

Instant Analyses in **AFNI** and **SUMA**: Clusters and Correlations

Data for this presentation:
AFNI_data5/ directory

All data herein
from Alex Martin,
et al. [NIMH IRP]

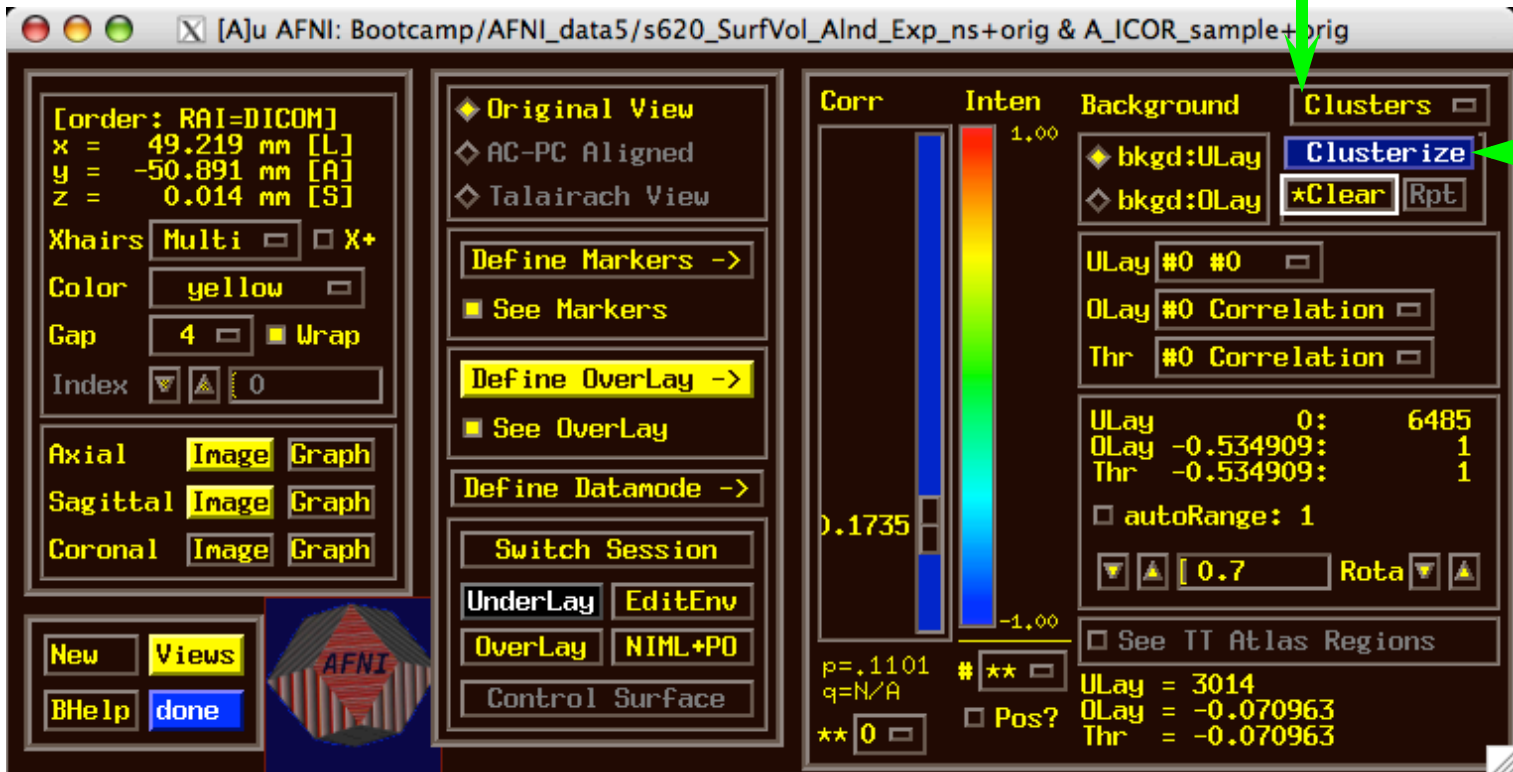


“Insta” Functions

- 3 new capabilities added to the interactive AFNI
- Each one: compute new dataset volumes **instantly** to replace the Overlay volume for image viewing
- **Clusters** = interactive clustering
 - remove clusters below a user-chosen size
 - display a table of clusters
- **InstaCorr** = interactive exploration of inter-voxel time series correlation
 - choose a seed voxel and see correlation map
 - SUMA version also exists
- **InstaCalc** = interactive version of **3dcalc**
 - e.g., display ratio of 2 datasets

AFNI! Clusters: Setup

- Open **Define Overlay**, choose **Clusters** from menu in top right corner



- Then press **Clusterize** to get the clusters control menu

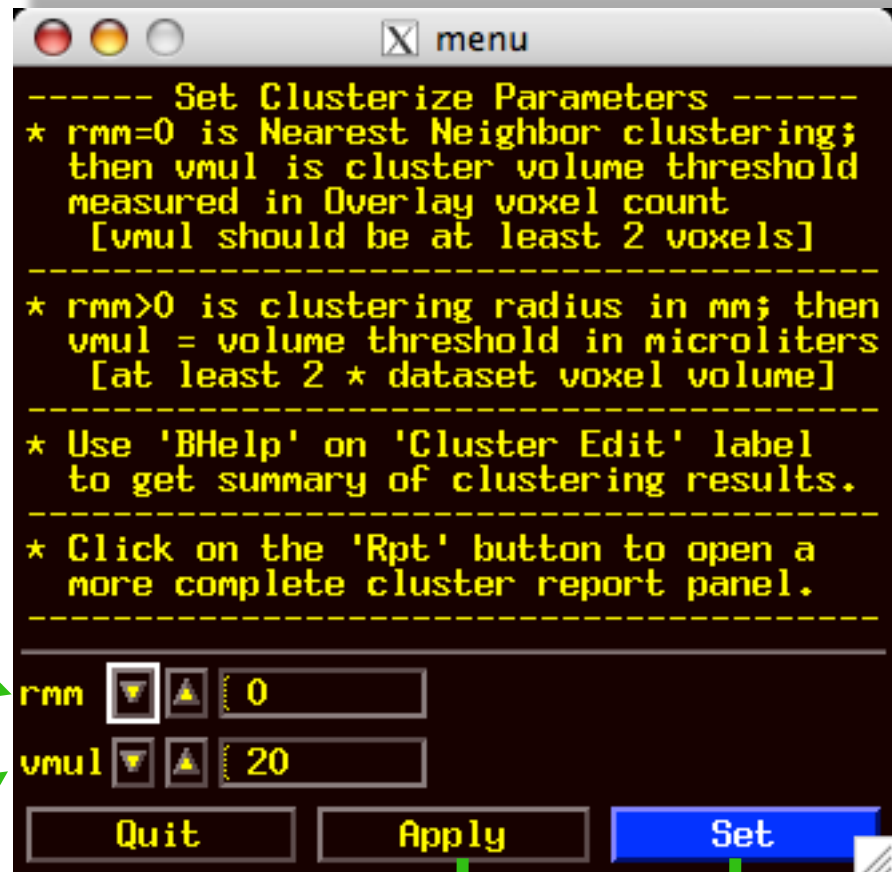
Clusters Control Menu

Operates on user's chosen **Overlay** dataset at the user's threshold;
Next slide example:
AFNI_ICOR_sample

Default: NN clustering

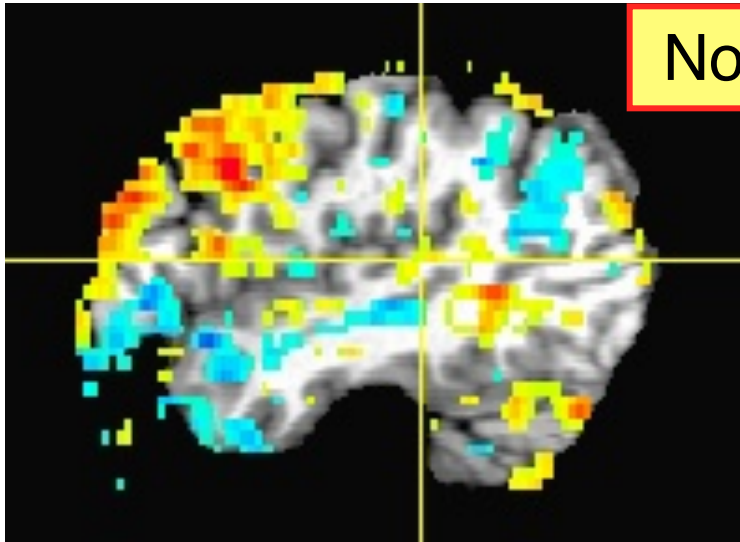
Default: 20 voxel minimum cluster size

Clustering is done in 3D

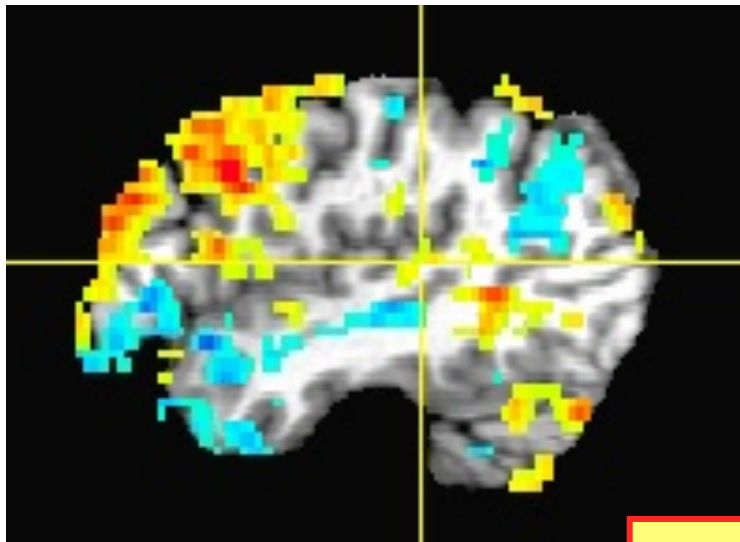


Press one of these buttons to create clusterized volume for display as new **Overlay**

Clusters Results

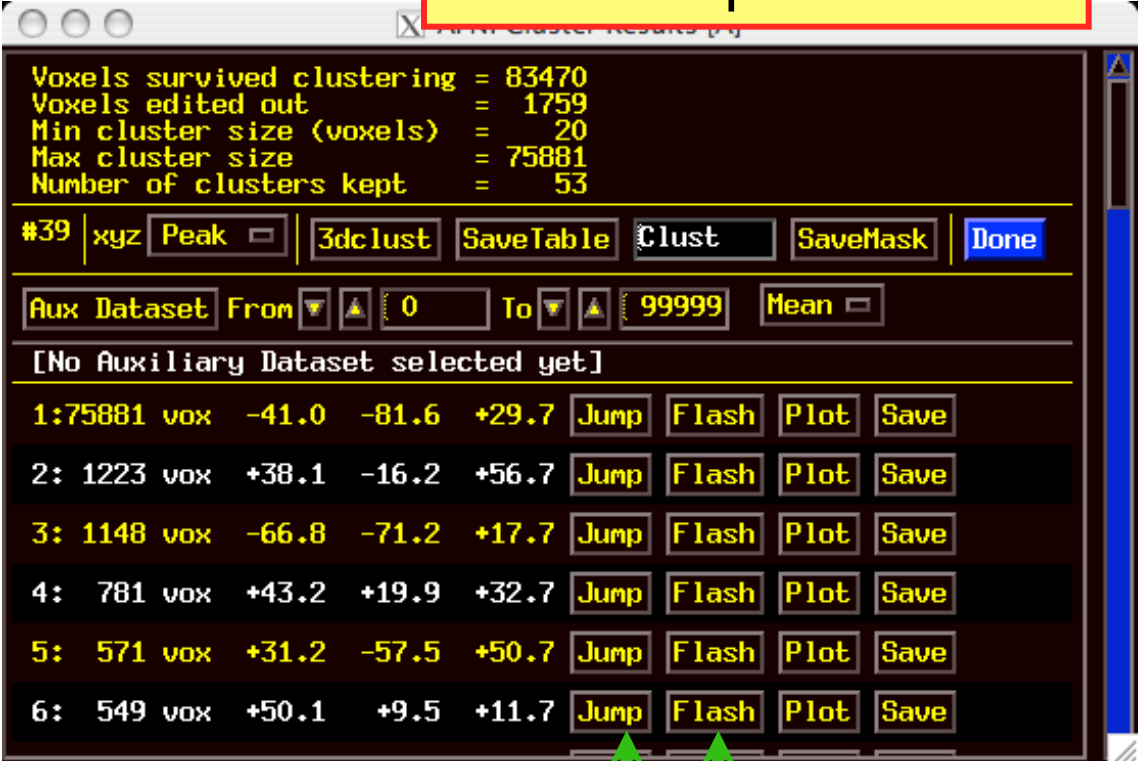


No clustering



With clustering

Cluster report window



Voxels survived clustering = 83470
Voxels edited out = 1759
Min cluster size (voxels) = 20
Max cluster size = 75881
Number of clusters kept = 53

#39 | xyz Peak | 3dc lust SaveTable Clust SaveMask Done

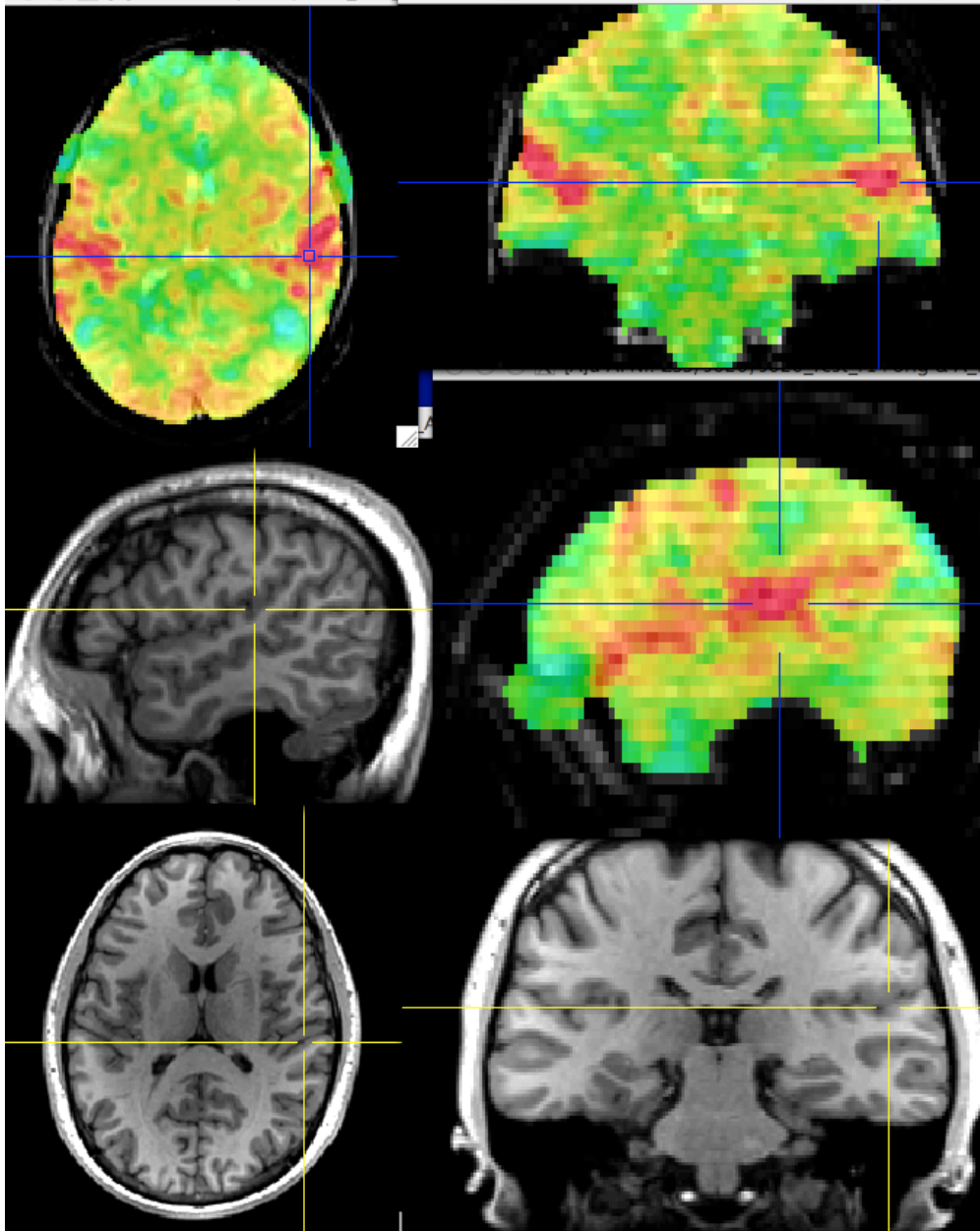
Aux Dataset From To Mean

[No Auxiliary Dataset selected yet]

1:	75881 vox	-41.0	-81.6	+29.7	Jump	Flash	Plot	Save
2:	1223 vox	+38.1	-16.2	+56.7	Jump	Flash	Plot	Save
3:	1148 vox	-66.8	-71.2	+17.7	Jump	Flash	Plot	Save
4:	781 vox	+43.2	+19.9	+32.7	Jump	Flash	Plot	Save
5:	571 vox	+31.2	-57.5	+50.7	Jump	Flash	Plot	Save
6:	549 vox	+50.1	+9.5	+11.7	Jump	Flash	Plot	Save

Jump: crosshairs move
Flash: colors on & off

AFNI! InstaCorr



- On-the-fly instantaneous correlation map of resting state data with interactively selected seed voxel
- **Setup phase:** prepares data for correlations (several-to-10+ seconds)
- **Correlation phase:** you select seed voxel, correlation map appears by *magic*

InstaCorr: Outline of 2 Phases

- **Setup phase:**

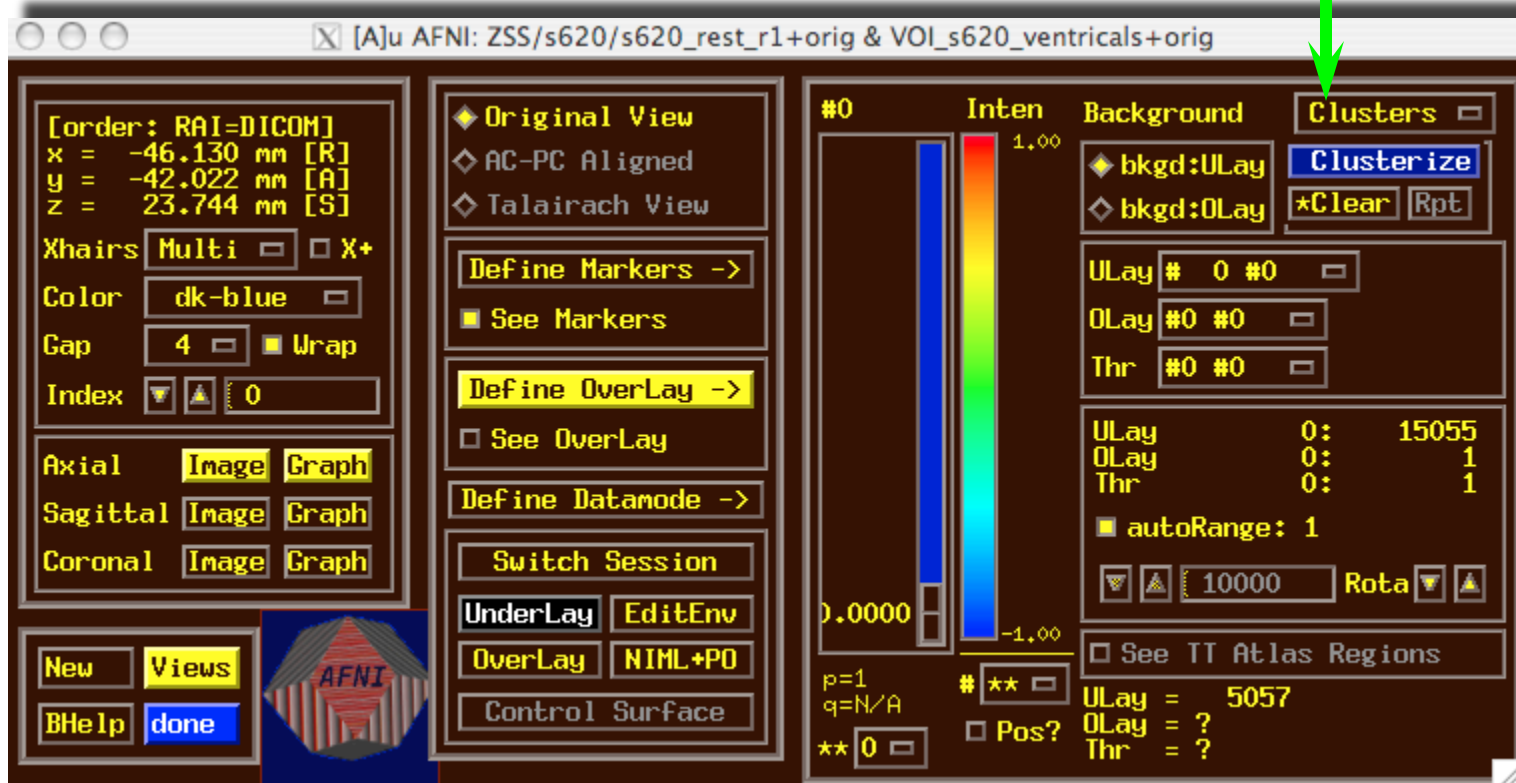
- Masking: user-selected *or* Automask
- Bandpass and other filtering of voxel time series
- Blurring inside mask = the slowest part

- **Correlation phase:**

- Correlate selected seed voxel time series with all other prepared voxel time series
- Make new dataset, if needed, to store results
- Save seed time series for graphing
- Redisplay color overlay
- Optional: compute FDR curve for correlations
 - Calculation is slow, so FDR is not turned on by default

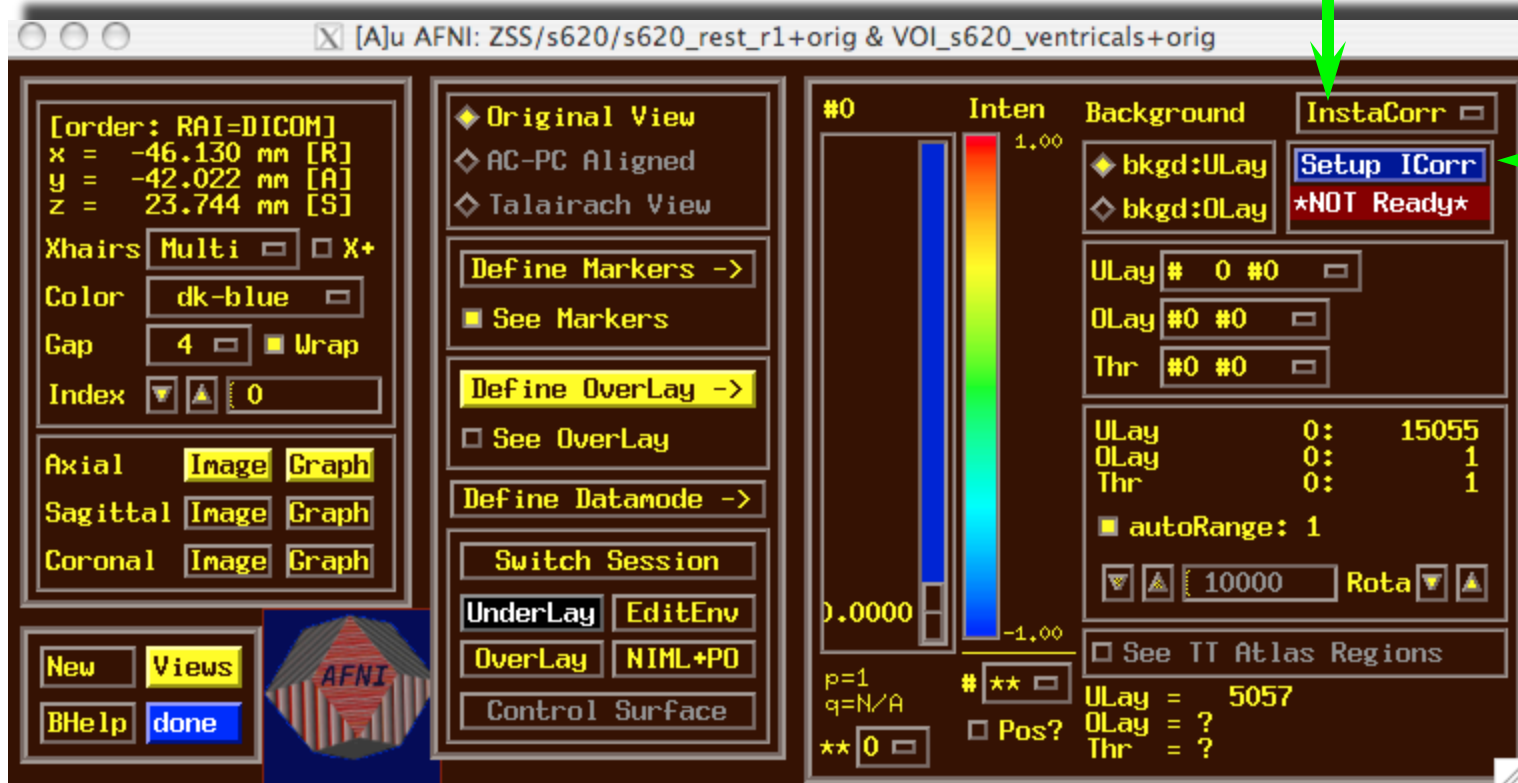
InstaCorr: Setup

- Open **Define Overlay**, choose **InstaCorr** from menu in top right corner



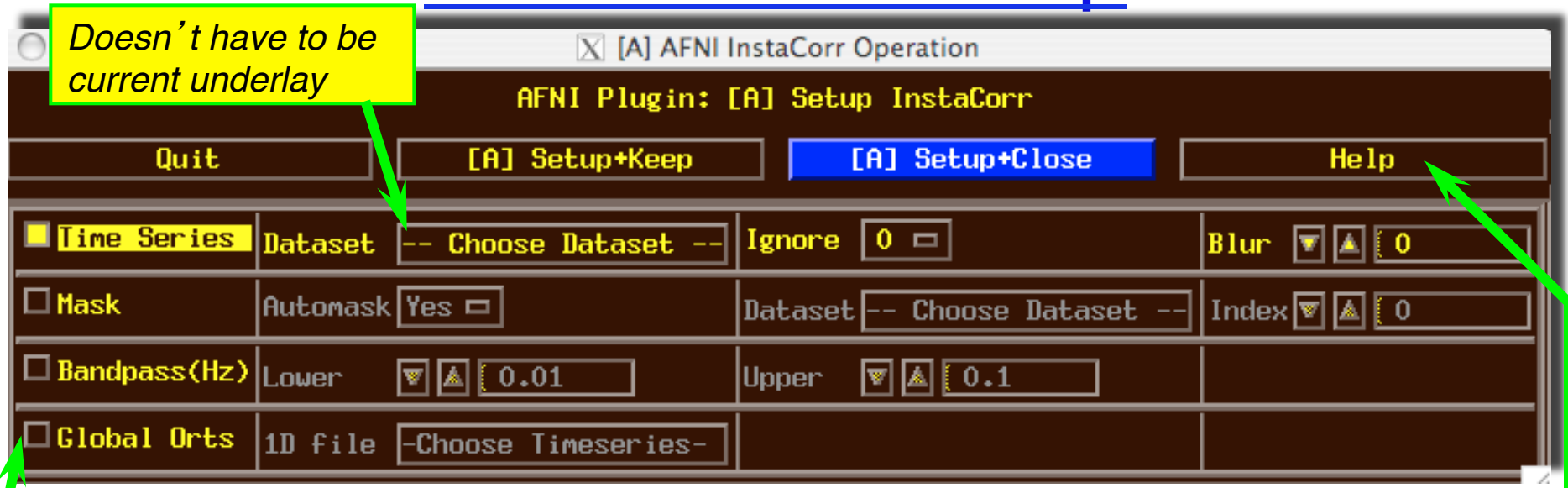
InstaCorr: Setup

- Open **Define Overlay**, choose **InstaCorr** from menu in top right corner



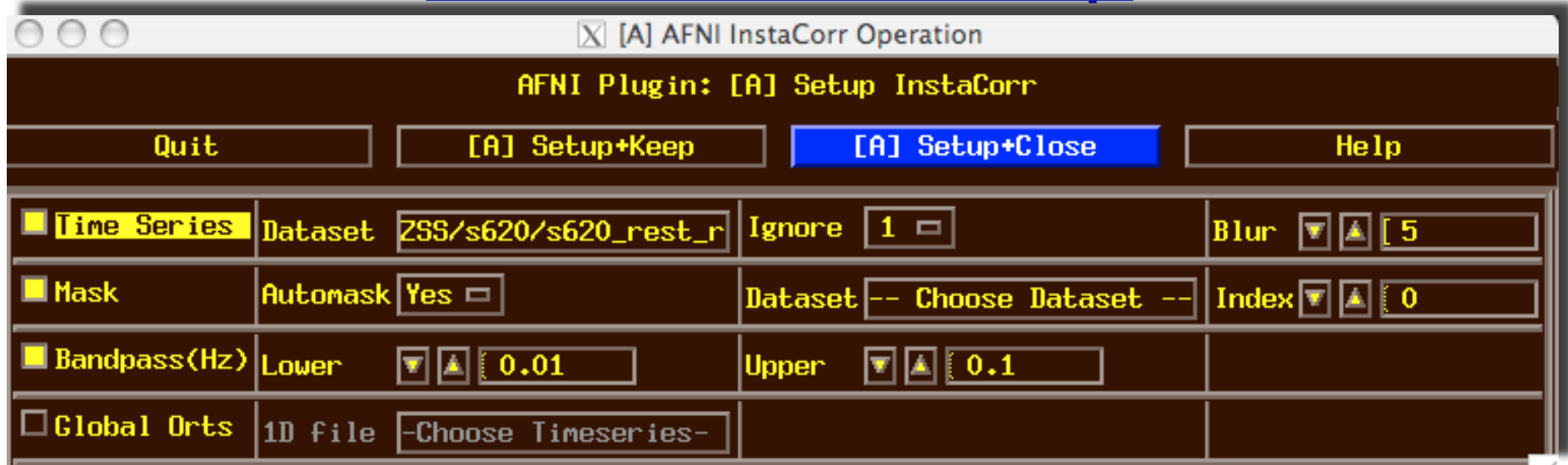
- Then press **Setup ICorr** button to get control panel

InstaCorr: Setup



- Mostly self-explanatory (I hope) — *cf.* **Help**
- **Global Orts** = extra time series to be projected out of dataset before correlation
 - All columns in selected 1D file
 - *e.g.*, movement parameters
 - The first **Ignore** rows (time points) will be skipped
- When ready, press one of the **Setup** buttons

InstaCorr: Setup



- Text output to shell window details the setup procedures:

++ InstaCorr preparations:

```
+ Automask from '/Users/rwcox/data/Resting/ZSS/s620/s620_rest_r1+orig.BRIK' has 197234 voxels
+ Extracting dataset time series
+ Filtering 197234 dataset time series
+ bandpass: ntime=139 nFFT=160 dt=3.5 dFreq=0.00178571
  Nyquist=0.142857 passband indexes=6..56
+ Spatially blurring 139 dataset volumes
+ Normalizing dataset time series
```

Dataset being analyzed

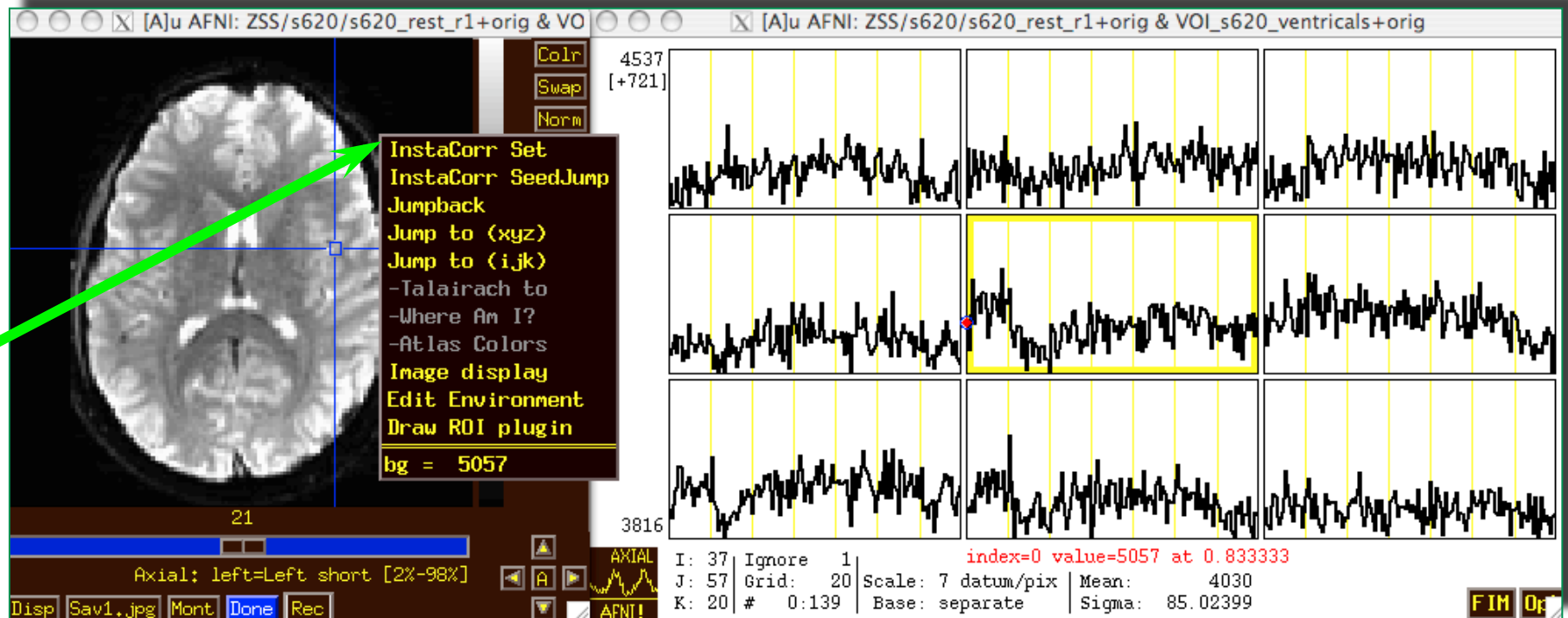
Most of the CPU time:
Uses BlurInMask

++ InstaCorr setup: 197234 voxels ready for work: 15.43 sec

or Preprocess via afni_proc.py

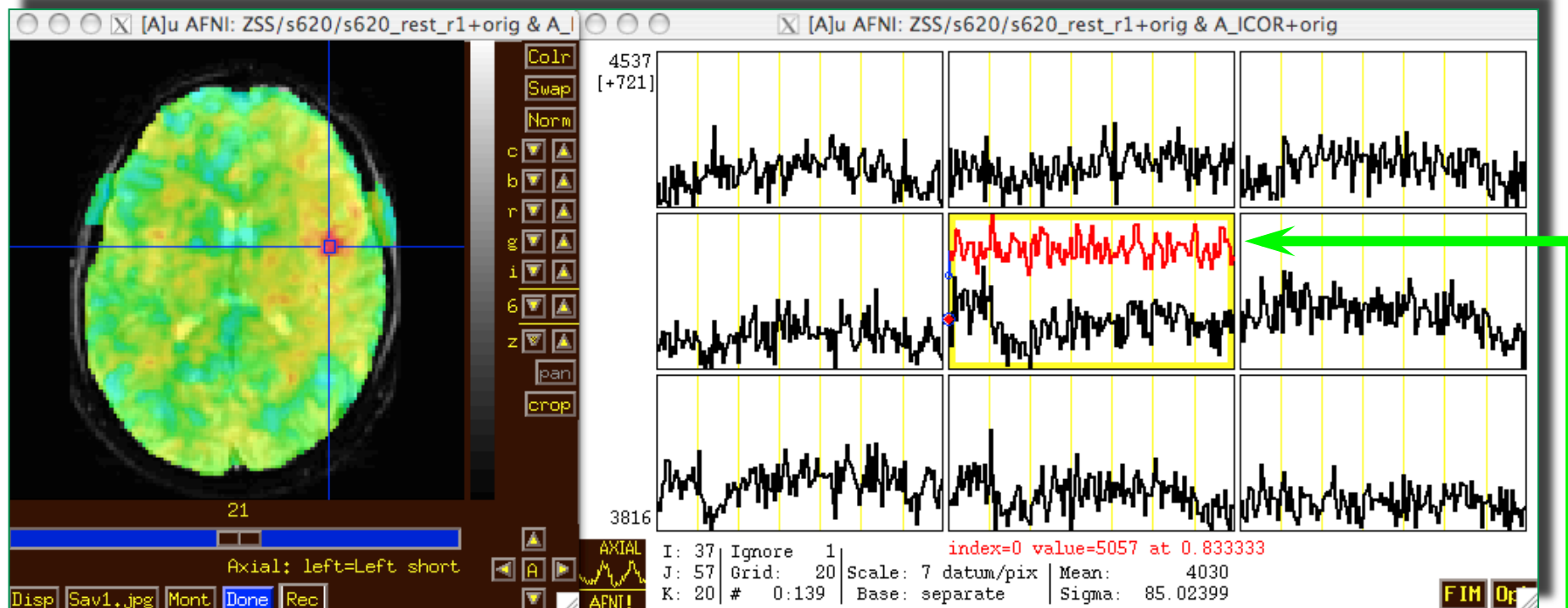
```
## Adapted from Example 9b in afni_proc.py -help
## Eliminated tlrc block; add blur_size of 5mm
afni_proc.py -subj_id s620 \
  -dsets s620_rest_r1+orig.HEAD \
  -copy_anat s620_t1_al2epi+orig.HEAD \
  -blocks despike tshift align volreg blur mask regress \
  -tcat_remove_first_trs 2 \
  -volreg_align_e2a \
  -blur_size 5 \
  -regress_anaticor_fast \
  -regress_censor_motion 0.2 \
  -regress_censor_outliers 0.1 \
  -regress_bandpass 0.01 0.2 \
  -regress_apply_mot_types demean deriv \
  -regress_run_clustsim no -regress_est_blur_errts
```

InstaCorr: The Fun Part



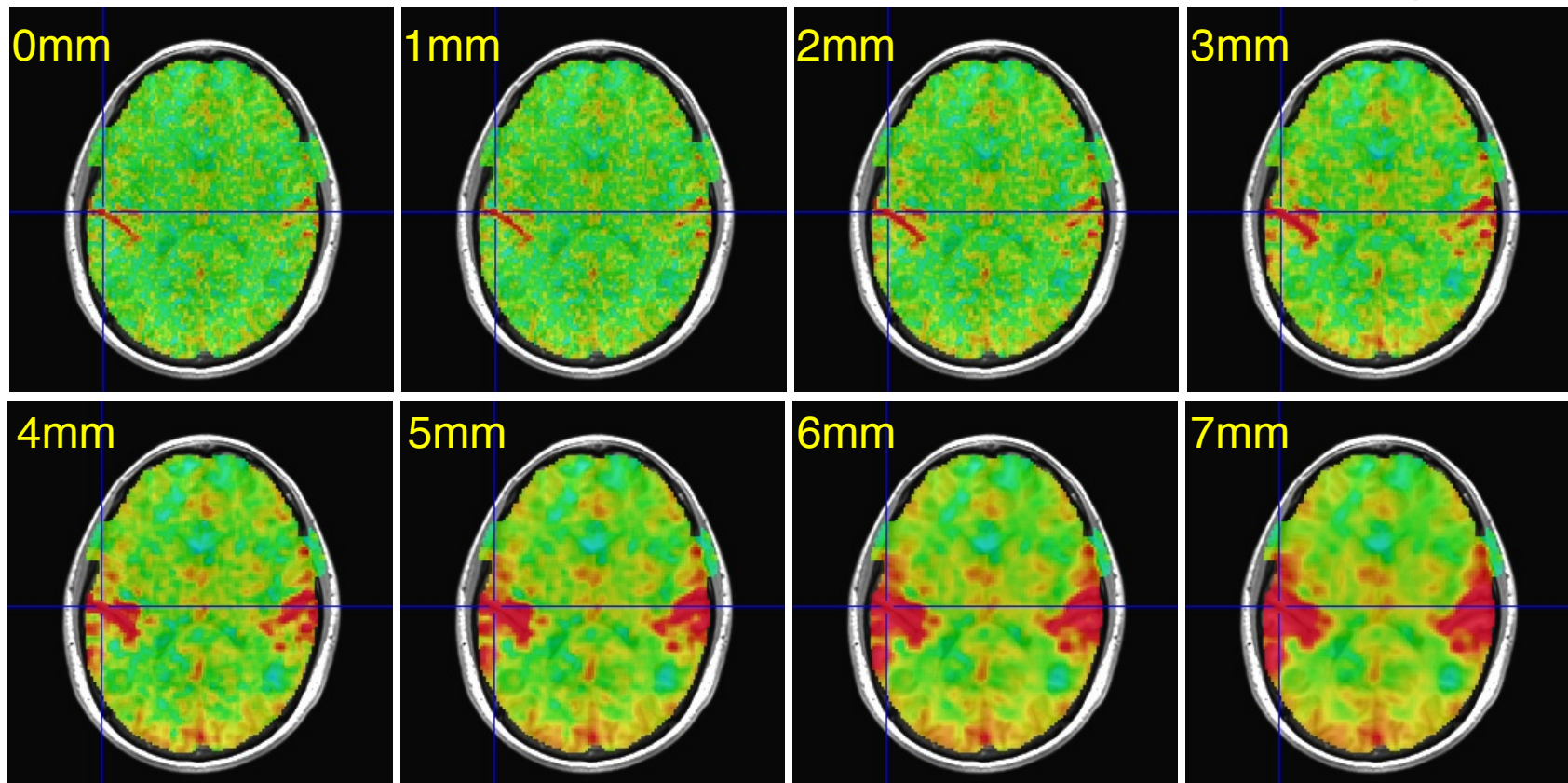
- In image viewer, set crosshairs to desired seed voxel
- **Right-click** popup menu → **InstaCorr Set**
 - Creates new dataset **A_ICOR** for Overlay
 - **Shortcut: Shift+Ctrl+Left-click** sets new crosshair location, then does **InstaCorr Set**
 - Can also hold down **Shift+Ctrl+Left-click** and drag seed around
- **InstaCorr SeedJump** jumps focus to current seed

InstaCorr: The Fun Part



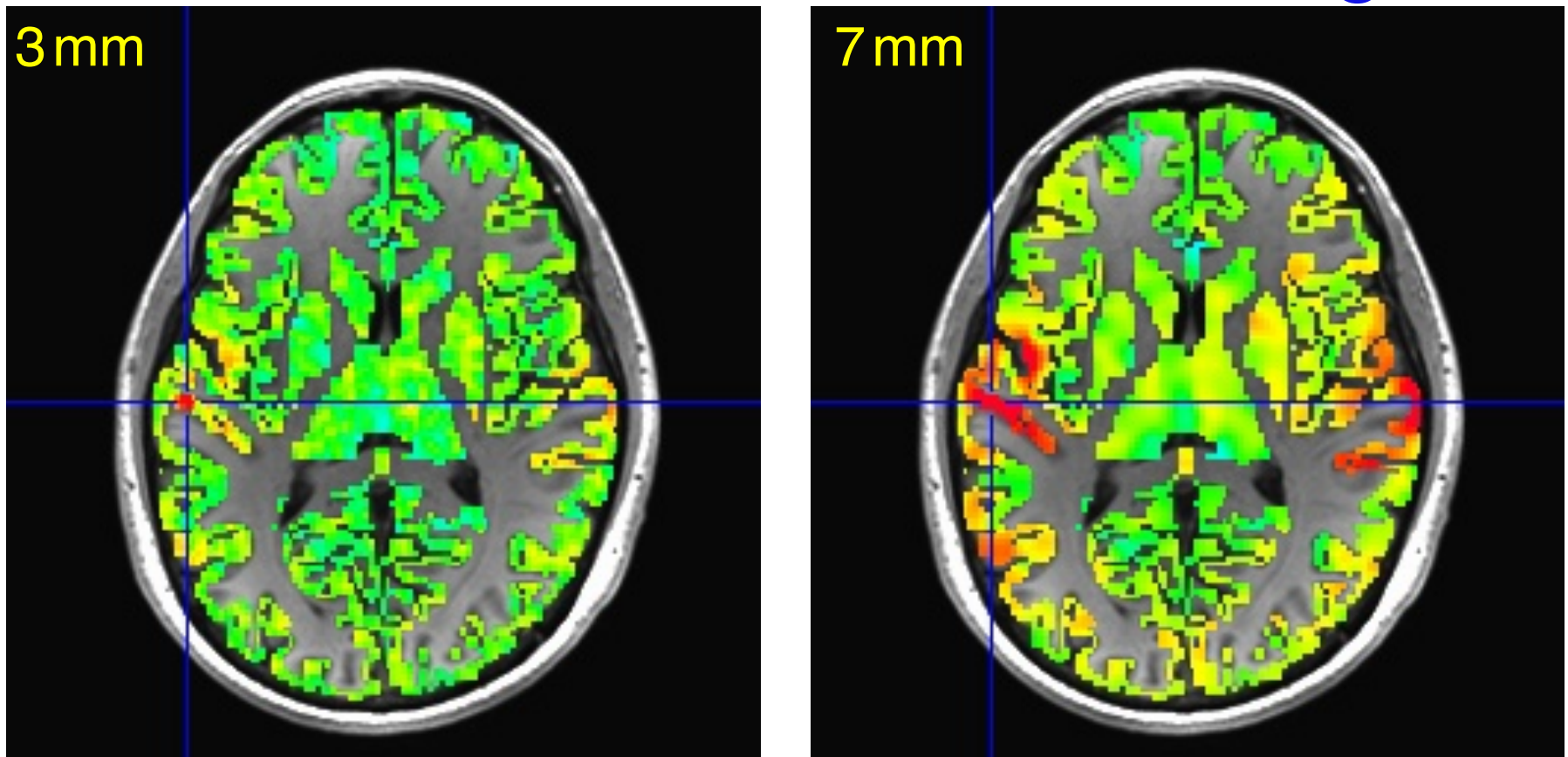
- In graph window:
 - Set Ignore with **FIM**→**Ignore** menu (or **I** key)
 - Set seed overlay with **FIM**→**Pick Ideal** menu
- When you change seed voxel, saved overlay time series will change (but you have to refresh graph to see it)

InstaCorr: Effects of Blurring



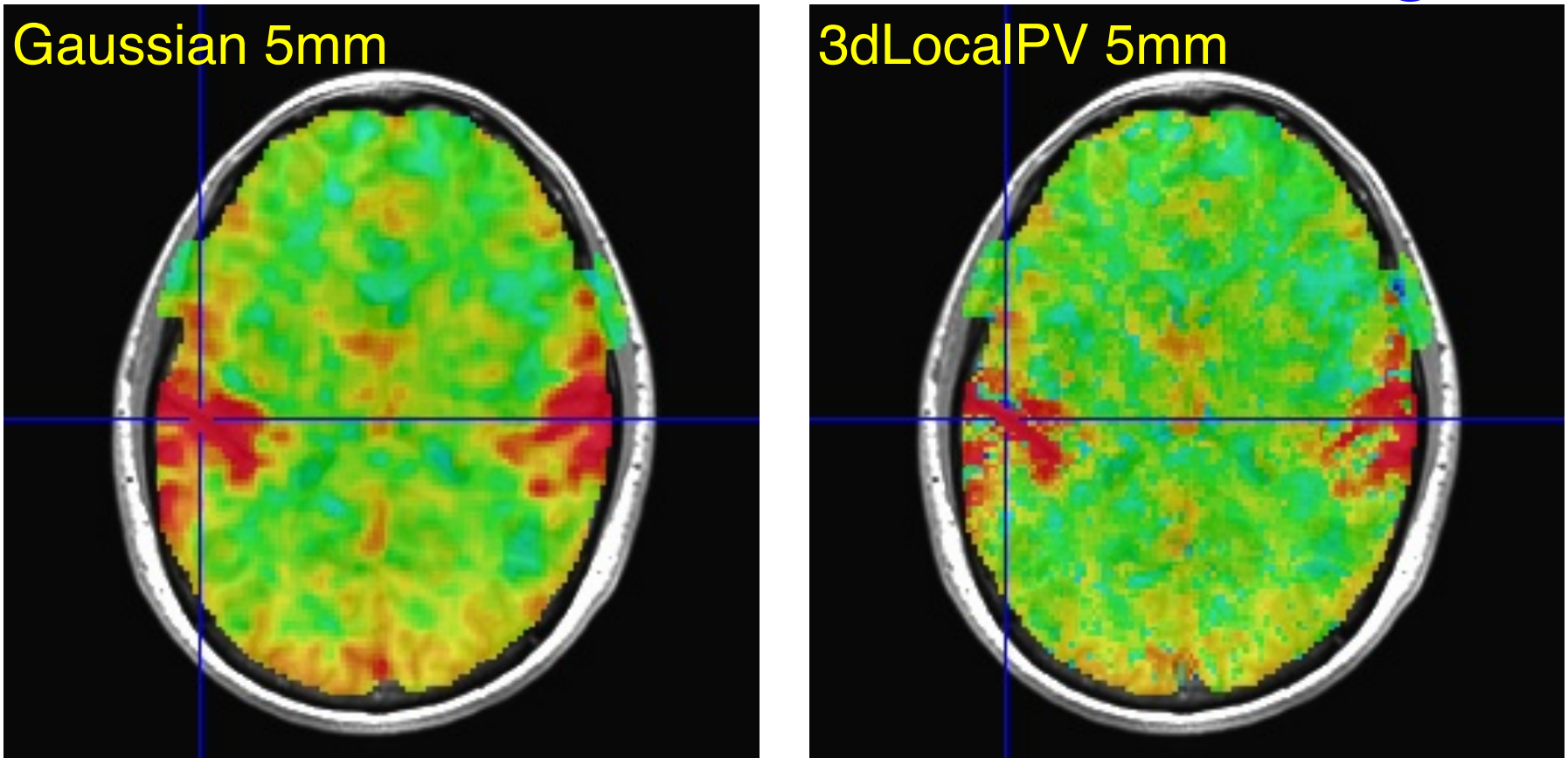
- Is this a pure vascular/cardiac effect being progressively smeared? Or real neural correlations seen via BOLD? Or some of both? *Venograms?*
 - Dataset was RETROICOR-ized; mask is whole brain

InstaCorr: Effects of Blurring



- Similar calculations, but with FreeSurfer-generated gray matter mask instead of Automask from EPI data
 - Blurring is done only inside the mask (**3dBlurInMask**)
 - Using a discrete PDE-based iterative approach

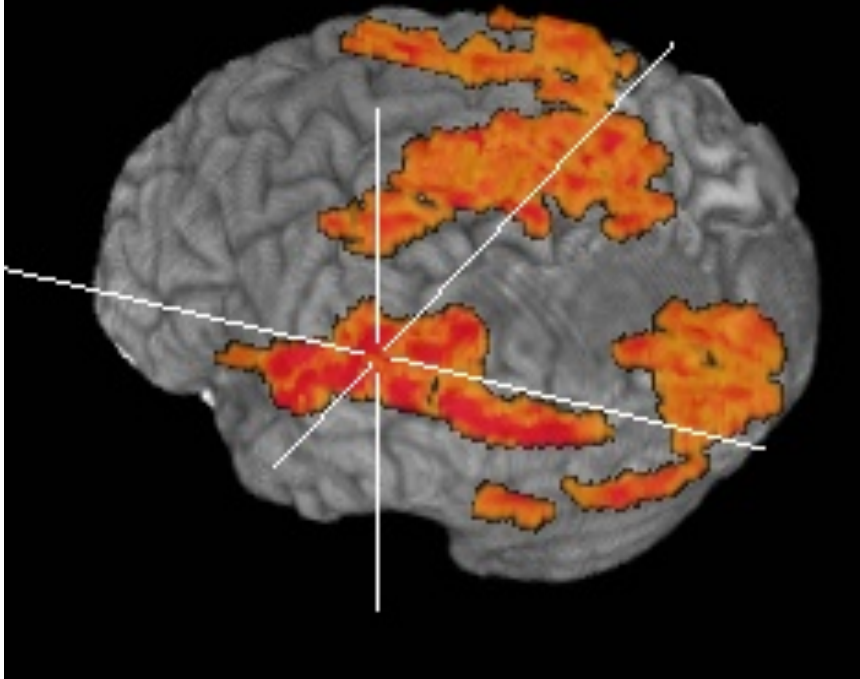
InstaCorr: SVD-based “Blurring”



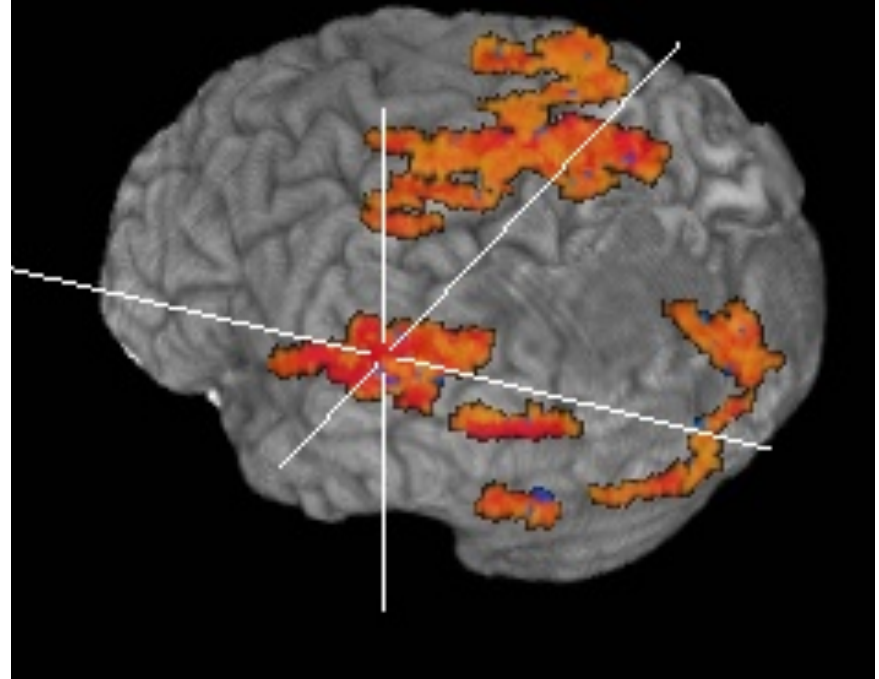
- Similar calculations, with Automask from EPI data, using **3dLocalPV** over 5 mm radius sphere (67 voxels)
 - Project each vector onto 2-dim principal subspace
 - Far too slow to calculate interactively (at this time)

InstaCorr: SVD-based “Blurring”

Gaussian 5mm



3dLocalPV 5mm



- Volume rendering of InstaCorr maps (threshold at $r=0.5$)
 - Renderer updates automatically if **DynaDraw** is on
- SVD smoothing has cleaner spatial structure?
 - Or has it lost some information? *I don't know.*

InstaCorr: Options and Plans

- Underlay doesn't have to be EPI data; could be anat
 - Can use InstaCorr in multiple AFNI controllers
 - FDR: **setenv AFNI_INSTACORR_FDR YES**
 - Will slow things down by a significant factor
 - Saving **A_ICOR** dataset: overwrites previous copies
-
- Future Possibilities:
 - Select ROI-based Orts to be detrended?
 - Based on ROIs from FreeSurfer or atlases?
 - Or multiple seeds (partial + multiple correlations)?
 - Interactive local SVD “smoothing”? (needs speedup)
 - ▪ Group analysis InstaCorr (in standardized space)
 - Not quite “Insta” any more; $\approx 0.1 \times \# \text{Subjects}$ sec per seed
 - External script to do subject setups
 - Use time series subsets? (e.g., for block design data)

Group InstaCorr

- If you have a robust enough system (multiple CPUs, several gigabytes of RAM), you can explore the *group* analysis of resting state seed-based correlations
- **Setup Phase:**
 - Unlike individual InstaCorr, the setup is done outside the AFNI GUI with command line programs
 - Step 1: pre-process all datasets (including into template space) by using `afni_proc.py`
 - We recommend something like Example 9b in the `afni_proc.py` help output
 - Step 2: collect groups of time series datasets into one big file = `3dSetupGroupInCorr`
- **Interactive Phase:** point-and-click to set seed voxel

3dGroupInCorr: Setup #2

- **3dSetupGroupInCorr** reads all filtered & blurred resting state EPI datasets, masks & normalizes them, and writes them to one *big* file for **3dGroupInCorr**
 - Sample tcsh script fragment below: 2 groups of subjects

```
set AAA = ( s601 s604 ... s644 s646 )
set BBB = ( s611 s612 ... s652 s654 )
set ggg = ( )
foreach fred ( $AAA )
    set ggg = ( $ggg errts.${fred}.fanaticor+tlrc.HEAD )
end
```

```
3dSetupGroupInCorr -mask ALL_am50+tlrc -prefix AAA $ggg
```

```
set ggg = ( )
foreach fred ( $BBB )
    set ggg = ( $ggg errts.${fred}.fanaticor+tlrc.HEAD )
end
```

```
3dSetupGroupInCorr -mask ALL_am50+tlrc -prefix BBB $ggg
```

3dGroupInCorr: Interactive Phase

- Start server program (2-sample *t*-test here):

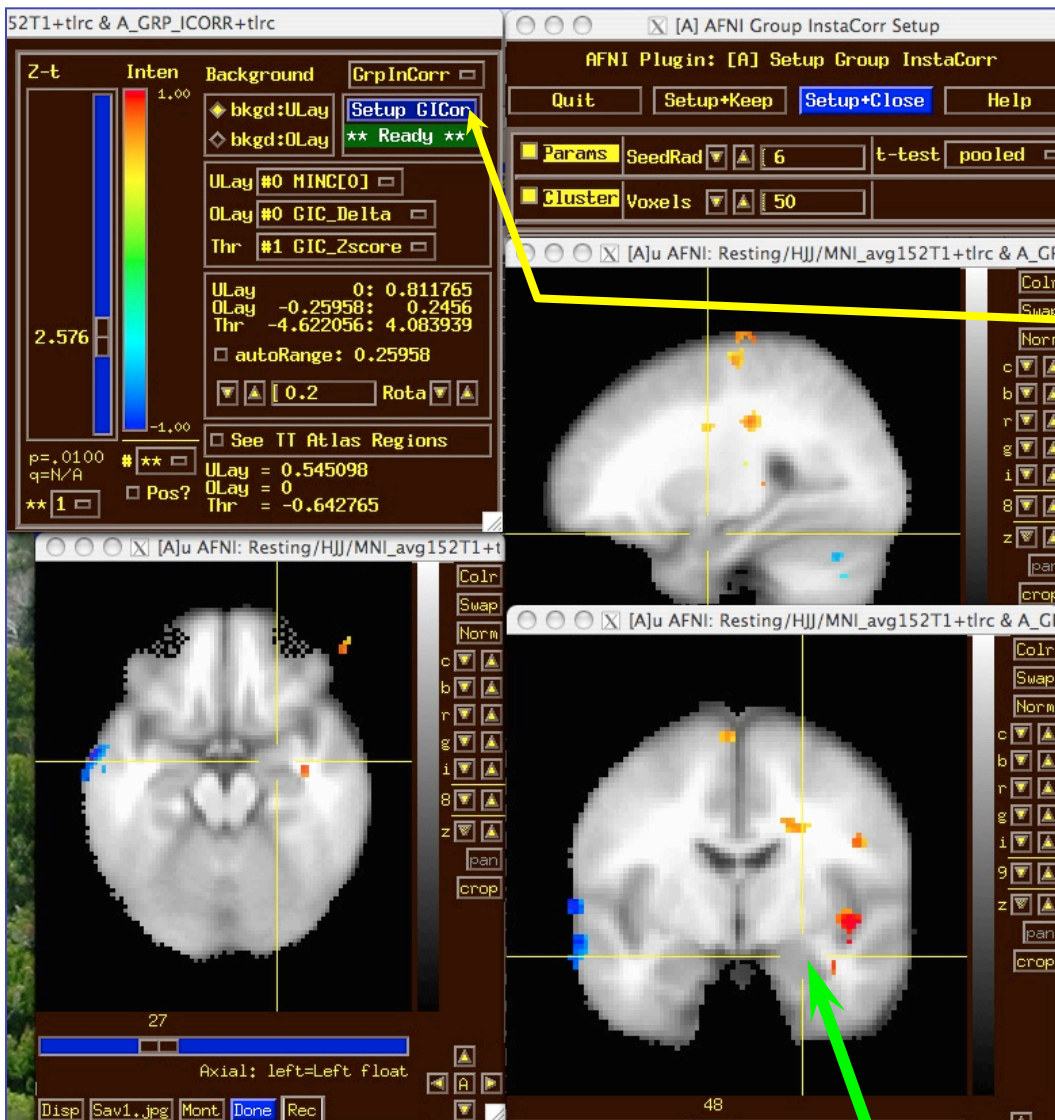
```
3dGroupInCorr -setA AAA.grpincorr.niml \  
              -setB BBB.grpincorr.niml
```

 - Startup takes a little while, as all data must be read into RAM (in this example, 3.2 Gbytes)
 - After data is read, connects to AFNI using a NIML socket
 - Server will use multiple CPUs if compiled with OpenMP (currently on Mac OS X 10.5 and 10.6)
- In a separate terminal window, start AFNI:

```
afni -niml ~/abin/MNI_avg152T1+tlrc.HEAD
```

 - Then open the **Define Overlay** control panel
 - Select **GrpInCorr** from the **Clusters** menu

3dGrpInCorr: Interactive Results



Seed voxel

- Use same buttons as individual subject InstaCorr to set seed
- Use **Setup GICor** panel to set the few options available interactively
 - **SeedRad** = extra smoothing radius for seed voxel time series (flat average)
 - **Cluster** = min number of voxels to keep above thresh

3dGrpInCorr: What It Computes

- Step 1: Extract seed time series from each dataset; correlate with all voxel time series in that dataset
- Step 2: Group analysis: *t*-test among correlation bricks
- 1-sample *t*-test (**-setA** only) gives 2 sub-bricks:
 - mean of \tanh^{-1} (correlation with seed)
 - *Z*-score of *t*-statistic of this mean
- 2-sample test (**-setA** and **-setB**) gives 6 sub-bricks:
 - difference of means (**A-B**) of \tanh^{-1} (correlation)
 - *Z*-score of *t*-statistic of this difference
 - Pooled or unpooled variance, or paired *t*-test (your option)
 - Plus 1-sample results for **-setA** and **-setB** separately
 - View these in AFNI **[B]** and **[C]** controllers, to see it all!

3dGrpInCorr: To Do It By Hand?

- After preprocessing all datasets, you would have to do the following steps on each resting state dataset:
 - Extract seed time series from each dataset [**3dmaskave**]
 - Correlate seed time series with all voxels from its dataset [**3dDeconvolve** or **3dfim**]
 - Convert to \tanh^{-1} (correlation) [**3dcalc**]
- Then do the following on the results from the above
 - Compute the t -test [**3dttest**]
 - Convert to Z -score [**3dcalc**]
 - Read into AFNI for display
- Even with a script, this would be annoying to do a lot
 - Just ask Daniel Handwerker!

Group InstaCorr: Final Notes

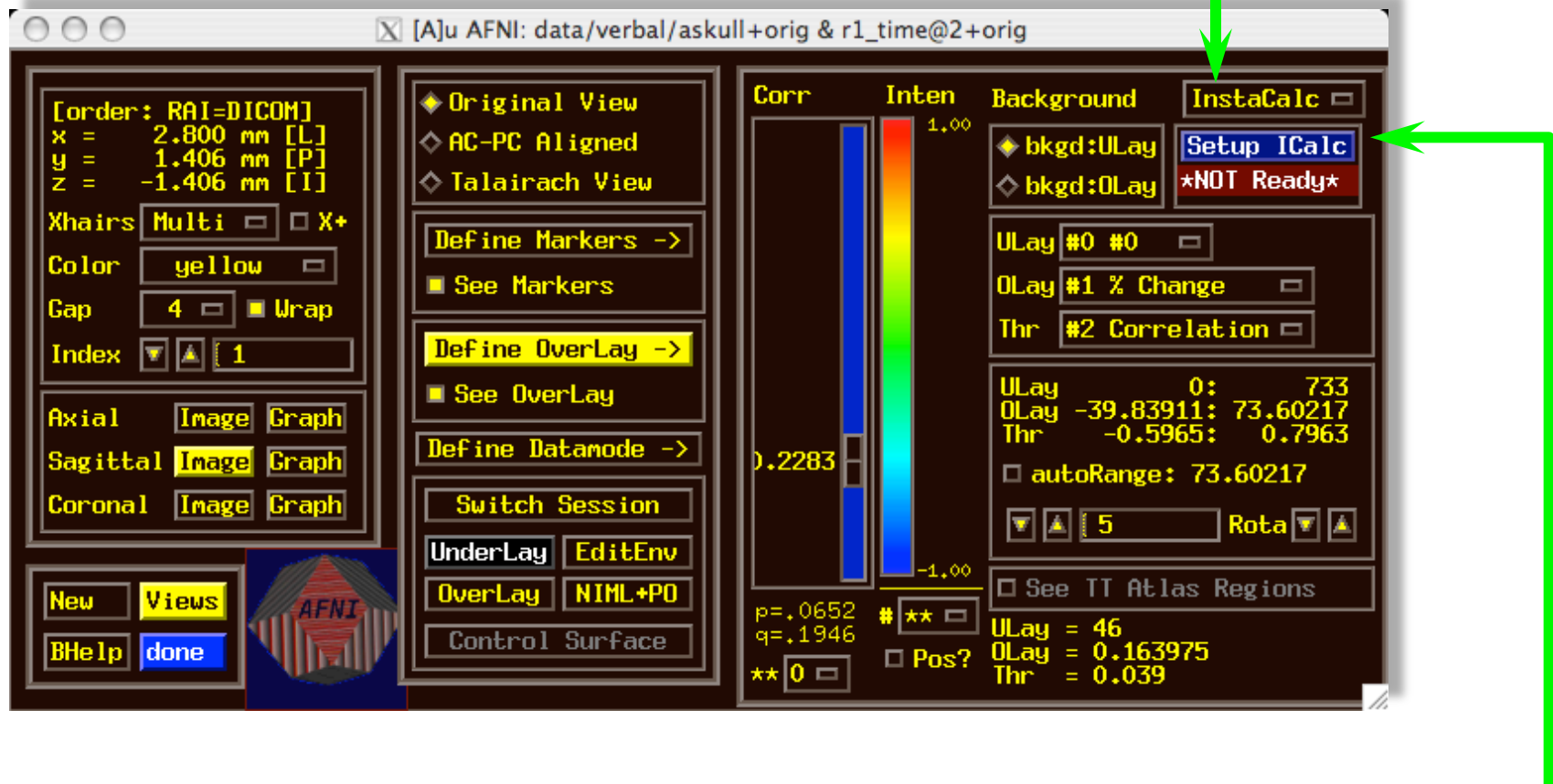
- Time series datasets can have different lengths
 - But all must have the same spatial grid and use the same mask!
- **Fun Stuff:** volume render results with **DynaDraw**
- Sometimes AFNI drops the shared memory connection to **3dGroupInCorr**
 - Due to unknown bugs somewhere in AFNI
 - Program tries to reconnect when this happens
 - If this gets bad, use the **-NOshm** option to **3dGroupInCorr** to force it to use TCP/IP only
 - Slower data transfer, but more reliable

Group InstaCorr: Finalest Notes

- Shift+Ctrl+Click+Drag method for dynamically setting the seed voxel also works with **Group InstaCorr**
 - But speed of interaction can be **slow**
- Can also include subject-level covariates (e.g., IQ, age) in the analysis *at the group step*
 - To regress them out (nuisance variables), and/or to test the slope of $\tanh^{-1}(\text{correlation})$ vs. covariate
- Can also run in batch mode (no talking to AFNI)
- Further ideas (i.e., to dream the impossible dream):
 - Granger-ize: correlate with lag-0 **and** lag-1 of seed and test Granger causality
 - Allow user to set other seeds to be "partialled out" of the analysis

InstaCalc: Dataset Calculator

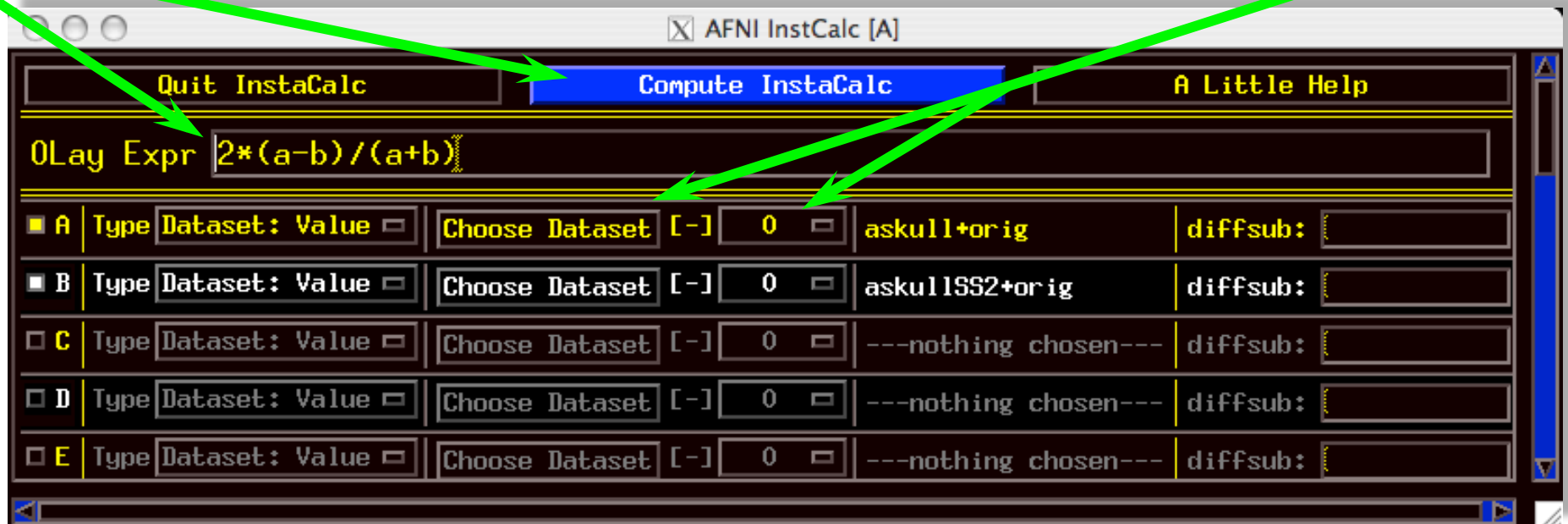
- Open **Define Overlay**, choose **InstaCalc** from menu in top right corner



- Then press **Setup ICalc** button to get control panel

InstaCalc: Setup

- Select datasets with **Choose Dataset** buttons
 - and sub-bricks with the **[-]** controls
- Enter symbolic expression
- Press **Compute InstaCalc**
- Creates new 1-brick dataset **A_ICALC** for Overlay
 - voxel-by-voxel calculations

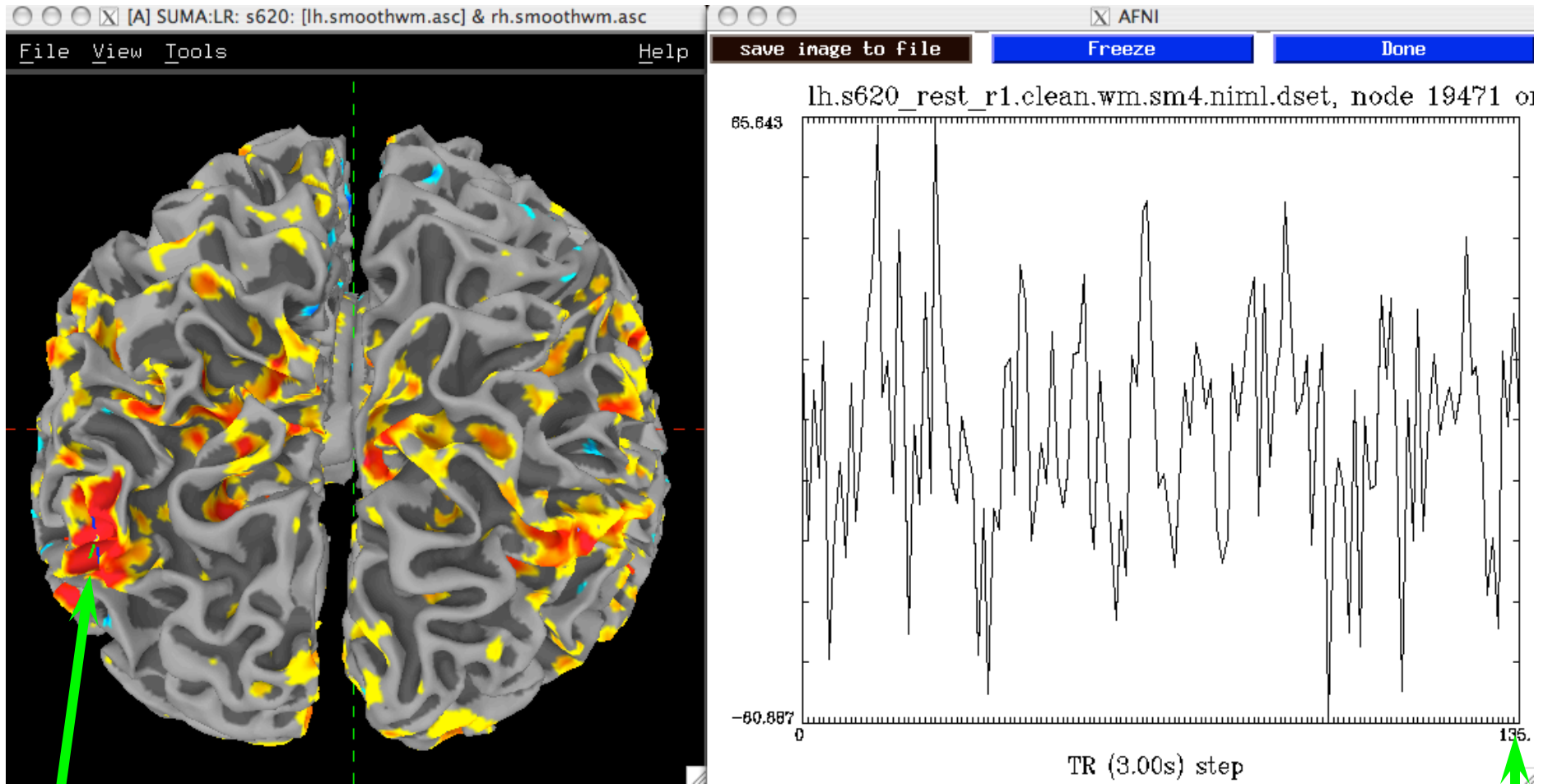


 **InstaCorr**
The logo for SUMA! features the word "SUMA!" in a colorful, blocky font where each letter is a different color (S: yellow, U: purple, M: green, A: red, !: blue). To the right of this is the word "InstaCorr" in a blue, sans-serif font, underlined.

- Similar in concept to AFNI **InstaCorr** but requires external pre-processing of time series datasets
 - Removal of baseline, projection to surface, blurring
- In the **AFNI_data5/** directory, run the script
tcsch ./@run_REST_demo
 - starts SUMA with 2 hemispheres
 - loads pre-processed datasets into SUMA
 - sets up SUMA's **InstaCorr**
- After all the setup is ready, right-clicking on the surface will do the **InstaCorr** calculations
- **3dGroupInCorr** also works with SUMA



InstaCorr: Sample



• Seed voxel and Seed voxel time series graph