s do some examples! Starting w/ func



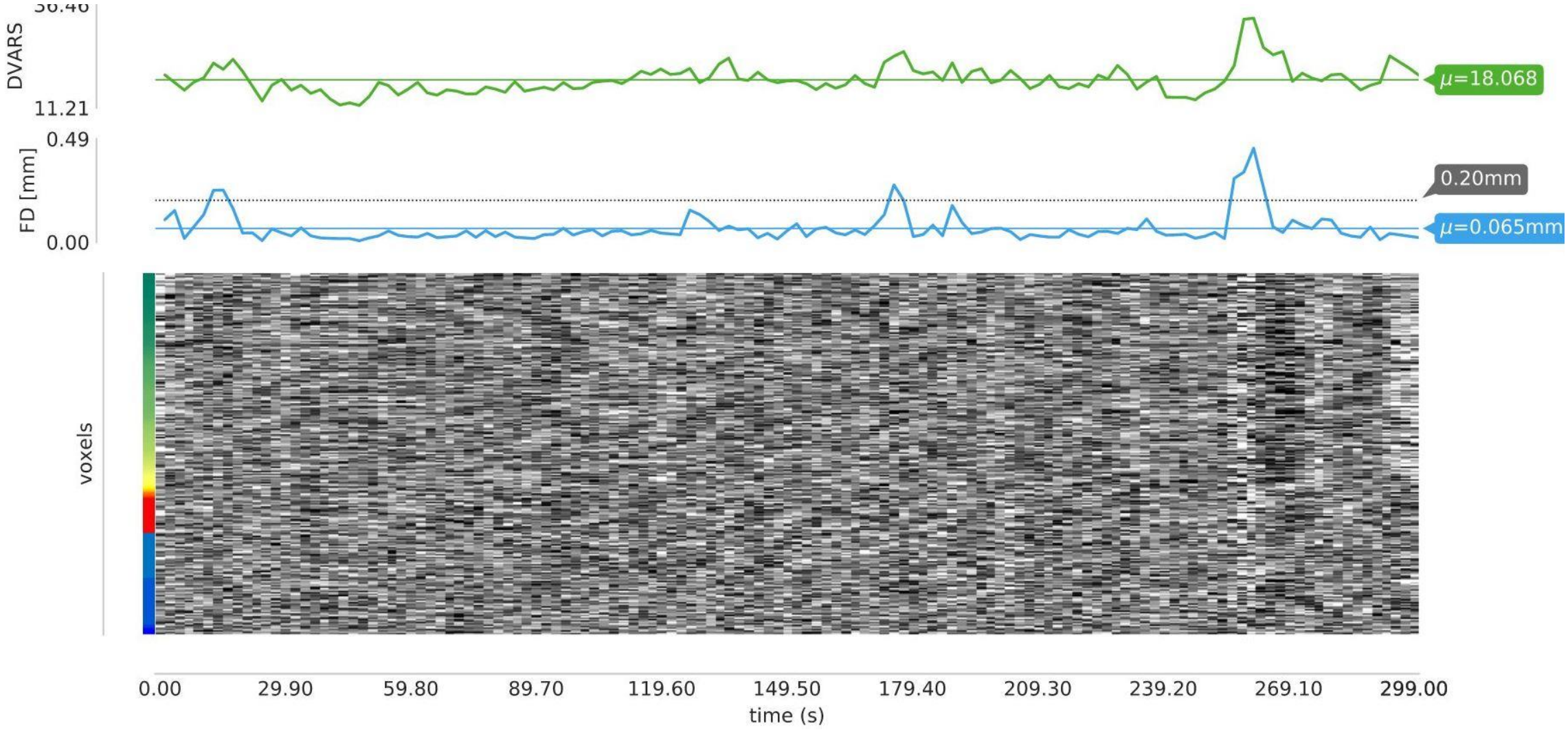
Problematic participant (resting)



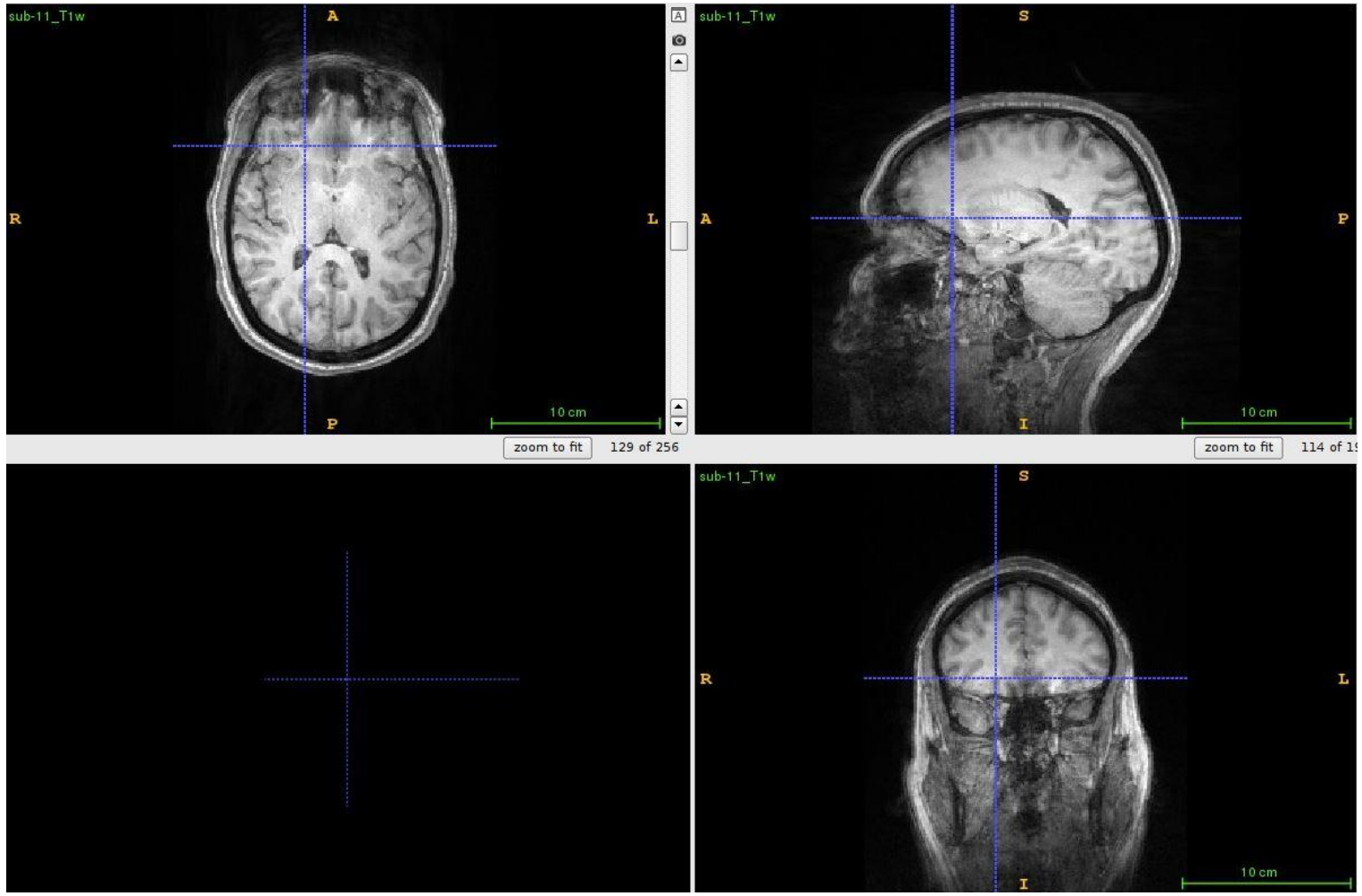
Maybe participant (resting)



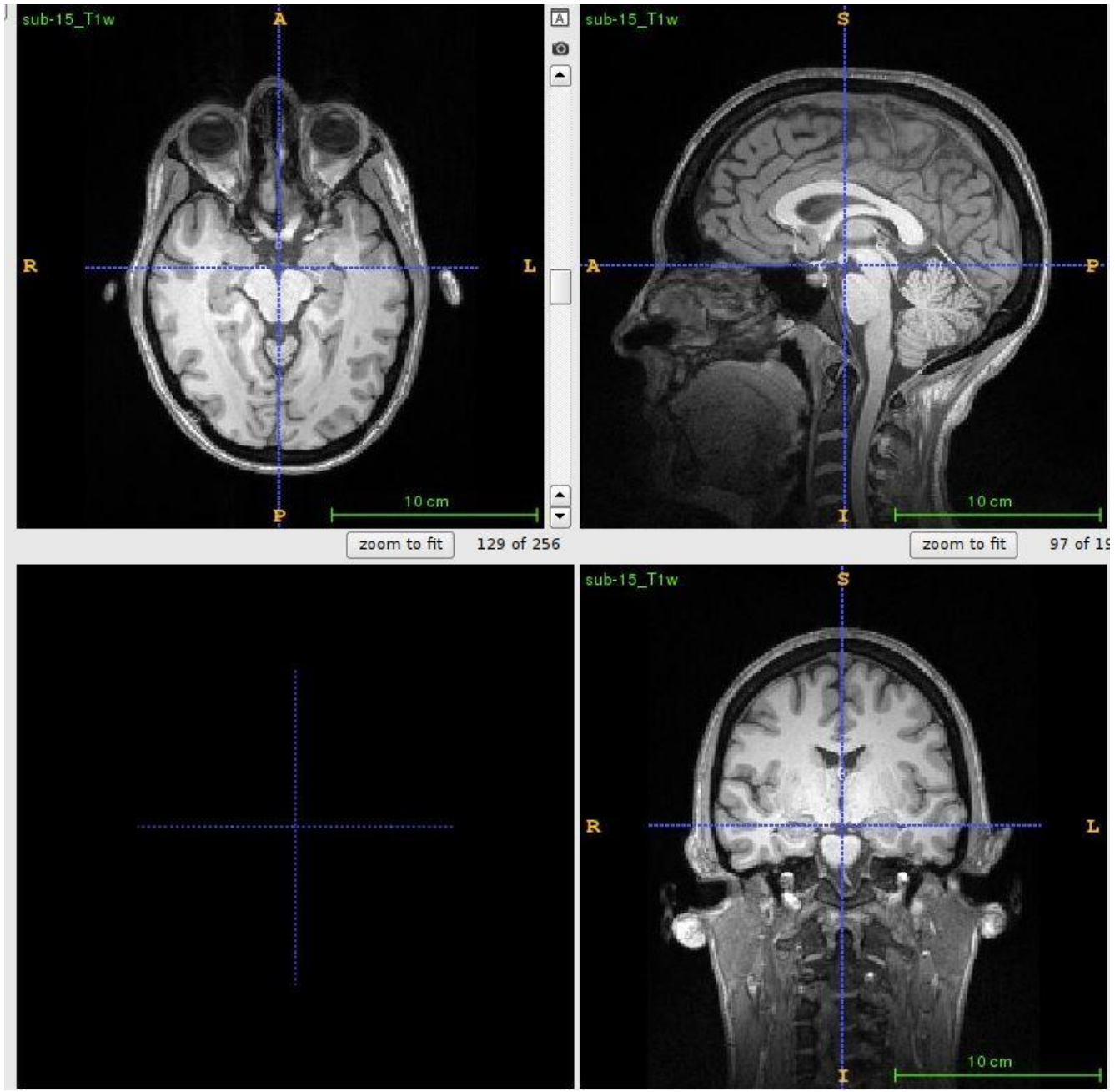
Great participant (resting)



Now T1, identified by classifier as exclude



Great T1

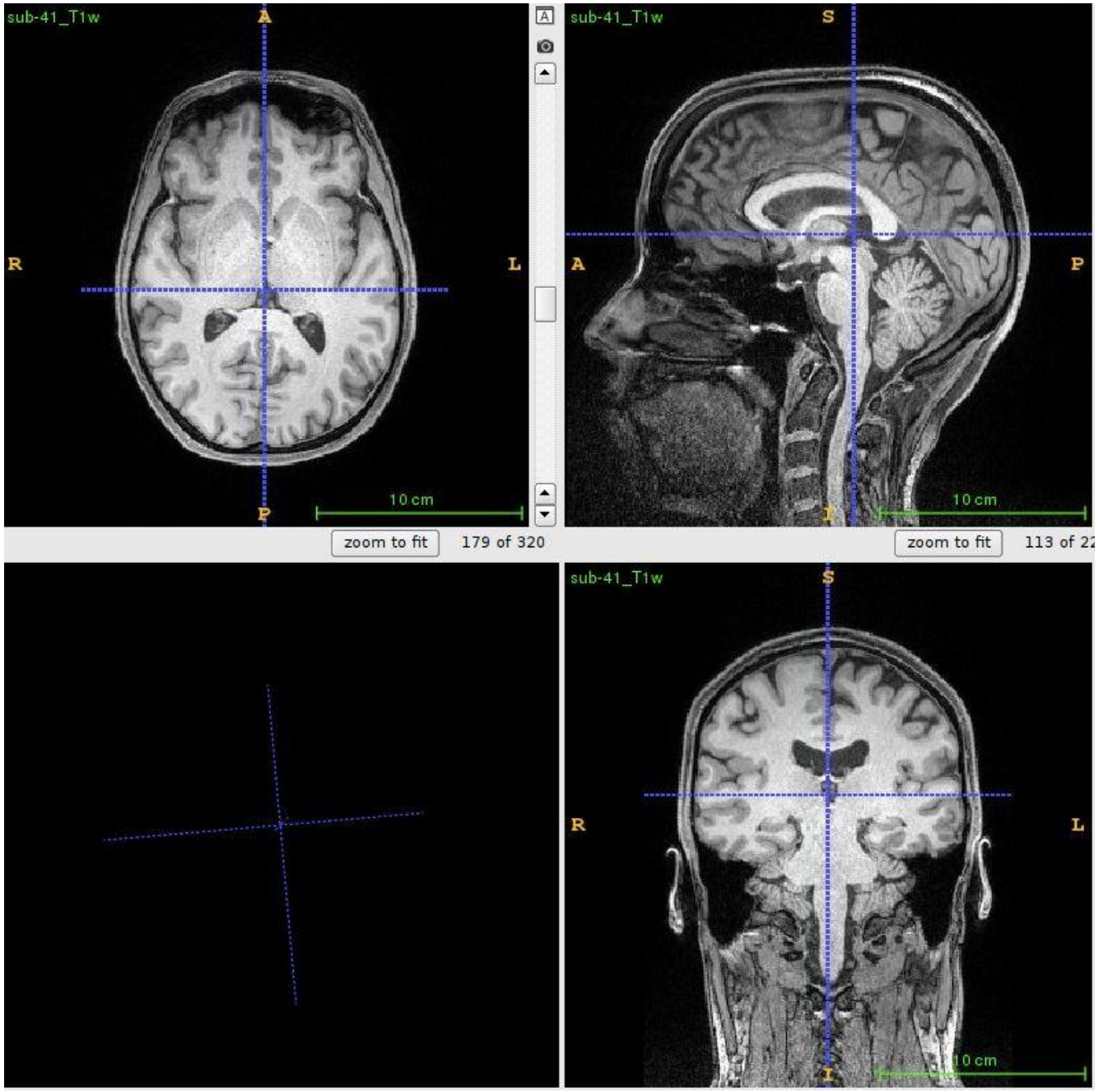

sub-15_T1w
A
R
L
P
10 cm
zoom to fit
129 of 256
sub-15_T1w
S
A
P
I
10 cm
zoom to fit
sub-15_T1w
S
R
L
I
10 cm

Maybe T1

Not suggested to exclude

No real problems, include!

Depends on your own criteria and scientific question!



Discussion

- 1. Any other steps/ tricks or tips people suggest?
- 2. What are some ways the field can begin to be consistent as far as QC across studies?
- Strategies for when to run initial QC versus more detailed QC?