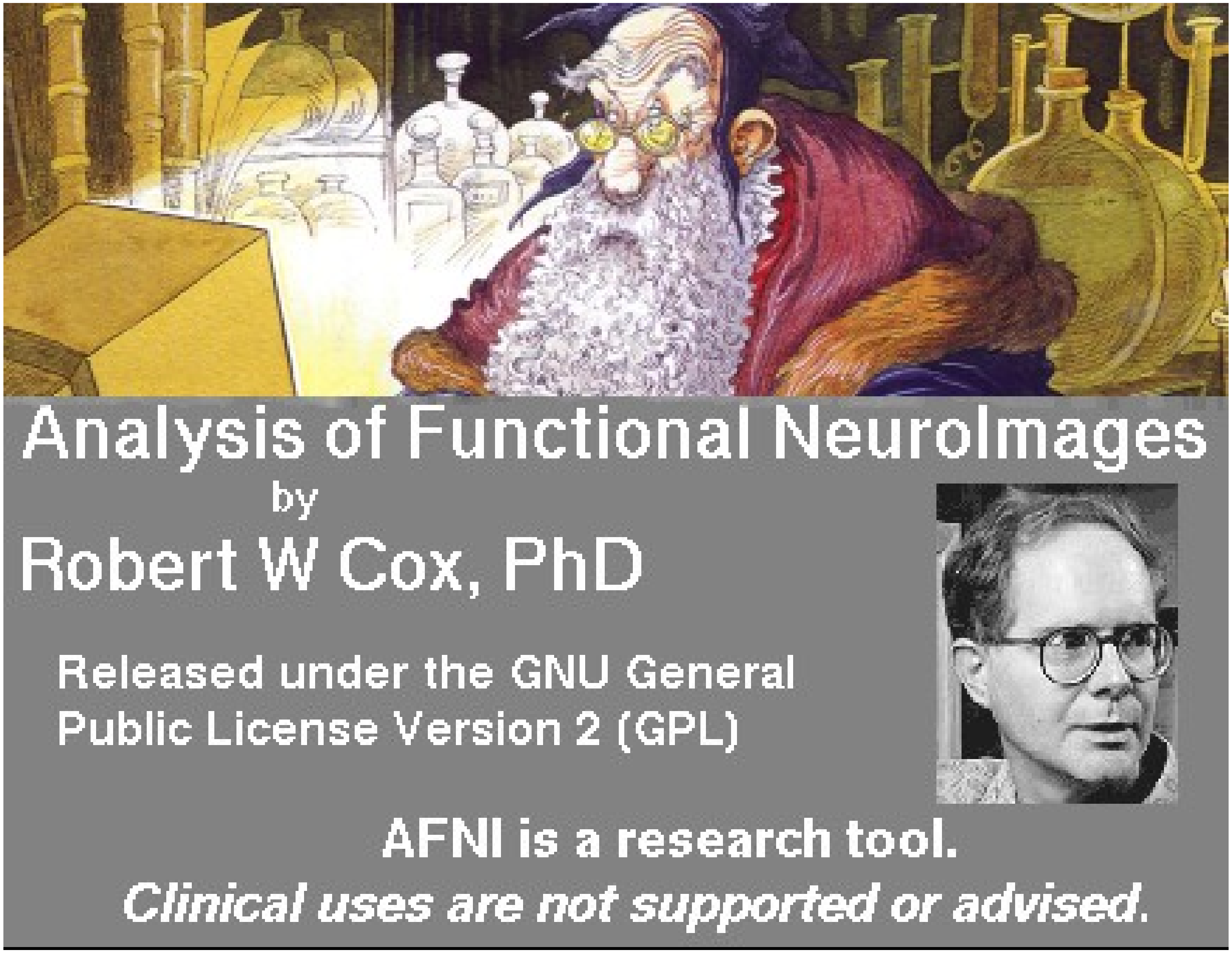
AFNI: Introduction, Concepts, and Principles



Analysis of Functional NeuroImages

by

Robert W Cox, PhD

Released under the GNU General Public License Version 2 (GPL)

AFNI is a research tool.

Clinical uses are not supported or advised.

`http://afni.nimh.nih.gov/afni/` Main Web page

`http://afni.nimh.nih.gov/afni/edu/` Educational materials

FMRI Data Analysis Principles

- Data analysis always takes place in the context of a mathematical and statistical model for the relationship between the observations (the numbers in the images) and the desired information (the location, timing, . . . , of neural activity)
- Things that make FMRI data analysis intricate:
  - ▷ The data↔activity/physiology relationship is complex with many facets, and is only partially understood
  - ▷ Different experimental designs interact to varying degrees with different facets of the data↔activity/physiology relationship:
    - ◇ Block paradigms:
      Low frequency noise in FMRI data is often larger than high frequency noise
    - ◇ Event-related paradigms:
      Additivity of the BOLD response is open to question when the responses from closely spaced stimuli overlap;
      BOLD signal changes are smaller than for block paradigms
    - ◇ Pharmacologic challenges:
      Usually at very low frequencies (mHz or less);
      Mathematical model of response is not clear;
      Challenge itself may alter the hemodynamic response

◇ Patient vs. Normal studies:
Patients tend to move their heads more, making their data different from controls in a non-interesting way that can confound interesting differences (e.g., smaller areas of activation in patients may be due to poorer activation detectability due to larger/more frequent head movements)

◇ Inventiveness of Neuroscientists and Psychologists:
It seems that practically every time I turn around, they come up with a new experiment, or a new type of information about the activation that they want to extract from the data

- As a result, FMRI data analysis is not now (nor likely to be soon) a stereotyped activity that can be packaged into a 'do everything' software package that solves all your problems for you

- Instead, the neuroscientist must understand the mathematical models of the signal and the noise in order to pick the tools to use, to know when the existing tools aren't adequate, and to prompt the development of new tools

  ▷ 'Tools' here means mathematical models, algorithms for extracting information from data using the models, and software that implements the algorithms

- A very solid grounding in statistics is the most important non-neuroscience/non-psychology skill that a successful FMRI investigator needs

  ▷ Otherwise you will use the wrong analysis tools, or misuse the correct tools

AFNI = Analysis of Functional NeuroImages

- Developed to provide an environment for FMRI data analyses, and a platform for development of new software to fit into this environment

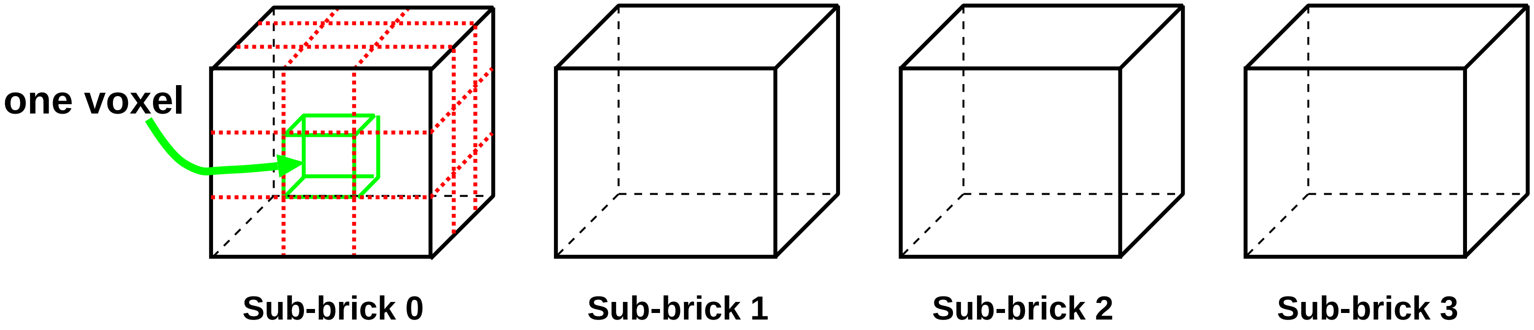
- AFNI refers to both the program of that name and the entire package of auxiliary programs and plugins (more than 100 of them now; some trivial, some complex)

- Important principles I had in mind when creating AFNI (starting back in 1994):
  - Allow user to stay close to data and to view it in many different ways
  - Allow user to put pieces together in different ways to make customized analyses
  - "Provide mechanism, not policy"
  - Allow other programmers to add features that can interact/cooperate with the rest of the package

- Important principles I try to follow:
  - Fix significant bugs as soon as possible ⇒ but I get to define 'significant'
  - Nothing is secret or hidden ⇒ but maybe not very well documented
  - "Release early and often" ⇒ all users are beta-testers for life
  - Help the user ⇒ until my patience runs thin

Fundamental AFNI Concepts

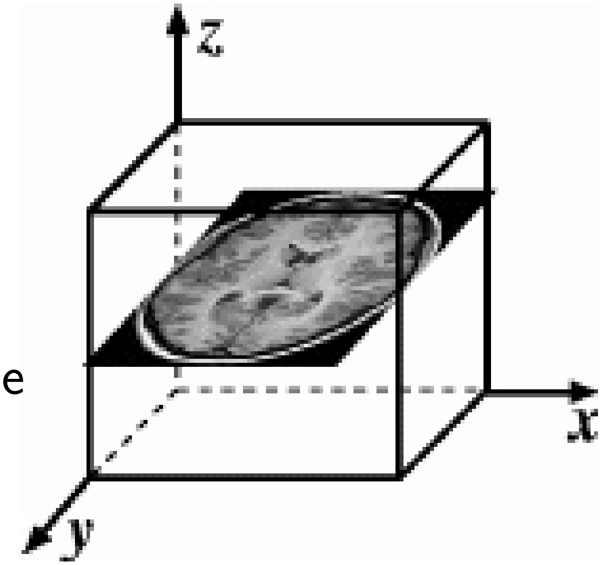
- AFNI is the name of both the original interactive program and the collection of batch (command line) programs that have grown up around it in the last 7 years.
- Basic unit of data in AFNI is the dataset:
  - ▷ A collection of 1 or more 3D arrays (tables) of numbers
    - ◇ Each entry in the array is in a particular spatial location in a 3D grid (a voxel)
    - ◇ Image datasets: each array holds a collection of slices from the scanner; each number is the signal intensity reported by the scanner for that particular voxel
    - ◇ Derived datasets: each number is computed from one or more other datasets (e.g., each voxel value is the average intensity at that spatial location from a number of different image acquisitions)
  - ▷ Each 3D array in a dataset is called a sub-brick; there is one number in each voxel in each sub-brick:



[Cartoon of a 3×3×3 AFNI Dataset with 4 Sub-Bricks]

- Different types of numbers can be stored in datasets: 8 bit bytes (grayscale photos); 16 bit shorts (MR images); 32 bit floats (calculated values); 64 bit complex
- Besides the voxel values, a dataset also contains auxiliary information, including (some of which is optional):
  - ▷ xyz-dimensions of each voxel (in mm)
  - ▷ Orientation of dataset axes;
    for example, x-axis $=$ R-L, y-axis $=$ A-P, z-axis $=$ I-S ⇒ Axial slices

    R=Right, L=Left,
    A=Anterior, P=Posterior,
    S=Superior, I=Inferior



  - ▷ Location of dataset in scanner coordinates;
    in particular, the slice offset $=$ z-coordinate of 1st slice
  - ▷ Time/date of creation
  - ▷ Time between sub-bricks (TR), for 3D+time datasets
    - ◇ 3D+time datasets are the basic unit of FMRI data (one per imaging run)
  - ▷ Type of data stored in dataset (anatomical? functional?)
  - ▷ Statistical parameters associated with each sub-brick;
    e.g., a t-statistic sub-brick has the degrees of freedom (a single number) stored

- Datasets are stored in 2 files:
  - ▷ The `.BRIK` file holds all the numbers in all the sub-bricks (and nothing else): a 100×100×100 dataset would have 1,000,000 numbers per sub-brick
  - ▷ The `.HEAD` file holds all the auxiliary information
- Datasets can be in one of 3 coordinate systems (also called views):
  - ▷ Original or `+orig` view: The original (scanner) coordinate system
  - ▷ AC-PC Aligned or `+acpc` view: Where the dataset is rotated so that the anterior commissure (AC) and posterior commissure (PC) are horizontal, the AC is at $(x, y, z) = (0, 0, 0)$, and the longitudinal (inter-hemispheric) fissure is vertical
  - ▷ Talairach or `+tlrc` view: Where the dataset has been further squeezed/stretched to have its size conform to the Talairach-Tournoux atlas;
coordinates in this view are often called Talairach or stereotaxic coordinates
- Dataset filenames consist of 3 parts:
  - ▷ The user-selected prefix (almost anything)
  - ▷ The view (one of `+orig`, `+acpc`, or `+tlrc`)
  - ▷ The suffix (one of `.HEAD` or `.BRIK`)
  - ▷ Example: files `ElvisPresley+tlrc.HEAD` and `ElvisPresley+tlrc.BRIK`
  - ▷ When creating a dataset with an AFNI program, you specify the prefix — the program supplies the rest

- Datasets are stored in directories called sessions
  - ▷ All the datasets in the same session, with the same view, are presumed to be aligned in xyz-coordinates — voxels with the same value of (x, y, z) are presumed to correspond to the same brain location
  - ▷ In the interactive AFNI program, you can overlay one dataset on top of another, even if their orientations and voxel sizes differ — this is how low-resolution functional data is combined with high-resolution anatomical data to produce cool looking images
  - ▷ Typical AFNI contents of a session directory are all the data derived from a single scanning session for one subject:
    - ◇ Anatomical dataset (1 sub-brick, typically a SPGR or MP-RAGE volume)
    - ◇ 10–20 3D+time datasets from EPI functional runs (100? sub-bricks each)
    - ◇ Statistical datasets derived from the 3D+time datasets, showing functional activation (you hope)
    - ◇ Datasets transformed from `+orig` coordinates to `+tlrc` coordinates — for comparison and integration with datasets from other subjects
- AFNI will load all the datasets in all the directories specified on its command line

  — Might be 100s or 1000s of datasets

  — current limit: max of 1024 datasets per session

  — current limit: max of 80 sessions per AFNI run

Installing AFNI on Your Computer

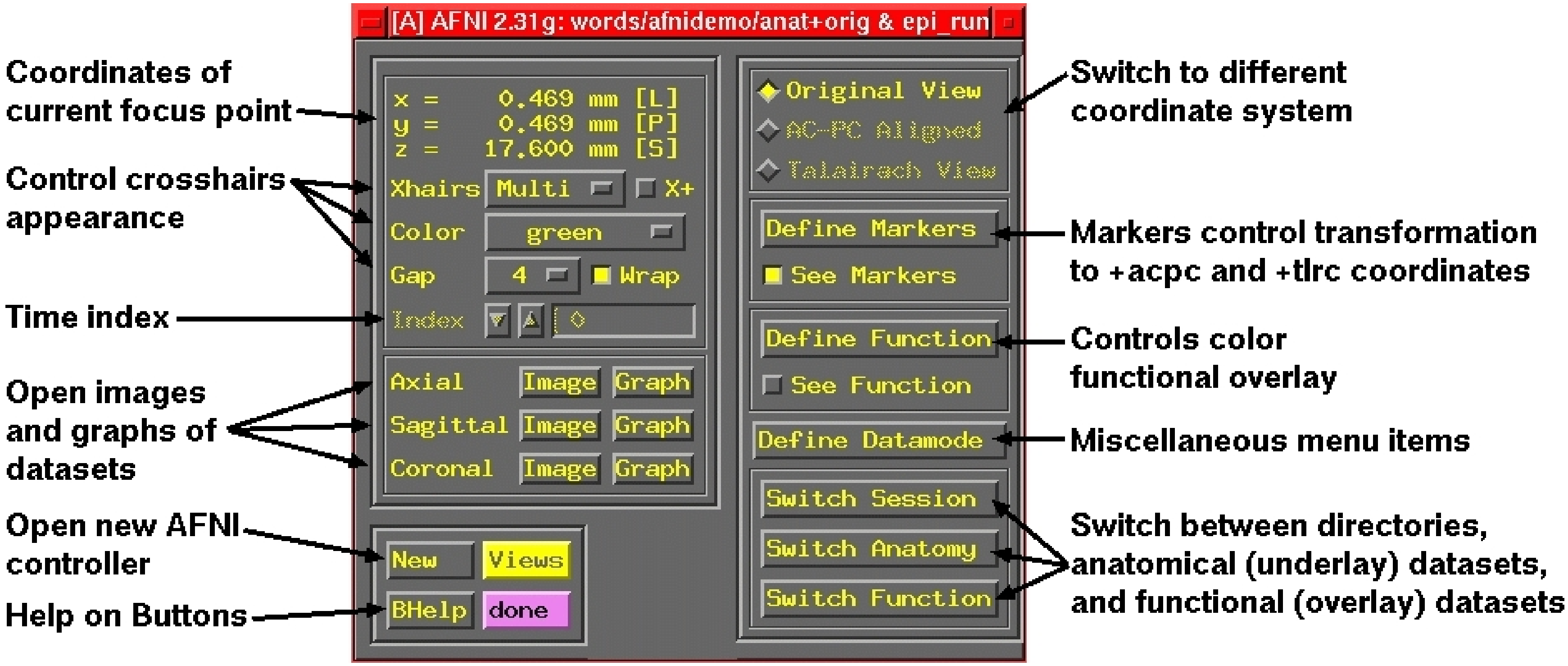
- AFNI runs on Unix+X11 systems: Linux, Sun, SGI, HP, Mac OS X
  — Can run under Microsoft Windows if you install free Cygwin Unix emulator
- `Best way to install`: have a computer that somebody else manages for you, and get him/her to do this task
- `Second best way`: if you are at the NIH, then I can auto-install AFNI upgrades onto your system, if you give me an account with rsh, ssh, or ftp access
- `Third best way`: install a pre-compiled binary package from the AFNI distribution computer using program wget;
  — the following command installs the Linux (Mandrake 7.2) binaries:

```
    wget -nv -m -np -nH -P /usr/local/abin --cut-dirs=3 \
         ftp://afni.nimh.nih.gov/AFNI/bin/linux_mdk72
```

  — Solaris 2.6/2.8, IRIX 6.5 binaries also available at this time
- `Fourth best way`: download the source code and compile it
- AFNI Web site: http://afni.nimh.nih.gov/afni
  — Documentation, FAQ list, Installation instructions, Presentations, Humor, . . .
- AFNI is updated fairly frequently (monthly or so)
  — the auto-install and wget methods are good ways to stay current

Using AFNI Interactively

- Start AFNI from the command line
  - ▷ `afni` reads datasets from current directory
  - ▷ `afni dir1 dir2 ...` reads datasets from directories listed
  - ▷ `afni -R` reads datasets from current directory and from all directories below it
- AFNI reads the file named `.afnirc` from your home directory, if it is present
  - ▷ Used to change many of the defaults (cf. file README.environment)



[A] AFNI 2.31g: words/afnidemo/anat+orig & epi_run

```
x =     0.469 mm [L]
y =     0.469 mm [P]
z =    17.600 mm [S]
```

Xhairs: Multi □ X+

Color: green

Gap: 4 ■ Wrap

Index 0

◆ Original View
◇ AC-PC Aligned
◇ Talairach View

■ See Markers

□ See Function

Coordinates of current focus point

Control crosshairs appearance

Time index

Open images and graphs of datasets

Open new AFNI controller

Help on Buttons

Switch to different coordinate system

Markers control transformation to +acpc and +tlrc coordinates

Controls color functional overlay

Miscellaneous menu items

Switch between directories, anatomical (underlay) datasets, and functional (overlay) datasets

AFNI controller window at startup

- Miscellaneous features of the AFNI controller window:
  - ▷ xyz-coordinate display in upper left corner shows current focus location
    - ◇ By default, the coordinates are in RAI order (from the DICOM standard):

      x = Right (negative) to Left (positive)
      y = Anterior (negative) to Posterior (positive)
      z = Inferior (negative) to Superior (positive)
    - ◇ This display order can be changed to the neuroscience imaging order LPI:

      x = Left (negative) to Right (positive)
      y = Posterior (negative) to Anterior (positive)
      z = Inferior (negative) to Superior (positive)
  - ▷ The `BHelp` button: when pressed, the cursor changes to a hand shape; use it to click on any AFNI button and you will get a small help popup
    - ◇ AFNI also has ‘hints’ (AKA ‘tooltips’)
  - ▷ Press the `New` button to open a new AFNI controller
    - ◇ Used to look at more than one dataset at a time
    - ◇ `Define Datamode`→`Lock` can be used to lock controllers together by co-ordinates

      ↪ All viewing windows within a controller are always locked together
  - ▷ Press the `Views` button to close/open the control panel at right

- ▷ Press the `done` button twice within 5 seconds to exit AFNI
  - ◇ First press changes ‘`done`’ to ‘`DONE`’
    - ↪ Fail to press second time in 5 s changes back to ‘`done`’
  - ◇ Whatever you do, don’t press a mouse button in the blank area to the right of `done`.
    - ↪ I won’t be responsible for the consequences.
- ▷ The `Switch` buttons let you control which datasets are being viewed
  - ◇ `Switch Session` controls which directory datasets are drawn from
  - ◇ `Switch Anatomy` controls the background (grayscale) dataset
    - ↪ Current anat dataset determines resolution of and 3D region covered by image viewers
  - ◇ `Switch Function` controls the overlay (color) dataset
    - ↪ Func datasets will be interpolated (if needed) to anat resolution, and flipped (if needed) to anat orientation
  - ◇ Current datasets are named in AFNI controller titlebar

- Touring the Image Viewer

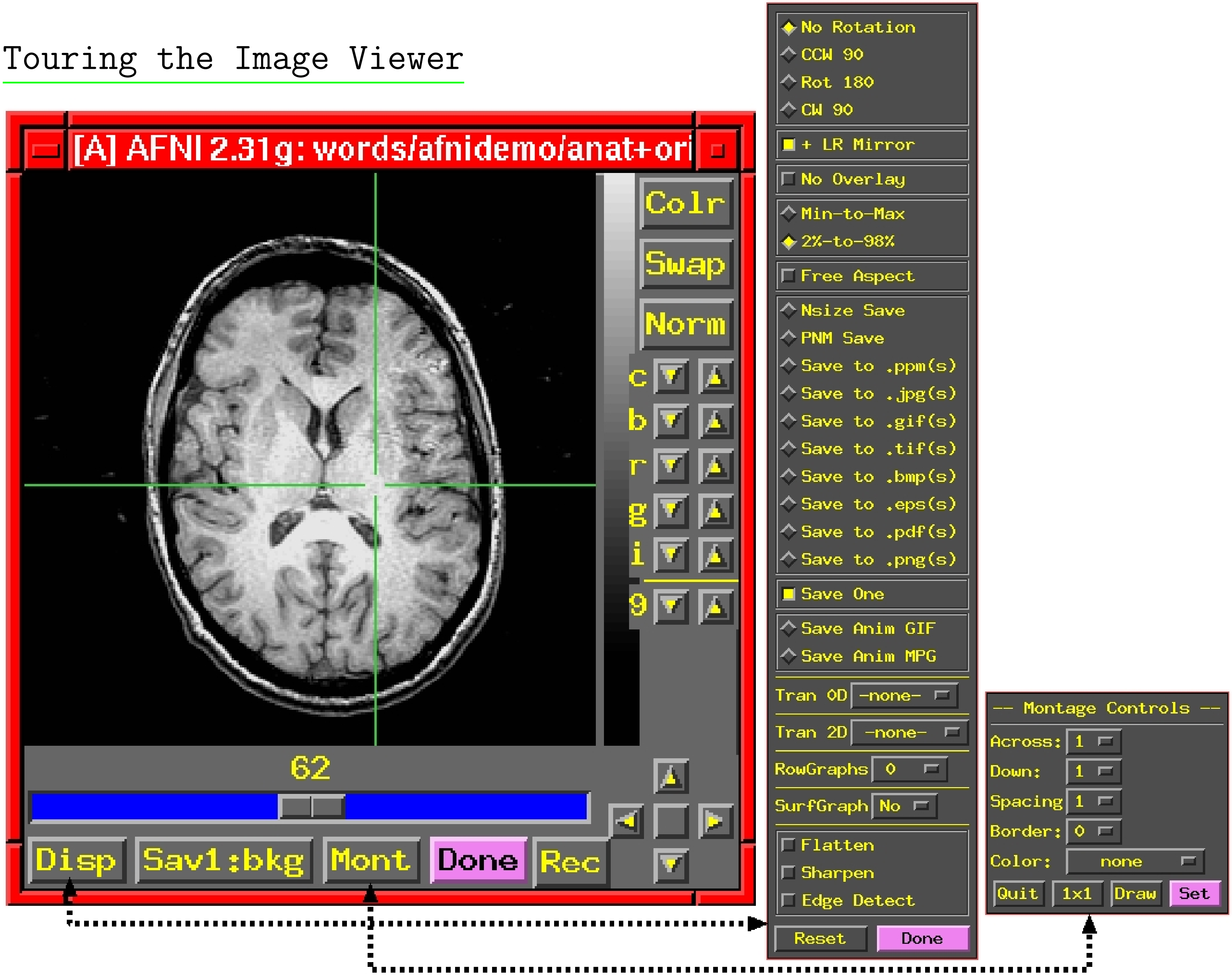


Image viewer; Disp and Mont control panels

▷ Crosshairs show the current focus location

- ◇ Also show the cut planes for the other image viewers
- ◇ When using image montage, other viewers show multiple crosshairs
- ◇ Can control crosshair color and gap size from main AFNI controller

▷ Slider below image lets you move between slices

- ◇ Left-click and drag 'thumb' to move past many slices
- ◇ Left-click ahead or behind thumb to move 1 image at a time
  - ↪ Hold click down to scroll continuously through slices
- ◇ Middle-click in 'trough' to jump quickly to a given location

▷ Vertical intensity bar to right of image shows mapping from numbers stored in image to colors shown on screen

- ◇ Bottom of intensity bar corresponds to smallest numbers displayed
- ◇ Top corresponds to largest numbers displayed
- ◇ Smallest-to-largest display range is selected from `Disp` control panel
  - ↪ Or from hidden popup menu on intensity bar
- ◇ All image viewers from all AFNI controllers use the same intensity bar
  - ↪ Unless AFNI is started with the `-uniq` command line option, in which case each AFNI controller's viewers have independent intensity bars
  - ↪ But all image viewers from the same controller always share the same intensity bar

▷ Buttons at right of intensity bar control image display (mostly colors)

- ◇ `Colr` changes grayscale to color spectrum, and back
- ◇ `Swap` swap top of intensity bar with bottom
- ◇ `Norm` returns the intensity bar to normal (after you mess it up)
- ◇ `c` controls contrast
- ◇ `b` controls brightness
  - ↪ Useful combination: `c` ▲ 2–3 times, `b` ▼ 2–3 times
- ◇ `r` rotates the intensity bar (useless, but very fun)
- ◇ `g` changes the gamma factor (nonlinearity) for the intensity bar
- ◇ `i` changes the size of the image in the window
- ◇ `9` changes the opacity of the color overlay
  - ↪ This control only present for X11 TrueColor displays

▷ At bottom right, the arrowpad controls the crosshairs

- ◇ Arrows move 1 pixel in that direction for that window
  - ↪ Sagittal ◀ is same as Axial ▲
- ◇ Central button closes and opens crosshair gap
- ◇ Items on AFNI controller (below xyz display) also alter crosshairs
  - ↪ Can change color, gap size, . . .

- ▷ Buttons along bottom provide various services
  - ◇ `Disp` controls the way images are displayed and saved
    - ↪ Pops up its own control window: most controls change image immediately
    - ↪ Orientation controls at top let you flip image around
    - ↪ `No Overlay` lets you turn color overlays off (crosshairs; function)
    - ↪ `Min-to-Max` ⇒ intensity bar is data min-to-max
    - ↪ `2%-to-98%` ⇒ intensity bar is smallest 2% of data to largest 98%
    - ↪ `Free Aspect` lets you distort image shape freely
    - ↪ `Save` panel controls how images are saved to disk:
      - All buttons off ⇒ saved image file contains slice raw data
      - `Nsize Save` ⇒ same, but images are 2^N in size
      - `PNM Save` ⇒ images are saved in PPM/PGM format (color/gray)
      - `Save to .xxx(s)` ⇒ saves image(s) to specified format
      - `Save One` ⇒ for saving montage
    - ↪ `Tran 0D` lets you transform voxel values before display
      - `Log10` and `SSqrt` useful for images with extreme values
    - ↪ `Tran 2D` provides some 2D image filters (underlay only)
      - `Median9` smoothing can be useful for printing images
    - ↪ `Rowgraphs` lets you graph the voxel values from image rows
      - If you want columns, flip the image with `CCW 90`
    - ↪ `Surfgraph` lets you graph the voxel values in a surface graph

↪ Three extra image processing filters are provided at the bottom

- Sharpen is sometimes useful for deblurring images

↪ Reset sets controls back to what they were when you opened Disp

↪ Done closes this control window

▷ Save lets you save images from viewer to disk files

◇ Warning: Images are saved as sent to the viewer, not as displayed

↪ Means that aspect ratio of saved image may be wrong (non-square pixels)

↪ Can fix this with Define Datamode→Warp Anat on Demand

◇ Save:bkg means it will save the background image data itself, whatever format it may be in (bytes, shorts, floats, complex numbers, RGB byte triples)

◇ Save:pnm means it will save the displayed image in PNM format

↪ PPM for color, PGM for gray-only images

↪ You might have to convert this to some other format

↪ See AFNI FAQ #57 for instructions on image format conversion

◇ Sav1:xxx means it will save the entire Montage in format xxx

↪ This is the only way to save a Montage layout (within AFNI)

↪ Save options will only save single slice images (one or more)

◇ Save.xxx means it will save the image in the "xxx" format

↪ You can also set this using a hidden right-click popup on the Save button

↪ Formats depend on presence of image conversion programs on your system

◇ After you press `Save`, then it asks for a filename prefix

◇ Except for `Sav1.xxx`, it then asks for ‘from’ and ‘to’ slice indexes

↪ You can save many images this way

↪ Filenames are like `prefix.0037.ppm`, for slice number 37, ppm format

↪ `Sav1.xxx` immediately saves its one file after prefix is entered

▷ `Mont` lets you display a rectangular layout of images

◇ Pops up its own little control window

↪ Controls at top do nothing until action is selected at bottom

◇ `Across` and `Down` determine number of sub-images shown

◇ `Spacing` determines how far apart the selected slices are

↪ Every n^{th} slice, for $n = 1, 2, \ldots$

↪ Multiple crosshairs in other image viewers will show montage slices

◇ `Border` lets you put some blank pixels between the sub-images

↪ `Color` lets you choose the color of the border pixels

◇ At the bottom, the action controls cause something to happen

↪ `Quit` closes the Montage control window

↪ `1x1` changes `Across` and `Down` back to 1

↪ `Draw` actually causes the montage to be drawn

↪ `Set` ⇔ `Draw` then `Quit`

▷ `Rec` lets you record images for later `Save`-ing

- ◇ So you can build a sequence of images from any set of AFNI controls
  - ↪ Change colormaps, functional thresholds, datasets, . . .
- ◇ Then save them to disk for animation, etc.
  - ↪ If Unix programs whirlgif and/or gifsicle are installed on your system, AFNI can write GIF animations directly (e.g., for fun Web pages)
  - ↪ If program mpeg_encode is installed, AFNI can write MPEG-1 animations
  - ↪ Source code for these free programs is included with AFNI source code
- ◇ `Rec` button pops down a menu that sets the record mode
  - ↪ `Off` ⇒ recording is off
  - ↪ `Next One` ⇒ next image displayed is recorded, then goes back to `Off`
  - ↪ `Stay On` ⇒ record each image when displayed
  - ↪ Controls below the line determine where in the recording sequence the saved images will be stored
- ◇ Recorded images go into a new image viewer, with its own controls
  - ↪ Its slider moves between recorded images
  - ↪ `Kill` will delete an image from the recorded sequence
  - ↪ `Save` will save record images
    - • Right-click on `Save` to bring up menu of format options
  - ↪ `Done` to close the recorded image viewer

▷ Hidden image popup menu (using Button 3 or right-click)

- ◇ `Jumpback` lets you jump the focus position back to its last place
  - ↪ For when you click in the wrong place and get lost
- ◇ `Jump to (xyz)` lets you enter xyz-coordinates (in mm) and then the focus position will jump there
  - ↪ External program 3dclust can generate xyz coordinates of interest
  - ↪ Once you have +tlrc dataset, can jump to regions from Talairach atlas
- ◇ `Jump to (ijk)` lets you jump to a particular voxel index location
- ◇ `Image display` lets you turn control widgets on and off
  - ↪ Can unclutter screen a little
  - ↪ Useful if you want to make a screenshot

▷ Hidden intensity bar popup menu

- ◇ `Choose Display Range` lets you pick the range of numbers that are mapped to intensity bar colors
  - ↪ Normally, each image is mapped to colors separately when it is displayed
    - Using `Min-to-Max` or 2%`-to-`98% from `Disp`
  - ↪ If you want each image to be mapped the same, then must give bottom-to-top values via this menu item (separate them with spaces)
  - ↪ If you set third (optional) input ‘ztop’ to 1, values above ‘top’ are set to 0
  - ↪ To restore normal auto-mapping, set ‘bot’ and ‘top’ both to 0

◇ `Choose Zero Color` lets you choose the color that is displayed for voxel values that are exactly 0

↪ Can be useful for filling in regions that were set to 0 by some program

↪ For example, values below ‘bot’ from `Choose Display Range` (and above ‘top’ if ‘ztop’ was set to 1)

↪ Choose the ‘none’ color to return to normal display

◇ `Choose Flatten Range` is used to control the `Flatten` filter from the `Disp` control window

↪ This is almost useless — don’t bother to try it

◇ `Choose Sharpen Factor` is used to control the `Sharpen` filter from the `Disp` control window

↪ Larger values mean more sharpening (and more image graininess)

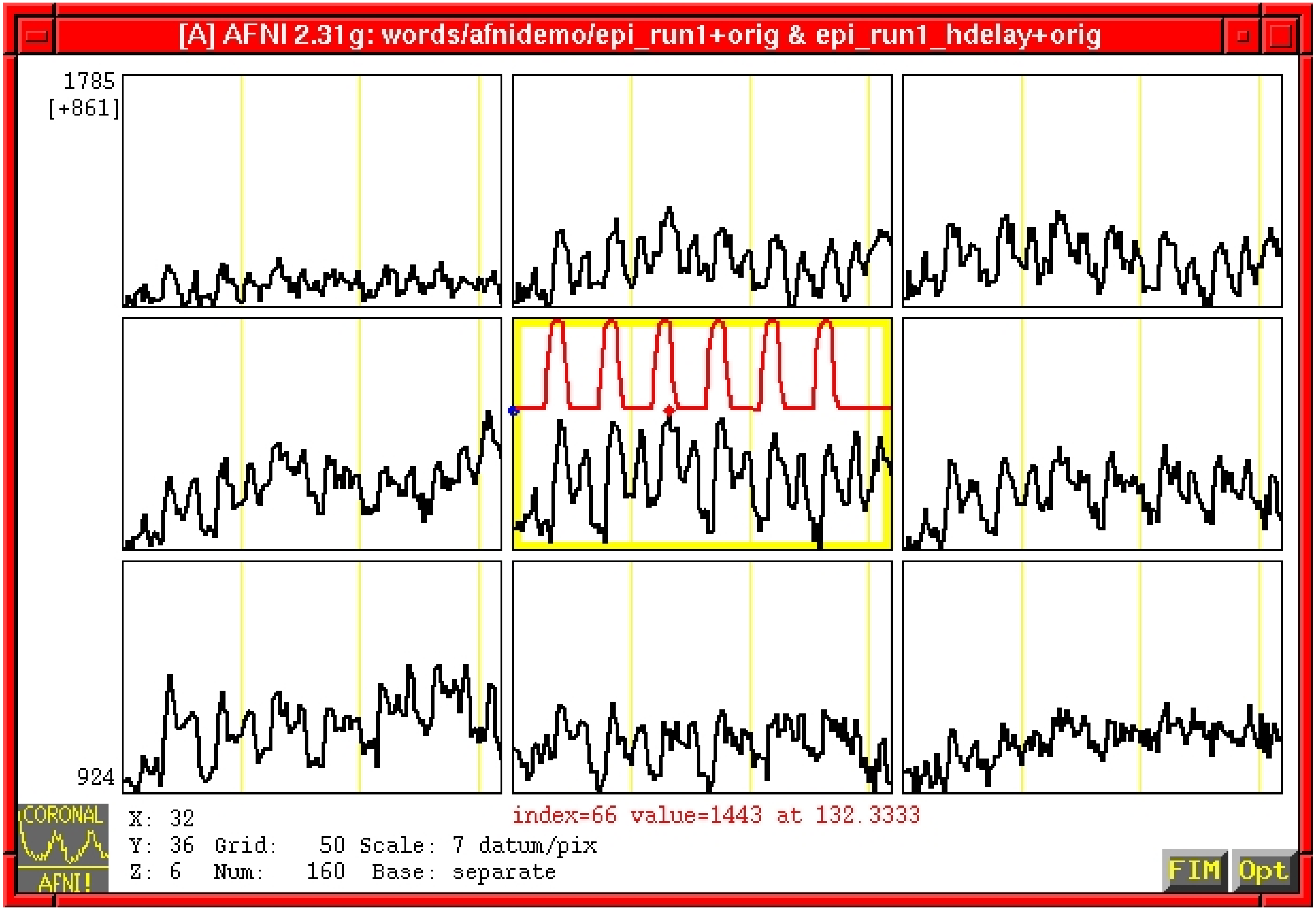
◇ `Plot Overlay Plots` turns overlay graphs on or off

↪ In future, will control overlay of cortical surface geometry.

↪ This feature is experimental now, and not documented.

◇ `Label` and `Size` controls display of slice coordinate overlay.

- Touring the Graph Viewer


[A] AFNI 2.31g: words/afnidemo/epi_run1+orig & epi_run1_hdelay+orig
1785
[+861]
924
CORONAL
AFNI!
FIM
Opt

```
X: 32                                      index=66 value=1443 at 132.3333
Y: 36  Grid:    50 Scale: 7 datum/pix
Z: 6   Num:    160  Base: separate
```

Graph viewer with data (black) and reference (red) waveforms

▷ Graph viewer takes voxel values from same dataset as image viewer

- ◇ If dataset has only 1 sub-brick, graph viewer only shows numbers
- ◇ To look at images from one dataset locked to graphs from another dataset, must use 2 AFNI controllers and `Define Datamode`→`Lock` on AFNI control panel

▷ If graph and image viewer in same slice orientation are both open, crosshairs in image window change to show a box containing dataset voxels being graphed

▷ Central sub-graph (current focus location) is outlined in yellow

- ◇ Current time index is marked with small red diamond
- ◇ Left-clicking in a non-central sub-graph moves that location to focus
- ◇ Left-clicking in central sub-graph moves time index to that point
  - ↪ Can also use `Index` control in AFNI controller
- ◇ Right-clicking in any sub-graph pops up some statistics of its data
- ◇ Left-clicking in icon (lower left corner) causes icon and menu buttons to disappear
  - ↪ Useful if you want to do a screenshot to save window
  - ↪ Left-clicking in same place will bring icon and buttons back

- `Opt` menu button lets you control how graphs appear
  - Many items have [`keyboard`] shortcuts
    - ↪ Make sure you are typing into the correct window!
  - `Scale` changes scale of graphs
    - ↪ Mapping from voxel values to screen pixels
    - ↪ `Down` [`-`] shrinks graphs vertically; `Up` [`+`] expands them
    - ↪ `Auto` [`a`] makes AFNI pick a nice scale factor
    - ↪ `Choose` lets you pick exact scale factor
      - Can choose positive values=`pix/datum` or negative=`datum/pix`
      - `pix/datum` = number of screen pixels for each change of 1 in data
      - `datum/pix` = size of change in data for each screen pixel
    - ↪ Current scale factor is shown below graphs
    - ↪ Scale factor does <u>not</u> change when you resize graph, change matrix, etc.
      - You usually have to auto-scale [`a`] afterwards
  - `Matrix` changes number of sub-graphs
    - ↪ `Down` [`m`] and `Up` [`M`] decrease and increase number
    - ↪ `Choose` lets you pick number exactly
      - Alternative: keyboard [`N`], type number, then [`Enter`] key
      - Range of allowable matrix size is 1..21

◇ `Grid` lets you change spacing of vertical grid lines

↪ Useful for showing regular timing interval (e.g., task timing)

↪ `Down` [g] and `Up` [G] decrease and increase spacing

↪ `Choose` lets you pick number exactly

↪ Current grid spacing is shown below graphs

↪ `Pin Num` lets you pick the horizontal length of the sub-graph

- Default length is number of sub-bricks in dataset
- Make it longer ⇒ graphs end before window
- Make it shorter ⇒ graphs are truncated
- Useful when switching between datasets of different lengths
- Set this to 0 to get back to default operation
- Current number of time points is shown below graphs

↪ `HorZ` [h] will put in a dashed line at the $y = 0$ level in sub-graphs

- Only useful if data range spans negative and positive values!

◇ `Slice` lets you change slices

↪ `Down` [z] and `Up` [Z] move one slice

↪ Can also choose slice directly from menu

↪ Current voxel indexes are shown below graphs

- Corresponds to `Voxel Coords?` display in AFNI controller

- ◇ `Colors, Etc.` lets you alter the colors/lines used for drawing
  - ↪ Lines used for sub-graph frame boxes, grid lines, data graphs, FIM orts/ideals, and double plots can have colors changed and can be made thicker
    - Grid color is also used to limn central sub-graph
  - ↪ Can choose to graph curves as lines, points, or both together
  - ↪ Can change color of background and text
  - ↪ Can change gap between sub-graph boxes
- ◇ `Baseline` [b] changes how the sub-graphs are plotted
  - ↪ All sub-graphs have same scale factor, to convert values into vertical pixels
  - ↪ Baseline is value that gets plotted to bottom of sub-graph
  - ↪ Individual: all sub-graphs have different baselines
    - Baseline = smallest value in each displayed time series
    - This can be confusing; same vertical location doesn't mean same value
    - Shown below graphs as `Base: separate`
  - ↪ Common: all sub-graphs shown at any one time get same baseline
    - Baseline = smallest value in all displayed time series
    - Shown below graphs as `Base: common`
    - Usually need to rescale [a] after changing baseline
  - ↪ Global: all sub-graphs get same baseline even when spatial position changes
    - Set from `Baseline`→`Set Global` menu item
    - Default global level is smallest value in entire dataset

↪ Range of central sub-graph is shown at left of graph region

- Central bottom (baseline) value is shown at lower left
- Upper left shows value at top of central sub-graph box
- Number in [brackets] shows data range of one sub-graph box's height
- If baselines are separate, bot/top values only apply to central sub-graph!!

◇ `Show text?` [t] allows you to see text display of values instead of graphs

◇ `Save PNM` [S] lets you save a snapshot of window to a PNM image file

◇ `Write Center` [w] lets you write data from central sub-graph to a file

↪ File is in ASCII format ⇒ can be imported into other programs

↪ Filename is of form `xxx_yyy_zzz.suffix.1D` (using voxel indexes)

↪ `suffix` is chosen using `Set 'w' suffix` button

◇ `Tran 0D` and `Tran 1D` let you transform the data before graphing

↪ `Log10` and `SSqrt` useful for images with extreme values

↪ `Median3` and `OSfilt3` are for smoothing time series

↪ Other choices are functions controlled by/from plugins

↪ `Double Plot` lets you plot output of `Tran 1D` and original data together

- Color of transformed data from `Dplot` on the `Colors, Etc.` menu
- `Dataset#2` transformation lets you plot two datasets together

- ◇ `X-axis` menu lets you choose how graph x-axis is chosen
  - ↪ Default: x is linear in time
  - ↪ Can instead choose x from a `.1D` format file from disk
  - ↪ Useful only in very limited circumstances
- ◇ `Done` [q] closes the graph viewer window

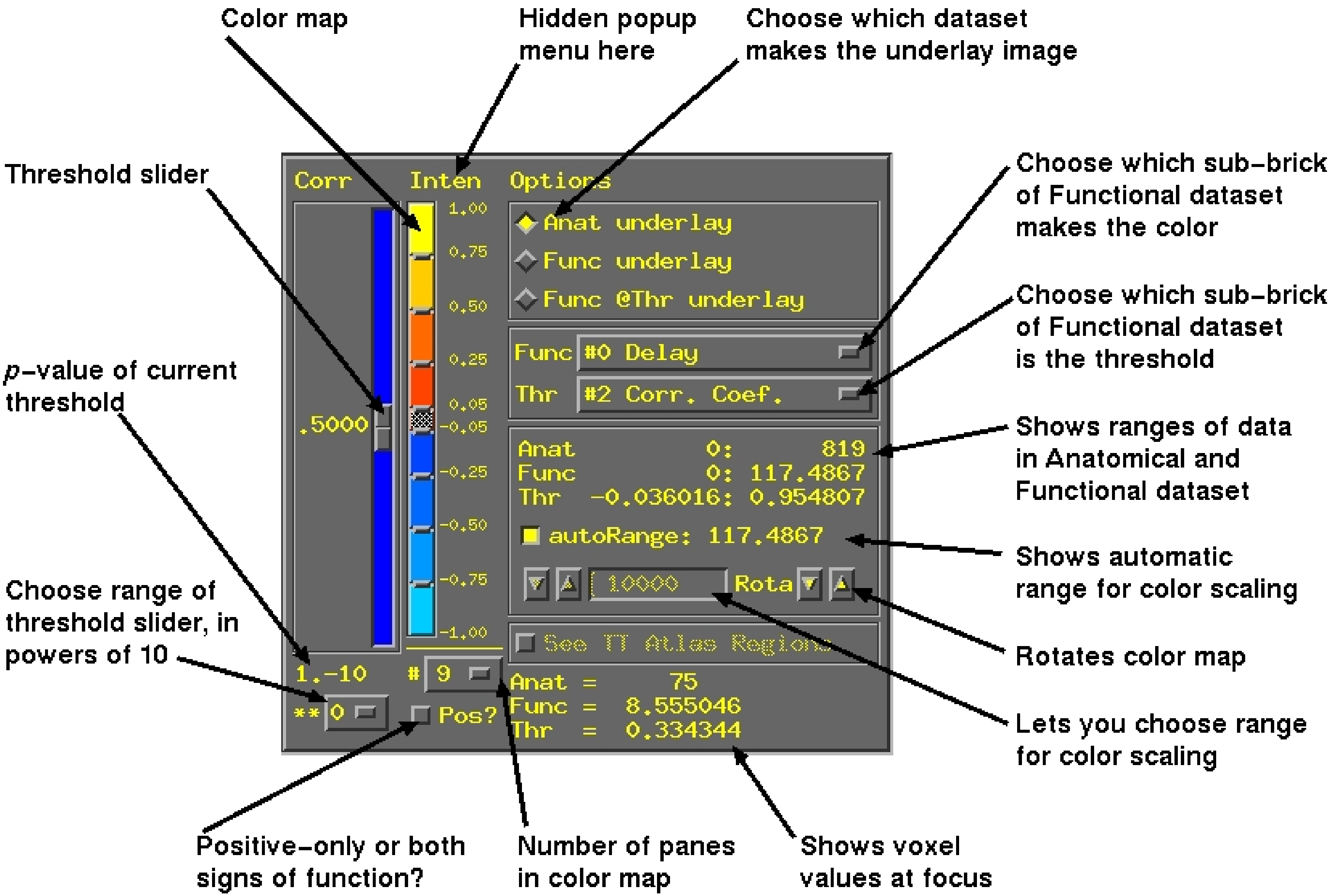
▷ [Keystrokes] in graphs that have no menu items are

- ◇ [<] moves time index down by 1
- ◇ [>] moves time index up by 1
- ◇ [1] moves time index to beginning (time index $= 0$)
- ◇ [l] moves time index to end
- ◇ [L] turns off/on the AFNI logo in the corner

▷ `FIM` menu controls interactive functional image calculations

- ◇ Not documented here; see 'Educational material' pages at AFNI Web site

- Brief Tour of the Functional Color Overlay Controls
  - ▷ Open with Define Function button on AFNI controller



Color map

Hidden popup menu here

Choose which dataset makes the underlay image

Threshold slider

Choose which sub-brick of Functional dataset makes the color

Choose which sub-brick of Functional dataset is the threshold

p–value of current threshold

Shows ranges of data in Anatomical and Functional dataset

Shows automatic range for color scaling

Choose range of threshold slider, in powers of 10

Rotates color map

Lets you choose range for color scaling

Positive–only or both signs of function?

Number of panes in color map

Shows voxel values at focus

Func #0 Delay
Thr #2 Corr. Coef.

```
Anat          0:      819
Func          0: 117.4867
Thr   -0.036016: 0.954807
```

■ autoRange: 117.4867

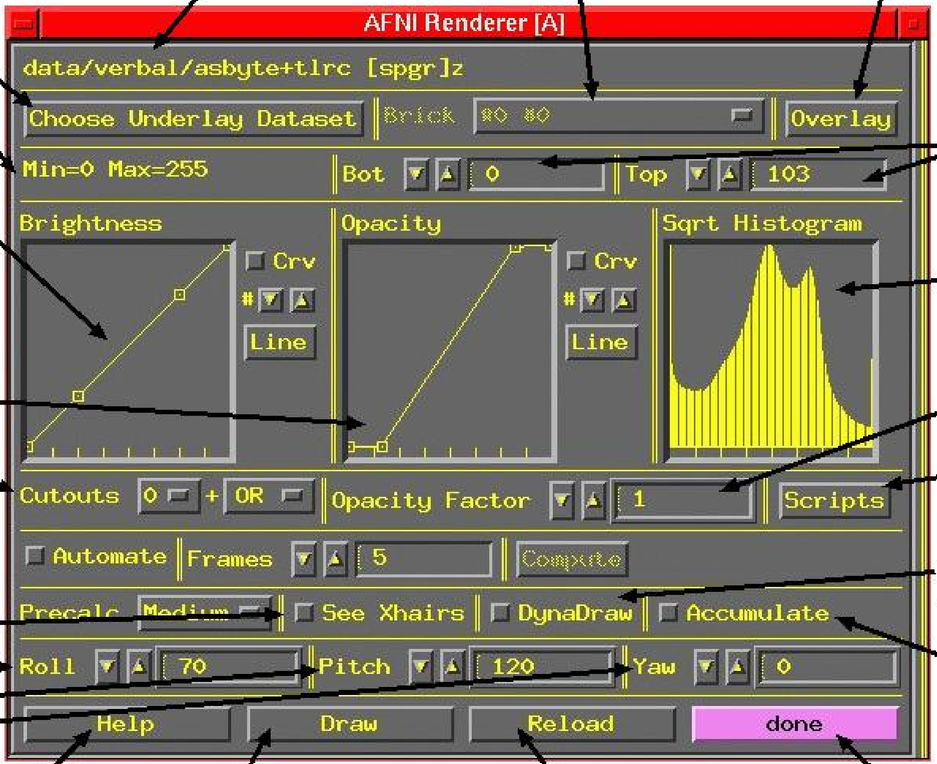
10000

```
Anat =      75
Func =  8.555046
Thr  =  0.334344
```

AFNI Plugins

- Plugins are modules that attach themselves to AFNI and add some interactive capabilites to the GUI program
- There is a (somewhat dated) manual for writing plugins
- Useful plugins:
  - ▷ 3D Registration:
    Provides a GUI control for time series registration (same as 3dvolreg)
  - ▷ Dataset Copy:
    Copy a dataset (useful as a start for ROI drawing)
  - ▷ Dataset NOTES:
    Add arbitrary text notes to a dataset header
  - ▷ Draw Dataset, Gyrus Finder:
    Draw regions-of-interest (ROIs) on 2D slices
  - ▷ Histogram:
    Graph the histogram of a sub-brick, or some parts of it
  - ▷ Deconvolution, NLfit & NLerr:
    Do linear and nonlinear regression interactively on the dataset time series being displayed in a graph viewer

▷ Render Dataset: Volume rendering with functional overlays



AFNI Renderer [A]

data/verbal/asbyte+tlrc [spgr]z

Min=0 Max=255

Bot 0 Top 103

Opacity Factor 1

Frames 5

Precalc Medium

Roll 70 Pitch 120 Yaw 0

Pick new underlay dataset

Name of underlay dataset

Sub-brick to display

Open color overlay controls

Range of values in underlay

Range of values to render

Change mapping from values in dataset to brightness in image

Histogram of values in underlay dataset

Mapping from values to opacity

Maximum voxel opacity

Cutout parts of the 3D volume

Menu to control scripting (control rendering from a file)

Compute many images in a row

Render new image immediately when a control is changed

Show 2D crosshairs

Accumulate a history of rendered images (can later save to an animation)

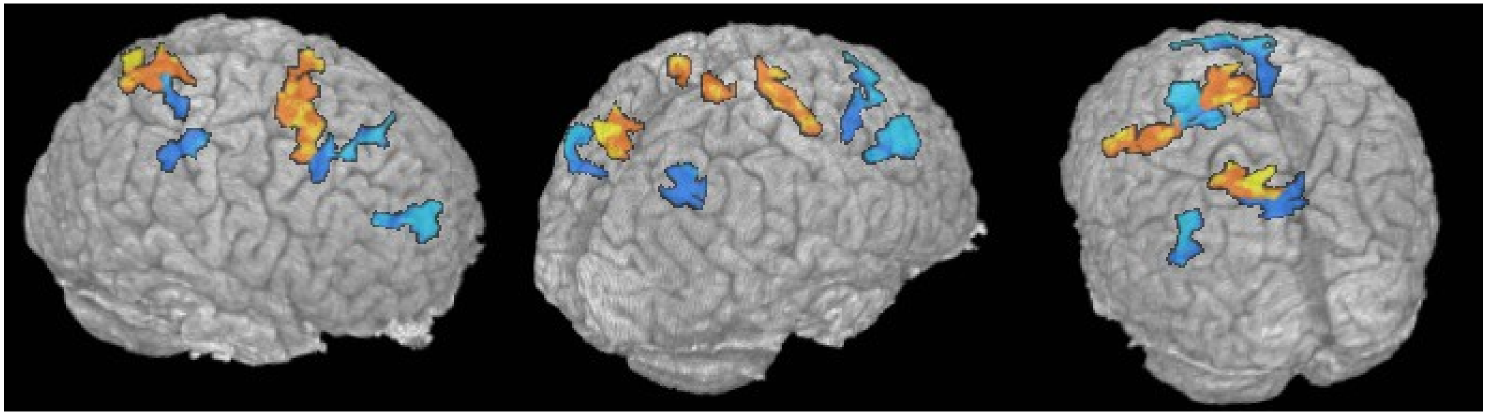
Control viewing angles

Detailed instructions

Force a new image to be rendered

Reload values from the dataset

Close all rendering windows



Being close to your FMRI data doesn't get better than this!

Using AFNI in Batch Mode

- Batch mode programs are run by typing commands directly to the computer, or by putting these commands into text files (scripts) and later executing them
- Advantages of batch mode (over graphical user interface)
  - ▷ Can process new datasets exactly the same way as previous ones
  - ▷ Can link together a series of programs to produce custom results
  - ▷ Programs that take a long time to operate are easier to 'fire and forget' from a script than if they had a GUI
  - ▷ It's easier to write a batch mode program
- Disadvantages of batch mode
  - ▷ Requires typing, rather than pointing-and-clicking
  - ▷ Requires learning/remembering how a program works all at once, rather than (re)discovering it through a kinder gentler interface
  - ▷ Many younger (born after 1970) researchers have virtually no experience with a command line interface, or anything like it
- Many significant AFNI capabilities are only available in batch mode programs
  - ▷ This is especially true of functions that combine data from multiple datasets to produce new datasets

- The 3d* series of programs (generally) take as input one or more AFNI datasets, and produce as output one (or more) new AFNI datasets
- `Time series activation analysis programs`:
  - ▷ 3dfim, 3dfim+, 3ddelay:
    Variations on 'classical' correlation analysis of each voxel's time series with a single reference (ideal) waveform
  - ▷ 3dDeconvolve:
    Multiple linear regression and/or linear deconvolution to fit each voxel's time series to a multi-dimensional signal model (similar models are found in SPM)
  - ▷ 3dNLfim:
    Nonlinear regression to fit each voxel's time series to an arbitrary functional model provided by the user
- `Time series utility programs`:
  - ▷ 3dFourier:
    Fourier domain filtering of voxel time series
  - ▷ 3dTcorrelate:
    Compute correlation coefficient of 2 datasets, voxel-by-voxel
  - ▷ 3dTsmooth:
    Smooth voxel time series

- ▷ 3dTqual, 3dToutcount:
  Examine voxel time series for statistical 'outliers'
- ▷ 3dTcat:
  Shift voxel time series to a common temporal origin
- ▷ 3dTstat:
  Basic statistics on voxel time series
- ▷ 3dvolreg:
  Volume registration to suppress motion artifacts, and to align same-subject data from different scanning sessions

- `Multi-dataset statistical operations`:
  - ▷ 3dttest:
    Voxel-by-voxel t-tests
  - ▷ 3dANOVA, 3dANOVA2, 3dANOVA3:
    1-, 2-, and 3-way voxel-by-voxel ANOVAs, including random effects and nested models
  - ▷ 3dFriedman, 3dKruskalWallis, 3dMannWhitney, 3dWilcoxon:
    Voxel-by-voxel nonparametric statistical tests analogous to ANOVAs
  - ▷ 3dRegAna:
    General linear regression models and tests derived therefrom

- Miscellaneous operations on datasets:
  - ▷ 3dAnatNudge:
    Try to align high-resolution anatomical volume with low-resolution EPI volume
  - ▷ 3dClipLevel:
    Find the voxel value to threshold an EPI volume at so as to remove most of the non-brain tissue
  - ▷ 3dIntracranial:
    Strip the scalp and other non-brain tissue from a high-resolution T1-weighted anatomical volume
  - ▷ 3dMean:
    Compute the mean of a collection of datasets, voxel-by-voxel
  - ▷ 3dmaskdump, 3dmaskave, 3dROIstats:
    Extract values from datasets and write to ASCII files
  - ▷ 3dUndump:
    Take values from ASCII files and write into a dataset
  - ▷ 3dmerge:
    Lots of options to edit datasets and combine then in multifarious and nefarious ways
  - ▷ 3dZeropad, 3dZcutup, 3dZcat, 3dZregrid:
    Utilities to add/subtract/resample datasets in the slice (z) direction

- ▷ 3daxialize:
  Re-write a dataset in a new slice orientation
- ▷ 3dcalc:
  General purpose voxel-by-voxel dataset calculator
- ▷ 3dfractionize:
  Resample a binary mask dataset from one resolution to another
- ▷ 3drotate:
  Rotate a dataset to a new orientation in space
- ▷ 3dpc:
  Extract principal components from a collection of datasets
- ▷ 3dWinsor:
  Spatially filter a T1-weighted anatomical dataset to reduce noise and make the gray-white matter boundary a little more distinct
- ▷ 3dclust:
  Find clusters of activated voxels and print out statistics about them
- ▷ 3dExtrema:
  Find local extrema (maxima or minima) in a dataset—intended for functional activation maps