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 <u>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.</u>

(Reynolds et al., 2023, Frontiers Open QC Project article)

Processing + QC = afni_proc.py (AP)

- AFNI's recommended pipeline for FMRI processing
  - Python program that generates full pipeline processing script
  - The "proc script" is 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

Creates 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

- Tabulate/compare dataset basics
  - Orientation, data type, voxel size, ...
  - min, max values, NIFTI properties, ...
- *Check consistency* across runs, sessions, subjects; missing/extra data
- Programs:
  - 3dinfo, nifti_tool, 3dBrickstat
  - gtkyd_check.py, gen_ss_review_table.py

older (more work for user)

newer (more convenient)

GTKYD - Getting information

Example: *Tabulate raw dataset properties (e.g., all subject EPIs).*

```
gtkyd_check.py                                        \
    -infiles  group_study/sub*/func*/epi*.nii.gz      \
    -outdir   group_summary
```

```
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

Example: *Which subject’s data are not like the others?*

- “infiles” are a set of simple text file dictionaries of properties
- “report_outliers” are list of properties to compare/search
  - Includes: >, >=, <, <=, !=, ==, varying across dsets
- Scriptable, shareable QC command to make a table of outlier subjects

```
gen_ss_review_table.py                              \
    -outlier_sep space                              \
    -report_outliers 'ad3'                 VARY     \
    -report_outliers 'ad3'                 GT 2.8   \
    -report_outliers 'orient'              VARY     \
    -report_outliers 'is_slice_timing_nz'  EQ 0     \
    -infiles group_summary/dset*.txt                \
    -write_outliers group_summary.vary.txt
```

AP Quantitative review

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, ...

- Particularly useful after running afni_proc.py (outputs useful summary file)
  - Check motion, censoring counts, ...
  - Compare GCOR, TSNR, ...
- Programs:
  - gen_ss_review_table.py

AP Quant – tabulate/search quantities

Example: *Single, scriptable command of quantitative exclusion criteria*

- Too few final degrees of freedom (DFs), too many time points censored
- Too much motion even after censoring, too high instance(s) of motion
- Too high global correlation (GCOR), appears to have left-right flipping

```
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}
```

→

AP Quant – tabulate/search quantities

→ **Output:** Simple output table listing subjects

- Might use to exclude or warn about
- Can rerun as more subjects are acquired
- Share with others and/or publish

```
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
```

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 generates HTML report
  - **open_apqc.py -infiles QC/index.html**

AP Qual – QC HTML

Versatile to use, and “open_apqc.py” starts local server running

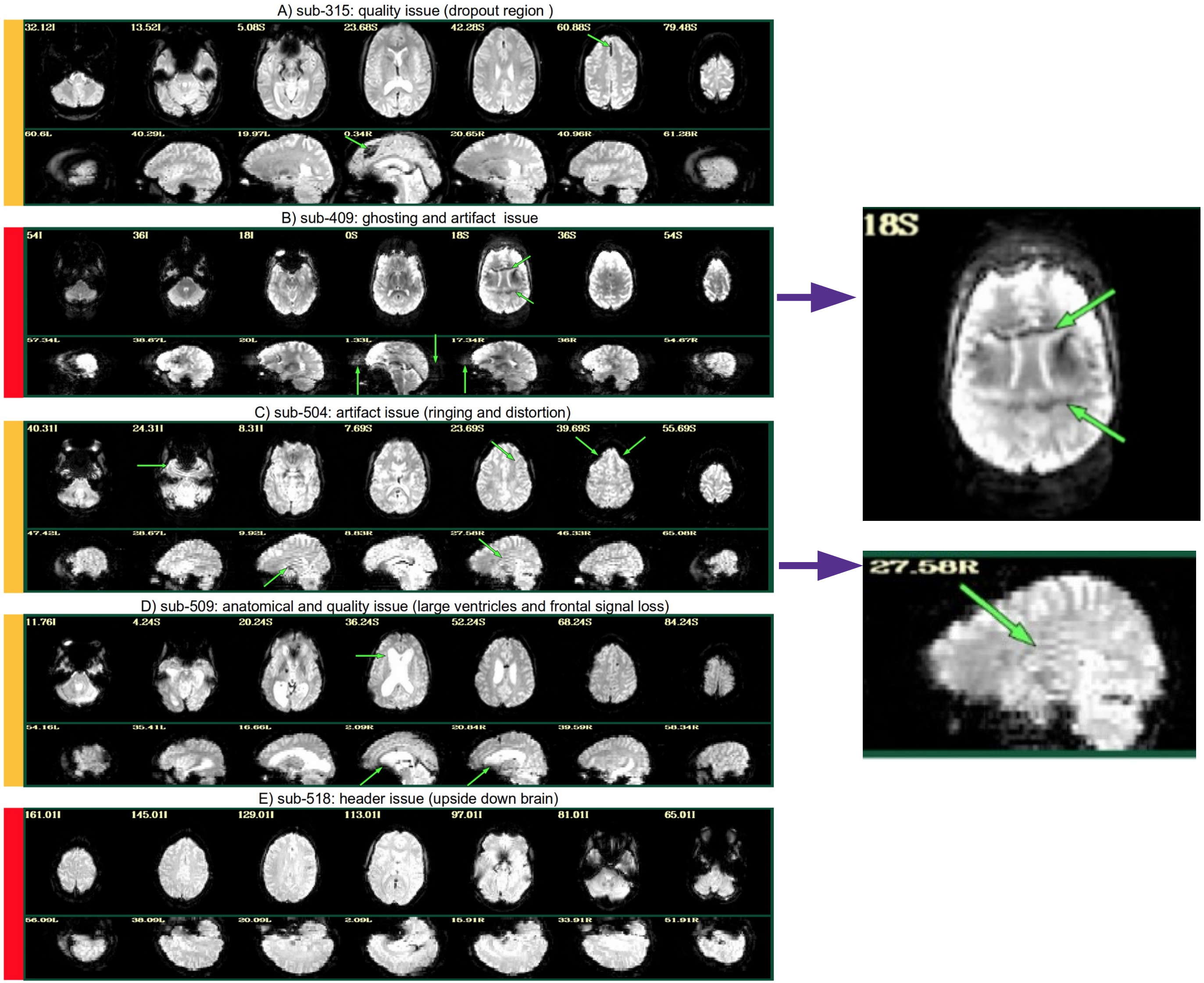
- View many properties across a given subject
- View the same property across many subjects (double-click gold text)
- Save QC ratings and notes instantly
- Fire up AFNI GUI and NiiVue viewer instances with single button clicks

```
# basic usage: open one HTML with local server running
open_apqc.py                                               \
  -infiles data_21_ap/sub-001/*results/QC_*/index.html

# open many subject HTMLs, jump all to same starting 
location:
open_apqc.py                                               \
  -infiles data_21_ap/sub*/*results/QC_*/index.html        \
  -jump_to vstat
```

Some APQC HTML examples (from the Open QC Project data)

APQUAL: *vorig*

APQUAL: vorig (views of original data)



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

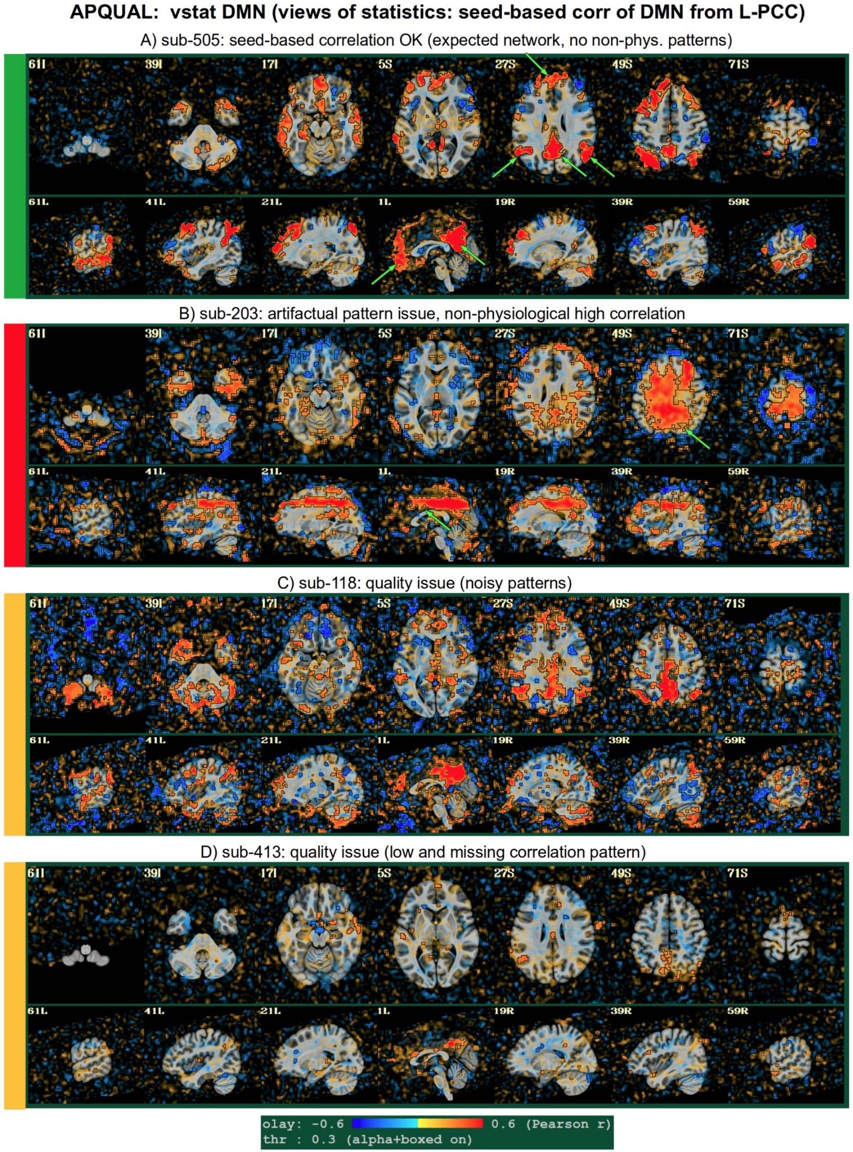
B) sub-409: ghosting and artifact issue

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)

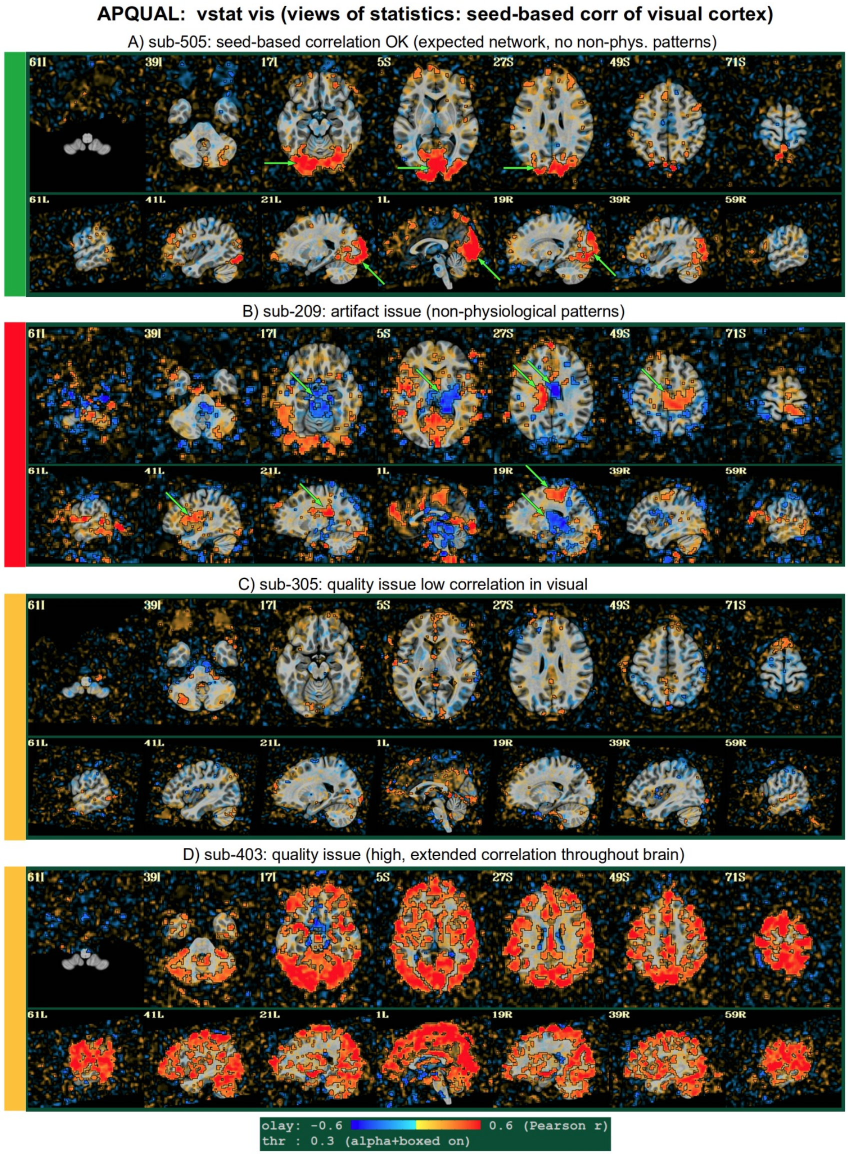
B) sub-203: artifactual pattern issue, non-physiological high correlation

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

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

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)

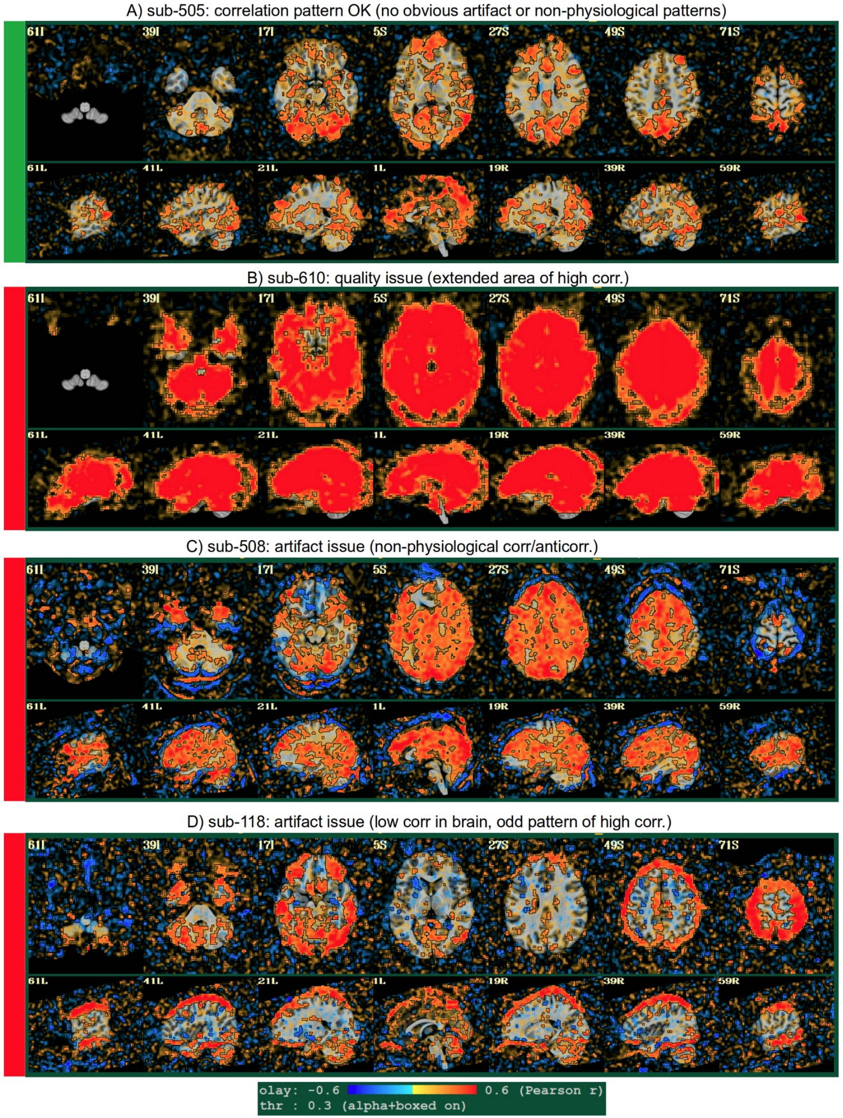
B) sub-209: artifact issue (non-physiological patterns)

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

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

APQUAL: *corr_brain*

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



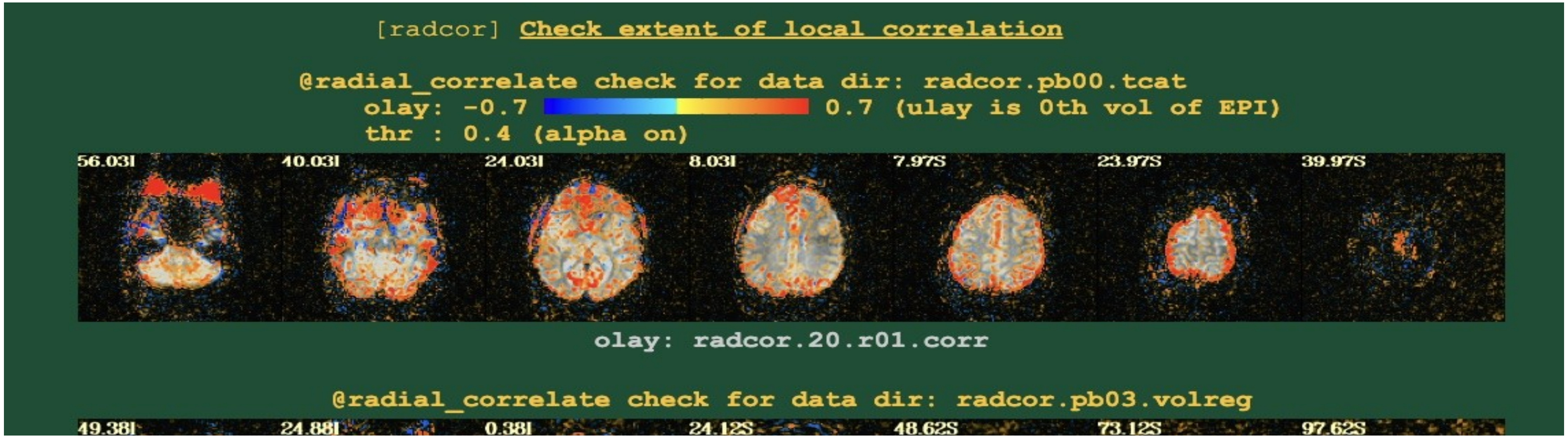
A) sub-505: correlation pattern OK (no obvious artifact or non-physiological patterns)

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.)

APQUAL:
radcor

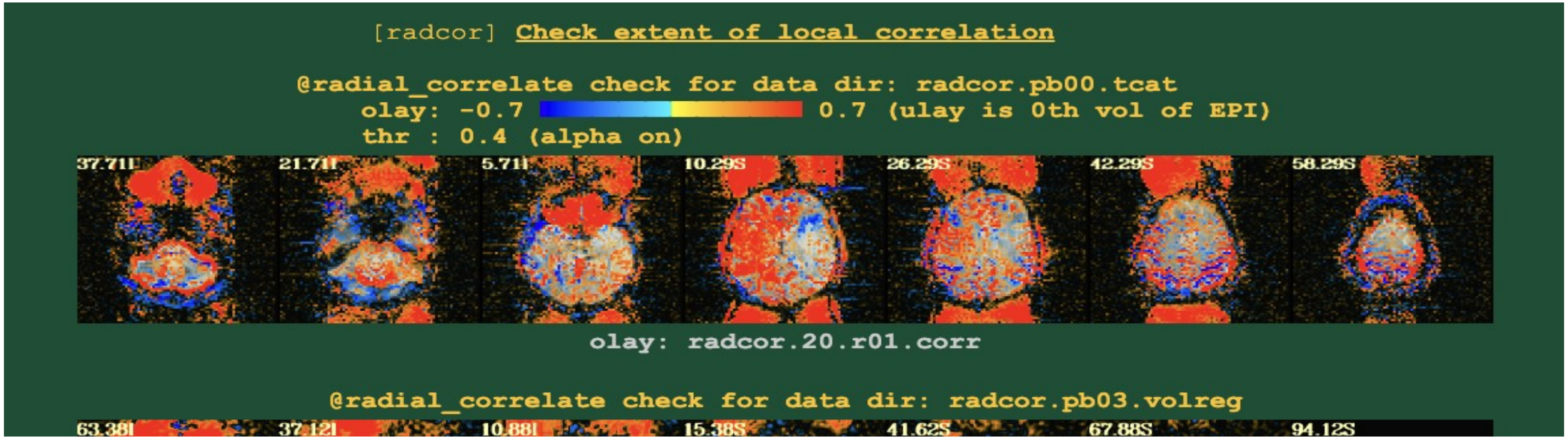


[radcor] Check extent of local correlation

@radial_correlate check for data dir: radcor.pb00.tcat
olay: -0.7 0.7 (ulay is 0th vol of EPI)
thr : 0.4 (alpha on)

olay: radcor.20.r01.corr

@radial_correlate check for data dir: radcor.pb03.volreg



[radcor] Check extent of local correlation

@radial_correlate check for data dir: radcor.pb00.tcat
olay: -0.7 0.7 (ulay is 0th vol of EPI)
thr : 0.4 (alpha on)

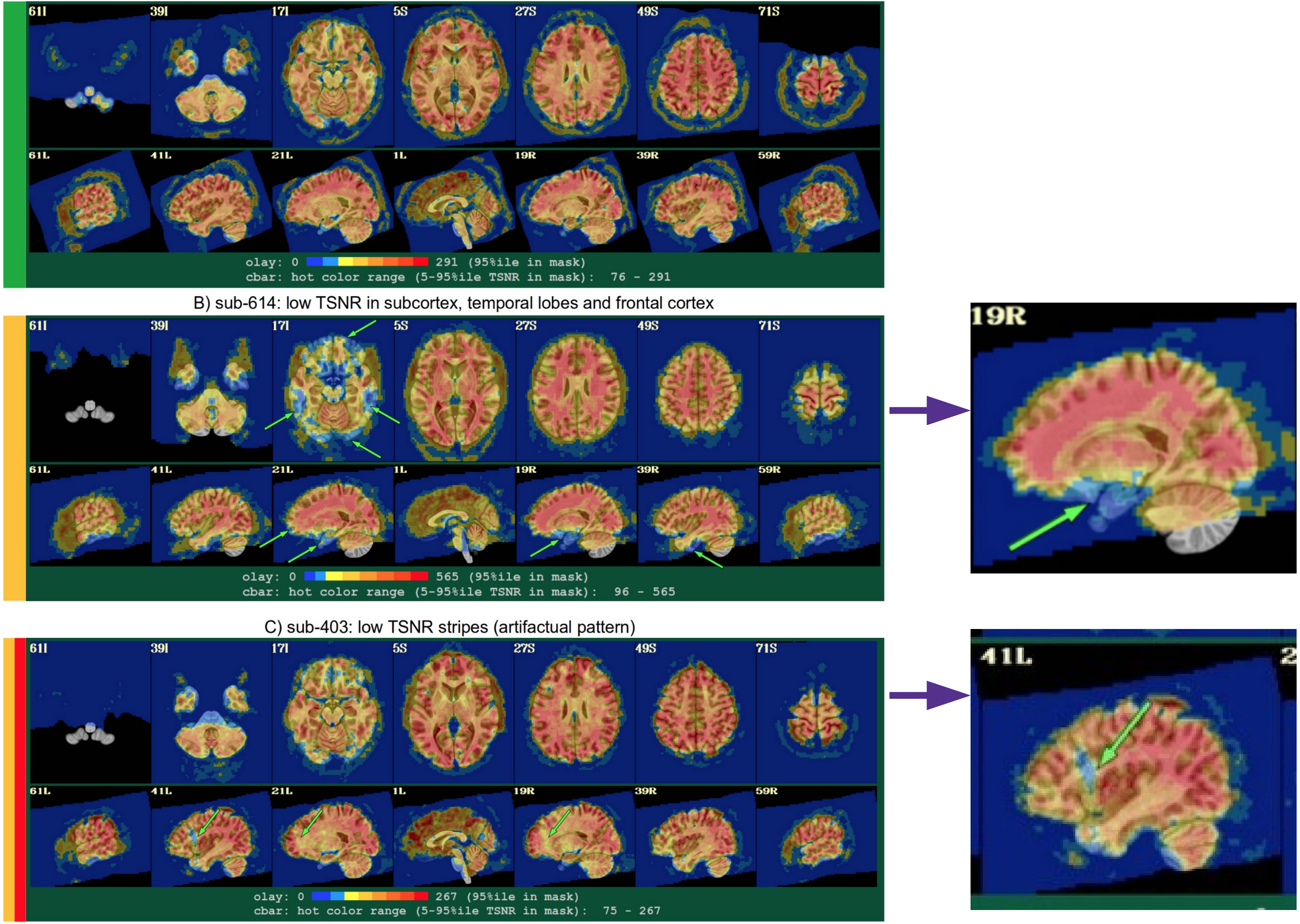
olay: radcor.20.r01.corr

@radial_correlate check for data dir: radcor.pb03.volreg

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)

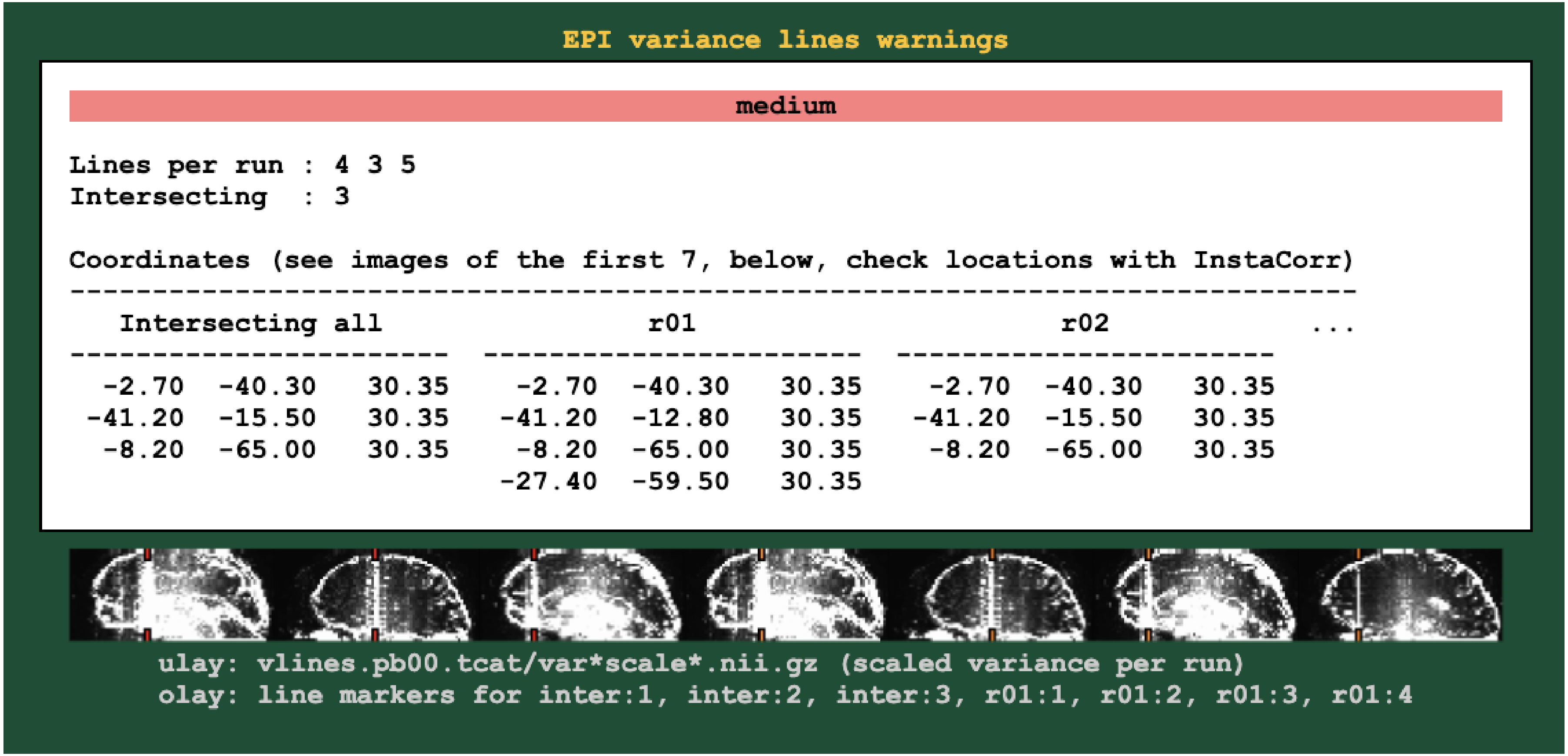


B) sub-614: low TSNR in subcortex, temporal lobes and frontal cortex

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

61I
39I
17I
5S
27S
49S
71S
61L
41L
21L
1L
19R
39R
59R
olay: 0 291 (95%ile in mask)
cbar: hot color range (5-95%ile TSNR in mask): 76 - 291
olay: 0 565 (95%ile in mask)
cbar: hot color range (5-95%ile TSNR in mask): 96 - 565
olay: 0 267 (95%ile in mask)
cbar: hot color range (5-95%ile TSNR in mask): 75 - 267
19R
41L

APQUAL:
variance lines

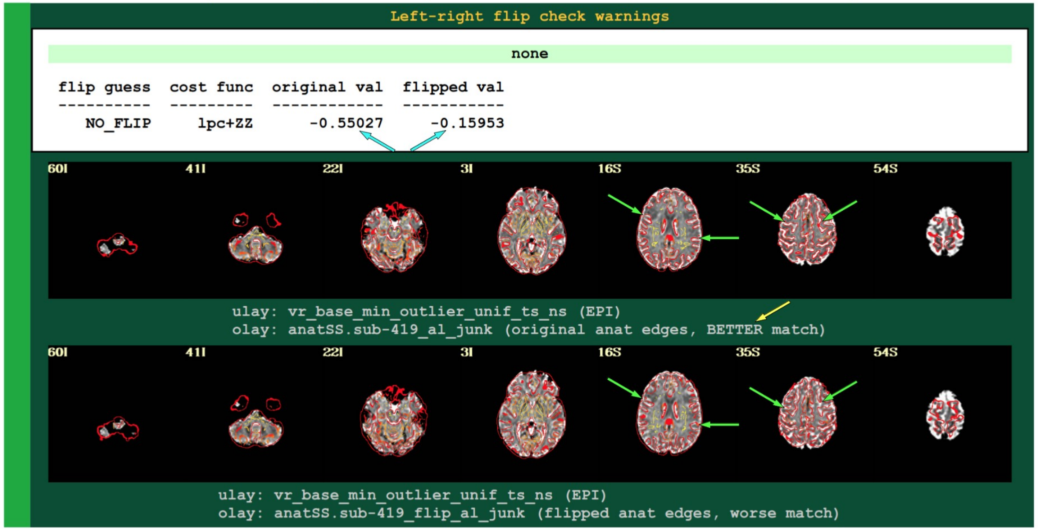


```
                         EPI variance lines warnings

                                   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)



Left-right flip check warnings

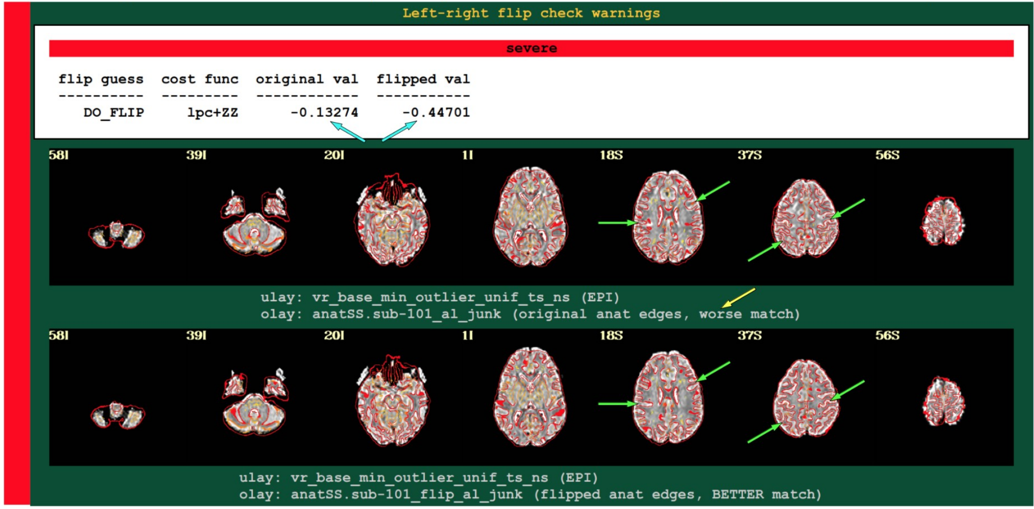
none

```
flip guess  cost func  original val  flipped val
----------  ---------  ------------  -----------
   NO_FLIP     lpc+ZZ      -0.55027     -0.15953
```

```
ulay: vr_base_min_outlier_unif_ts_ns (EPI)
olay: anatSS.sub-419_al_junk (original anat edges, BETTER match)
```

```
ulay: vr_base_min_outlier_unif_ts_ns (EPI)
olay: anatSS.sub-419_flip_al_junk (flipped anat edges, worse match)
```

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



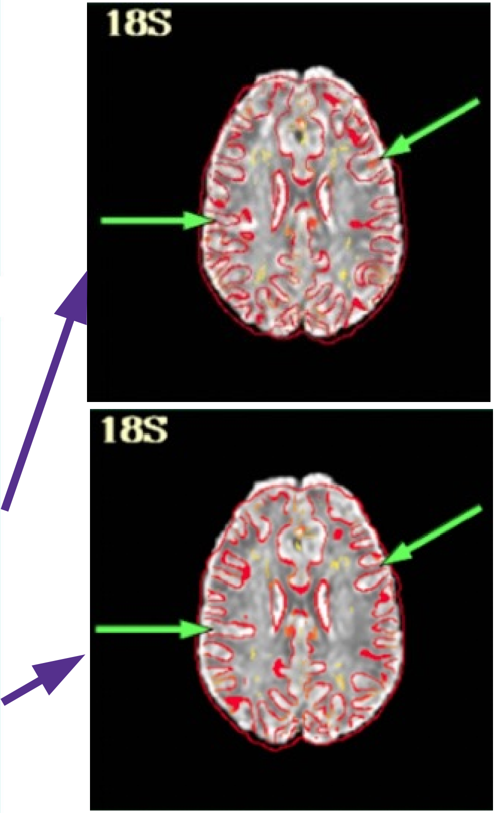
Left-right flip check warnings

severe

```
flip guess  cost func  original val  flipped val
----------  ---------  ------------  -----------
   DO_FLIP     lpc+ZZ      -0.13274     -0.44701
```

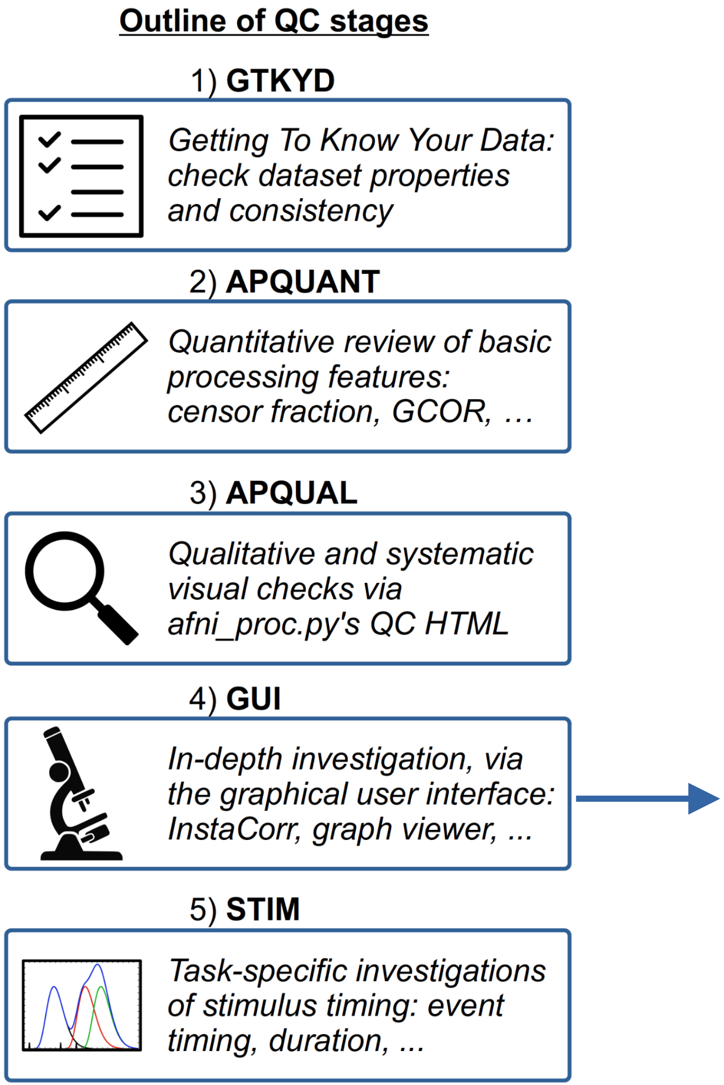
```
ulay: vr_base_min_outlier_unif_ts_ns (EPI)
olay: anatSS.sub-101_al_junk (original anat edges, worse match)
```

```
ulay: vr_base_min_outlier_unif_ts_ns (EPI)
olay: anatSS.sub-101_flip_al_junk (flipped anat edges, BETTER match)
```



(Glen et al., 2020)

Graphical User Interface



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**

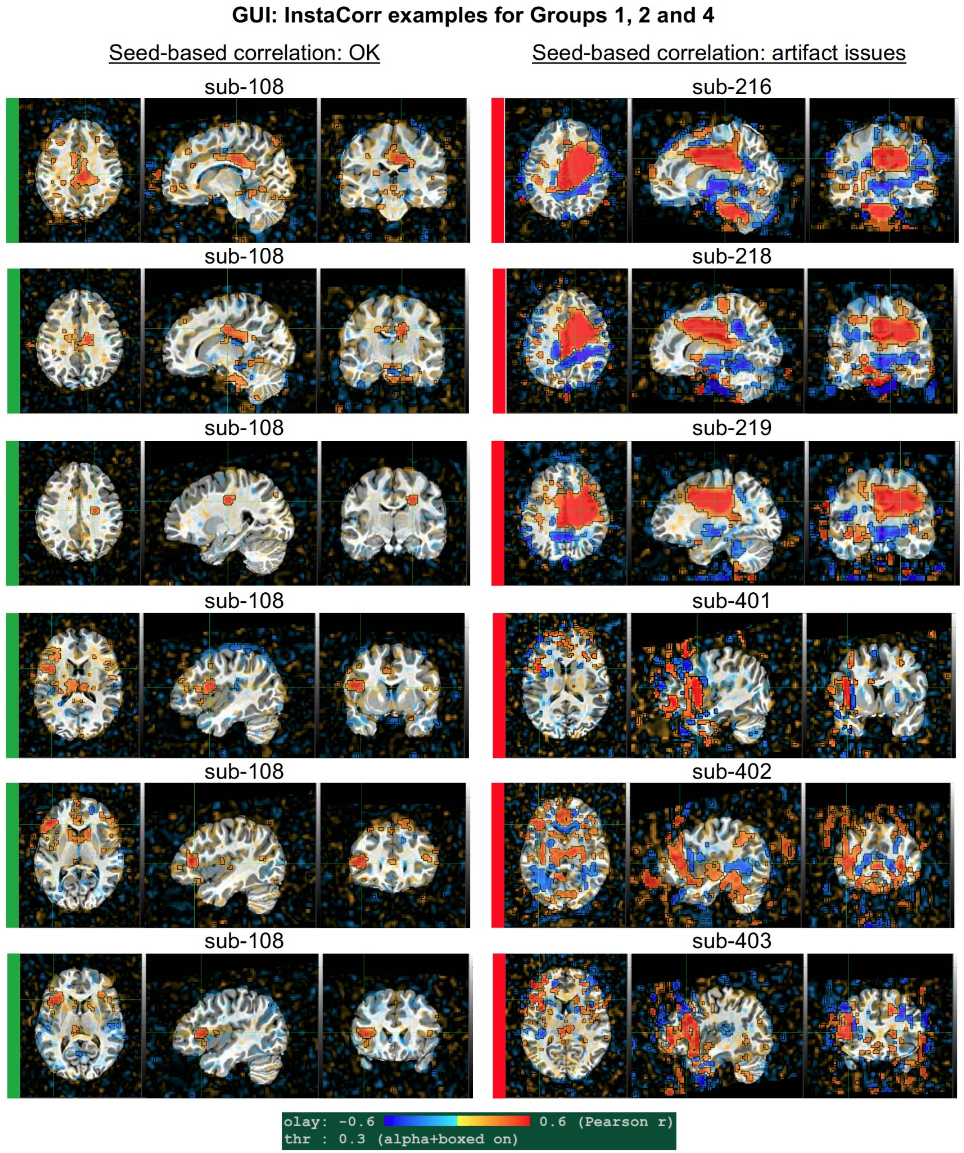
In-depth investigation, via the graphical user interface: InstaCorr, graph viewer, ...

5) **STIM**

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

- 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
  - APQC HTML pages, *afni* or *NiiVue*
  - *afni* GUI, *suma* GUI, *InstaCorr*

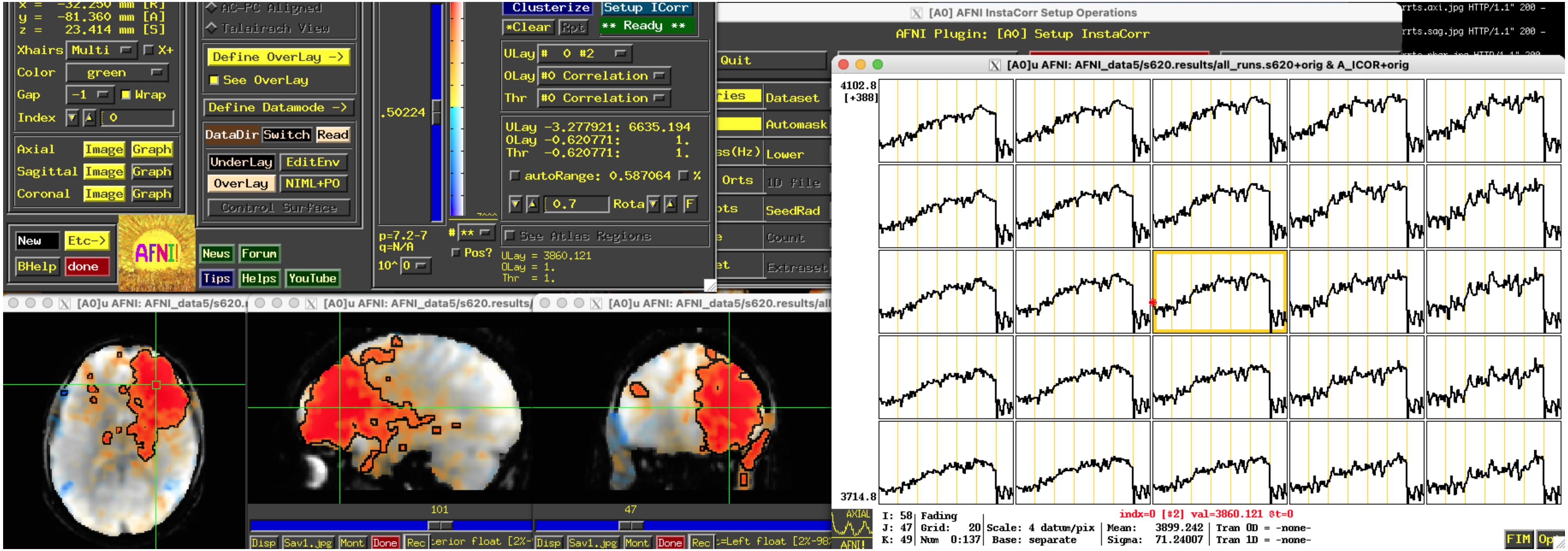
GUI: *InstaCorr*



GUI: InstaCorr examples for Groups 1, 2 and 4

Seed-based correlation: OK
Seed-based correlation: artifact issues
sub-108
sub-216
sub-108
sub-218
sub-108
sub-219
sub-108
sub-401
sub-108
sub-402
sub-108
sub-403
olay: -0.6 0.6 (Pearson r)
thr : 0.3 (alpha+boxed on)

GUI:
InstaCorr graphs



x = -32.250 mm [R]
y = -81.360 mm [A]
z = 23.414 mm [S]

Xhairs Multi □ X+
Color green
Gap -1 ■ Wrap
Index 0

ULay # 0 #2
OLay #0 Correlation
Thr #0 Correlation

```
ULay -3.277921: 6635.194
OLay -0.620771:        1.
Thr  -0.620771:        1.
```

□ autoRange: 0.587064 □ %

0.7 Rota

.50224

p=7.2-7
q=N/A

10^ 0

ULay = 3860.121
OLay = 1.
Thr = 1.

[A0] AFNI InstaCorr Setup Operations

AFNI Plugin: [A0] Setup InstaCorr

[A0]u AFNI: AFNI_data5/s620.results/all_runs.s620+orig & A_ICOR+orig

4102.8
[+388]

3714.8

indx=0 [#2] val=3860.121 @t=0

```
I: 58  Fading            Scale: 4 datum/pix  Mean:   3899.242  Tran 0D = -none-
J: 47  Grid:      20     Base: separate      Sigma:  71.24007  Tran 1D = -none-
K: 49  Num    0:137
```

[A0]u AFNI: AFNI_data5/s620.

[A0]u AFNI: AFNI_data5/s620.results/

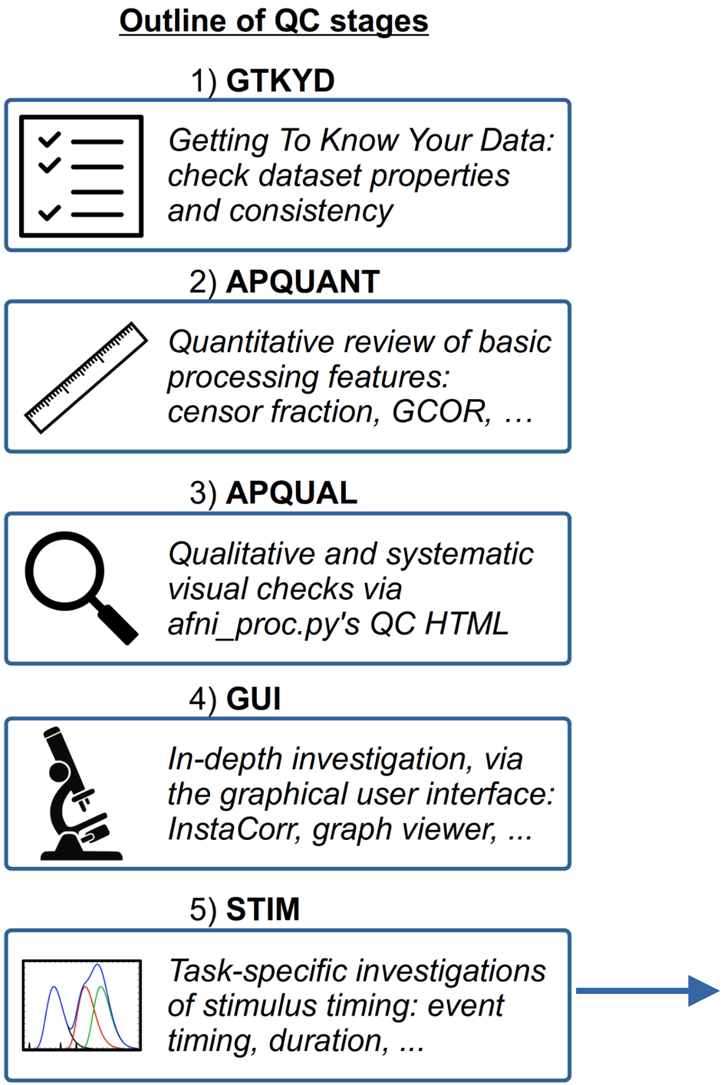
[A0]u AFNI: AFNI_data5/s620.results/all

101

47

rrts.axi.jpg HTTP/1.1" 200 -
rrts.sag.jpg HTTP/1.1" 200 -

Stimulus timing (task FMRI)



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, ...

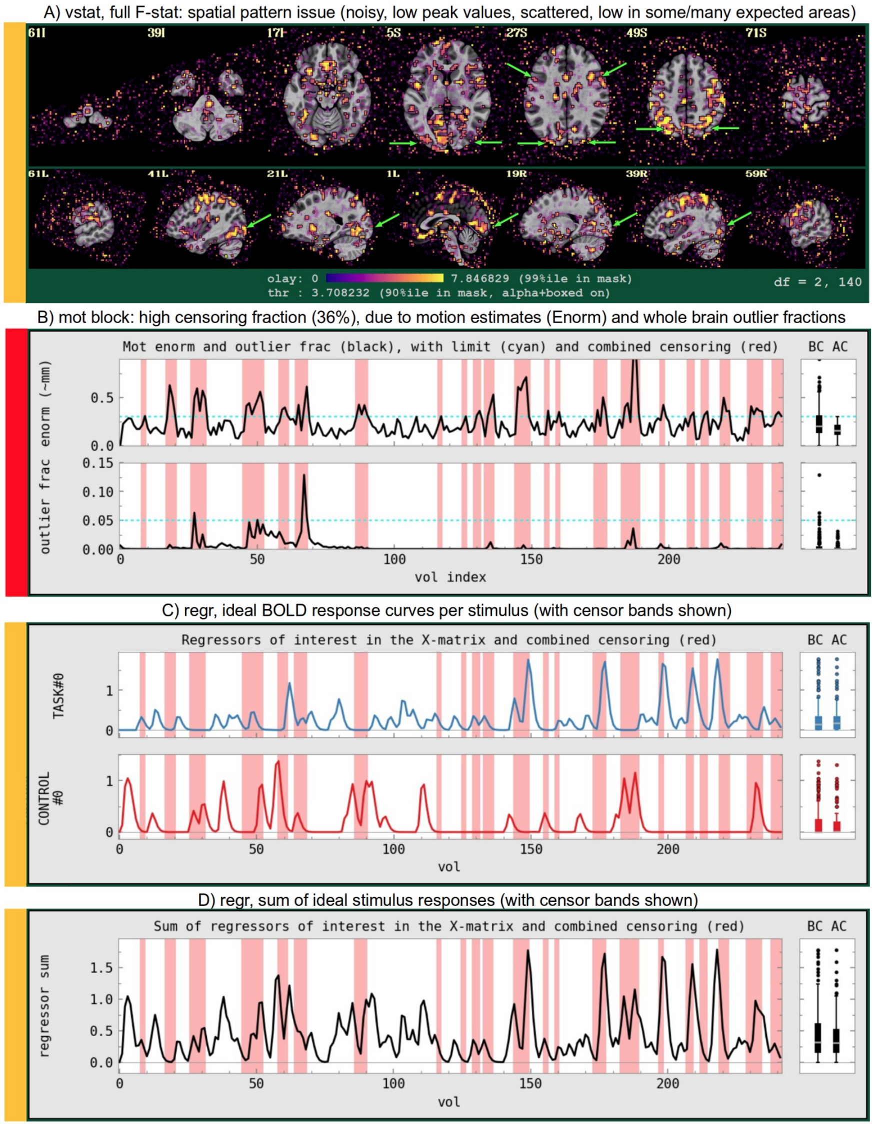
5) **STIM**

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

- 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?)



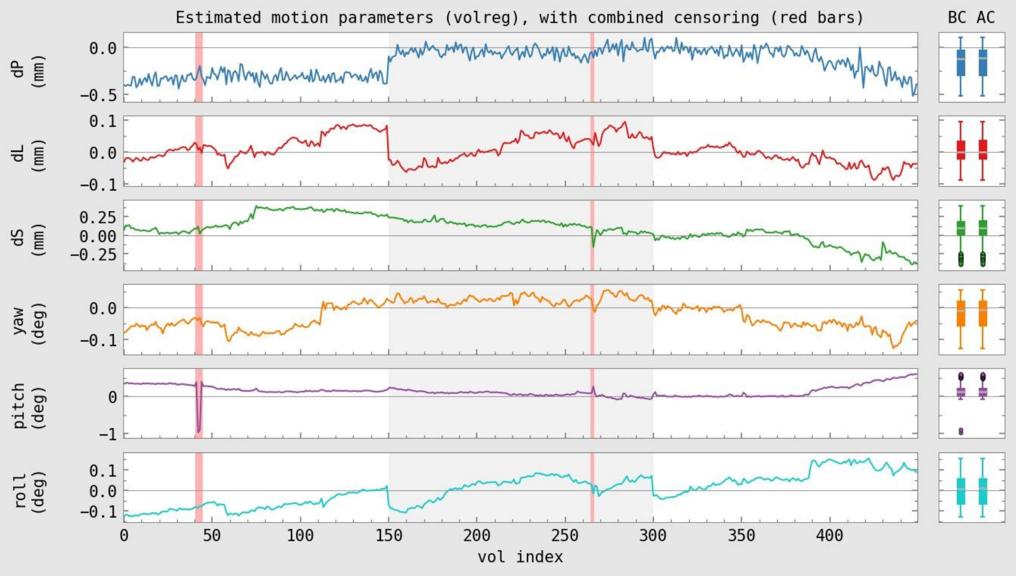
A) vstat, full F-stat: spatial pattern issue (noisy, low peak values, scattered, low in some/many expected areas)

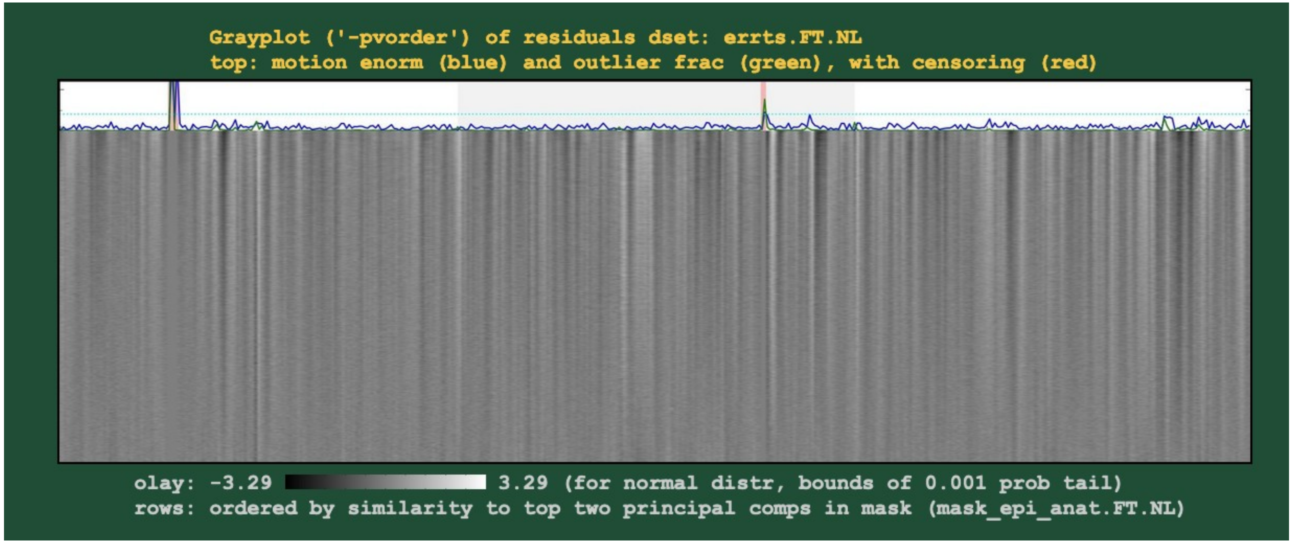
B) mot block: high censoring fraction (36%), due to motion estimates (Enorm) and whole brain outlier fractions

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