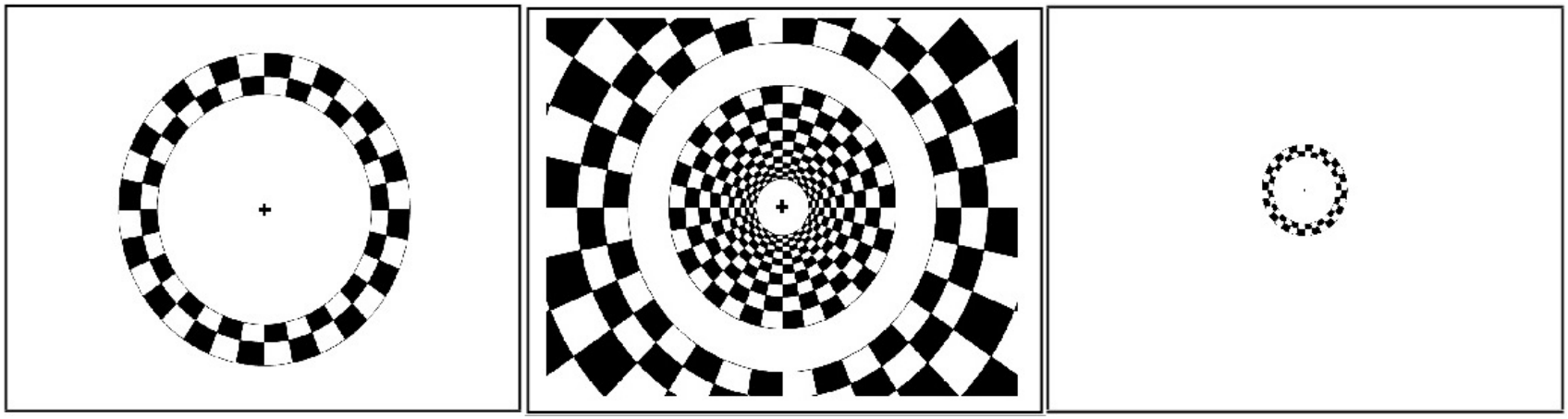
Sample Data Analysis with 3dDeconvolve

- Brief description of experiment (dataset `epi07+orig`):
  - ◇ Three classes of active stimuli
    - ↪ R = Right hand only sequential finger-thumb opposition
    - ↪ A = 8 Hz flickering annular checkerboard
    - ↪ a = 8 Hz flickering anti-annular checkerboard
    - ↪ r = no motor action
    - ↪ + = visual fixation on non-flickering image
    - ↪ Stimulus time series (TR=2 s, 3.125×3.125×5 mm^3, 21 slices, 100 volumes, 3 Tesla):

```
Visual ++++++++++AAAAA+++++aaaaa+++++AAAAA+++++aaaaa+++++AAAAA+++++aaaaa+++++AAAAA+++++aaaaa+++++++++++++++
Motor  rrrrrrrrrrRRRRRRRRRRRRRRRRRRrrrrrrrrrrrrrrrrrrrrRRRRRRRRRRRRRRRRRRrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
```



Annulus, Anti-Annulus, and Fixation Stimuli

- First analysis: Simple regression with fixed hemodynamic model
  - ◇ Will use 3 regressors to correspond to the three stimulus classes
  - ◇ Step 1: Create 0-1 time series files to correspond to absence-presence of annular, anti-annular, and motor stimuli:

    Files: `ann_stim_07.1D` `antiann_stim_07.1D` `righthand_stim_07.1D`

  ↪ Created with a text editor (in this case, `vi`)

  - ◇ Step 2: Create hemodynamic response to stimuli time series, via program waver:

```
waver -GAM -dt 2.0 -input ann_stim_07.1D > ann_response_07.1D
waver -GAM -dt 2.0 -input antiann_stim_07.1D > antiann_response_07.1D
waver -GAM -dt 2.0 -input righthand_stim_07.1D > righthand_response_07.1D
```

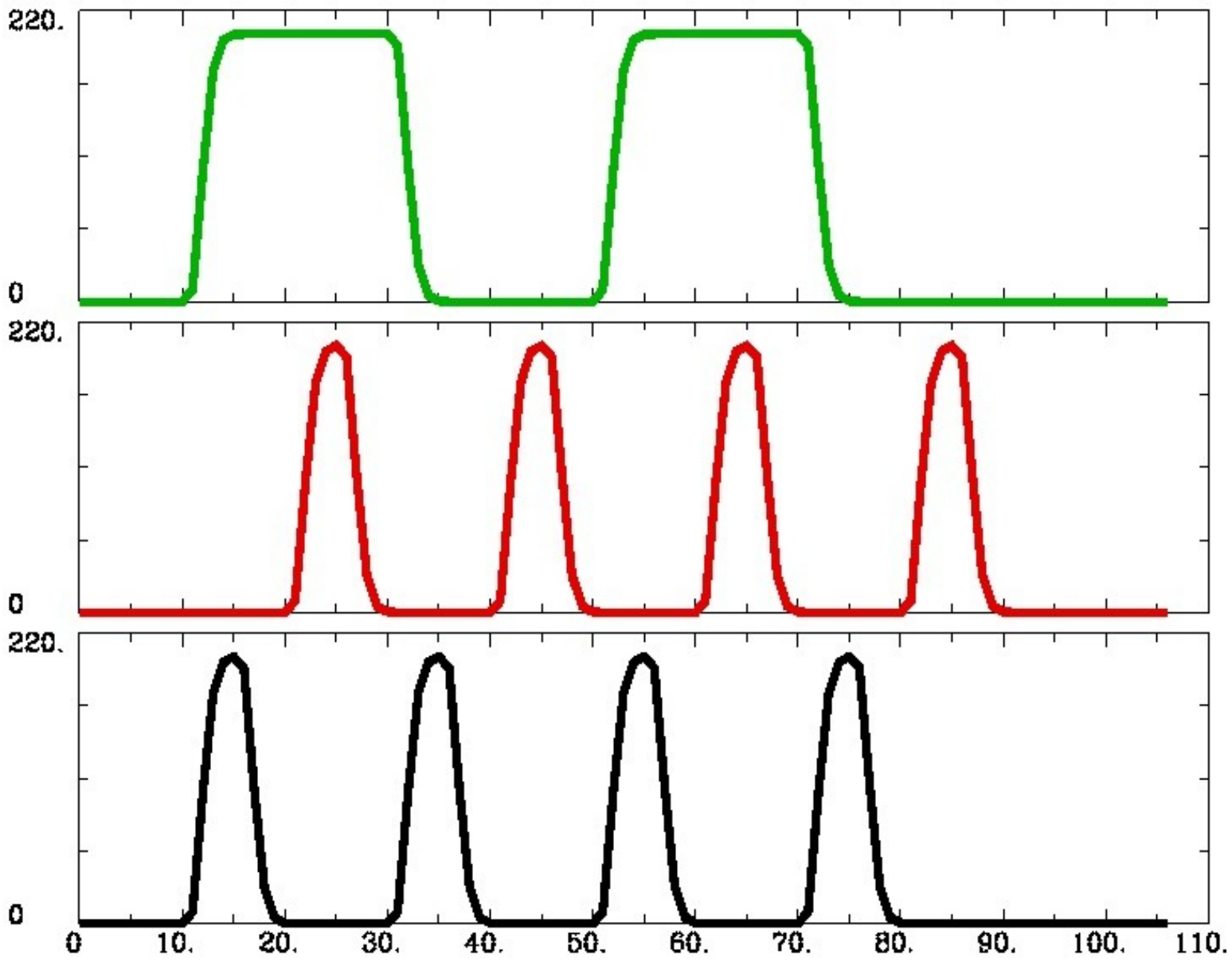
    (cf. Script File `waver_07a`)

  ↪ Put all 3 responses together into 1 file with 3 columns:

```
1dcat ann_response_07.1D antiann_response_07.1D \
      righthand_response_07.1D > all3_response_07.1D
```

↪ Plot them (to make sure we haven't made a stupid mistake yet):

```
1dplot all3_response_07.1D
```



◇ Carry out regression analysis (cf. Script File `decon_07a`):

```
3dDeconvolve -input epi07+orig.HEAD                    \
             -num_stimts 3                             \
             -stim_file 1 ann_response_07.1D           \
             -stim_file 2 antiann_response_07.1D       \
             -stim_file 3 righthand_response_07.1D     \
             -stim_label 1 annulus                     \
             -stim_label 2 antiann                     \
             -stim_label 3 motor                       \
             -stim_minlag 1 0  -stim_maxlag 1 0        \
             -stim_minlag 2 0  -stim_maxlag 2 0        \
             -stim_minlag 3 0  -stim_maxlag 3 0        \
             -fitts epi07a_fitts                       \
             -fout -bucket epi07a_stats
```

↪ Each stimulus has minimum and maximum lags set to 0
⇒ Signal model is just the sum of the response time series we input

↪ We keep the stimulus labels short since the AFNI menus don't like long labels

↪ `-fitts` option ⇒ We get the fitted model as an output dataset

↪ `-fout -bucket` options ⇒ We get the F-statistics in a bucket dataset

↪ We skip the individual regressor t-statistics (no `-tout`): these are redundant with the partial-model F-statistics when we have only 1 lag

▷ Each t tests one coefficient; each partial-F tests all coefficients in that sub-model; 1 degree of freedom in a sub-model ⇒ $t^2 = F$

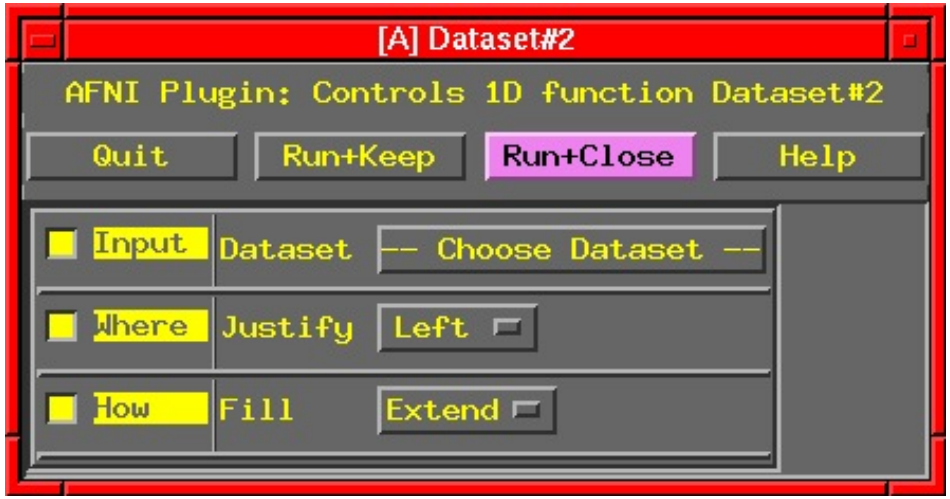
↪ We don't need `-ignore` option, since this dataset doesn't have the T1-equilibration spike at the beginnning (must have been pre-saturated)

◇ Look at results with interactive AFNI program:

↪ Since data slices were axial, open Axial image and graph for dataset `epi07`

↪ Let's look at output of `-fitts` graph overlaid on data time series:

▷ In AFNI controller: `Define Datamode→Plugins→Dataset#2`

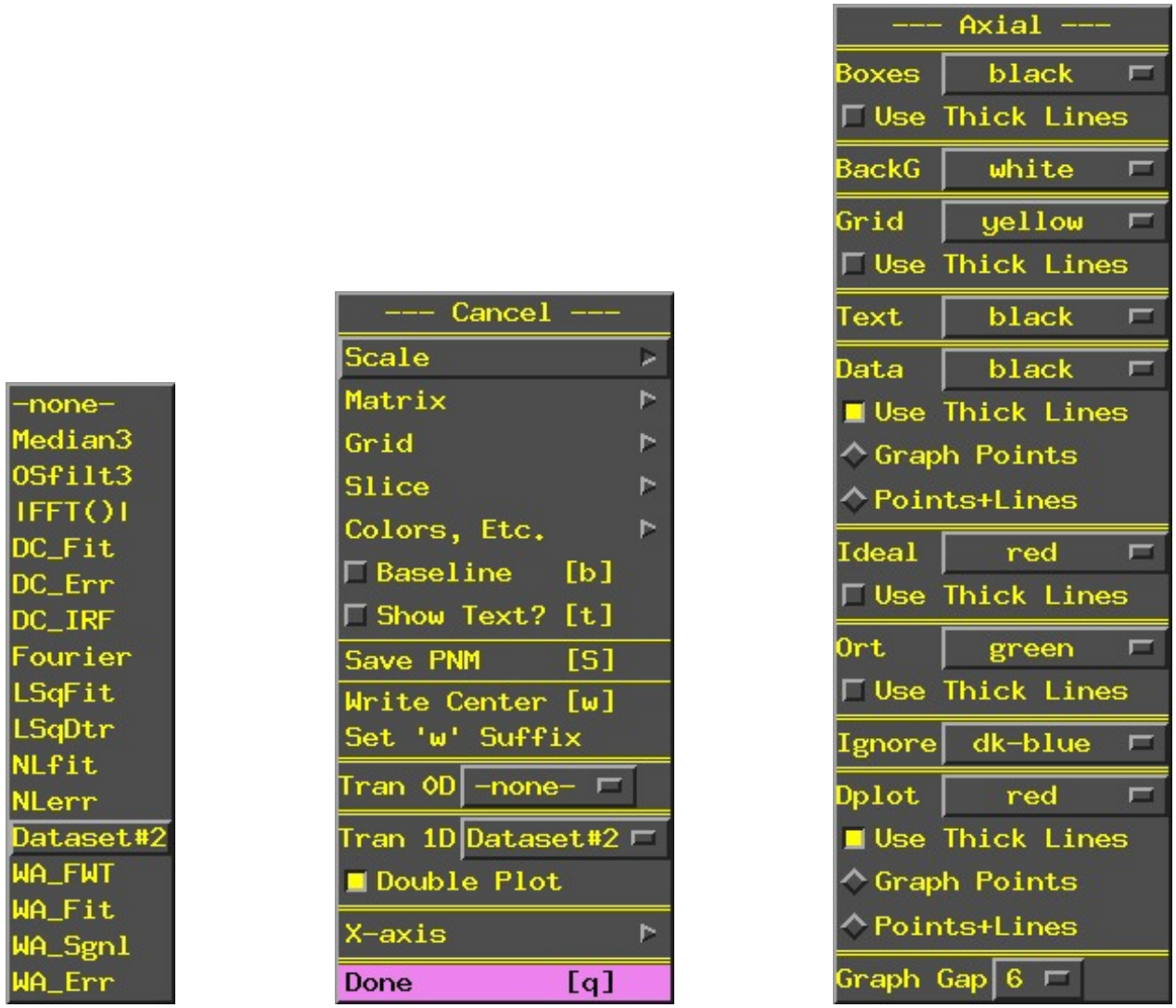


Dataset#2 Plugin Control window

Dataset Chooser window

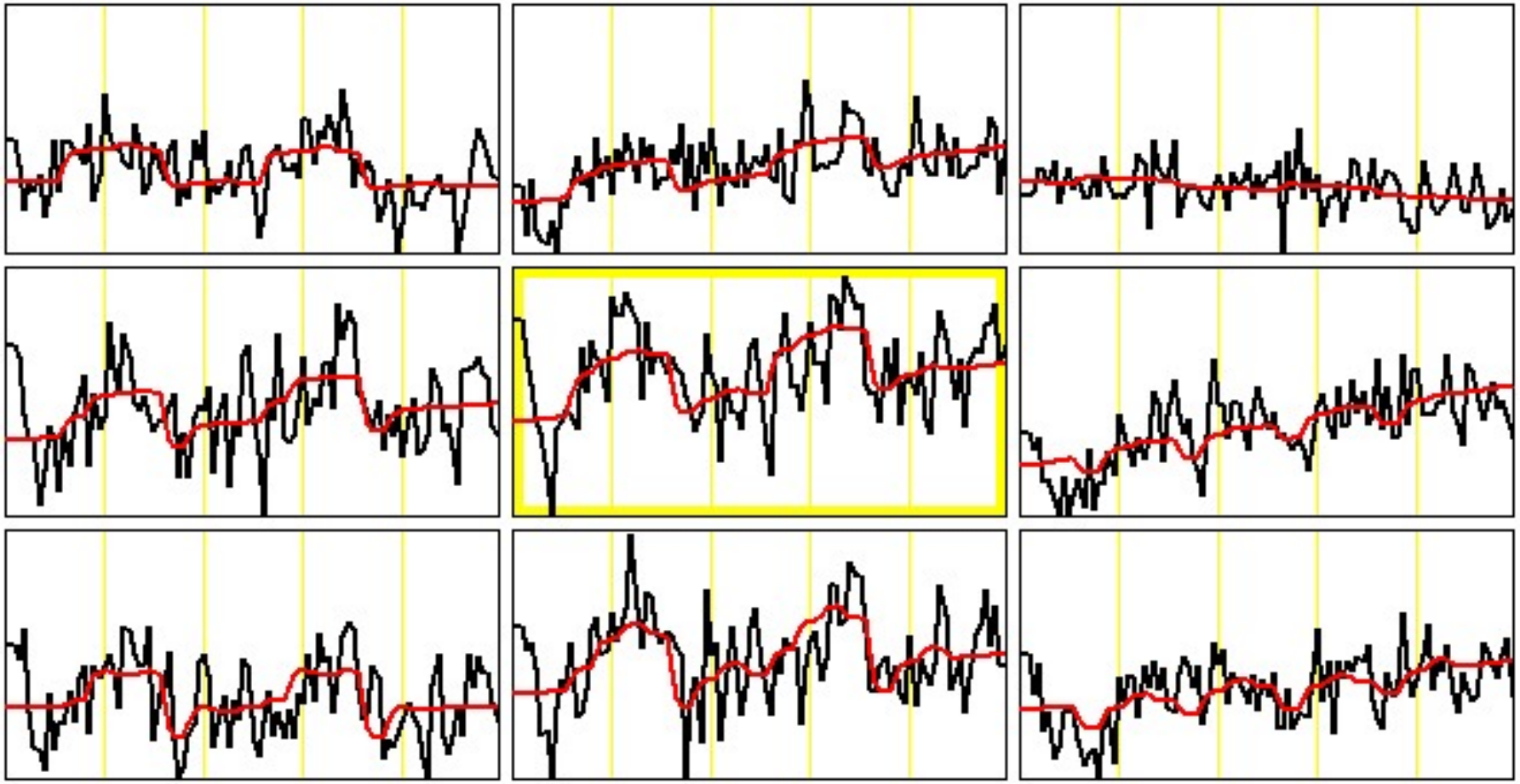
▷ Choose the fit dataset by clicking `-- Choose Dataset --`
Then click on the `epi07a_fitts+orig` choice in the chooser
Then `Set` in the chooser (will close chooser window)
Then `Run+Close` in the plugin's window (tells AFNI about your choice)

▷ From the graph window's Opt menu:
select Tran 1D→Dataset#2
press in Double Plot
press in Opt→Use Thick Lines (under the Dplot item)



The menus you will encounter when setting up for Double Plot

▷ Sample Dataset#2 `Double Plot` (x,y,z=–48 [R], 14 [P], –26 [I]):

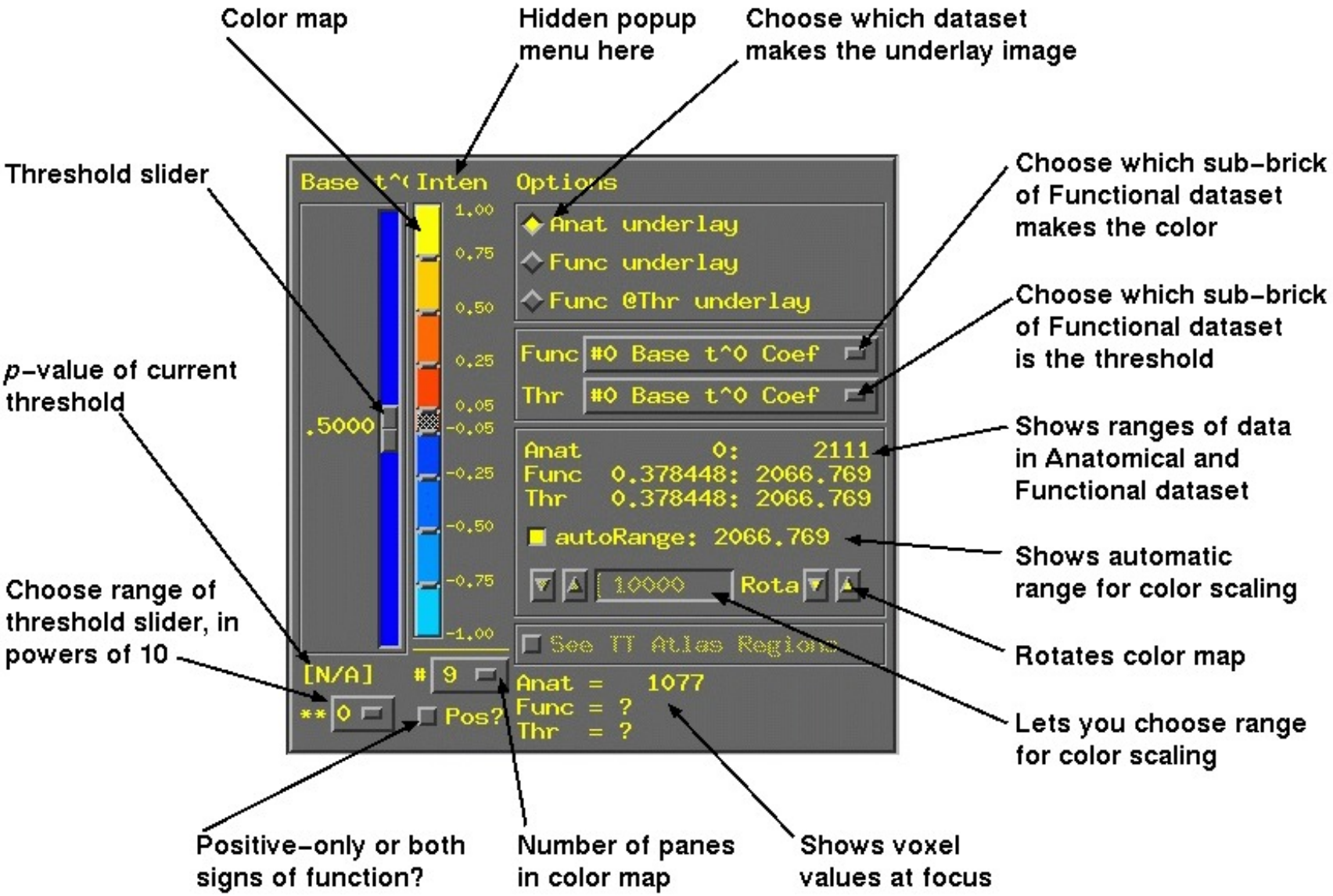


▷ Can get identical graphs using the `Deconvolution` plugin and `Double Plot`-ing with the `Tran 1D⟶DC_Fit` transformation function

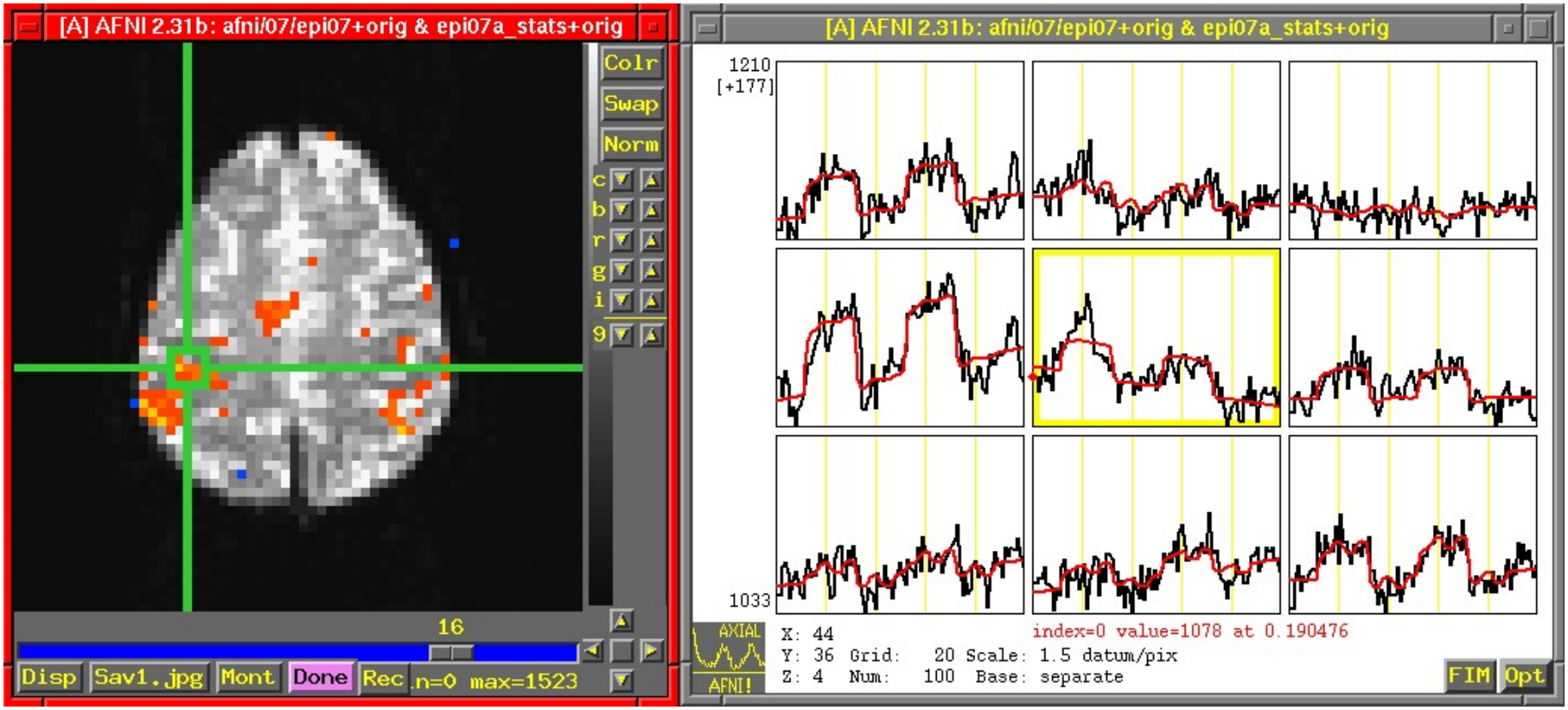
- Regression analysis done "on-the-fly" for voxels shown in graph window
- Button3 graph popup shows fit parameters/statistics for sub-graph fit

↪ Let's look at the color overlay from the functional dataset

▷ Open `Define Function` control panel



▷ Set the `Func` control to `#6 motor[0] Coef`
Set the `Thr` control to `#8 Full F-stat`
Set threshold slider range to `** 1`
Set threshold slider to about 9.8 (p-value about `1.0-5` $= 1.0 \times 10^{-5}$)
Click on `See Function`
Move Axial image viewer to slice 16 (where we'll see some motor activity)



▷ Exercises:
- Find motor voxels that are modulated by visual stimuli
- Find visual activation areas

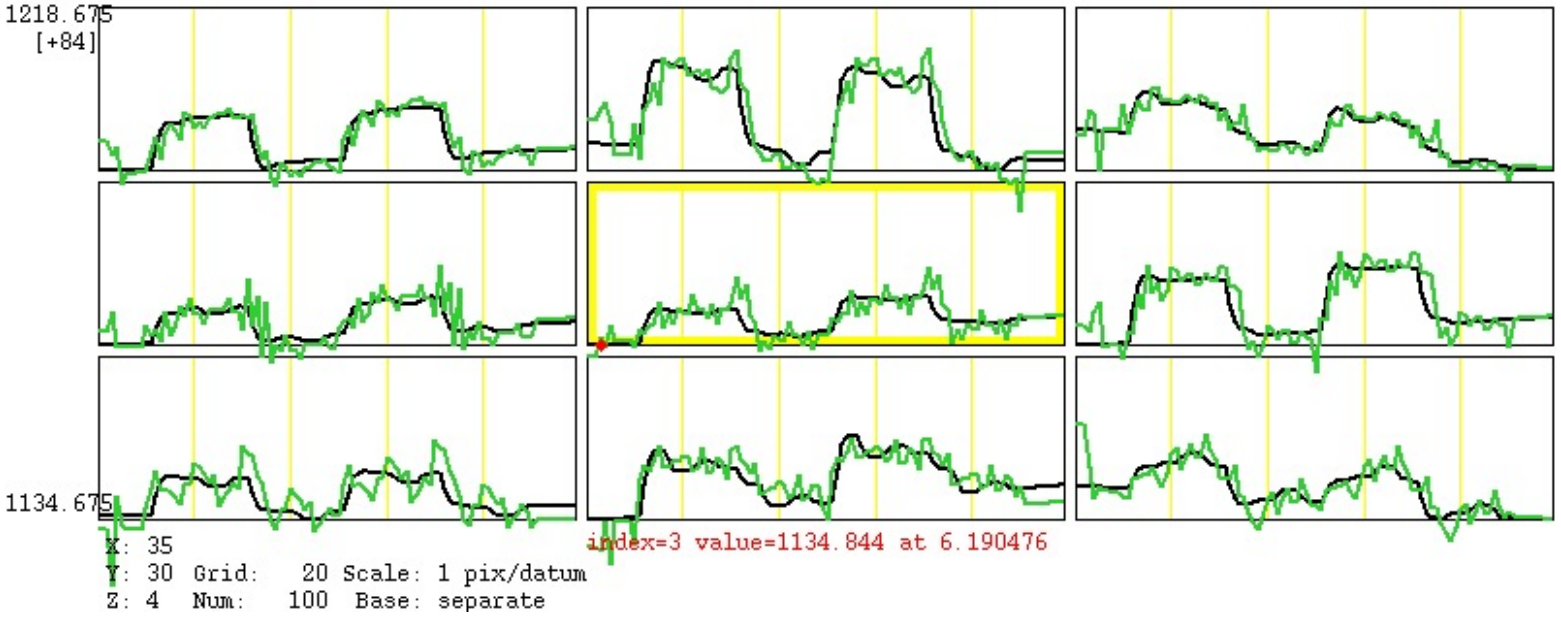
- Second analysis: Deconvolution
  - ◇ Since each stimulus interval for each stimulus class is the same in this experiment (4×10 s for visual, 2×40 s for motor), could try deconvolve whole response for each class
    - ↪ Would treat experiment as having 4 visual events of each class (with about 10 lags), and 2 motor events (with about 25 lags)
    - ↪ But wouldn't work very well, since we'd be estimating 2+10+10+25=57 parameters from 100 time points ⇒ garbage results (over-fitting the data)
      - ▷ Exercises:
        - Explain this calculation
        - Carry out this analysis anyway, and see what happens
  - ◇ Instead, will treat each TR with a stimulus as a discrete event, and try to deconvolve hemodynamic response function
    - ↪ Not an ideal experiment for this analysis, since only have 2 "turn on" and 2 "turn off" transitions for motor response
    - ↪ Will use lags from 0 to 6 (0 s to 12 s), since expect hemodynamics to be about that long
    - ↪ Will have 2+7+7+7=23 unknown parameters (vs. 2+1+1+1=5 in the simple regression analysis)
    - ↪ Input functions will be the 0-1 stimulus time series files created earlier

- Carry out deconvolution analysis (cf. Script File `decon_07b`):

```
3dDeconvolve -input epi07+orig.HEAD                \
             -num_stimts 3                         \
             -stim_file 1 ann_stim_07.1D           \
             -stim_file 2 antiann_stim_07.1D       \
             -stim_file 3 righthand_stim_07.1D     \
             -stim_label 1 annulus                 \
             -stim_label 2 antiann                 \
             -stim_label 3 motor                   \
             -stim_minlag 1 0  -stim_maxlag 1 6    \
             -stim_minlag 2 0  -stim_maxlag 2 6    \
             -stim_minlag 3 0  -stim_maxlag 3 6    \
             -fitts epi07b_fitts                   \
             -iresp 1 epi07b_iresp_ann             \
             -iresp 2 epi07b_iresp_antiann         \
             -iresp 3 epi07b_iresp_motor           \
             -sresp 1 epi07b_sresp_ann             \
             -sresp 2 epi07b_sresp_antiann         \
             -sresp 3 epi07b_sresp_motor           \
             -fout -tout -bucket epi07b_stats
```

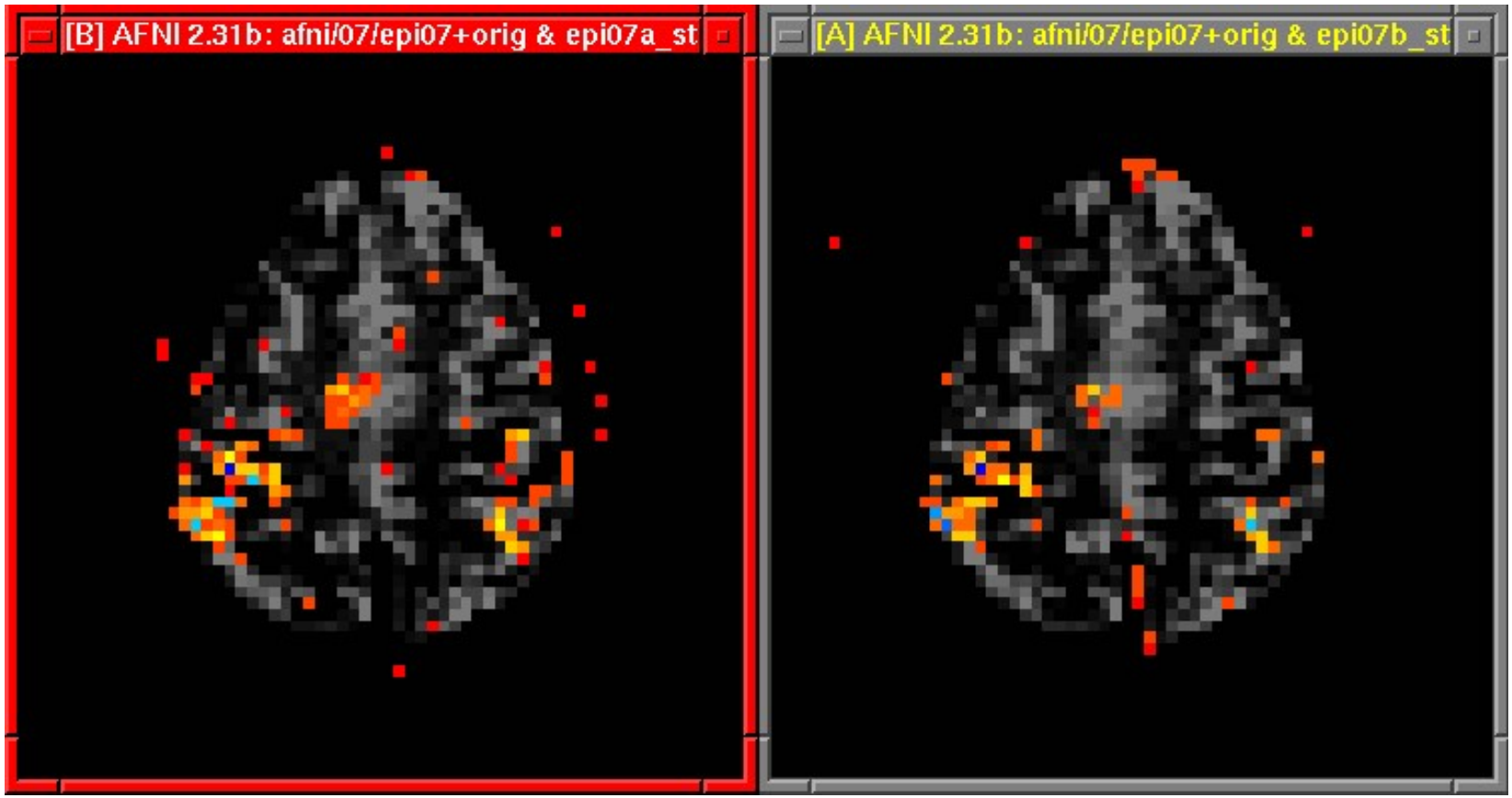
  - ◇ Saving impulse response and its standard error for graphing (`-iresp`, `-sresp`)
  - ◇ Saving individual lag coefficient t-statistics this time (`-tout`)

- Different ways to look at the results with AFNI:
  - ◇ Graph `epi07b_iresp_*` time series with `epi07b_sresp_*` `Double Plot`-ed to see estimate hemodynamic response and it standard error together
  - ◇ Set the threshold sub-brick to the full F-statistic to see where any activation was detected
  - ◇ Set the threshold sub-brick to the partial F-statistics to see where individual stimuli correlated activity was detected
  - ◇ Compare the simple regression and deconvolution time series fits



↪ Looks like deconvolution is "over-fitting" the data somewhat

◇ Compare the simple regression and deconvolution activation maps



Motor: Simple Regression

Motor: Deconvolution

↪ Displaying Axial slice 16 (again)

↪ Thresholding on `Full F-stat`, set to p=`1.0e-4`=10^{-4} (per voxel)

↪ Color overlay is from `motor F-stat` (using `Pos`, `autoRange`, 11 colors)

↪ Exercise: Why is there somewhat less activation in the Deconvolution map?