



QC – the AFNI way

Always looking for trouble...



What is quality control (QC), particularly in FMRI?

- Finding good and bad (and “other”) datasets
 - Checking the *consistency* of initial dataset
 - Evaluating the success of processing steps
 - Determining if the data are suitable for this particular study
- ... and more.

What is quality control (QC), particularly in FMRI?

- Finding good and bad (and “other”) datasets
 - Checking the *consistency* of initial dataset
 - Evaluating the success of processing steps
 - Determining if the data are suitable for this particular study
- ... and more.

Rather than viewing QC as simply filtering datasets into “good” or “bad” bins, we regard it as the larger procedure of being as sure as possible about the contents of the data collection, from acquisition properties to artifact checking to regression evaluation.

(Reynolds et al., 2023, Frontiers QC Project contribution)

Processing + QC = afni_proc.py (AP)

- AFNI's recommended pipeline for fMRI processing
 - Python program that generates system command shell script
 - it's a commented and readable form of *full* pipeline
 - full provenance at code *and user understanding* levels
- Can be used for full / partial processing
- Simplifies choices by choosing processing blocks:
`tshift align volreg regress ...`
- Flexible with hundreds of options for *your* analysis needs

→ Automatic quantitative summaries and HTML reports that we need for QC ...

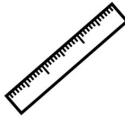
Outline of QC stages

1) **GTKYD**



*Getting To Know Your Data:
check dataset properties
and consistency*

2) **APQUANT**



*Quantitative review of basic
processing features:
censor fraction, GCOR, ...*

3) **APQUAL**



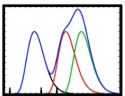
*Qualitative and systematic
visual checks via
afni_proc.py's QC HTML*

4) **GUI**



*In-depth investigation, via
the graphical user interface:
InstaCorr, graph viewer, ...*

5) **STIM**



*Task-specific investigations
of stimulus timing: event
timing, duration, ...*

Outline of QC stages

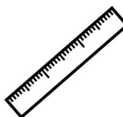
1) GTKYD



*Getting To Know Your Data:
check dataset properties
and consistency*



2) APQUANT



*Quantitative review of basic
processing features:
censor fraction, GCOR, ...*

3) APQUAL



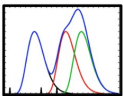
*Qualitative and systematic
visual checks via
afni_proc.py's QC HTML*

4) GUI



*In-depth investigation, via
the graphical user interface:
InstaCorr, graph viewer, ...*

5) STIM



*Task-specific investigations
of stimulus timing: event
timing, duration, ...*

Getting To Know Your Data

- The basics of datasets
- Orientation, data type, voxel size
- min, max values
- Consistency across runs, sessions, subjects; missing/extra data
- Tools:
 - 3dinfo, nifti_tool, 3dBrickstat
- Example scripts

GTKYD - Getting information

```
3dinfo -n4 -orient -ad3 -obliquity -tr -slice_timing -datum -prefix ... \
mysubjects/func/*.nii.gz
```

```
$ 3dinfo -header_line -n4 -orient -ad3 -obliquity -tr \
data_00_basic/sub-5??/ses-01/func/sub-5??_ses-01_task-rest_run-01_bold.nii.gz
```

Ni	Nj	Nk	Nv	orient	ADi	ADj	ADk	oblq	TR	prefix
128	128	34	144	RPI	1.875004	1.874996	4.000003	23.147	2.000000	sub-501.nii.gz
128	128	34	144	RPI	1.874997	1.874997	4.000003	3.695	2.000000	sub-502.nii.gz
128	128	34	144	RPI	1.874999	1.874997	3.999999	6.245	2.000000	sub-503.nii.gz
128	128	34	144	RPI	1.874996	1.875002	3.999995	4.795	2.000000	sub-504.nii.gz
80	80	34	144	RPI	3.000002	3.000005	3.999999	1.887	2.000000	sub-505.nii.gz
80	80	35	144	RPI	3.000001	2.999996	3.999999	6.980	2.000000	sub-506.nii.gz
80	80	35	144	RPI	3.000001	3.000000	4.000004	9.429	2.000000	sub-507.nii.gz
80	80	39	144	RPI	2.999999	2.999998	4.000004	9.190	2.000000	sub-508.nii.gz
...										

*inconsistent
matrix sizes*

*inconsistent
voxel sizes*

GTKYD - Getting information

```
nifti_tool -disp_hdr -field datatype \  
-field sform_code -field qform_code \  
-infiles mysubjects/func/*.nii.gz
```

```
$ nifti_tool -disp_hdr -field datatype \  
-infiles sub-5??/ses-01/func/sub-5??_ses-01_*.nii.gz | \  
grep datatype
```

```
datatype          70          1          16  
datatype          70          1          16  
datatype          70          1          16  
datatype          70          1          16  
datatype          70          1           4  
datatype          70          1           4  
datatype          70          1           4  
datatype          70          1           4
```

```
...
```

*inconsistent
data types
(float and
signed short)*



GTKYD - which subject is
not like the others?

```
gen_ss_review_table.py          \  
-outlier_sep space             \  
-report_outliers 'AFNI version' VARY \  
-report_outliers 'num regs of interest' VARY \  
-report_outliers 'final voxel resolution' VARY \  
-report_outliers 'num TRs per run' VARY \  
-infile sub*/s*.results/out.ss*.txt \  
-write_outliers outliers.vary.txt
```

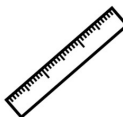
Outline of QC stages

1) GTKYD



*Getting To Know Your Data:
check dataset properties
and consistency*

2) APQUANT



*Quantitative review of basic
processing features:
censor fraction, GCOR, ...*



3) APQUAL



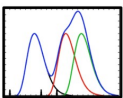
*Qualitative and systematic
visual checks via
afni_proc.py's QC HTML*

4) GUI



*In-depth investigation, via
the graphical user interface:
InstaCorr, graph viewer, ...*

5) STIM



*Task-specific investigations
of stimulus timing: event
timing, duration, ...*

AP Quantitative review

- Motion/censoring counts
- GCOR estimates, TSNR, ...
- Tools:
 - `gen_ss_review_table.py`
- Example scripts

APQUANT - exclusion/warning

```
gen_ss_review_table.py          \  
-outlier_sep space              \  
-report_outliers 'final DF fraction'    LE 0.6    \  
-report_outliers 'censor fraction'      GE 0.2    \  
-report_outliers 'average censored motion' GE 0.15  \  
-report_outliers 'max censored displacement' GE 8    \  
-report_outliers 'global correlation (GCOR)' GE 0.20  \  
-report_outliers 'flip guess'          EQ DO_FLIP  \  
-infiles      ${all_infiles}          \  
|& tee GSSRT/table_gssrt_Group_${gg}_exclude.dat
```

```
Subject final DF fraction censor fraction ... flip guess  
SHOW    LE:0.6          GE:0.2          EQ:DO_FLIP  
sub-507 0.277778      0.618056  
sub-511 0.479167      0.416667  
sub-512          0.250000  
sub-518          DO_FLIP  
sub-519 0.409722      0.486111      DO_FLIP
```

Single, scriptable command
of quantitative exclusion
criteria:

too many points censored,
too few degrees of freedom,
high global correlation, ...
for individual subjects or
groups

Simple output table listing
subjects to exclude or
warning limits

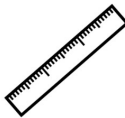
Outline of QC stages

1) GTKYD



*Getting To Know Your Data:
check dataset properties
and consistency*

2) APQUANT



*Quantitative review of basic
processing features:
censor fraction, GCOR, ...*

3) APQUAL



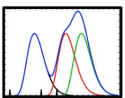
*Qualitative and systematic
visual checks via
afni_proc.py's QC HTML*

4) GUI



*In-depth investigation, via
the graphical user interface:
InstaCorr, graph viewer, ...*

5) STIM



*Task-specific investigations
of stimulus timing: event
timing, duration, ...*

AP Qualitative review

- Visualize data for “quick review”
 - data properties/coverage
 - processing success (e.g., align)
 - spatiotemporal aspects (stats/corr)
- Tools
 - afni_proc.py generated html pages
 - afni_open QC/index.html
 - open_apqc.py !

open_apqc.py

HTML interface with server for python and javascript

allows for in-browser viewing, saving forms and driving the AFNI GUI

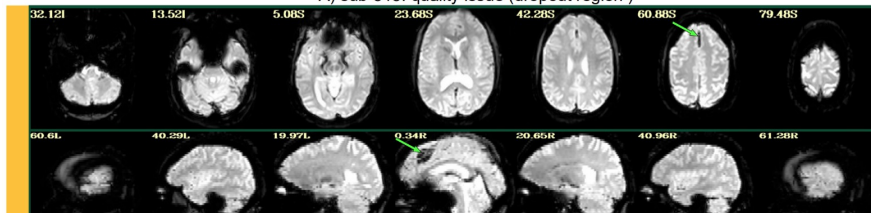
Simple example to open many tabs for every subject in a study all on the same kind of image.

```
open_apqc.py \
  -infile data_21_ap/sub*/*results/QC_*/index.html \
  -jump_to vstat
```

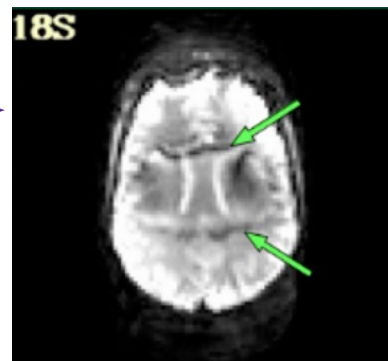
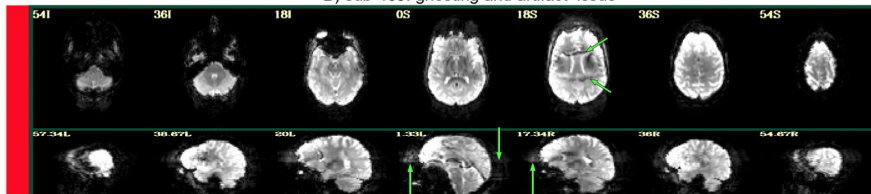
APQUAL: *vorig*

APQUAL: *vorig* (views of original data)

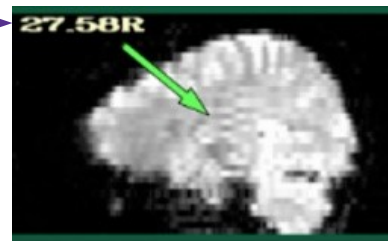
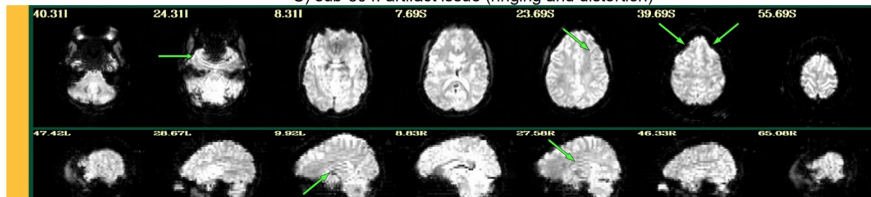
A) sub-315: quality issue (dropout region)



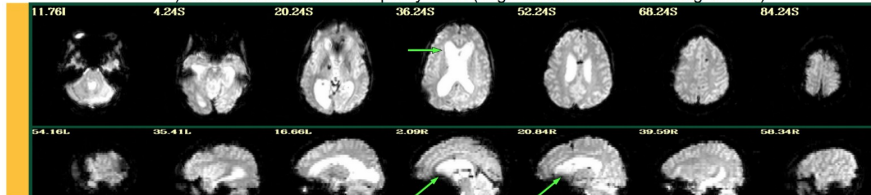
B) sub-409: ghosting and artifact issue



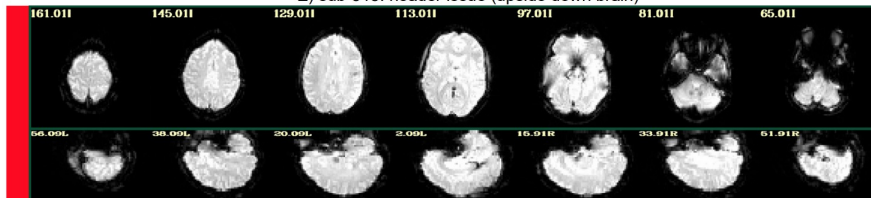
C) sub-504: artifact issue (ringing and distortion)



D) sub-509: anatomical and quality issue (large ventricles and frontal signal loss)



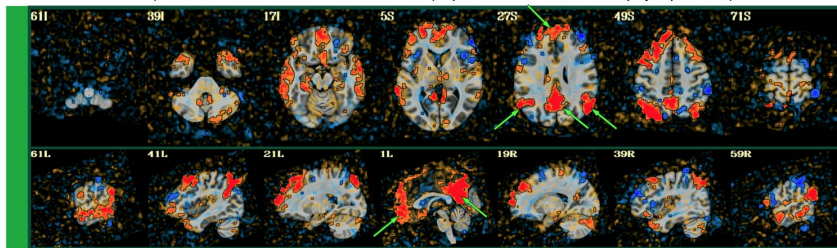
E) sub-518: header issue (upside down brain)



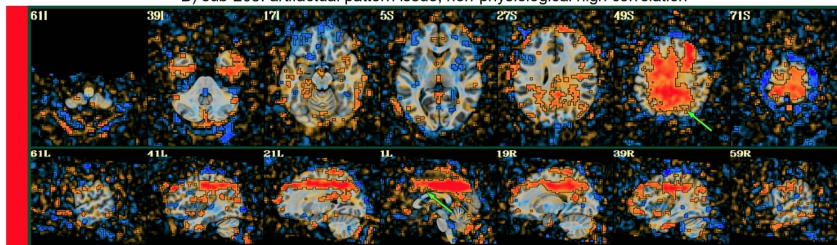
APQUAL: *vstat* (DMN)

APQUAL: *vstat* DMN (views of statistics: seed-based corr of DMN from L-PCC)

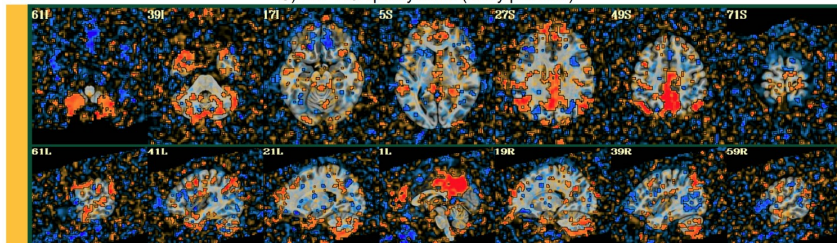
A) sub-505: seed-based correlation OK (expected network, no non-phys. patterns)



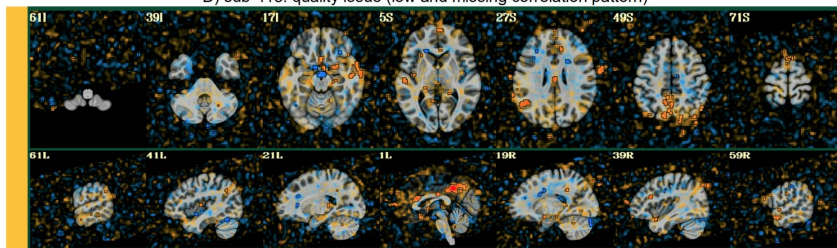
B) sub-203: artifactual pattern issue, non-physiological high correlation



C) sub-118: quality issue (noisy patterns)



D) sub-413: quality issue (low and missing correlation pattern)

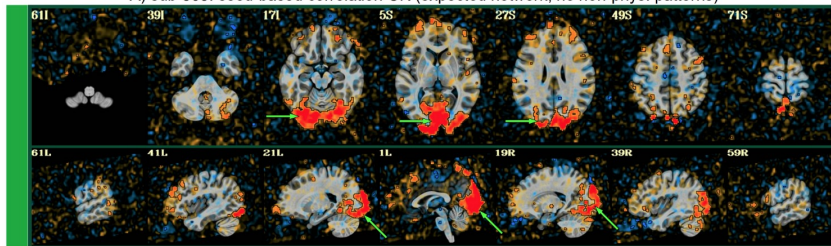


clay: -0.6 0.6 (Pearson r)
thr : 0.3 (alpha+boxed on)

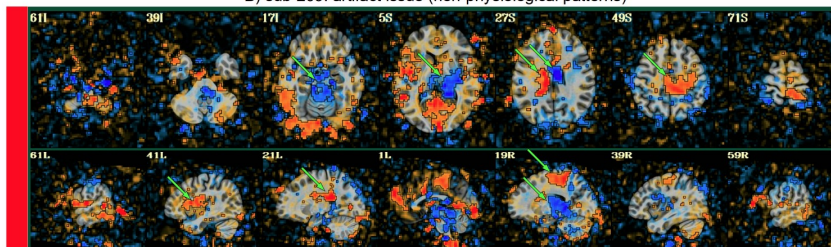
APQUAL: *vstat (vis)*

APQUAL: *vstat vis* (views of statistics: seed-based corr of visual cortex)

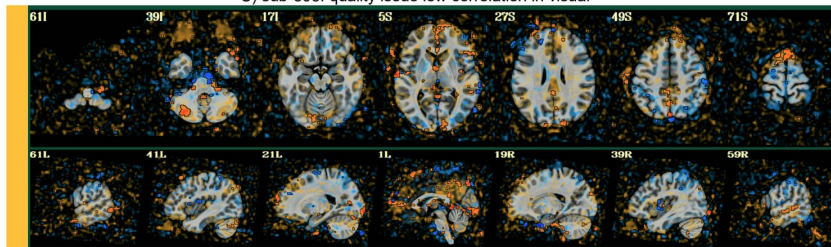
A) sub-505: seed-based correlation OK (expected network, no non-phys. patterns)



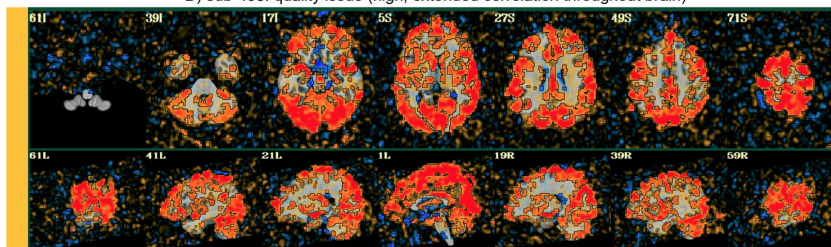
B) sub-209: artifact issue (non-physiological patterns)



C) sub-305: quality issue low correlation in visual



D) sub-403: quality issue (high, extended correlation throughout brain)

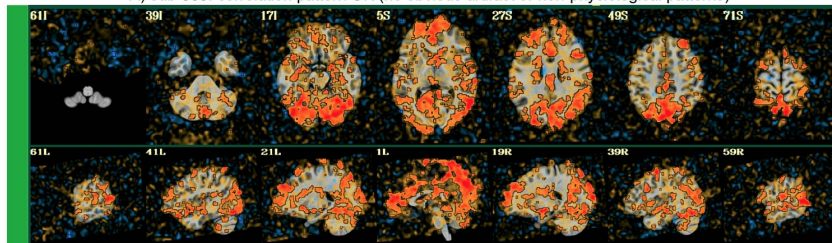


clay: -0.6 0.6 (Pearson r)
thr : 0.3 (alpha+boxed on)

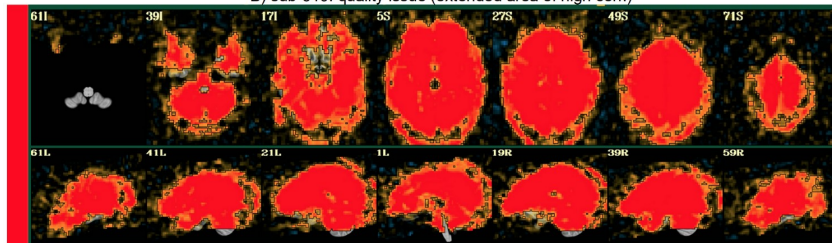
APQUAL: *corr_brain*

APQUAL: regr. corr_brain (correlation of WB-ave residual time series)

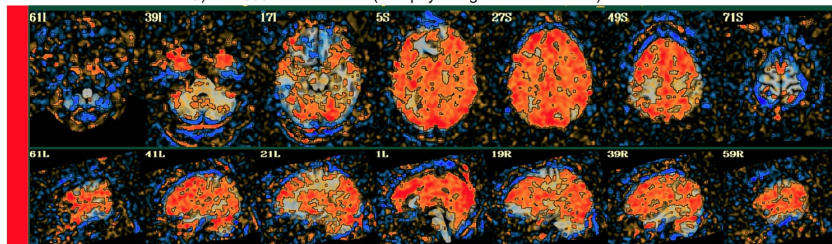
A) sub-505: correlation pattern OK (no obvious artifact or non-physiological patterns)



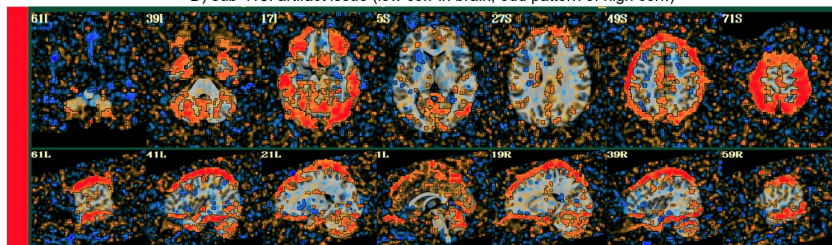
B) sub-610: quality issue (extended area of high corr.)



C) sub-508: artifact issue (non-physiological corr/anticorr.)

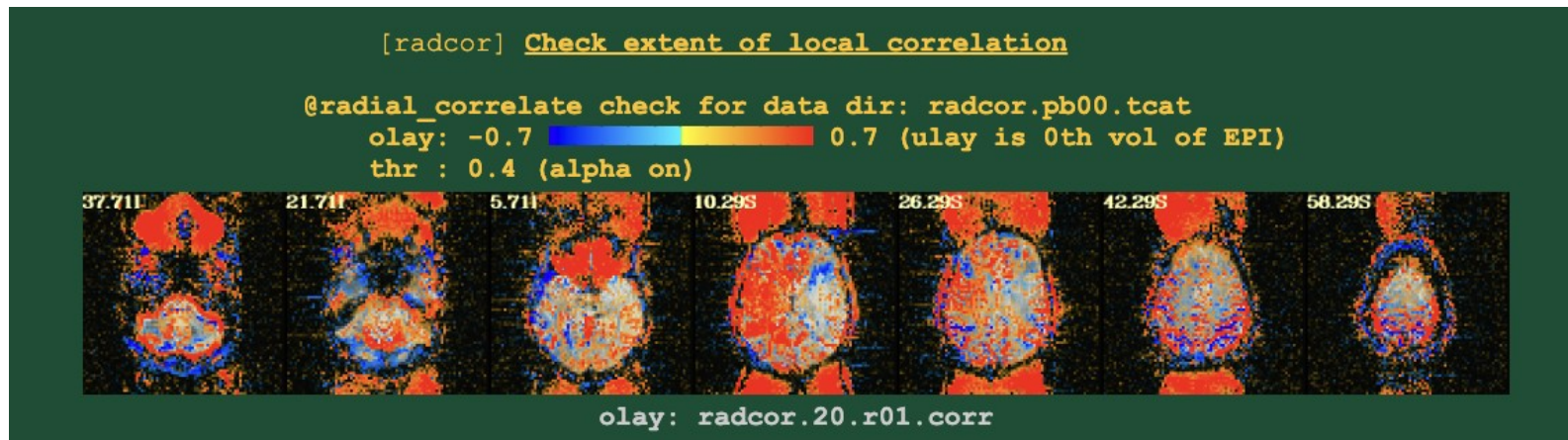
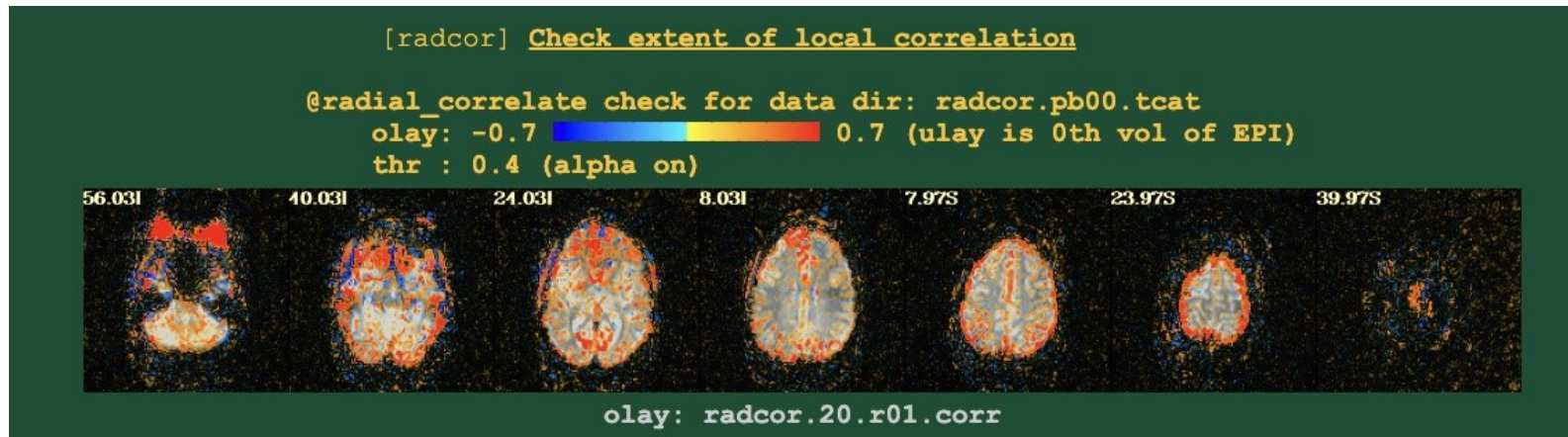


D) sub-118: artifact issue (low corr in brain, odd pattern of high corr.)



play: -0.6 0.6 (Pearson r)
thr : 0.3 (alpha+boxed on)

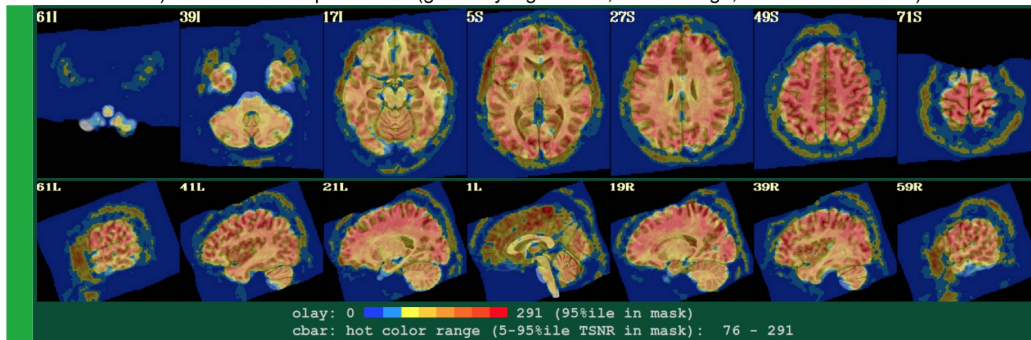
APQUAL: *radcor*



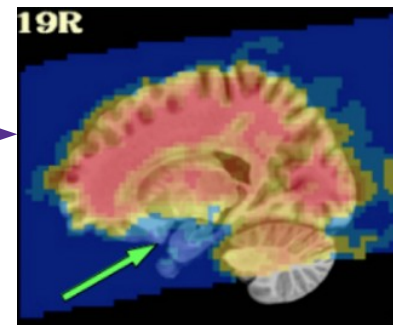
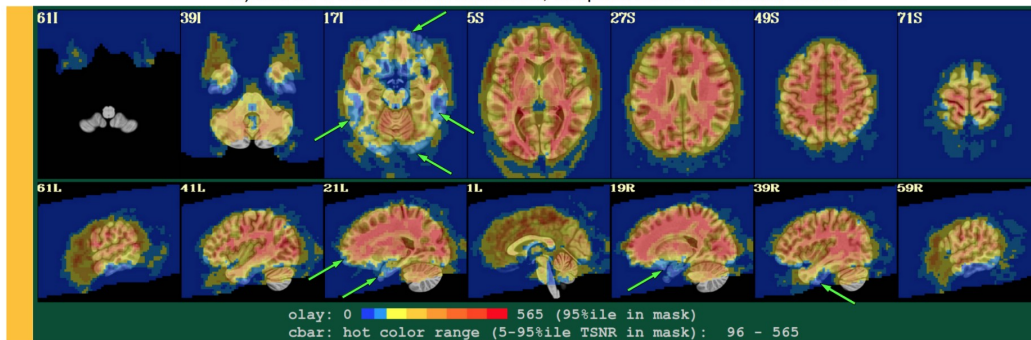
APQUAL: TSNR

APQUAL: regr. TSNR (temporal SNR for regressed time series)

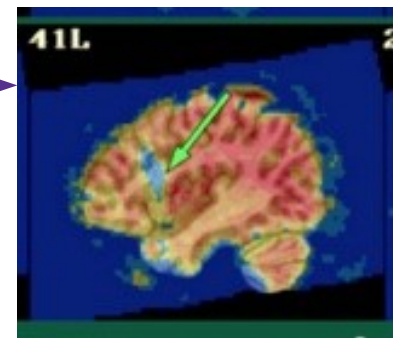
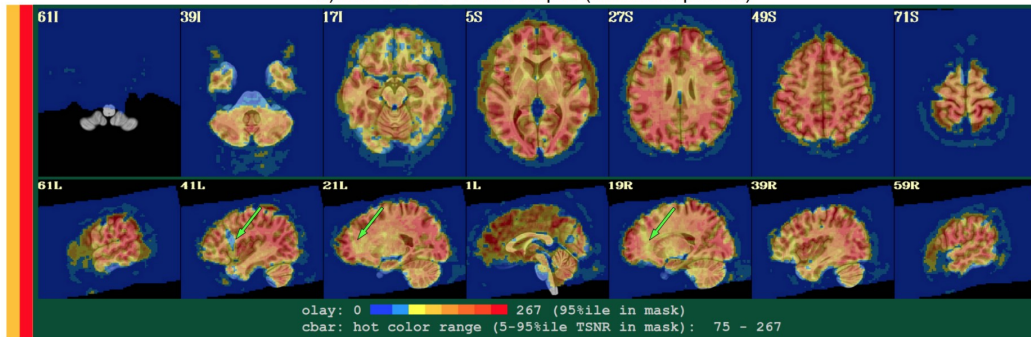
A) sub-313: TSNR pattern OK (generally high TSNR, full coverage, no obvious artifacts)



B) sub-614: low TSNR in subcortex, temporal lobes and frontal cortex



C) sub-403: low TSNR stripes (artificial pattern)



APQUAL: *variance lines*

EPI variance lines warnings

medium

Lines per run : 4 3 5
Intersecting : 3

Coordinates (see images of the first 7, below, check locations with InstaCorr)

```
-----  
Intersecting all          r01          r02          ...  
-----  
-2.70 -40.30 30.35    -2.70 -40.30 30.35    -2.70 -40.30 30.35  
-41.20 -15.50 30.35    -41.20 -12.80 30.35    -41.20 -15.50 30.35  
-8.20 -65.00 30.35    -8.20 -65.00 30.35    -8.20 -65.00 30.35  
-27.40 -59.50 30.35
```

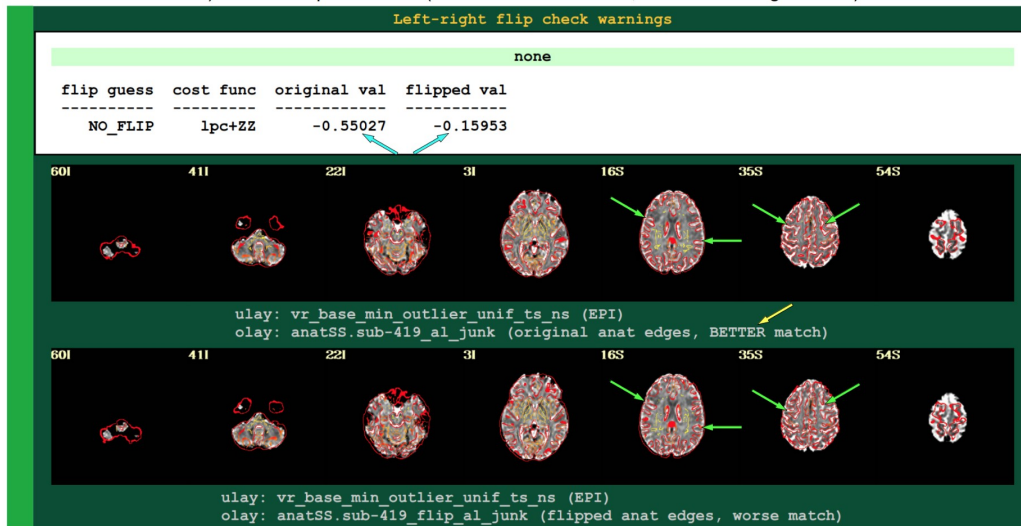


```
ulay: vlines.pb00.tcat/var*scale*.nii.gz (scaled variance per run)  
olay: line markers for inter:1, inter:2, inter:3, r01:1, r01:2, r01:3, r01:4
```

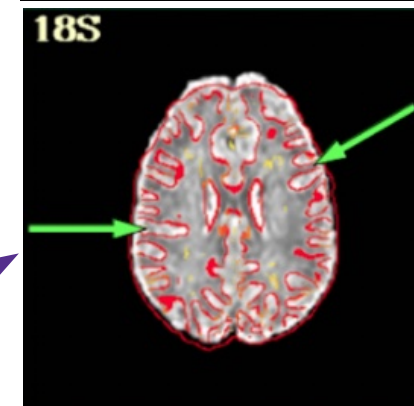
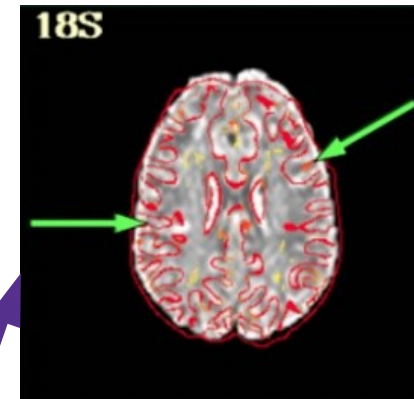
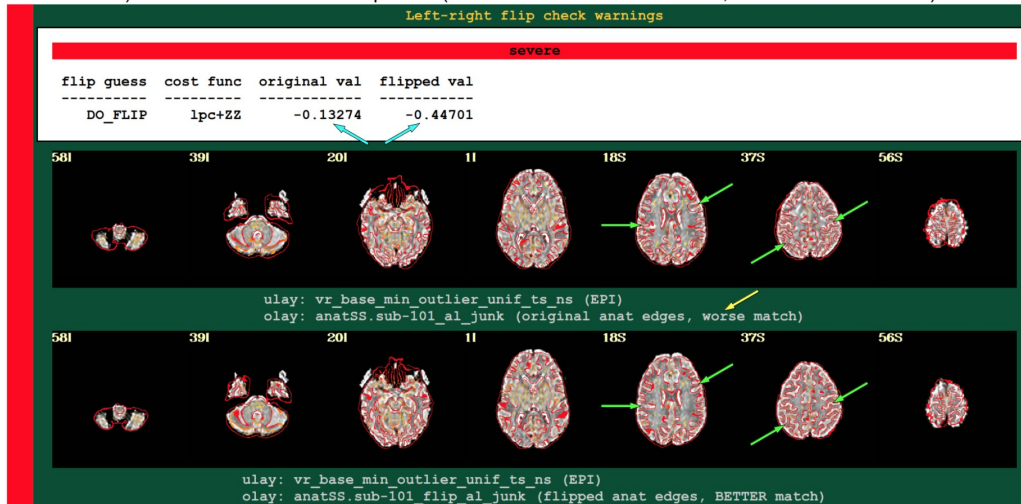
APQUAL: LR flips

APQUAL: Left-right flip check warns (EPI-anatomical agreement)

A) sub-606: flip check OK (cortical features match, cost function agreement)



B) sub-101: EPI-anatomical flip issue (relative cortical structure match, cost function evidence)



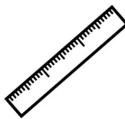
Outline of QC stages

1) GTKYD



*Getting To Know Your Data:
check dataset properties
and consistency*

2) APQUANT



*Quantitative review of basic
processing features:
censor fraction, GCOR, ...*

3) APQUAL



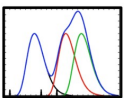
*Qualitative and systematic
visual checks via
afni_proc.py's QC HTML*

4) GUI



*In-depth investigation, via
the graphical user interface:
InstaCorr, graph viewer, ...*

5) STIM



*Task-specific investigations
of stimulus timing: event
timing, duration, ...*

Graphical User Interface

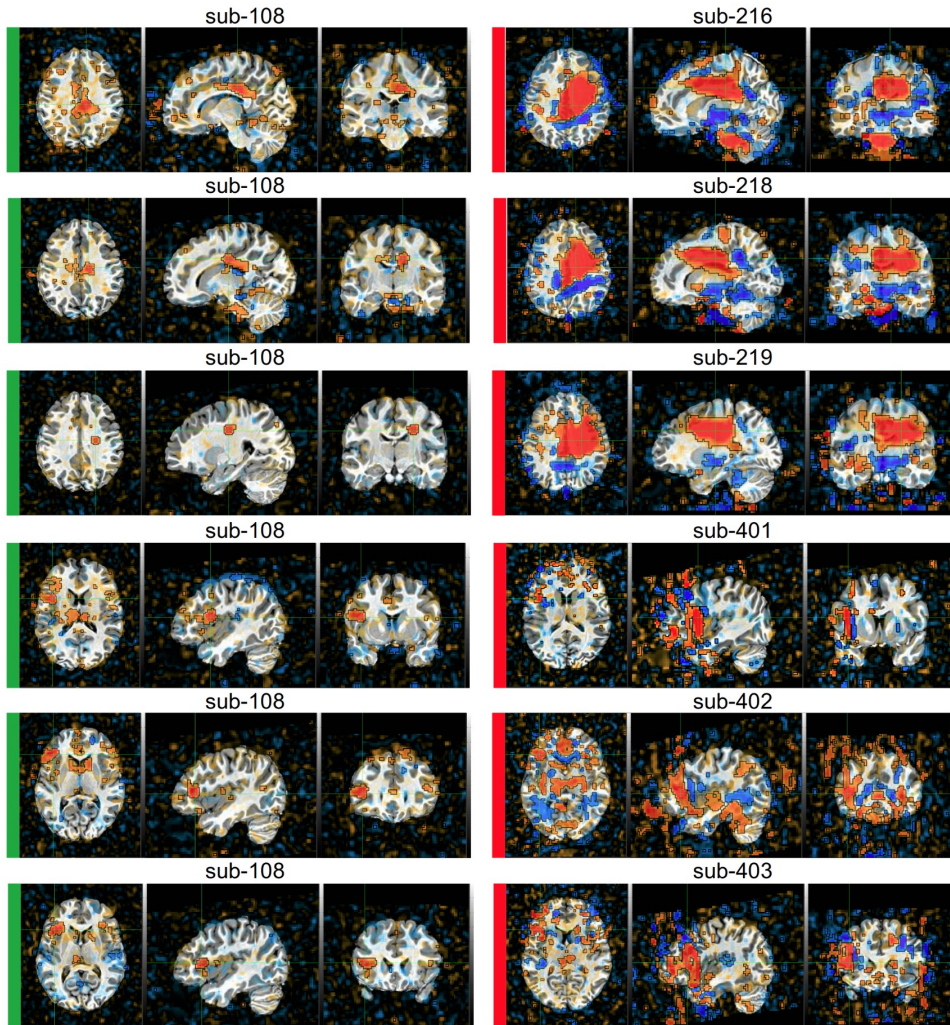
- Some datasets need more in-depth follow-ups to determine what is there
- Visualize with deep dives, now more efficiently connected within AFNI QC HTML
- Tools
 - from HTML pages, afni or NiiVue
 - afni GUI, suma GUI, InstaCorr

GUI: InstaCorr examples for Groups 1, 2 and 4

GUI: InstaCorr

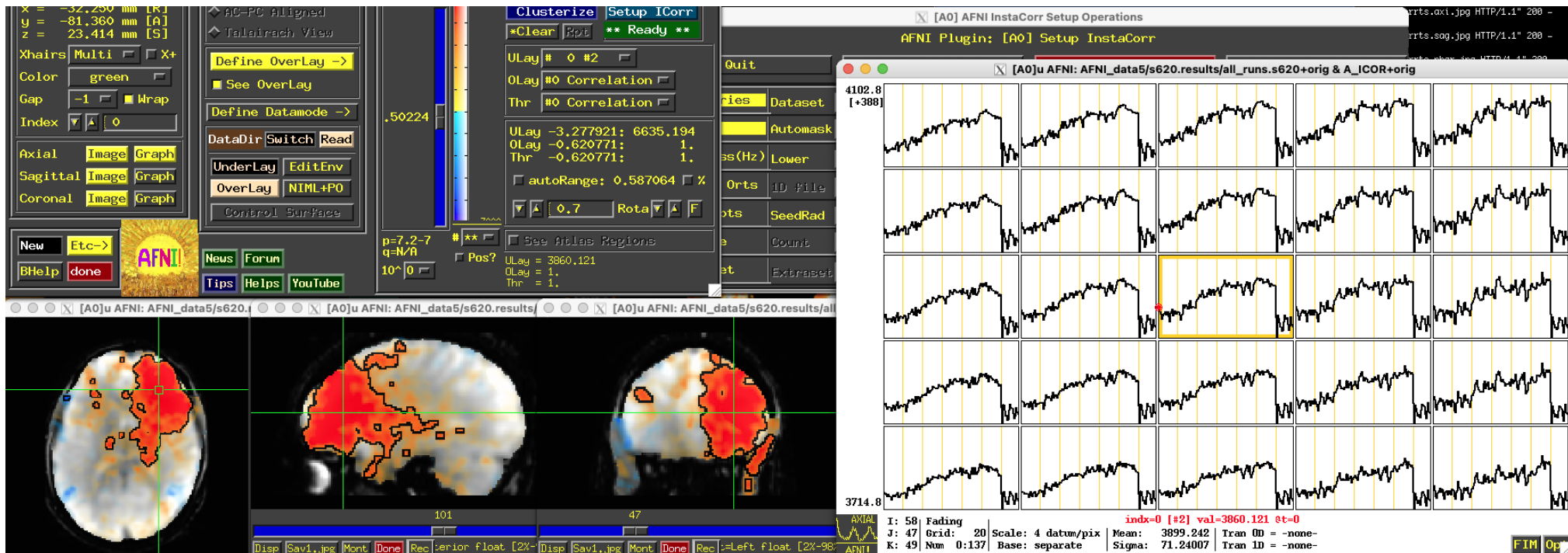
Seed-based correlation: OK

Seed-based correlation: artifact issues



play: -0.6 0.6 (Pearson r)
thr : 0.3 (alpha+boxed on)

GUI: InstaCorr graphs



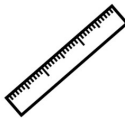
Outline of QC stages

1) GTKYD



*Getting To Know Your Data:
check dataset properties
and consistency*

2) APQUANT



*Quantitative review of basic
processing features:
censor fraction, GCOR, ...*

3) APQUAL



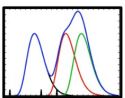
*Qualitative and systematic
visual checks via
afni_proc.py's QC HTML*

4) GUI



*In-depth investigation, via
the graphical user interface:
InstaCorr, graph viewer, ...*

5) STIM



*Task-specific investigations
of stimulus timing: event
timing, duration, ...*



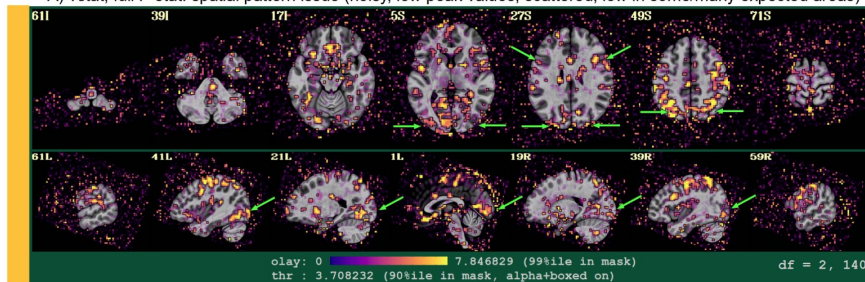
Stimulus timing (task FMRI)

- Motion plots
- Stimulus timing
- Stimulus correlation: was the model setup OK?
- Participant responses: was the task done?
- F-stats, t-stats: full model, individual stimulus response mapping

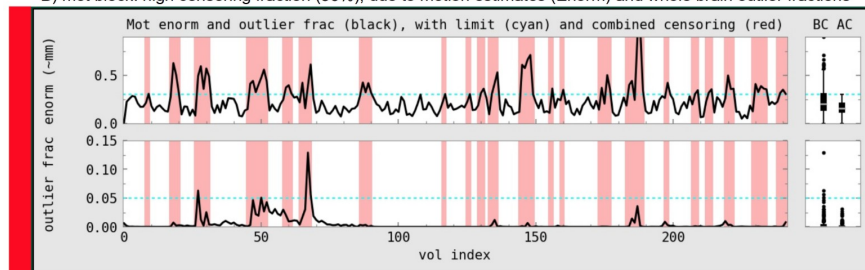
STIM: Full F-stat

APQUAL: sub-024 (stimulus-correlated motion?)

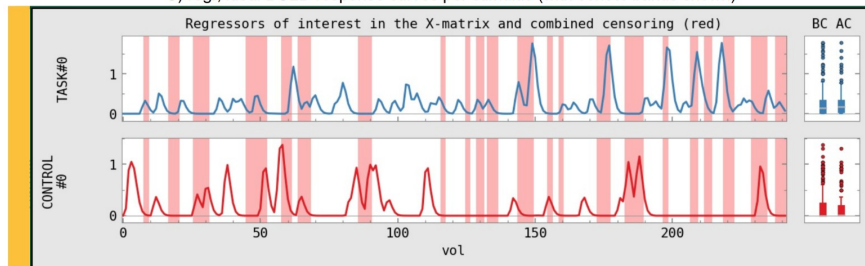
A) vstat, full F-stat: spatial pattern issue (noisy, low peak values, scattered, low in some/many expected areas)



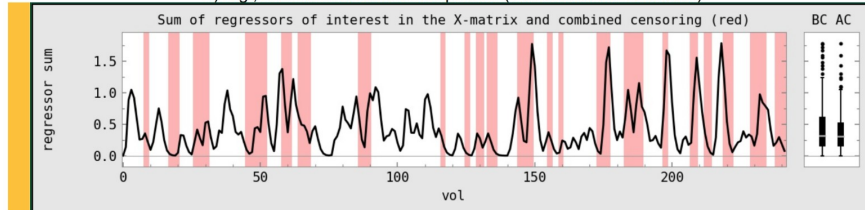
B) mot block: high censoring fraction (36%), due to motion estimates (Enorm) and whole brain outlier fractions



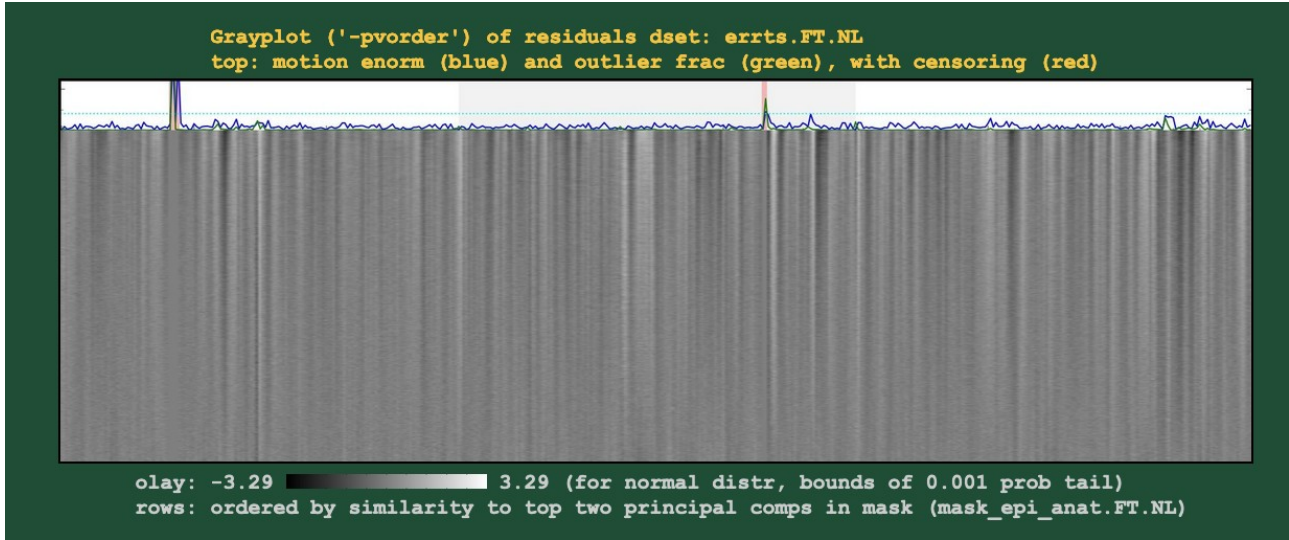
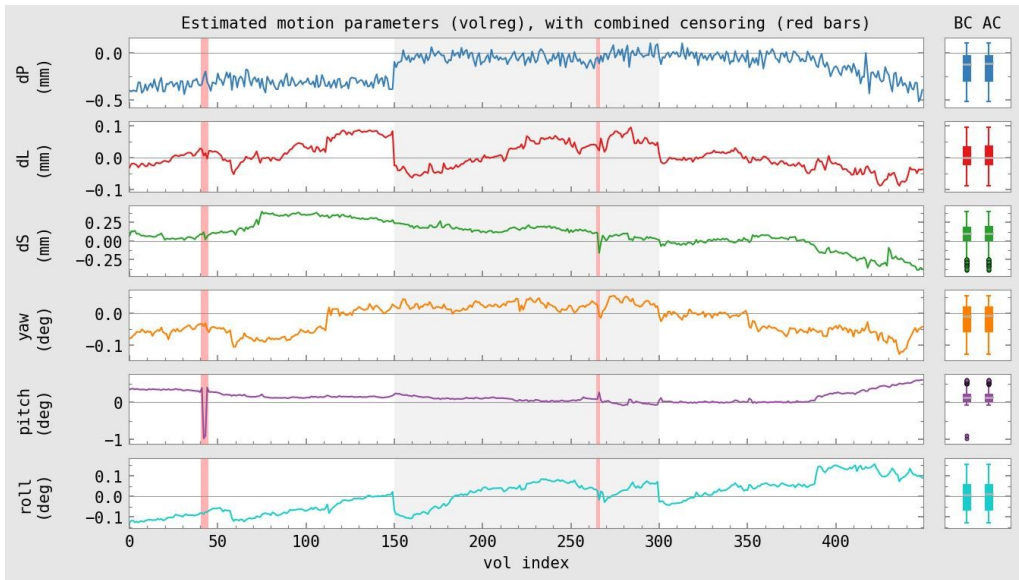
C) regr, ideal BOLD response curves per stimulus (with censor bands shown)



D) regr, sum of ideal stimulus responses (with censor bands shown)



other QC: *motion plots*



Summary

- Real-world data quality issues - often missed
- There are more problems out there than shown here
- There will be new problems with data
- AFNI has many tools, generally designed with the motto of:
“Helping users stay close to their data”
 - **Use QC steps to stay close to your own data!**

Acknowledgments

Rick Reynolds - the chief architect of afni_proc.py

Paul Taylor - quality is his middle name - chief QC reporter

Bob Cox - AFNI founder, developer of GUI and InstaCorr

Taylor Hanayik and Chris Rorden - NiiVue authors