A Handi Guide to AFNI*

Our Motto: "We already suffered so you won't have to!"
Version: 14 FEB 2009

TABLE OF CONTENTS

To be continued...

Problems? Contact Tom Christensen: 621-8618

Tips on using this guide:

Things in *italics* show you the command line syntax

Things in **bold** are programs and examples of things you type in the command line

Things underlined are examples you need to replace with real info (e.g. replace *filename* with actual filename)

# = a number you need to supply

Using the MACs: login=neustu; password=neurolab!

MACs: Ctrl-scrollwheel magnifies the screen

MACs: open an X-window first, then open AFNI in a regular terminal window

UNIX Commands: http://www.u.arizona.edu/udocs/unix-cmds.html

cd *foldername* = take one step up in the directory tree to selected folder

cd .. = take one step down in directory tree

cd ../.. = take two steps down in directory tree

cp [file1 or *file*] [destination] = copy a file(s) to a different place

ctrl-c = get back to command prompt in X window

lcd = local dir

ls = list contents of directory

ls -l (letter 'l' not 'one') = list all info for each file directory

ls –r = list one screen of files

ls –rlt = reverse list time

more *filename* prints the contents of *filename* to the screen

mv *file1 file2* (e.g., **mv S1_fse+orig /struct/fse/S1_fse+orig**) = move *file1* in current directory to another directory, overwrite *file2* if it exists

pwd = present working directory

rm [file] = deletes file(s)

- Press the **Up Arrow** key: it gives you access to all your previous command lines
- to run a program or script, type **./** in front of the filename

AFNI tricks in terminal command line:

- if you get an error "Can't open dataset S1_voice_blur" try adding **+orig** to the filename
- if you get "permission Denied" type: **chmod ugo+x** *filename*
- **chmod –R 775*** (R = recursive; include all files)
- **chgrp –R fmri*** (change all group permissions to fmri)
- type **3dinfo** *filename* to get info on how a file was created
- **afni &**: by typing the "&" after "afni" you can keep the UNIX shell in the background

AFNI tricks once you've opened the program:

- AFNI sometimes refers to preprocessed functional (3d+time) datasets as "anatomical" files if no statistical analysis has yet been done on them
- the "graph" buttons only show data for the underlay
- hit "Rescan" and "This" if you add files to directory while running AFNI
- to see header info in AFNI: Define Datamode>Misc>ULay Info or OLay Info
- to see header info in command line use: **3dinfo** *filename*
- useful links: http://afni.nimh.nih.gov/afni/doc/howto
  http://afni.nimh.nih.gov/afni/doc/program_help/index.html

NB: BEFORE USING THE MACS!!!!!!!!!!!!!!

- First, take a deep breath...

- Check to see that AFNI runs. If you get 'command not found' NOTHING will work! You'll need to logout and login again!

-If you see the "*Pinwheel of Calamity*" spinning for a long time, logout and login again (or you might be better off just re-booting the machine - don't fight it, just do it!)

In fMRI, all X, Y, Z coordinates [right-left; anterior-posterior; inferior-superior] are expressed in a particular order and in millimeters. The two most common ones are RAI/DICOM or LPI/SPM. The only difference between the two is the sign of the first two (x,y) coordinates. In AFNI's main controller, these coordinates along with their order, are specified in the top left corner. The default is RAI/DICOM. Right Hemisphere, Anterior, & Inferior are all expressed as NEGATIVE numbers.

In AFNI you can set the coordinate order to your liking with the envrionment variable AFNI_ORIENT. Programs (except 'whereami') will report coordinates in the order specified by this environment variable, unless they are specifically instructed to do otherwise. At any rate, the new version of the programs dealing with coordinates, such as 'whereami', '3dclust' and '3dExtrema', have been modified to output a string specifying the coordinate order.

The program 'whereami' will output coordinates in the LPI/SPM order, regardless of the AFNI_ORIENT variable because the report is given in multiple coordinate systems such as Talairach, MNI, and MNI Anatomical. The latter two are almost always in LPI order in the literature so we decided to minimize confusion and report whereami's coordinate in LPI all the time.

Do you have any other useful tips? Write them in here:

I. EXPERIMENTAL DESIGN

Proper research design and experiment evaluation requires some serious time and planning. However, it pales in comparison to the amount of time, effort, and resources that are wasted on poorly designed experiments, which result in low statistical power and cloudy results. Fortunately, there's help! Check out the AFNI HOW-TO #3: CREATING OPTIMAL STIMULUS TIMING FILES FOR AN EVENT-RELATED FMRI EXPERIMENT. The AFNI script '@stim_analyze' has been reformatted in a script called '6stim_timing' so that it is easy to enter your own parameters, create wave files, and run a deconvolution program (3dDeconvolve - more on that later) that performs a statistical verification of your design. It is good to get into the practice of running this script BEFORE you scan your first subject!

1. Open and edit '6stim_timing' in a text editor. This script allows you to validate experimental designs using up to six different stimulus conditions (plus a *Null* condition).
2. refer to instructions in the script '6stim_timing'

II. STUDY ORGANIZATION

- **Structural MRI Scans**

Three types of structural MRI scans are collected as part of each study. The participant can listen to music during these scans and will be invited to bring their own CDs. The total time taken to complete these scans is approx. 15 minutes.

1. Three-plane scout: This scan is used to plan all of the other scans during the neuroimaging session.

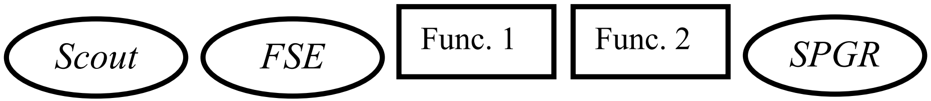
Fast-Spin Echo (FSE) Scan: A T1-weighted structural MRI scan that uses the same slice selection parameters as the functional MRI scans, in order to provide a high resolution image to overlay the functional data.

Volume 3D (SPGR). This is a high resolution structural scan that is collected AFTER the functional scans to allow structural and functional data to be easily transformed to a standard coordinate system (see "Transfer anatomical datasets to Talairach coordinates") for post-processing.

- **Functional MRI Scans**

During fMRI scans, participants will be asked to listen to and evaluate stimuli presented through headphones. Participants will press response buttons to indicate decisions concerning the stimuli. Up to three functional scans, each lasting between 15 and 30 minutes are typically collected.

- **Typical Scan Timeline:**



- **Organization of study directories**

Uniform organization across studies facilitates having multiple people contribute to different aspects of the study and keeps everyone on the same page. It also is invaluable if we have to go back to the study years later and try to find stuff for re-processing in a different way.

Create subdirectories on the local machine to receive data. These should include:

1. parent experiment directory (e.g. **/LocalImages/tom/stroop**)
2. subdirectories: **ANOVA**, for the group data; **programs** (with scripts for running steps in AFNI)
3. subdirectories for each subject (e.g. **S1**, **S2**, etc)

4. subdirectories within each subject directory for:

a. the raw datafiles used to make epan (functional) files (e.g. **run1_voice**, **run2_word**, etc.)

b. **struct**: for the structural files - **struct** contains subdirs for **fse**, **scout**, and **spgr** files, as well as a subdirectory **tal** where the *fse+orig* and *spgr+orig* files are copied in preparation for transforming anatomical files to Talairach space (p. 12). Then, the newly created *spgr+acpc* and *spgr+tlrc* files will also be put here.

c. other subdirectories you may want to make inside the subject directory:
3dD (for 3dDeconvolve files)

prep (for -anat, -reg –dt –cat versions of functional scans, as needed)
fim (for fim datasets, if used) (fim = *f*unctional *im*age)
clus (for clustered data, if used)
    waves (for wave files, if used)

Sequence & Directory Structure: EXAMPLE

```
/stroop
          /ANOVA
          /programs
          /S1
               /S1_voice.results
                         /stimuli
               /S1_word.results
                         /stimuli
               /struct
                         /fse
                         /scout
                         /spgr
                         /tal
               /run1_voice
               /run2_word

/stroop/S2
               /S2_voice.results   etc...
```

These directories are made by the script 'afni_proc.py' (see p. 21)

- **NB: If you're using the Macs: 1)** open an X11 terminal window **first.** Then open a **REGULAR** terminal window, and in this second window, type the path: **cd /images/tom** and make a new study directory (e.g., **mkdir stroop**) there

- **NB: On PCs**, you can ssh a directory template to any other machine that has the data. Be sure to set permissions so others can read your files!

- **Files are named as follows:** *Subject#_condition_subcondition_filetype* [NB: not all studies have subconditions] (e.g., **S1_select_noise+orig**) NB: use of UNDERSCORES, not DASHES, is HIGHLY recommended because DASHES are reserved for command-line options (e.g., **-prefix** or **–stim_times**). AFNI adds additional components to the file name (e.g., +orig, +acpc +tlrc) to tell itself what type of file it is. It will add them as needed, so you do not need to designate these in the filenames you create. However, if you are using a file as an input to an AFNI command line, you must provide the name through the +orig or +tlrc extension.

III. I MAGE EXTRACTION

NB: make sure you have the current versions of the preprocessing software.
NB: consider making a batch file to execute these commands without your babysitting them.

1. **COPY** P-files (functional) and e-files (structural) to local machine for safe storage and **BACKUP**, then **DELETE** from **mrisun** or **ultimate** server at UMC.
   a. from the local computer, cd to the directory you want the files to land in (e.g. /Stroop/S13)
   b. connect to **mrisun** or **ultimate**
      sftp plante@mrisun.radiology.arizona.edu (pwd = sw8rdf6sh)
      sftp christensen@ultimate.ahsc.arizona.edu (pwd = tc1111tc)
   c. cd to the subdirectory on **mrisun** or **ultimate** that contains the sliced data
   d. transfer the original P-files and e-files to safe storage
      mget P*
      mget e*
   e. once SAFELY backed up, delete P-files from **mrisun** or **ultimate**

2. Now, we want to transfer a COPY of the P-files from safe storage to the center (UNIX) server for unpacking
   a. from the local computer, cd to the directory you want to get the files from
   b. **sftp thomas@center (pwd = neurolab!)**
   c. cd to directory on **center** where you want the files to go
   d. transfer the P-files
      mput P*

- **Extraction of raw P-files**

This can be done on **mrisun** or **ultimate** (at UMC) or it can be done using **ssh telnet** from a lab computer

1. telnet to **center**:
   ssh thomas@center (pwd = neurolab!)

2. cd to the directory of the P-file to be extracted
   e.g., **cd /stroop/S32**

3. run **prepio** to extract the image files and then re-number them – **follow the on-screen instructions**:
   Syntax: prepio <Pfile> <nfs> <nas> <output name> (where nfs = #functional time pts; nas = # anatomical slices)
      e.g. **prepio P09785.7 214 26 run1 (takes about 10 min)**

4. While you're waiting, check that subdirectories on the local machine have been created to receive data. These should include a parent experiment directory named for the study (e.g., stroop), subdirectories for each subject (e.g., S1, S2, etc), and subdirectories within each subject directory for the experimental conditions (e.g., run1_voice, run2_word), the structural files (struct, with subdirectories for fse, spgr, & scout), and the analysis (combos).

5. **prepio** renames the image files (214 x 26 = 5,564 files) and they are now ready to put into **to3D** for further analysis (see later)

6. if you need to, **tar** zips all these files into a single .tgz file: tar zcvf *output filename.tgz input files.**
   e.g. **tar zcvf S1_voice.tgz S1_voice.***

7. move the .tgz file to correct folder (Stroop/S1/run1) on reckless or any local machine for further processing
8. unpack the .tgz file: tar zxvf *.tgz file*

e.g. **tar zxvf S1_voice.tgz**

Other useful sftp commands:
lcd=local directory
pwd=present working directory
rm –fr <directory>

9. **Repeat steps 3-8 for all functional P-files**

10. leave ssh: **exit**

11. **BACKUP!!**

- **Extraction of raw anatomical files**

1. open new X window
2. transfer the anatomical files to the lab computer; from local computer, cd to directory you want the files to go in:
   cd /mul/raid2/tom/stroop/S1/struct/fse

3. connect to **mrisun** or **ultimate**:
   sftp plante@mrisun.radiology.arizona.edu (pwd = sw8rdf6sh)
   sftp christensen@ultimate.ahsc.arizona.edu (pwd = tc1111tc)

4. cd to subdirectory on **ultimate** that contains sliced data
   cd /data4/plante/tom/S1

5. transfer the structural data files
   mget e3908s2* [scout=s1; fse=s2; spgr=s5]

6. exit sftp: **quit**
7. make sure you are in correct directory
   cd /mul/raid2/tom/stroop

8. copy **rename_anat** program to the fse, scout, and spgr subdirectories (for UNIX window ONLY: otherwise, click & drag!)
   cp rename_anat /mul/raid2/tom/stroop/S1/struct/fse (and /scout and /spgr)

9. Change directory to the subdirectory with the anat (e) files, e.g. **cd S1/struct/fse**
   type **./rename_anat** and at prompt: type (e.g.) **e3908s2**
   NB: This will create 26 fse slice files

Repeat for scout and spgr -- Remember scan number codes [scout=s1; fse=s2; spgr=s5]
NB: Program will create 54 scout files and 128 spgr files

IV. IMAGE PROCESSING

The following section describes the sequence of steps that goes into the analysis of fMRI data. Also note that for all the AFNI command line programs, a list and description of all the command line options can be viewed by typing the name of the command by itself or followed by "-help"

e.g: **3dvolreg** *or* **to3d –help**

In UNIX, command lines can be written as follows:

```
3dvolreg –prefix S1_attn_silence_reg -tshift 0  -1Dfile S1_attn_silence_reg.1D  –Fourier –base 8 -1Dfile
S1_attn_silence.1D  –rotparent S1_attn_noise_reg+orig  S1_attn_silence_anat+orig
```

or like this:

```
3dvolreg \
-prefix S1_attn_silence_reg \
-tshift 0 \
-1Dfile S1_attn_silence_reg.1D \
-Fourier -base 8 \
-1Dfile S1_attn_silence.1D \
-rotparent S1_attn_noise_reg+orig S1_attn_silence_anat+orig
```

and they mean **exactly** the same thing. (the '\' at the end of each line simply tells AFNI that what follows is still part of the same command line - note that there is no '\' at the very end, signifying the end of the command line). The command can be saved as a script in either format using a text editor (in UNIX or X11 use *pico* and save the script in the programs directory). Thereafter, it is an easy task to simply read the script file, edit the filenames, and then run the command again on another dataset by typing ***./filename***. This prevents having to re-write the entire script every time you use it.

STEP 1. Process Individual Subjects' Data (modified from AFNI HOW-TO #5, Part 1)

A. Create anatomical datasets with 'to3d -fse' and 'to3d –spgr'

When the dataset comes off the scanner it's in "k-space"~ k-space is a **HUGE** complex array of numbers whose Fourier transform is the MR image. Every value in k-space represents a wave that has frequency, phase, and amplitude information. This step does the Fourier transformation of the raw k-space files.

- **Build FSE datasets**

1. **cd /image directory/subject/struct/fse**
2. *to3d -fse -prefix output_filename input_files*
   (e.g., in terminal window type: **to3d -fse -prefix S1_fse e***)

A **to3d** window should open up (NB: it does not always pop up the first time. This is an unfortunate quirk in the program, but AFNI will nevertheless make xxx_fse+orig.HEAD and BRIK files! Run **to3d** again; you'll have to put in a new value for "Prefix for 3D Dataset file", then after Saving, immediately erase the first fse anatomy files.)

In pop-up window:

a. x orientation: right to left
b. y orientation: anterior to posterior
c. z orientation: inferior to superior

[NB: you must press "View Image" key to see image and confirm how it loads;
above x, y, z directions may not apply!]

d. view: original view
e. set FOV (from scan parameters; e.g. **240**)
f. irregular
g. voxel x and y dimensions will automatically calculate if you set the FOV

h. input z voxel size [from log book, e.g. **5** mm]
i. click 3 buttons on right to center ALL axes
j. click 'Save Dataset' / Quit (press twice) / Delete old 'fse' files

- **Build SPGR datasets**

1. **cd /image directory/subject/struct/spgr**
2. *to3d -spgr -prefix* ***output_ filename input_ files***
   (e.g., **to3d -spgr -prefix S1_spgr e***)

A **to3d** window should open up (NB: it does not always pop up the first time! Run it again; you'll have to put in a new value for "Prefix for 3D Dataset file"!)

a. x orientation: anterior-posterior
b. y orientation: superior-inferior
c. z orientation: left-right

[NB: you must press "View Image" key to see image and confirm how it loads;
above x, y, z directions may not apply!]

d. view: original view
e. set FOV (from scan parameters, e.g. **240**)
f. irregular
g. voxel x and y dimensions will automatically calculate if you set the FOV
h. input z voxel size [from log book, e.g. **1.5**]
i. click 3 buttons on right to center ALL axes
j. click 'Save Dataset' / Quit (press twice) / Delete old 'spgr' files

- **Copy fse and spgr files to struct/tal directory for next step**
  (e.g., **cp S1_fse* /Stroop/S1/struct/tal**)

B. Transform anatomical datasets to Talairach coordinates

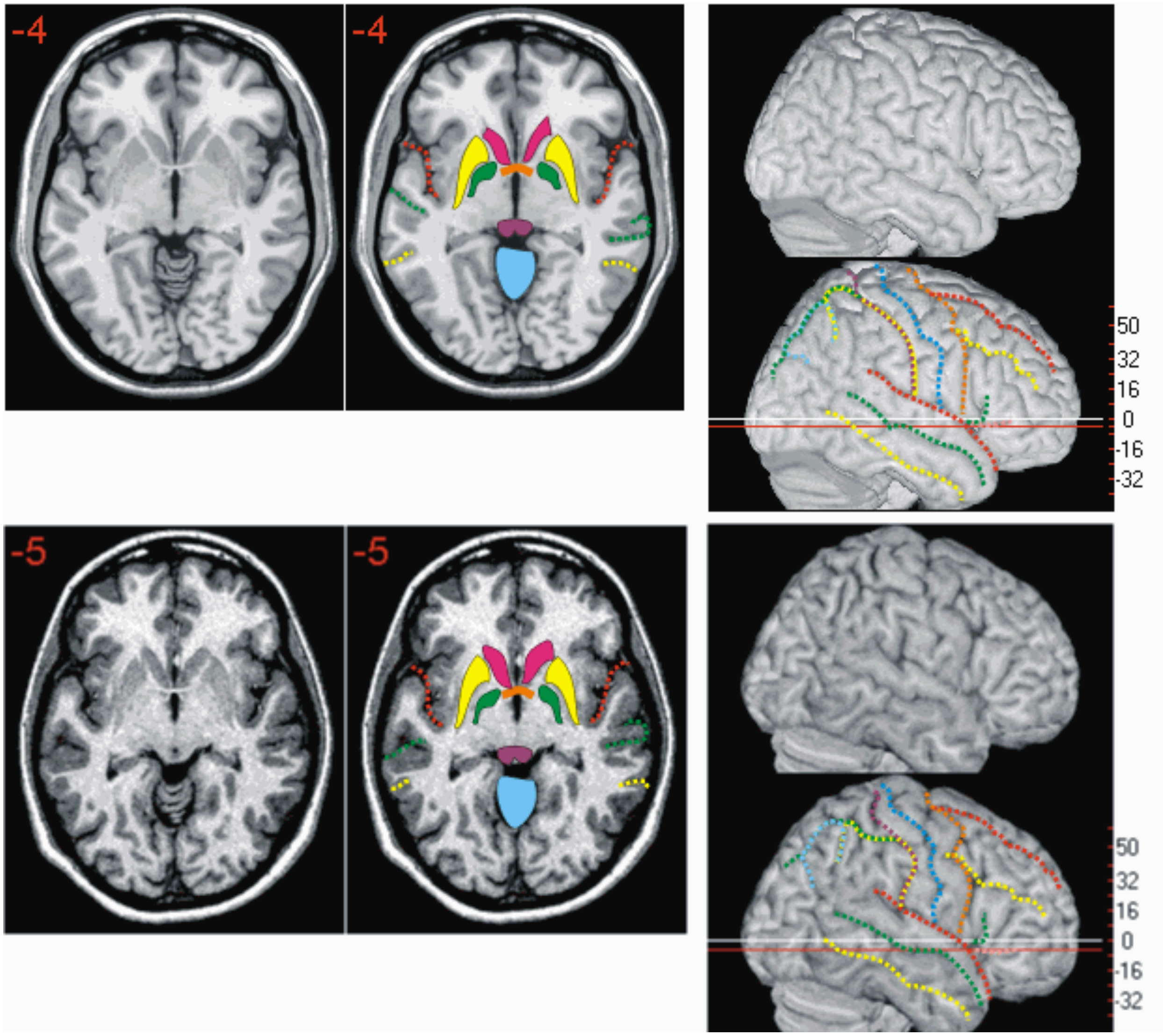
(note: some of the following text is taken from a Waisman lab how-to "Analysis of fMRI data with AFNI")

The functional data from each subject must be transformed to a standard space so that they can later be aligned to other brains for the group average analysis. This transformation involves two steps. First, we need to calculate the transformation from the individual subject's brain space to the standard (Talairach or MNI) space, which is based on the high-resolution anatomical scans for each subject. Later, we will apply the same spatial transform to our functional datasets, thus bringing them into the same standard space. To perform both of these steps, we can choose from two options:

(1) Manual (traditional) technique, or (2) Automated (newer) technique (skip to p. 16).

Nudge Dataset: In either case, the scan that comes after the functional data runs (spgr) must first be aligned with the scan that precedes the functional data runs (fse) because movement may have occurred over the course of the experiment. **It is important to do this so that the functional data overlay is in correct alignment with the underlying anatomical data.**

Some landmarks to look for:



- anterior commissure **(orange)**
- caudate nucleus **(magenta)**
- globus pallidus **(green)**
- putamen **(yellow)**
- quadrigeminal plate **(purple)**
- vermis **(ice blue)**
- sylvian fissure **(dotted red)**
- superior temproal sulcus **(dotted green)**
- inferior temporal sulcus **(dotted yellow)**

http://www.sph.sc.edu/comd/rorden/anatomy/na_ac.html

Aligning SPGR files with FSE files using AFNI plugin 'Nudge Dataset'

1. change directories to the directory with both fse+orig & spgr+orig files
   e.g., cd ../../struct/tal
2. open afni
3. display the fse image
   a. click on "switch underlay" (bottom right)
   b. select fse file (e.g. S1_fse)
   c. click on "Axial Image" (middle left) to display
4. display the spgr image
   a. click on "New" (bottom left) to open a new window
   b. click on "switch underlay" (bottom right)
   c. select spgr file (e.g. S1_spgr)
   d. click on "Axial Image" (middle left) to display

NB: if these two images are aligned, they will look like the same slice in the two windows and x,y,z coordinates will be almost identical in the two afni windows. If not, they need to be aligned with 'Nudge Dataset'

5. Get the x, y, z coordinates for both images:
   a. Unlock the cross-hairs: click 'Define Datamode' then 'Lock' then 'Clear All'
   b. Change slices until you find a point, such as the anterior commissure (the origin [0,0,0] of the Talairach coordinate system), that is small and appears in both images.
   c. Place the cursor over that point (change the cursor gap size to 1 for better accuracy of placement (top left "Gap" button)
   d. Compare other aspects of the image to make sure the anatomy is the same for the two images, adjust as needed
   e. For the fse image, write down the values associated with the x,y,&z coordinates, including R/L, A/P, and S/I information
   f. For the spgr image, write down the values associated with the x,y,&z coordinates, including R/L, A/P, and S/I information

NB: these values represent distance along an x,y,z grid where 0 is the center of each axis

6. Nudge the **spgr** (last anatomical scan) to align with the **fse** (first anatomical scan)
   a. click 'Define Datamode' then 'Plugins' then 'Nudge Dataset'
   b. click 'Choose Dataset' then select your -spgr file and click 'Set'
   c. note row of 'Shifts': there are windows for '**S**uperior' , '**L**eft' and '**P**osterior'
   d. note that both files should be in 'RAI' orientation
   e. The difference between the coordinates (see 5e & f) is the distance the spgr must be moved in each direction to align it with the fse image. For each of the coordinates (x, y, & z) individually:
      1. If the values are of like signs (- - or + +), subtract the smaller coordinate from the larger. This is the <u>distance</u> that the spgr must change to align.
      2. If the values are of different signs (-/+ or +/-), add the two coordinates. This is the <u>distance</u> that the spgr must change to align.
   f. now determine the <u>direction</u> (R/L, A/P, S/I) that the spgr must move to align with the fse. E.g., for x axis, if fse > spgr (e.g. 4.008 > 2.25), calculate the difference (1.758) and insert that number into the 'L' window. In this example, the spgr is not far enough to the left hemisphere (that means the <u>image</u> needs to be shifted to the RIGHT; remember left=right) so the value is POSITIVE; if the spgr were too far to the left (fse < spgr), you would enter a NEGATIVE number to nudge the image to the right (the image would be shifted LEFT).

   g. click 'Nudge' you should see the <u>image</u> shift under the cross-hairs

h. now move on to the y-axis and z-axis **NB: Every time you nudge, you move the axes by whatever values are set in the three windows. Therefore, if you want to nudge one axis at a time and look at your results, you MUST reset the other two windows to 0, or nudge will execute the same movement again!**

i. click 'Do All' when done to save the new axis info to the spgr header file

j. exit afni, go home and have a beer

Put SPGRs into Talairach space (manual technique)

To perform group studies using functional imaging data, the images from individual subjects must be transformed into a common coordinate space before you can perform voxel-wise comparisons across subjects. The two most widely used spaces in the neuroscience community are the Talairach space and the Montreal Neurological Institute (MNI) space. The Talairach coordinate system is the standard reference for reporting brain locations in scientific publications.

NB: SPGRs must first be 'nudged' and aligned with FSE images to do this
NB: You may want to use 3ddup to duplicate file before altering

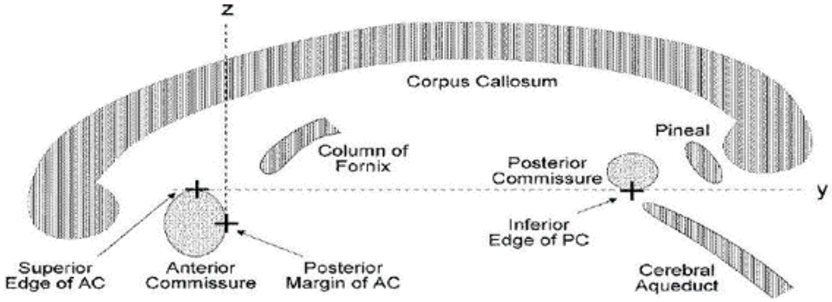
1. Transform original 3d image into AC-PC view (see AFNI screenshot on next page)
   a. open afni
   b. click 'Switch Underlay' to open your spgr file
   c. change Xhairs Gap to 1
   d. bring up ALL views
   e. push 'See Markers'
   f. push 'Define Markers' (new panel will open)
   g. on new panel:
      1. push "allow edits" button
      2. click first marker (AC superior edge)
      3. on axial image, place cursor on upper edge of AC and left-click (I find clicking on the 4 arrow buttons in the lower right-hand corner easiest)
      4. press "Set" button - a box with cross-hairs will mark your selection
      5. repeat steps 2-4 for subsequent markers; use all views to help guide your selections; to set mid-sag points look at a relatively superior axial slice
      6. When all markers are set, push "quality" button
         1. if error message appears, adjust markers to solve (anglular deviation must be < 2 degrees)
         2. if no error message, push "Transform Data" button. This creates the *xxxx*_spgr+acpc.HEAD file and puts it in the same directory as the *xxxx*_spgr+orig.HEAD file

AC-PC Placement on spgr:
1) find midline ~ sag65
2) find AC in axial slice
3) PC will be about 5 slices below AC

Define your markers using sagittal view:

NB:
Commissures are not really this large!

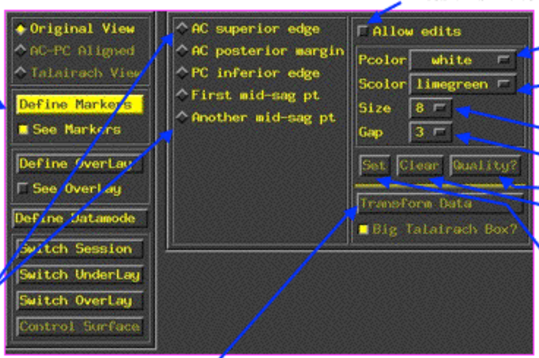
-21-



Press this IN to create or change markers

Click Define Markers to open the "markers" panel

Select which marker you are editing



Color of "primary" (selected) marker

Color of "secondary" (not selected) markers

Size of markers (pixels)

Size of gap in markers

Clear (unset) primary marker

Set primary marker to current focus location

Carry out transformation to +acpc coordinates

Perform "quality" check on markers (after all 5 are set)

2. Tranform AC-PC view into Talairach space
    a. afni
    b. select "AC-PC Aligned" button
    c. push "See Markers"
    d. push "Define Markers" (new panel will open)
        1. push "Allow edits" button
    2. click first marker (Most anterior point - start with sagittal and finish with coronal view)
        3. click image on corresponding location - press "Set"
        4. repeat for subsequent markers
        - Most posterior point - start with sagittal, finish with coronal view
        - Most superior point - use axial view, compare with other views;

NB: that's most superior point in the BRAIN not the meninges!

- Most inferior point - start with coronal view, finish with sagittal
- Most left point; NB: LEFT = RIGHT and vice versa! So to find the

leftmost point, use the coronal view, click on the far RIGHT, then switch to sagittal view and click until brain disappears; likewise

- for most right point

5. When all markers are set, push "quality" button

- if error message appears, adjust markers to solve
- if no error message, push "Transform Data" button. This creates the *xxxx*_spgr+tlrc.HEAD file and puts it in the same directory as the *xxxx*_spgr+orig.HEAD file

- **Repeat for each subject.**

(2) Automated (new) technique

NB: SPGRs must first be 'nudged' and aligned with FSE images before proceeding!

In this newer technique, AFNI can automatically calculate the transform of each subject's anatomy to a template anatomy of your choosing. A number of templates are included in the AFNI package, including:
TT_N27+tlrc: This is the "Colin" brain – one subject scanned 27 times and averaged
TT_icbm452+tlrc: This is the International Consortium for Brain Mapping template, created by averaging 452 normal brains
TT_avg152T1+tlrc: The Montreal Neurological Institute template, created by averaging 152 normal brains
It might also be desirable to use a different template, for example, a custom-made template if examining a childhood population.

IMPORTANT NOTE: Please consult **@auto_tlrc –help** for a description of the templates and differences between automatic and manual transformation techniques. It is very important that you understand the differences in coordinates that arise from using different templates, and that you use only a single template for your study. It is also worth noting that the manual technique might give better alignment across subjects for structures close to the anterior and posterior commissures, such as the amygdala, striatum, etc.

For automatic transformation, the first step is to calculate and apply the transform from individual subject space to template space based on the subject's high resolution anatomical scan:

e.g., **@auto_tlrc -base TT_icbm452+tlrc –suffix_icbm452 -input S1_spgr+orig**

This step should strip the skull off the anatomical image, and then calculate the best spatial transform to warp the individual's anatomical data into the same space as the template image (in this case the icbm452 template). A few things can go wrong at this stage, so you should open up AFNI and display your transformed anatomical as an OverLay on the icbm452 template to make sure the alignment looks okay. Also, look at the transformed anatomical as an Underlay to make sure that the skull stripping has done a decent job (e.g. hasn't cut off large chunks of brain). If the stripping wasn't good, it will be necessary to manually skull strip your anatomical, either using AFNI programs or perhaps the BET program from the FSL package.

Assuming all went well with transforming the anatomical, you can now apply the same transformation to your functional data (see "**Build functional datasets"** below). This step is performed in the AFNI GUI using "Warp on Demand" (see below)

C. Build functional datasets (spirals, 3D+time) with 'to3d -epan'

NB: verify that the parameters like field of view (FOV), voxel dimensions, # of timepoints, # of slices, scan duration (TR), are correct from the log book.
Extracted datafiles (several thousand *input files*) must be renamed first (see p. 8).

- cd */image directory/subject/session*
  (e.g., **cd /mul/disk11/S1/run1_voice**)

- make a 3D+time file from the functional (echo planar or 'epan') data. Type:
  to3d -epan -prefix *name* -save_outliers *name*.1D -time:tz *#timepoints #slices TRduration @offsets.*1D *input files**
  NB: if there are too many slices, substitute *input files*.* for *input files**

NB: Slices are not always acquired during the entire TR. In our sparse scan protocol, for example, slice acquisition is complete after 2.8 s, so AFNI needs to know this. *Offsets filename* is a *.1D file that lists the time offsets for each slice. On the 3.0T scanner, slices are acquired in the -z direction (bottom to top). Thus, if you're using a sparse scan protocol, collecting 26 slices in 2800 ms out of the total TR of 3500, then your offset file (offsets.1D) will look like this:

```
2700 2592 2484 2376 2268 2160 2052 1944 1836 1728 1620 1512 1404 1296 1188 1080 972 864 756 648
540 432 324 216 108 0
```

NB: Copy this file 'offsets.1D' into the same directory with all the 1000's of raw functional files (run1_voice, etc.)

EXAMPLE:
to3d -epan -prefix S1_voice_epan -save_outliers S1_voice_out.1D -time:tz 214 26 3500 @offsets.1D S1_voice.* [program reads all several thousand data files]

NB: If you are not running a sparse scan protocol, replace **@offsets.1D** with **seqminus**
Seqminus tells AFNI that the sequence of acquiring slices on the GE 3.0T goes from bottom-to-top.

NB: IGNORE DICOMM ERRORS if you see them
NB: use -ncolor 16 option if errors arise

- **Wait while program checks for outliers:**

NB: Subjects move in the scanner and we have to remove these data points from further analysis if the brain displacement is too large. The following steps show how AFNI determines the time points at which movement or some other type of artifact occurred: these artifacts are called **outliers**.

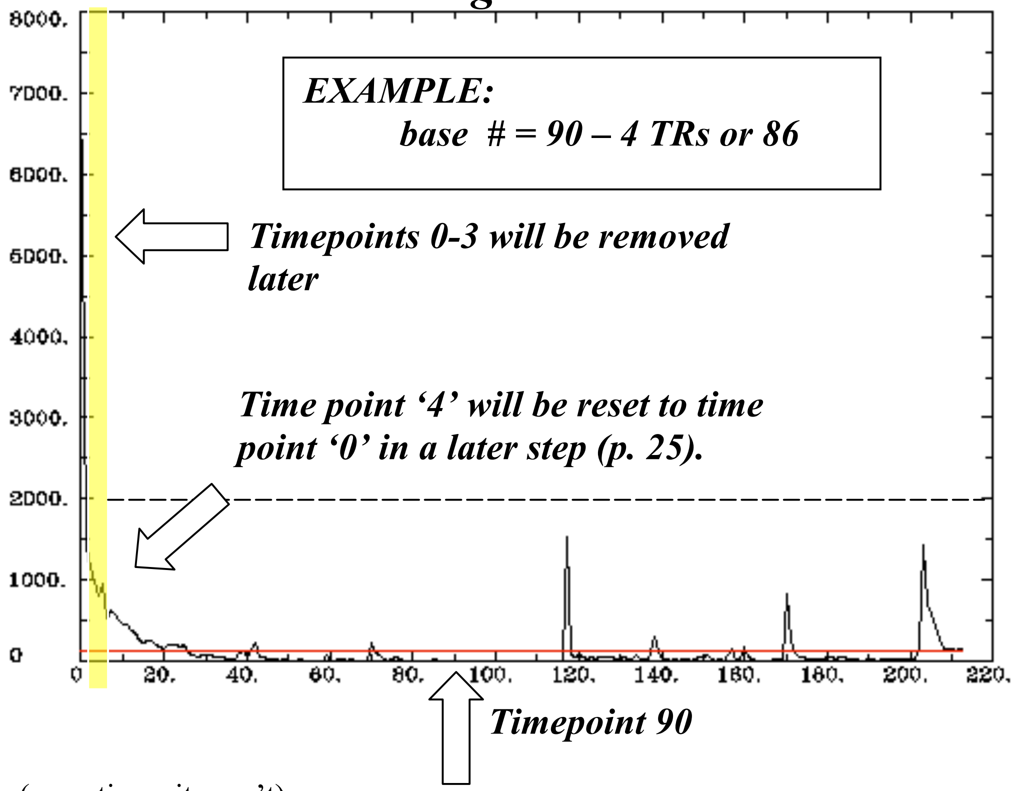
A pop-up window should open. Check:
a. x orientation: right to left
b. y orientation: anterior to posterior
c. z orientation: inferior to superior
NB: you must press "View Images" key to see image and confirm x, y, z directions!
d. view: Original view
e. set FOV (from scan parameters, e.g. **240**)
f. irregular
g. voxel x and y dimensions will automatically calculate when you set the FOV
h. input z voxel size [from log book, e.g. **5**]
i. click 3 buttons on right to center ALL axes

j. click 'Save Dataset'
k. Quit (click twice) **NB: these axis values will be the same as those used for the fse files**

- **Save outlier information and find the base # for movement correction:**
  if a graph pops up:

a. click *save* button
b. type in S#*_run_outliers.ps* (e.g. S1_voice_outliers.ps)
c. save
d. Look at the 'outliers.ps' file below. It shows 214 time points (TRs or scan periods). Later we will be removing the first 4 TRs because it takes the scanner some time to come up to equilibrium after it is started. We generally remove the first 4 TRs.
e. To find the **base #**: Find the time point at which the signal first falls to a near zero level. In this case (see graph below), the base # would be 90-4 or 86. Save this number for running the pre-processing script (see below, p. 20).

Finding the base #



EXAMPLE:
base # = 90 – 4 TRs or 86

Timepoints 0-3 will be removed later

Time point '4' will be reset to time point '0' in a later step (p. 25).

Timepoint 90

If no graph pops up (sometimes it won't):

a. type in the command line: **1dplot –one** ***outlier_file.*1D**
(e.g., **1dplot –one S1_attn_outliers.1D**)
b. save image as a *outlier_file*.ps file
(e.g., **S1_attn_outliers.ps**)

- Save text file:

a. in command window, highlight all the numbers appearing under the text 'Significant outliers detected in these sub-bricks:' These are the time points flagged as outliers.

b. click on 'edit' at top of screen; click 'Copy'; open TextEdit (3rd icon from left at bottom); right-click and select 'paste'; click 'File' at top of screen; click 'Save' using the filename *S1_xxxxxx_out.txt* and save in same directory as functional files; can also highlight and use
CTR-C.

c. Look at the 'outliers.txt' file. Find the most stable time point collected closest in time to the FSE anatomical scan – this is a point not identified as an outlier after running 'to3d' (compare to the outliers graph). This time point is used as the '**base #**' in '3dvolreg' below (see p. 20).

- **Determine whether outliers are sufficiently few & small to process data further. Values under 2000 are OK (see dotted line in graph above). The large values at the beginning of each scan are OK because they will be removed from the analysis.**

****DO NOT PROCEED IF THERE ARE TOO MANY OUTLIERS - THE SCAN PROBABLY CAN'T BE USED****

Optional: Concatenate functional (3D+time) runs with AFNI '3dTcat'

Concatenating files from a Single Subject's Session from the command line:

3dTcat -prefix *outfilename_cat input_file1 input_file2 etc...*
(e.g., **3dTcat -prefix S1_both_cat S1_voice_norm+orig S1_word_norm+orig**)

To make the optional *.1D concatenation file (only if files are concatenated); use with '-concat' option in 3dDeconvolve

- Using a text editor
  a. type 0 to indicate that the data in the first concatenated file start at timepoint 0
  b. on the next line, type the time point number where the next concatenated file starts.
  c. on subsequent lines (one line per number) add where additional concatenated files start.
  d. save as **concat.1D**
  e. copy to each subject directory where it will be needed

D. Pre-process functional data with the script 'afni_proc.py'

http://afni.nimh.nih.gov/pub/dist/doc/program_help/afni_proc.py.html

AFNI provides *afni_proc.py* as a tool that creates a basic Unix analysis script. This script is then used to run the entire sequence of steps (10 sections) that must be performed in the analysis of the functional data. We have created such scripts for various projects and placed them in the root subject directories (S1, S2, etc.). These can easily be edited in a text window for your own needs. Ensure that the following variables are correct before running the script:

- ${subj} – subject # (S1, S2, etc.)
- ${run} – condition (voice, word, etc.)
- ${base} – base# for movement correction
- Things that must be edited are shown in purple. Make sure the correct names of the timing files

and filenames with the correct subject, condition & run are typed into sections **1, 8 & 9** of the script.

- Save the edited script.

Running a script:

- Open a regular terminal window; make sure you are in the correct directory.
- Enter the following in the command line:

```
./scriptname[space]Subject#[space]Run[space]Base#
```

e.g., **./StroopScript.txt S21 voice 86**.

This sequence lets you set the subject number, the experiment condition (run), and the base# all from the command line, so you won't have to edit the script to change these parameters for every subject and run. If you have a large number of subjects and multiple runs, this can cause a real editing headache - this is where all your previous hard work pays off (extra thanks to Kyle Almryde for much time spent building and troubleshooting these scripts!).

EXAMPLE SCRIPT:

```
#!/usr/bin/env tcsh
echo "auto-generated by afni_proc.py, Mon Jun 30 11:23:00 2008"
echo "(version 1.28, Jan 28, 2008)"

# execute via : tcsh -x S#-run-script |& tee output.S#-run-script

# 1-------------------------------------------------
# script setup

# the user may specify a single subject to run with

set images_home = /LocalImages

if ( $#argv > 0 ) then
   set subj = $argv[1]
else
#    set subj
   set subj = $1
endif
```

```
# set run
set run = $2

# set Base Number
set base = $3

# assign output directory name
set output_dir = ${subj}_${run}_results

# verify that the results directory does not yet exist
if ( -d $output_dir ) then
   echo output dir "${subj}_${run}_results" already exists

   exit

endif

# create results directory
mkdir $output_dir

# copy structural fse into folder to allow for resampling

cp *fse* $images_home/Stroop/$subj/${subj}_${run}_results
```

NB: Make sure these filenames match the correct subject, run and condition

```
# create stimuli directory, and copy stim files
mkdir $output_dir/stimuli
cp S22_voice_mvfw_res.1D.txt S22_voice_fvmw_res.1D.txt S22_voice_mvmw_res.1D.txt \
   S22_voice_mvnw_res.1D.txt S22_voice_fvnw_res.1D.txt S22_voice_fvfw_res.1D.txt \
   S22_voice_null_res.1D.txt $output_dir/stimuli

# 2----------------------------------------------------
# apply 3dTcat to copy input dsets to results dir, while
# removing the first 4 TRs
3dTcat -prefix $output_dir/${subj}_${run}_tcat ${subj}_${run}_epan+orig'[4..$]'

# and enter the results directory
cd $output_dir

# 3----------------------------------------------------
# run 3dToutcount and 3dTshift for BRIK and HEAD File

3dToutcount -automask ${subj}_${run}_tcat+orig > outcount_${subj}_${run}_1D

3dDespike -prefix ${subj}_${run}_despike -ssave ${subj}_${run}_spike ${subj}_${run}_tcat+orig
3dTshift -tzero 0 -rlt+ -quintic -prefix ${subj}_${run}_tshift                          \
          ${subj}_${run}_despike+orig

3dToutcount -automask ${subj}_${run}_tshift+orig > outcount2_${subj}_${run}_1D

# 4----------------------------------------------------
# align each timepoint to the base volume
```

```
   3dvolreg -verbose -zpad 1 -base ${subj}_${run}_tshift+orig"[${base}]"          \
         -1Dfile dfile_${subj}_${run}_1D -prefix ${subj}_${run}_volreg             \
         -cubic                                                                    \
         ${subj}_${run}_tshift+orig


# make a single file of registration params
cat dfile_${subj}_${run}_1D > ${subj}_${run}_1D

# 5---------------------------------------------------
# blur each volume

   3dmerge -1blur_fwhm 8.0 -doall -prefix ${subj}_${run}_blur          \
        ${subj}_${run}_volreg+orig


# 6---------------------------------------------------
# create 'full_mask' dataset (union mask)

   3dAutomask -prefix ${subj}_${run}_automask ${subj}_${run}_blur+orig

# 7---------------------------------------------------

# scale each voxel time series to have a mean of 100
# (subject to maximum value of 200)

   3dTstat -prefix ${subj}_${run}_mean ${subj}_${run}_blur+orig
   3dcalc -a ${subj}_${run}_blur+orig -b ${subj}_${run}_mean+orig  \
        -c ${subj}_${run}_automask+orig                            \
        -expr 'c * min(200, a/b*100)'                                      \
        -prefix ${subj}_${run}_norm

# 8---------------------------------------------------
# run the regression analysis
3dDeconvolve -input ${subj}_${run}_norm+orig.HEAD                  \
   -polort 5                                                       \
   -mask ${subj}_${run}_automask+orig                                      \
   -num_stimts 7                                                           \
  -stim_times 1 stimuli/S22_voice_mvfw_res.1D.txt 'GAM'            \
  -stim_label 1 S22_voice_mvfw_res                                 \
  -stim_times 2 stimuli/S22_voice_fvmw_res.1D.txt 'GAM'            \
  -stim_label 2 S22_voice_fvmw_res                                 \
  -stim_times 3 stimuli/S22_voice_mvmw_res.1D.txt 'GAM'            \
  -stim_label 3 S22_voice_mvmw_res                                 \
  -stim_times 4 stimuli/S22_voice_mvnw_res.1D.txt 'GAM'            \
  -stim_label 4 S22_voice_mvnw_res                                 \
  -stim_times 5 stimuli/S22_voice_fvnw_res.1D.txt 'GAM'            \
  -stim_label 5 S22_voice_fvnw_res                                 \
  -stim_times 6 stimuli/S22_voice_fvfw_res.1D.txt 'GAM'            \
  -stim_label 6 S22_voice_fvfw_res                                 \
  -stim_times 7 stimuli/S22_voice_null_res.1D.txt 'GAM'            \
```

NB: Make sure these filenames match the correct subject, run and condition

```
  -stim_label 7 S22_voice_null_res                                    \
#   -stim_file 8 dfile.$run.1D'[0]' -stim_base 8 -stim_label 8 roll
#   -stim_file 9 dfile.$run.1D'[1]' -stim_base 9 -stim_label 9 pitch        \
#   -stim_file 10 dfile.$run.1D'[2]' -stim_base 10 -stim_label 10 yaw       \
#  -stim_file 11 dfile.$run.1D'[3]' -stim_base 11 -stim_label 11 dS         \
#  -stim_file 12 dfile.$run.1D'[4]' -stim_base 12 -stim_label 12 dL         \
#  -stim_file 13 dfile.$run.1D'[5]' -stim_base 13 -stim_label 13 dP         \
```

NB: These lines are commented out, but you may want to use these movement parameters in a GLM analysis at some point.

```
  -iresp 1 S22_voice_mvfw_norm.irf                                        \
  -iresp 2 S22_voice_fvmw_norm.irf                                        \
  -iresp 3 S22_voice_mvmw_norm.irf                                        \
  -iresp 4 S22_voice_mvnw_norm.irf                                        \
  -iresp 5 S22_voice_fvnw_norm.irf
  -iresp 6 S22_voice_fvfw_norm.irf                                    \
  -iresp 7 S22_voice_null_norm.irf                                    \
```

NB: Make sure these filenames match the correct subject, run and condition

```
  -num_glt 2                                                          \
  -glt_label 1 mvfw_vs_fvfw                                           \
  -gltsym 'SYM: +S22_voice_mvfw_res -S22_voice_fvfw_res'                  \
  -glt_label 2 fvmw_vs_mvmw                                           \
  -gltsym 'SYM: +S22_voice_fvmw_res -S22_voice_mvmw_res'              \
  -fout -tout -x1D X.xmat.1D -xjpeg X.jpg                                 \
  -fitts ${subj}_${run}_fitts
  -bucket ${subj}_${run}_stats
```

NB: Make sure these filenames match the correct subject, run and condition

```
# create an all_runs dataset to match the fitts, errts, etc.
3dTcat -prefix ${subj}_${run}_all_runs ${subj}_${run}_norm+orig.HEAD


# create ideal wave files for each stim run
1dcat X.xmat.1D'[6]' > ideal_mvfw-voice-res.1D
1dcat X.xmat.1D'[7]' > ideal_fvmw-voice-res.1D
1dcat X.xmat.1D'[8]' > ideal_mvmw-voice-res.1D
1dcat X.xmat.1D'[9]' > ideal_mvnw-voice-res.1D
1dcat X.xmat.1D'[10]' > ideal_fvnw-voice-res.1D
1dcat X.xmat.1D'[11]' > ideal_fvfw-voice-res.1D
1dcat X.xmat.1D'[12]' > ideal_null-voice-res.1D
```

NB: Make sure these filenames match the correct run and condition

```
# 9-----------------------------------------------------

# remove temporary rm.* files
# rm -f rm.*

# return to parent directory
cd ..

cd ${subj}_${run}_results

3dbucket -prefix ${subj}_${run}_mvfw-tstat-script -fbuc ${subj}_${run}_stats+orig'[23]'

3dbucket -prefix ${subj}_${run}_fvmw-tstat-script -fbuc ${subj}_${run}_stats+orig'[26]'

3dbucket -prefix ${subj}_${run}_mvmw-tstat-script -fbuc ${subj}_${run}_stats+orig'[29]'
```

NB: Make sure these filenames match the correct condition

```
3dbucket -prefix ${subj}_${run}_mvnw-tstat-script -fbuc ${subj}_${run}_stats+orig'[32]'

3dbucket -prefix ${subj}_${run}_fvnw-tstat-script -fbuc ${subj}_${run}_stats+orig'[35]'

3dbucket -prefix ${subj}_${run}_fvfw-tstat-script -fbuc ${subj}_${run}_stats+orig'[38]'

mv *tstat* $images_home/stroop/AnovaData/SingleSubjectTSTAT

# return to parent directory
cd ..

cp *spgr* $images_home/stroop/AnovaData/SingleSubjectTSTAT

end
```

Files needed to run the Script:

Before you can run a script you'll need to have the following files in the subject (S#) root directory:

- **one or more 'xxxx_epan+orig' (3D+time) functional datasets (see above)**
- **'xxxx.txt' timing files - one for each stimulus category (see below)**

E. Make timing files:

Each timing file is a single column of numbers corresponding to the onset times of every stimulus in a category (S#_stim_time.txt), or the response times (S#_stim_res.txt).

EXAMPLE for the first run:

			Corrected Onset Times		**Corrected Response Times (sec)**
Trial	SoundFile	ONSET	sub14.0s	SlideTarget.RT	
1	null.wav	14.0	0.0	0.0	0.0
2	null.wav	16.8	2.8	0.0	2.8
3	mudF8N3.wav	20.3	6.3	2086	8.39
4	westF6N5.wav	23.8	9.8	1004	10.8
5	moustacheM7.wav	27.3	13.3	988	14.29

1) The first column of numbers shows the ACTUAL stimulus onset times (time from when E-Prime is started)
2) The next column (Corrected Onset Times) shows 14 sec (TR=3.5s x 4) subtracted
3) To build the response-time files for each stimulus category, take that number and add the response time from E-Prime column "Slide Target.RT" (see E-prime)
4) e.g., the third stimulus occurred at 6.3 sec after subtracting the first 3 TRs; add this to the response time (2.09 **sec**; you must convert ms to **sec**); this results in a CORRECTED response time of **8.39 sec**; this is the value that would go into the **'xxx_stim_res.txt'** file for this stimulus (FemaleVoice + NeutralWord).
5) repeat this for every stimulus in every stimulus category
6) repeat for all additional runs

F. Summary of steps performed by the script (numbered as in the script):

(1) Commands for Script Setup

(2) Remove first 4 TRs. This is necessary because it takes the scanner some time to come up to equilibrium after it is started. We generally remove the first 4 TRs. What follows is 2 lines from the script: the first is the message line (like all message lines, it begins with a '#'), followed by the line of code that performs this step:

```
    # removing the first 4 TRs
    3dTcat -prefix $output_dir/${subj}_${run}_tcat ${subj}_${run}_epan+orig'[4..$]'
```

**This # will remain 4 in all scans.

(3) Remove outliers (de-spike the data) and plot outliers before de-spiking.

```
3dToutcount -automask ${subj}_${run}_tcat+orig > precount_${subj}_${run}.1D

3dDespike -prefix ${subj}_${run}_despike -ssave ${subj}_${run}_spike        \
${subj}_${run}_tcat+orig
```

Time-shift slices (-tzero 0), remove slow shift in baseline (-rlt+), and re-plot outliers.

```
3dTshift -tzero 0 -rlt+ -quintic -prefix ${subj}_${run}_tshift              \
${subj}_${run}_despike+orig

3dToutcount -automask ${subj}_${run}_tshift+orig > postcount_${subj}_${run}.1D
```

(4) Register functional images (perform movement correction).

**NB: this base number changes for every run.

```
# align each timepoint to the base volume
3dvolreg -verbose -zpad 1 -base ${subj}_${run}_tshift+orig"[${base}]"             \
            -1Dfile dfile_${subj}_${run}.1D -prefix ${subj}_${run}_volreg       \
            -cubic ${subj}_${run}_tshift+orig
```

(5) Smooth (blur) each 3D+time dataset with '3dmerge' (all runs)

**fwhm=full-width half maximum and refers to voxel radius used for Gaussian smoothing. Generally, this number will scale according to voxel dimensions.

```
    # blur each volume

3dmerge -1blur_fwhm 8.0 -doall -prefix ${subj}_${run}_blur                  \
    ${subj}_${run}_volreg+orig
```

(6) Create a brain mask with 3dAutomask

A 'mask' is used to tell AFNI which voxels in the dataset to analyze and which to ignore. Obviously, we do not want to spend any time or computational resources analyzing voxels that have no information in them, so these are zeroed out using a mask. In general, making a perfect brain mask is very tricky. Essentially, what you want to achieve is a binary version of the functional data, where all voxels that have activation are set to a value=1 and all other voxels=0. AFNI then proceeds to analyze only those voxels with a value=1. This line of code makes a mask using 3dAutomask. It's a good idea to check it to see what it looks like and to edit it if necessary (see below).

```
# create 'full_mask' dataset (union mask)
   3dAutomask –prefix ${subj}_${run}_automask ${subj}_${run}_blur+orig
```

(7) Normalize each 3D+time dataset (Calculating % change in signal)

```
    # scale each voxel time series to have a mean of 100
    # (subject to maximum value of 200)
    3dTstat -prefix ${subj}_${run}_mean ${subj}_${run}_blur+orig 3dcalc -a
    ${subj}_${run}_blur+orig -b ${subj}_${run}_mean+orig           \
            -c ${subj}_${run}_automask+orig                              \
            -expr 'c * min(200, a/b*100)'                                \
            -prefix ${subj}_${run}_norm
```

(8) Run the regression analysis with 3dDeconvolve. You will create the following files using 3dDeconvolve:

a. impulse response file(s) (*.irf) for each condition that shows the waveform associated with each voxel

b. a bucket file that has multiple sub-briks with the processed data

(9) Extract one sub-brick from bucket file with '3dbucket' to use in Group ANOVA

For example:

```
3dbucket -prefix ${subj}_${run}_mvfw-tstat-script -fbuc              \
     ${subj}_${run}_stats+orig'[23]'
```

Etc... Then, all of the tstat files are moved to the `AnovaData/SingleSubjectTSTAT` **directory.**

When you've got everything ready...

RUN THE SCRIPT (REVIEW):

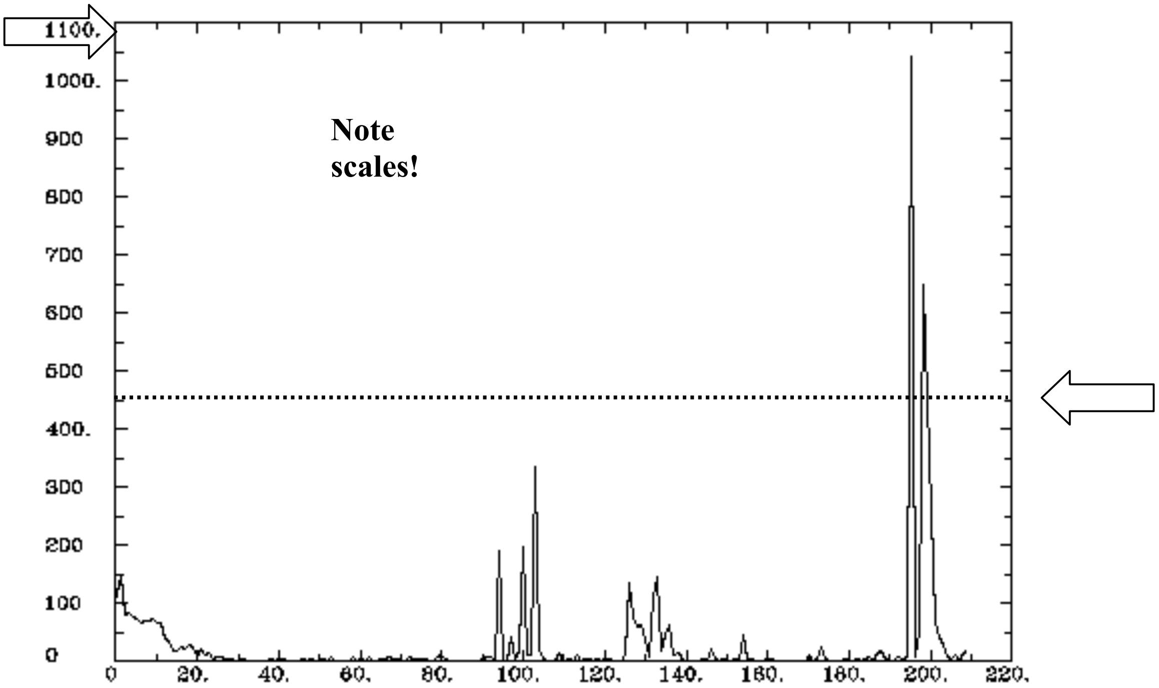
- Open a regular terminal window; make sure you are in the correct directory.
- Enter the following in the command line:
  ./scriptname[space]Subject#[space]Run[space]Base#
  e.g., **./StroopScript.txt S21 voice 86**.

This sequence lets you set the subject number, the experiment condition (run), and the base# all from the command line, so you won't have to edit the script to change these parameters for every subject and run.

G. AFTER THE SCRIPT RUNS:

VERY IMPORTANT!! : Check your data visually **before and after** despiking and registration (step 3)!
The script will plot the outlier files (`precount_${subj}_${run}.1D` and `postcount_${subj}_${run}.1D`) and the movement parameter file (`dfile_${subj}_${run}.1D)` in 3 separate windows. You can save these as **xxx.ps** files. Look at the `postcount` file: if you see any large spikes in these graphs (e.g. outlier intensities above 2000), these should be removed using the *-censor* option in 3dDeconvolve. **If there are too many spikes, you may not be able to use the data.**

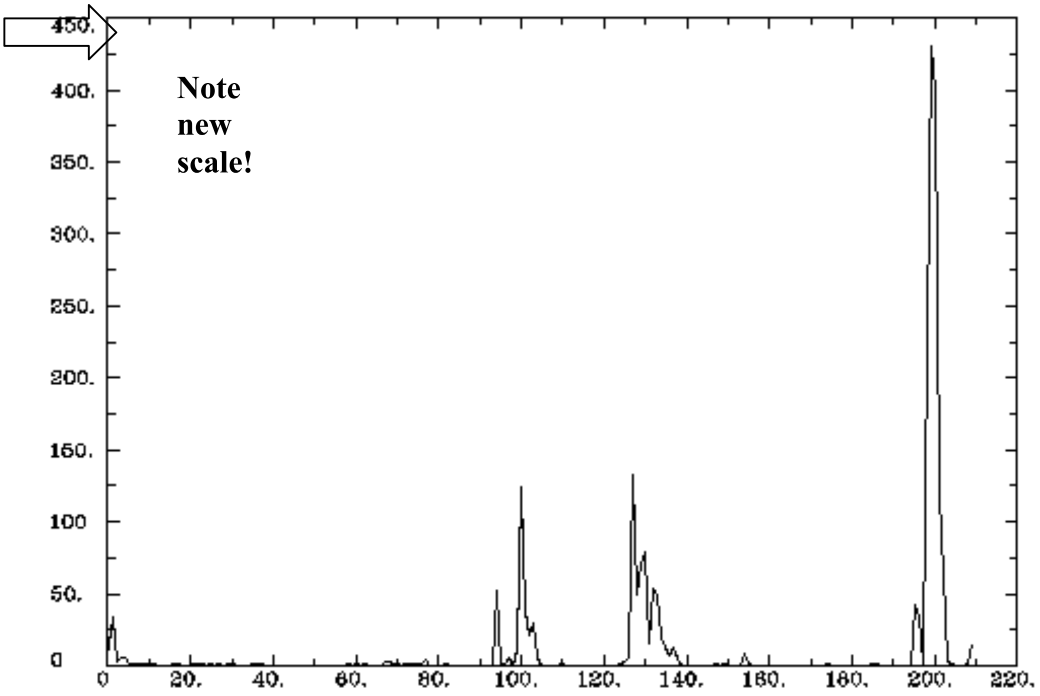
1. Typical output of `precount_${subj}_${run}.1D` **(before 3dDespike & 3dTshift)**



Note scales!

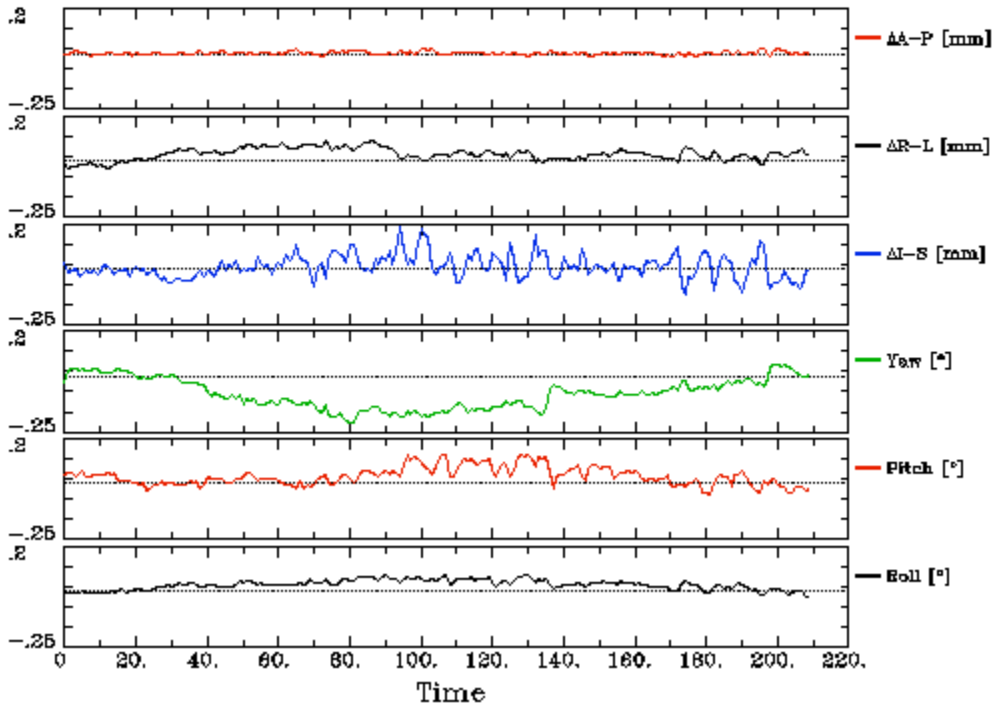
new maximum after removing outliers (see below!)

2. Typical output of `postcount_${subj}_${run}.1D` **(after 3dDespike & 3dTshift)**



Note new scale!

3. Typical output of `dfile_${subj}_${run}.1D` (movement parameters)



4. Use AFNI 'warp on demand' to resample functional datasets (*.irf*, *tstat* as needed) to same grid as our Talairached anatomical datasets.

In the example below, 3dANOVA2 uses *tstat* files;
Input: 'S10_voice_mvmw_tstat+orig' files
Output: 'S10_voice_mvmw_tstat+tlrc' files

1) Move all spgr files (BRIK & HEAD) from ..S#/struct/tal to same dir as tstat files (AnovaData/SingleSubjectTSTAT)
2) open AFNI
3) Click 'Talairach View'
4) click 'Define Datamode'
5) click 'Warp ULay on Demand' ; ULay resam mode = Li ; Resam (mm) = 1
6) click 'Warp OLay on Demand' ; Olay resam mode = Bk ; Stat resam mode = Bk ; Write 'Many'
7) select all of the '+tlrc' files you want to save and will need for the Group ANOVA (see below).

H. Notes on Other Steps in the Script:

Step (7): Notes on Normalizing Data

There has been some debate on the AFNI message boards regarding percent change and when it should be done. Some people prefer to compute the percent signal change after running '3dDeconvolve'. This method of normalizing the data involves dividing the IRF coefficients by the baseline. Other people, including Doug Ward (one of our message board statisticians), recommend doing it before '3dDeconvolve'. This method involves normalizing individual runs first, and then concatenating those runs. This normalized and concatenated file is then inputted into '3dDeconvolve'. For more information on this topic, search "Percent Signal Change" on the AFNI message board.

- **Why bother normalizing data in the first place?**

Normalization of the data becomes an important issue if you are interested in comparing your data across subjects. The main reason for normalizing FMRI data is because there is variability in the way subjects respond to a stimulus presentation. First, baselines or rest states may vary from subject to subject. Second, the level of activation in response to a stimulus event will also differ across subjects. In other words, the baseline ideal response function (IRF) and the stimulus IRF will most likely vary from subject to subject. The difference may be bigger for some subjects and smaller for others.

For example:

Subject 1:

Signal in hippocampus goes from 1000 (baseline) to 1050 (stimulus condition), resulting in a difference of 50 IRF units.

Subject 2:

Signal in hippocampus goes from 500 (baseline) to 525 (stimulus condition), resulting in a difference of 25 IRF units.

If we are simply comparing the absolute differences between the baseline and stimulus IRF's across subjects, one might conclude that Subject 1 showed twice as much activation in response to the stimulus condition than did Subject 2. Therefore, Subject 1 was affected significantly more by the stimulus event than Subject 2. However, this conclusion may be erroneous because we have failed to acknowledge that the baselines between our subjects differ. If an ANOVA were run on these difference scores, the change in baseline from subject to subject would artificially add variance to our analysis. More variance can lead to errors in interpretation of the data, and this is obviously a bad thing.

Therefore, we must control for these differences in baseline across subjects by somehow "normalizing" or standardizing the baseline so that a reliable comparison between subjects can be made. One way to do this is by calculating the percent change. That is, instead of using absolute difference scores (as the above example did), we can examine the percentage the IRF changes between a baseline/rest state and the presentation of an experiment stimulus. Does it increase (or decrease) by 1%? 5%? 10%? The percent signal change is calculated for each subject on a voxel-by-voxel basis, and these percentages replace our difference scores as our new dependent variable. Percent change values will be used later for both our deconvolution analysis and our ANOVA.

Normalization of the data is accomplished as follows:

If A = Stimulus IRF
B = Baseline IRF,

Then Percent Signal Change is: (A/B) * 100

Now let's compute the percent signal change for our two subjects:

Subject 1: (1050/1000) * 100 = 105 or 5% increase in IRF from baseline (which is set to 100)
Subject 2: (525/500) * 100 = 105 or 5% increase in IRF from baseline (which is set to 100)

These results suggest that the percent change from the baseline/rest IRF to the stimulus IRF is identical for both subjects. **In both cases, the change is 5%**. While the difference scores created the impression that Subject 1showed twice the activation in response to the stimulus than Subject 2, this impression is wrong. In reality, they both showed a 5% increase in activation to the stimulus, relative to the baseline state. Hopefully this example adequately expresses the importance and necessity of normalizing FMRI data in preparation for statistical comparisons across subjects.

Step (8): Notes on Regression analysis (3dDeconvolve)

The Hemodynamic Response Function (HRF) is estimated using a gamma function ‘GAM’ in 3dDeconvolve, and evoked neural responses are convolved with the HRF to give the measured impulse response functions (IRFs).

Graphing the Impulse Response Functions created with the script:

- Open AFNI, set ‘S#_xxxx_norm.irf’ file as Underlay AND OverLay;
- Set Index above zero (1-4). Define Overlay with 2 color panes (one + and one -) to see whether there are both % increases and decreases in activity. Adjust slider down until you see colored voxels. Select ‘Graph’ to look at the time course of individual voxels (must be in ‘Original View’). You'll see a 3x3 voxel matrix showing the 5-timepoint, 14-sec **IRF** function for each voxel
- Leftclick on each point (or change with 'index' button in AFNI). The 'value' at the bottom (in red) tells you the % change in intensity; e.g., value=102.0503 means a % change of +2.0503%

- If you want, you can change the Underlay to the SPGR (remember to select 'Talairach View'), but you will lose the graphing ability.

NOTES on 3dDeconvolve (see '3dDeconvolve –help' for other options):

`-censor` can be used to remove any additional time points from analysis

`-stim_times # timefile.1D 'GAM'` generates the gamma function for the estimated BOLD response for each stimulus category; where # = stimulus category #, timefile.1D=files from step C. (2) Make Timing Files, p. 18 above.

`-iresp` generates the IRF file for each stimulus condition

`-gltsym` calculates the linear contrast between two conditions

`-fitts` generates the full model-fitted time series for each voxel

`-fout, -rout, and -tout` commands indicate that the F-stats, R^2, and t-stats are included in the bucket output.

Step (9): Extract one sub-brick for each condition to use in Group ANOVA: '3dbucket'

Input files for the ANOVA do not have to be '*.irf' files, but they must specifiy ONLY ONE SUB-BRICK. Here's how to extract one sub-brick for each condition to use in the ANOVA:

1. Open afni
2. Select bucket as OverLay and spgr as UnderLay
3. Click 'Define Overlay' and click on the 'Olay' or 'Thr' menu
4. Find the number of the sub-brik for each condition (tstat or % change) and save the number to use in `3dbucket`

To see the activation resulting from the chosen condition:

1. set Olay AND Thr to same sub-brick
2. Slide slider up to desired P-value (read the max t-value (or % change) next to the slider)
3. Click on a voxel and read the t-value at the bottom of the window.

Once you have found the # of the sub-brick (e.g. 37) for each condition, you will use this to create a "slimmed-down" bucket file that will then be used in the group ANOVA. To do this, use the command '3dbucket':

e.g. **3dbucket -prefix S14_voice_-mvfw_tstat -fbuc S14_voice_norm_bucket+orig'[37]'**

- Talairach the tstat files (see **part E**, below). The output files of type 'S14_voice_mvfw_tstat+tlrc' will be used in the group ANOVA.
- **Repeat for each condition, and for each subject.**

I. Making ROI Masks from Anatomical Volumes:

ROI masks are used on a voxel-by-voxel basis to select parts of datasets (usually functional datasets)

1. On the main AFNI control panel, set the anatomical underlay dataset (with [UnderLay]) to be what you want to draw on -- usually a SPGR or MP-RAGE type of dataset. i.e., the anatomical underlay will serve as our guide or template for drawing the ROI mask. Open the Axial image.
2. Start the **Draw Dataset** plugin (this is our ROI-creating plugin):
   [Define Datamode] → [Plugins] → [Draw Dataset]
3. Create an all zero anatomical overlay dataset with the **Draw Dataset** plugin.

- [Choose dataset for copying] → select the spgr
- At this point, the spgr is copied and COPY is added to the beginning of the filename

- If value of ‘1’ is selected, voxels that are inside the shape drawn will be filled in red
- At this point, the ‘COPY’ anatomical overlay is empty - i.e., all the voxel values are zero. This anatomical overlay must have the same geometry as the anatomical underlay, i.e., it has the same voxel size as the underlay, the same xyz-grid spacing, etc…, since drawing/editing is done on a voxel-by-voxel basis. Think of the anat overlay as a blank piece of tracing paper that is the same size as the template underneath. The blank overlay will be used to trace portions of the underlay. Voxels inside the traced portion will make up the ROI mask. Values outside the traced region are irrelevant (zeros). Note: You could also edit an already-existing ROI mask (that you created earlier) at this step.

4. To view and begin drawing an ROI mask (or several ROIs) on this blank anatomical overlay dataset, make sure [See OverLay] is selected
5. Start drawing the ROI mask into this blank anatomical overlay dataset. Select the ‘pen’ box inside the image window. The cursor changes to a pen. You draw by holding down the left mouse button, clicking and dragging. Voxels inside the ROI mask will receive a non-zero value (you decide what value to give them – usually ‘1’). Values outside the ROI mask will remain zero. Be sure to save the results by pressing [Save]. [SaveAs] will create another file, but you can decide what to call it. Click [DONE] when you’re done. ([Quit] will exit the ROI plugin without saving
your work)
6. Convert the anatomical-resolution ROI dataset into a dataset at the resolution of the functional (statistical) datasets you want to analyze with the ROI

Notes:

- The ROI and functional datasets may already be at the same resolution, if you are operating in +tlrc coordinates
- Resolution conversion of masks is done with program **3dresample** (p. 25-6)
- Use programs **3dmaskave**, **3dmaskdump**, and **3dROIstats** to extract ROI-based information about functional datasets
- Also can use the **ROI Average** plugin in AFNI to extract interactively the average of a dataset over a ROI (does the same thing as **3dmaskave** in the command line)

STEP 2. Group Analysis

Combine data across subjects with 3dANOVA (modified from AFNI HOW-TO #5, Part II)

A. Run a 2-Way ANOVA with AFNI '3dANOVA2'

This AFNI program performs a two-factor analysis of variance (ANOVA) on 3D datasets.

```
Usage:  3dANOVA2
              -type  k
              -alevels  a
              -blevels  b
              -dset  a  b
              [-options]
```

(see also '3dANOVA2 –help')

----

* EXAMPLE of 3dANOVA2

In our sample experiment, we have two factors (or independent variables) for our analysis of variance: "Stimulus Condition" and "Subjects". As such, we are using the '3dANOVA2' program. The levels for each factor are shown below:

a) Stimulus Conditions = 12 levels:
VOICE TRIAL (6 conditions): Fword/Mvoice, Mword/Fvoice, Mword/Mvoice, Nword/Fvoice, MWord/Mvoice/Noise, Nword/Fvoice/Noise;
WORD TRIAL(6 conditions): Fword/Mvoice, Mword/Fvoice, Mword/Mvoice, Nword/Fvoice, MWord/Mvoice/Noise Nword/Fvoice/Noise

b) Subjects (n = 4)

Our script shows the following '3dANOVA2' command (*don't panic-* each argument and option will be explained in detail later):

# OF CONDITIONS

# OF SUBJECTS

```
3dANOVA2 -type 3 -alevels 12 -blevels 4   \
-dset 1 1 S1-voice-fwmv-tstat+tlrc         \
-dset 2 1 S1-voice-mwfv-tstat+tlrc         \
-dset 3 1 S1-voice-mwmv-tstat+tlrc         \
-dset 4 1 S1-voice-mwmvn-tstat+tlrc        \
-dset 5 1 S1-voice-nwfv-tstat+tlrc         \
-dset 6 1 S1-voice-nwfvn-tstat+tlrc        \
-dset 7 1 S1-word-fwmv-tstat+tlrc          \
-dset 8 1 S1-word-mwfv-tstat+tlrc          \
-dset 9 1 S1-word-mwmv-tstat+tlrc          \
-dset 10 1 S1-word-mwmvn-tstat+tlrc        \
-dset 11 1 S1-word-nwfv-tstat+tlrc              \
-dset 12 1 S1-word-nwfvn-tstat+tlrc             \
    -dset 1 2 S2-voice-fwmv-tstat+tlrc          \
-dset 2 2 S2-voice-mwfv-tstat+tlrc              \
-dset 3 2 S2-voice-mwmv-tstat+tlrc              \
-dset 4 2 S2-voice-mwmvn-tstat+tlrc             \
-dset 5 2 S2-voice-nwfv-tstat+tlrc              \
-dset 6 2 S2-voice-nwfvn-tstat+tlrc             \
-dset 7 2 S2-word-fwmv-tstat+tlrc               \
-dset 8 2 S2-word-mwfv-tstat+tlrc               \
-dset 9 2 S2-word-mwmv-tstat+tlrc               \
-dset 10 2 S2-word-mwmvn-tstat+tlrc             \
-dset 11 2 S2-word-nwfv-tstat+tlrc              \
-dset 12 2 S2-word-nwfvn-tstat+tlrc             \
    -dset 1 2 S10-voice-fwmv-tstat+tlrc                   \
-dset 2 2 S10-voice-mwfv-tstat+tlrc             \
-dset 3 2 S10-voice-mwmv-tstat+tlrc             \
-dset 4 2 S10-voice-mwmvn-tstat+tlrc            \
-dset 5 2 S10-voice-nwfv-tstat+tlrc             \
-dset 6 2 S10-voice-nwfvn-tstat+tlrc            \
-dset 1 3 S13-voice-fwmv-tstat+tlrc             \
-dset 2 3 S13-voice-mwfv-tstat+tlrc             \
-dset 3 3 S13-voice-mwmv-tstat+tlrc             \
-dset 4 3 S13-voice-mwmvn-tstat+tlrc            \
-dset 5 3 S13-voice-nwfv-tstat+tlrc             \
-dset 6 3 S13-voice-nwfvn-tstat+tlrc            \
-dset 1 4 S14-voice-fwmv-tstat+tlrc             \
-dset 2 4 S14-voice-mwfv-tstat+tlrc             \
-dset 3 4 S14-voice-mwmv-tstat+tlrc             \
-dset 4 4 S14-voice-mwmvn-tstat+tlrc            \
-dset 5 4 S14-voice-nwfv-tstat+tlrc             \
-dset 6 4 S14-voice-nwfvn-tstat+tlrc            \
-voxel 2000                                     \
-fa Fstat_all                                   \
-amean 1 mean-voice-fwmv                        \
-amean 2 mean-voice-mwfv                        \
-amean 3 mean-voice-mwmv                        \
-amean 4 mean-voice-mwmvn                       \
-amean 5 mean-voice-nwfv                        \
-amean 6 mean-voice-nwfvn                       \
-adiff 1 2 1vs2                                 \
-adiff 1 3 1vs3                                 \
-adiff 2 5 2vs5                                 \
-adiff 3 4 3vs4                                 \
-adiff 3 5 3vs5                                 \
-adiff 4 6 4vs6                                 \
-adiff 5 6 5vs6                                 \
-acontr  2  -1  -1  0  0  0  contr_1vs2+3
```

Explanations:
-type 3 is a mixed effects model, where ‘a’ is fixed, ‘b’ is random
-alevels is the number of conditions
-blevels is the number of subjects

NB: In each contrast (adiff) file there are two sub-briks: 0=% change; 1=t-value

B. Alpha Correction for Multiple Comparisons: Why it's CRUCIAL!

(from http://afni.nimh.nih.gov/sscc/gangc/mcc.html and http://afni.nimh.nih.gov/pub/dist/doc/manual/AlphaSim.pdf)
In FMRI studies data analysis is usually done voxel-wise with all statistical tests conducted separately and simultaneously. Although these voxel-by-voxel tests increase the precision of the conclusions in terms of clusters, they also lead to the increase of the chance that at least one of them is wrong. Therefore, a family of statistical tests suffers one serious problem: the probability --traditionally called 'alpha' - of at least one error (type I, or false positive) is greater than that of an error on an individual test. To control the severity of this problem of alpha escalation, some measure of multiple testing correction (similar to multiple comparisons correction in the traditional sense) is desirable during group analysis.

In order to achieve a specified level of statistical significance, allowance must be made when multiple statistical comparisons are required. This is typically accomplished by using the Bonferroni method: when making N simultaneous inferences, an overall significance level of alpha is achieved by using individual significance levels of alpha / N. For a typical AFNI 3D functional image, with N = 64 x 64 x 26 = 106,496 voxels, the Bonferroni correction is too conservative, causing a loss of power to detect areas of true activation.

There are three occurrences of multiple comparisons in FMRI analysis: individual subject analysis, group analysis, and conjunction analysis. Compared to the number of EPI voxels (on the order of 10^4 inside the brain out of total ~10^5 voxels), conjunction analysis with a few contrasts is much less a severe problem than the case for individual subject and group analysis, and can be simply dealt with Bonferroni correction if the researcher is willing to.

If an analysis fails to survive correction, a lot of factors could contribute to the failure since the analysis is such a long chain of steps, but one possibility is the power of the analysis. If statistical power is an issue, then you might consider optimizing the experiment design or increasing the number of subjects.

Familywise approach

Familywise approach fixes alpha for the whole family (brain) of tests. For example in a brain with 25,000 voxels, a fixed type I error of 0.05 would lead to false detection of 1,250 active voxels simply by chance. By significantly lowering individual type I error, we could achieve the control of total type I error. This is usually done with the consideration of cluster size in the brain, in which case a corrected type I error of p means that among 100 such brain activation maps on average there would have 100*p% of them having false detection. However the downside is that the cost of the approach is the loss of power on the individual tests, increasing type II error (beta error).

The Bonferroni correction (also called Fisher's method of alpha splitting) is such a familywise multiple-comparison correction used when multiple dependent or independent statistical tests are being performed simultaneously, but it is overly conservative in the case of fMRI analysis as it doesn't consider cluster size and spatial correlation among neighboring voxels. It works well if the brain behaves like Brownian motion of water molecules. Of course the brain has various well-organized structures.

Monte Carlo simulations

One approach to dealing with this problem in AFNI is to run Monte Carlo simulations with AlphaSim to obtain corrected type I error for the statistical analysis in the whole brain or an ROI. Such a dual thresholding of both type I error (i.e., alpha) and cluster size gives a reasonable correction for simultaneous tests during the group analysis.

In order to carry out the Monte Carlo simulation, we need some basic information about the data itself, as well as the uncorrected P-value that we plan to use to create clusters, as well as a few other parameters that control the clustering operation.

First, we need an estimate of how spatially smooth the noise in the dataset is. If we used spatial blurring earlier, we should use that same value for FWHM here. During single-subject analysis, we used a spatial blur of 8mm, so our estimate of spatial smoothness that we use in Monte Carlo simulations should also be 8mm.

The next step is to decide upon the uncorrected P-value that we will use to isolate clusters (option '–pthr'). This choice is fairly arbitrary. If we choose a fairly liberal -pthr, say P = 0.01, we will be able to detect large clusters of voxels that are activated at a fairly liberal threshold, but we will miss small clusters that are made up of highly significantly activated voxels. If, on the other hand, we use a conservative -pthr, say P = 0.0001, we will be able to detect small clusters of highly activated voxels, but not larger clusters of less activated voxels. The final choice of –pthr depends on the size of clusters that we are looking for, as well as considerations of what our overall statistical power is likely to be. We typically set –pthr = 0.005

The other parameter that we need to determine is what we want our cluster connection radius (-rmm) to be. This specifies how close two voxels need to be, measured center to center, in order to be considered part of the same cluster. This number is based upon the voxel size of the datasets being analyzed: in the case of Talairached data, voxels are always 1x1x1mm.

In original space, there are 3 different ways to define how voxels are considered neighbors: two voxels touch one face, one edge, or one point (vertex). Correspondingly, you can choose a number for –rmm that is equal to one voxel dimension, slightly larger than the in-plane diagonal, or slightly larger than the diagonal of a voxel. This means that any two voxels with a center-to-center distance smaller than the connectivity radius are treated as neighbors in a cluster. For example, with a voxel size of 3 x 3 x 3mm, the connectivity radius can be defined as 3.01 mm [face-to-face], 4.3 mm [edge –to –edge > sqrt(3^2 + 3^2)], or 5.2 mm [point-to-point > sqrt(3^2 + 3^2 + 3^2)]. The smaller the connectivity radius, the more difficult it is to form clusters.
In Talairach space, the connectivity radius can be 1.01 mm [face-to-face], 1.4mm [edge –to –edge > sqrt(1^2 + 1^2)], or 1.7mm [point-to-point > sqrt(1^2 + 1^2 + 1^2)].

Options:
(1) Create a common mask or ROI among all subjects. 3dAutomask and 3dcalc are the primary tools to make such a mask. You might have obtained such a mask for each subject during individual subject analysis with 3dDeconvovle. Create a group mask by warping all the masks to Talairach space, summing them, and thresholding by n-1 (n = number of subjects); Or, multiplying all individual Talairach masks. Applying this mask to group analysis, you would avoid some spurious activations along the edge of the brain. If a template was used in spatial normalization, you can obtain a group mask by running 3dAutomask on the template.

The mask will be used for multiple comparisons correction: With a restriction on the region of interest, less voxels would be considered for correction, thus avoiding unnecessary penalty. If you have a problem obtaining survival clusters in the end, consider a smaller mask. One possibility is to restrict the correction only on the grey matter.

Voxel-level approach
Instead of controlling false positives in conjunction with minimum cluster size, an alternative is to put a limit on the total number of false positives at the individual voxel level. In other words, a corrected type I error of p indicates that among those active voxels, some proportion of them are false positives. The conventional method

is determining the *false discovery rate* with the program 3dFDR, and focuses on the total number of detected voxels in a brain without considering minimum cluster size. Thus spatial filtering (smoothing) for FDR is not as important as for family-wise error correction.

A natural question is: Which method should be used, family-wise error correction or false discovery rate? Practically this might boil down to whichever method gives "nice" active maps, but the choice depends more on the nature of the study: If you want some extent of certainty (sensitivity) on cluster detection, go with family-wise error correction; If specificity combined with the control of false positives is more important, choose FDR.

Run AlphaSim - there are several ways to run it.

1st way: AlphaSim with coordinates typed in:
For STROOP study: FOV(X,Y)=240mm (3.75mm x 64 slices); FOV(Z)=130mm (5mm x 26 slices);
+tlrc voxel dimensions = 1 x 1 x 1mm

Example 1 command line:
AlphaSim -nx 240 -ny 240 -nz 130 -dx 1 -dy 1 -dz 1 -iter 1000 -rmm 1 -pthr 0.005 \
-fwhm 8.0 -out alphasim.txt (use '-out' option if you want to save output to a text file)

The '–iter 1000' option specifies that we want to run 1000 simulations. The program will take quite a few minutes to run, and then produce output like the following:

Example 1 output:

```
Program:          AlphaSim
Author:           B. Douglas Ward
Initial Release:  18 June 1997
Latest Revision:  10 Jan 2008

Data set dimensions:
nx =   240   ny =   240   nz =   130   (voxels)
dx =  1.00   dy =  1.00   dz =  1.00   (mm)

Gaussian filter widths:
sigmax =  3.40   FWHMx =  8.00
sigmay =  3.40   FWHMy =  8.00
sigmaz =  3.40   FWHMz =  8.00

Cluster connection radius: rmm =  1.00

Threshold probability: pthr = 5.000000e-03

Number of Monte Carlo iterations =  1000

Cl Size     Frequency    CumuProp     p/Voxel   Max Freq       Alpha
      1         16915    0.046580  0.00497567         0     0.999996
      2         10332    0.075032  0.00497341         0     0.999996
      3          7672    0.096159  0.00497065         0     0.999996
```

----

and so on. What you now need to do is read down the right-most column, which gives the corrected P-value for each minimum cluster size (left hand column).

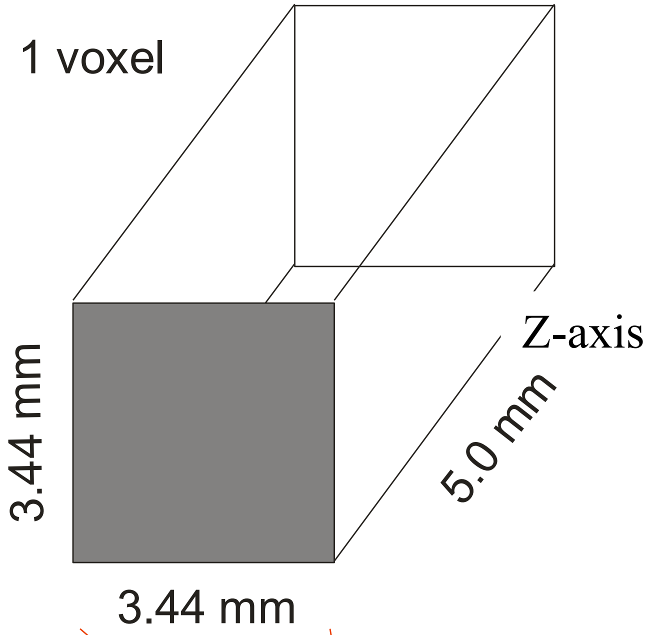
```
   1692             0    0.999855  0.00001291         0     0.051000
   1693             0    0.999855  0.00001291         0     0.051000
   1694             0    0.999855  0.00001291         0     0.051000
   1695             0    0.999855  0.00001291         0     0.051000
   1696             0    0.999855  0.00001291         0     0.051000
   1697             2    0.999861  0.00001291         2     0.051000
   1698             0    0.999861  0.00001246         0     0.049000
   1699             0    0.999861  0.00001246         0     0.049000
   1700             0    0.999861  0.00001246         0     0.049000
   1701             0    0.999861  0.00001246         0     0.049000
   1702             2    0.999861  0.00001246         2     0.049000
   1703             0    0.999861  0.00001246         0     0.049000
```

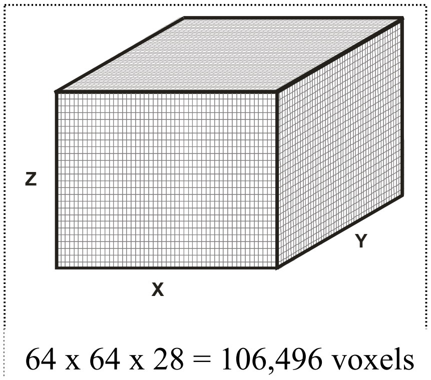
IMPORTANT: Remember to report these correction parameters in your publications!

```
   1704            0    0.999861  0.00001246         0    0.049000
   1705            0    0.999861  0.00001246         0    0.049000
```

So from this we can see that according to Monte Carlo simulations, thresholding the statistical images with an uncorrected *P* = 0.005, and clustering with a cluster connection radius of 1mm, the resulting clusters need to be at least 1702 voxels in size in order to achieve a corrected *P* < 0.05. We can now use this information to threshold and cluster our group analysis statistical maps – this is explained in the next section.

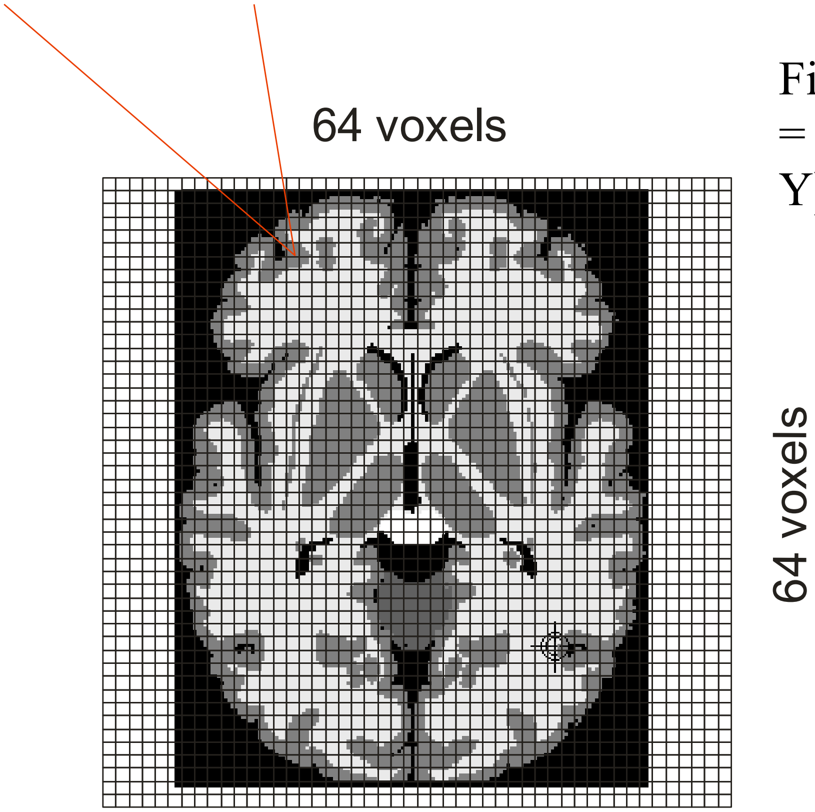
FUNCTIONAL DATA IN ORIGINAL VOXEL DIMENSIONS





64 x 64 x 28 = 106,496 voxels

FOV = 220 mm



Field of view = 64 x 3.44 = 220 x 220mm (X and Y)

V. DATA EXTRACTION / ROI Analysis

• DATA EXTRACTION

Step 1. Cleaning up the group functional files with plugin ‘DrawDataset’

You will want to look at your functional data (means and contrasts) on top of an anatomical file (spgr+tlrc) in order to make sure that the functional data do not include too many voxels that are actually outside or on the edge of brain tissue.

NB: The DrawDataset plugin will accept files that have only 1-subbrick. You can use *3dbucket* to extract the subbrick you want from the functional file.

a. edit functional files using **DrawDataset**. Deselect “Copy Dataset”
b. click on “Choose dataset to change directly”; select the OverLay file to edit
c. click “pen” in viewing window: this lets you use left mouse button to draw
d. Flip through each slice. When you find a slice with too much overlap, erase data by encircling it with Value=0.
e. repeat for each slice. It can be helpful to de-select “See OverLay”. Or, you can change the “Color overlay opacity” in the view window (right side, above “z” for “zoom”).
f. Save edited dataset by pressing “SaveAs”; use prefix like 'xxx_mean_clean'

Step 2. Clustering from the Command Line using ‘*3dmerge*’

Before we run the **3dmerge** program, we need to determine the threshold t-statistic that corresponds to an uncorrected *P*-value (typically 0.01-0.05). This can be read off the threshold slider in AFNI when viewing the functional statistical map. In this example, *P*=.05 and the correct t-value is 2.228.

NB: In each mean (-amean) and contrast (-adiff) file from the ANOVA there are two sub-briks: 0=% change; 1=t-value

Now we can use the **3dmerge** program to perform the clustering:
These examples are based on the Talairach data grid of 1 x 1 x 1 (in mm) and a threshold of 1702 voxels.

If you want to create a dataset that retains the intensity t-values at a fixed threshold, use the following:
EXAMPLE: 3dmerge –dxyz=1 -1clust 1 -1702 -1thresh 2.228 –prefix mean_voice_mvfw_clust \ mean_voice_mvfw+tlrc’[1]’ If instead you want to retain %-change values, use **mean_voice_mvfw+tlrc’[0]’** as the input file.

If you want to create a dataset that overwrites the intensity values and assigns a single number (and color) to each cluster, then use the following:
EXAMPLE: 3dmerge –dxyz=1 -1clust_order 1 -1702 -1thresh 2.228 –prefix \ mean_voice_mvfw_clust_order mean_voice_mvfw+tlrc’[1]’ (or use **mean_voice_mvfw+tlrc’[0]’** to retain %-change data**)**

The first command will identify clusters of voxels that exceed a t-value of 2.228, using a cluster connectivity radius of 1mm, and a minimum cluster size of 1702 voxels, using sub-brick 1 as the thresholding brick, saving the thresholded, clustered data in a new bucket dataset called ‘mean_voice_mvfw_clust +tlrc’. The second command will do the same thing, except all voxel intensities within a cluster will be replaced by the cluster size index (largest cluster=1, next largest=2, etc...). This will be useful for extracting data from specific clusters later on.

NB: The reason why we specify the minimum cluster size as a *negative* number is that it denotes minimum cluster size in *voxels*. If we were in original space, we would put a positive 1702 there, and that would denote a *cluster volume* of 1702 mm^3.

NB: Some of the other options in the command **3dmerge** might be useful. In particular, you can play around with the '–1erode' and '–1dilate' options, which are useful in separating clusters that are joined by a narrow "neck". You will also want to separate clusters that appear as mirror-images across the midline and separate larger clusters into smaller ones that reflect logical activation ROIs. **Do this cleanup with DrawDataset as in Step 1 above. NB: this plugin only allows you to undo the LAST thing you did. You may want to save frequently with alternate file names and then delete intermediate versions.**

Step 3. Output active clusters to a spreadsheet using '*3dclust*'

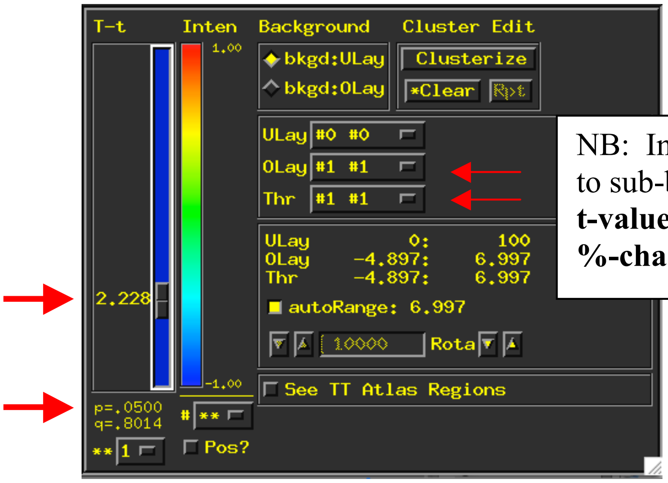
Usage: 3dclust [editing options] [threshold] [cluster connection radius (in millimeters)] [minimum cluster size] [input dataset]

EXAMPLE: 3dclust –dxyz=1 -1thresh 2.228 1 1702 mean_voice_fvmw+tlrc'[1]' > out.txt

The above command line tells **3dclust** to find potential cluster sizes for dataset *mean_voice_fvmw+tlrc*, sub-brick #1 (t-stat data), where the threshold has been set to t=2.228; connection radius = 1 (to be included in a cluster, voxels must be no more than 1mm center-to-center); and the cluster must be at least 1702 voxels in size (as determined by AlphaSim). Finally, output is sent to *out.txt*

Let's take a look at where these numbers came from:

First, you must select a *P*-value at which to analyze your data. Do this by adjusting the threshold slider in AFNI. AFNI will automatically calculate the corresponding t-statistic to put in the command line. **2.228** is the threshold t-value (to left of slider) that corresponds to p=.0500 in this dataset (see first window below). The p-value and q-value will depend on the sample size (how many voxels are used in the GLM).

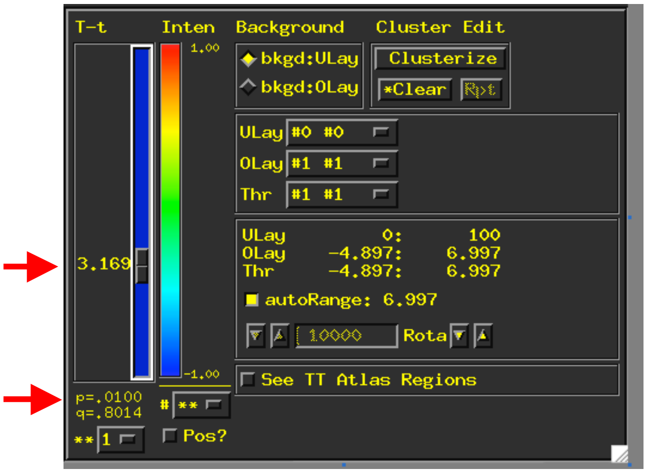


ULay #0 #0
OLay #1 #1
Thr #1 #1

ULay 0: 100
OLay -4.897: 6.997
Thr -4.897: 6.997
autoRange: 6.997

2.228
p=.0500
q=.8014

NB: In AFNI, Olay and Thr are both set to sub-brick #1 if you want to look at **t-values**, or #0 if you want to look at **%-change** values



ULay #0 #0
OLay #1 #1
Thr #1 #1

ULay 0: 100
OLay -4.897: 6.997
Thr -4.897: 6.997
autoRange: 6.997

3.169
p=.0100
q=.8014

At p=.0100 (see bottom window at left), the t-value you'd enter in the command line would be **3.169**
You'll notice that if your stats are based on smaller p-values, this will result in reporting fewer statistically significant clusters. In general, you should select the smallest p-value that gives you a few distinct clusters for each of your experimental conditions. Limiting clusters can also be achieved by changing the cluster volume threshold (VT). In this example, VT=1702, as calculated from AlphaSim. In this example, the smallest cluster found above threshold was **2109 voxels** in size (see output table below).

Talairach Coordinates, sub-brick #1 (t-values)

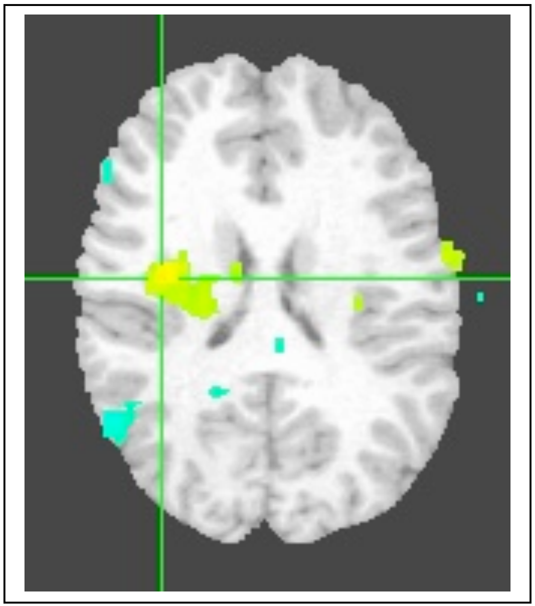
EXAMPLE of out.txt:

```
#
#Cluster report for file mean-voice-mwfv+tlrc[1]
#[Connectivity radius = 1.00 mm  Volume threshold = 1702.00 ]
#[Single voxel volume = 1.0 (microliters) ]
#[Voxel datum type    = short ]
#[Voxel dimensions    = 1.000 mm X 1.000 mm X 1.000 mm ]
#[Coordinates Order   = RAI ]
#[Fake voxel dimen    = 1.000 mm X 1.000 mm X 1.000 mm ]
#Mean and SEM based on Absolute Value of voxel intensities:
#
#Volume  CM RL  CM AP  CM IS  minRL  maxRL  minAP  maxAP  minIS  maxIS     Mean      SEM    Max Int  MI RL  MI AP
MI IS
#------  -----  -----  -----  -----  -----  -----  -----  -----  -----  -------  -------  -------  -----  -----
-----
  51629   12.4   61.6  -20.2  -61.0   63.0   26.0   96.0  -54.0   12.0   3.3636   0.0037   -9.245    0.0   68.0
-16.0
  10901  -38.4  -15.1  -14.7  -57.0  -15.0  -51.0   20.0  -35.0   -2.0     3.07   0.0054   -5.603  -46.0  -28.0
-18.0
   6420  -24.2   10.1   13.9  -40.0  -13.0  -22.0   29.0   -1.0   30.0   3.0417   0.0055    4.836  -35.0    7.0
21.0
   5977   -0.5   61.8   54.5  -15.0   13.0   43.0   81.0   40.0   76.0   2.9584   0.0046   -3.957   -3.0   63.0
54.0
   3161   10.7  -72.3    1.3  -12.0   31.0  -80.0  -60.0   -4.0    9.0   3.2464   0.0094    -4.69    8.0  -73.0
5.0
   2420   56.2   -9.3  -10.9   47.0   65.0  -23.0   12.0  -28.0   -3.0   3.0703   0.0094   -4.694   56.0  -17.0
-16.0
   2127  -15.0   54.4   27.2  -26.0   -9.0   42.0   66.0   21.0   39.0     3.15   0.0141     -5.4  -14.0   60.0
26.0
   2118  -45.3   56.6   25.0  -54.0  -34.0   46.0   78.0   17.0   34.0   2.9985   0.0081   -4.215  -46.0   52.0
25.0
   1976  -53.9  -24.0   18.2  -61.0  -45.0  -35.0  -13.0    3.0   41.0   2.9482   0.0082   -4.182  -49.0  -15.0
39.0
   1812  -27.3  -43.5   40.5  -36.0  -20.0  -61.0  -32.0   30.0   51.0   2.9059   0.0078   -4.276  -28.0  -38.0
45.0
#------  -----  -----  -----  -----  -----  -----  -----  -----  -----  -------  -------  -------  -----  -----
-----
# 88541    0.2   38.5   -7.6                                            3.2321   0.0024
```

Coordinates are as they appear in AFNI

One cluster with positive activation was identified (highlighted in yellow).

-35, 7, 21



RIGHT Hemisphere (AFNI neg.)

LEFT Hemisphere (AFNI pos.)

Explanation of **3dclust** Output:

Volume : **In original space, cluster volume (first column) is in mm^3; in Talairach space, it is the number of voxels in the cluster

CM RL : Center of mass (CM) for the cluster in the Right-Left direction (i.e., the coordinates for the CM)

CM AP : Center of mass for the cluster in the Anterior-Posterior direction

CM IS : Center of mass for the cluster in the Inf-Sup direction

minRL, maxRL : Bounding box for the cluster, min and max coordinates in the Right-Left direction

minAP, maxAP : Min and max coordinates in the Anterior-Posterior direction of the volume cluster

minIS, max IS: Min and max coordinates in the Inferior-Superior direction of the volume cluster

Mean: Mean value for the volume cluster

SEM: Standard Error of the Mean for the volume cluster

Max Int: Maximum Intensity value for the volume cluster

MI RL: Coordinate of the Maximum Intensity value in the
Right-Left direction of the volume cluster

MI AP: Coordinate of the Maximum Intensity value in the
Anterior-Posterior direction of the volume cluster

MI IS: Coordinate of the Maximum Intensity value in the
Inferior-Superior direction of the volume cluster

Notes from AFNI website:

* The program does not work on complex- or rgb-valued datasets!

* Using the -1noneg option is strongly recommended! **(by AFNI, that is. But if you use it, you won't see any negative activation!)**

* 3D+time datasets are allowed, but only if you use the -1tindex and -1dindex options.

* Bucket datasets are allowed, but you will almost certainly want to use the -1tindex and -1dindex options with these.

* SEM values are not realistic for interpolated data sets! A ROUGH correction is to multiply the SEM of the interpolated dataset by the square root of the number of interpolated voxels per original voxel.

* If you use the -dxyz=1 option, then rmm should be given in terms of voxel edges (not mm) and vmul should be given in terms of voxel counts (not microliters). Thus, to connect to only 3D nearest neighbors and keep clusters of 10 voxels or more, use something like:
3dclust -dxyz=1 1.01 10 dset+orig. In the report, 'Volume' will be <u>voxel count</u>, but the rest of the coordinate dependent information will be in actual xyz millimeters.

* The default coordinate output order is DICOM. If you prefer the SPM coordinate order, use the option '-orient LPI' or set the environment variable AFNI_ORIENT to 'LPI'. For more information, see file README.environment.

<u>Step 4</u>. Identifying Clusters with program '*whereami*'

The AFNI program **whereami** can provide the user with more detailed information regarding the output of **3dclust**. For instance, you can obtain information regarding the center of mass of each cluster, and then identify the brain structure where that center of mass is located using multiple brain atlases provided in AFNI. The program **whereami** will print a spreadsheet in the terminal window. The program 'whereami' will output coordinates in the LPI (not RAI) coordinate order, regardless of the AFNI_ORIENT variable because the report is given in multiple coordinate systems such as Talairach, MNI, and MNI Anatomical. The latter two are almost always in LPI order in the literature so AFNI decided to "minimize confusion" and report **whereami**'s coordinates in LPI all the time. The only difference between the LPI and RAI is the sign of the first two (x,y) coordinates. In AFNI's main controller, these coordinates along with their order, are specified in the top left corner (Left and Posterior are assigned positive [+] values). These should be inverted before attempting to locate structures in a Talairach or MNI map. Note that **whereami** does this for you!

EXAMPLE:

Step 1: Make 'cluster.txt' file

```
3dclust -dxyz=1 -1thresh 2.5 1 1702 mean_xxxx+tlrc > clusts.txt  [see spreadsheet above]
```

Step 2: Use this file to get structures:

```
whereami -coord_file clusts.txt'[1,2,3]' -tab | less
```

Sample output of whereami, showing ID for cluster #1 using Center-of-Mass:

```
++ Input coordinates orientation set by default rules to RAI
++ Input coordinates space set by default rules to TLRC
+++++++ nearby Atlas structures +++++++

Focus point (LPI)                    Coord.Space
 -12 mm [L], -62 mm [P], -20 mm [I]    {T-T Atlas}
 -13 mm [L], -62 mm [P], -28 mm [I]    {MNI Brain}
 -13 mm [L], -71 mm [P], -18 mm [I]    {MNI Anat.}
Atlas         Within  Label                      Prob.  Code
TT_Daemon      0.0    Left Declive                  ---    65
TT_Daemon      2.0    Left Fastigium                  ---    60
TT_Daemon      2.0    Left Culmen                     ---    66
TT_Daemon      2.0    Left Dentate                    ---   127
TT_Daemon      4.0    Left Nodule                     ---   61
TT_Daemon      5.0    Left Uvula                    ---   62
TT_Daemon      5.0    Left Pyramis                    ---   63
CA_N27_ML      0.0    Left Cerebelum (VI)                ---   99
CA_N27_ML      4.0    Left Cerebelum (Crus 1)              ---   91

******** Please use results with caution! ********
******** Brain anatomy is quite variable! ********
******** The database may contain errors! ********
```

SAME coordinates are highlighted in aqua in Table on p. 42: NOTE, HOWEVER that **whereami** has changed the sign of coordinates X and Y from positive to negative! This is done to match the Talairach-Tournoux atlas. Note also that AFNI does not know how to spell *cerebellum*!

Coordinates are **NOT** as they appear in AFNI!

NB: 'Where am I' can also be accessed from the AFNI GUI by right-clicking on the voxel of interest in any viewing window!

• ROI Analysis

What if *3dmerge*, *3dclust* & *whereami* return large clusters that cover several regions? You can do Region-of-Interest (ROI) analysis on specific brain regions. First, you'll need to create masks for each ROI:

How To Make a Master Dataset Combining Clusters Across Subjects

If you want to look at the complete dataset from ALL subjects, use the clustered files from the previous sections.

Alternatively, if you want to examine common areas of activation ROIs *across a subset of subjects*, use *3dcalc* as follows. For example, use this protocol to separate activation maps for male and female brains:
NB: you need to have all your data collected & processed up to the clustering step to make masks. All data must be in +tlrc format to combine.
NB: Reading the help files for the Draw Dataset plugin is Strongly recommended.
NB: To make a mask file from a functional file, you must have a functional file that has only one sub-brick.

Make a master template for a sub-set of subjects:

1. Change (cd) to the mask directory
2. Make sure you have a clean cluster file for each condition and each subject in the experiment
3. Combine clustered files by condition.

a. 3dcalc –prefix *name* –a *filename* –b *filename* –c *filename* –expr (a+b+c)
   (where a, b, c, etc are individual subjects within a condition)

e.g., **3dcalc –prefix all-attn-noise –a S1-attn-noise-clus+tlrc –b S2-attn-noise+tlrc \
–c S3-attn-noise+tlrc –expr (a+b+c)**

4. Decide how much overlap across subjects you want to consider a "common area of activation". This might be areas active in half of subjects, 2/3rds of subjects, etc.
5. Threshold clusters by this value; in this case, threshold=10

a. 3dcalc –prefix *name* –a *filename#* –expr astep(a,#)
(where # is the threshold chosen for number of subjects showing activation)
e.g., **3dcalc –prefix attn_noise_men –a attn_noise_all+tlrc –expr 'astep (a,10)'**

```
NB:    step(x)     = {1 if x>0        , 0 if x<=0},
       astep(x,y) = {1 if abs(x) > y , 0 otherwise} = step(abs(x)-y)
```

How to make MASKS from the Talairach Atlas:

Start with the **mask_template+tlrc** file:

a. Right-click in AFNI Image window
b. Select 'Draw ROI plugin'
c. uncheck 'Copy Dataset'
d. Choose dataset to change directly (e.g., TT_mask_template+tlrc)
e. Click on BAR beneath 'TT Atlas Region to Load'
f. Select a region of interest (e.g., putamen)
g. Select Hemisphere(s) [Left only, Right only, or Both]
h. Click 'Load: Overwrite' With Value=1, putamen should now be red.
i. SaveAs (e.g., **TT_mask_Lputamen** or **TT_mask_Rputamen** or **TT_mask_putamen**
   NB: for ease of later steps, make sure mask files are named for hemisphere (r/l), area (e.g., smg, prcs, stg, etc), and –mask.

You can also create a mask file from the command line using **'3dcalc'**
To assign all edited clusters a value=1,
EXAMPLE: 3dcalc –prefix foo_mask –a foo+tlrc –expr 'step(a)'

Using the mask to extract functional data:

a. 3dcalc –prefix *masked_datafile* –a *mask_file* –b *master_file* –expr (a*b)
e.g., **3dcalc –prefix attn_rsmg –a rsmg_mask+tlrc –b all_attn+tlrc –expr (a*b)**
Repeat this step for all ROIs.

b. Remove all clusters that are not the ROI:

1. afni
2. Switch function to the clustered image to clean up
3. Switch structural image to the +anat file so you can see artifacts
4. **Define Datamode**
5. **Plug Ins**
6. **Draw Dataset**

click Choose Dataset on Which to Draw

a. set Drawing Value to 0
b. set Drawing mode to Filled Curve
c. using middle mouse button, circle any clusters to be eliminated.
d. save

NB: this plugin only allows you to undo the LAST thing you did. You may want to save frequently with alternate file names and then clean up intermediate versions with rm.

Once all masks are made:

1. create a subdirectory called templates
(mkdir templates)
2. move all template files to it
(mv *files* templates)
3. remove original clustered files that are not mask files
(rm *clus*)
3. copy each mask file into the subject directory fim subdirectory
(e.g., cp *mask* ../S1/combos/fim)

We would typically want to carry out the same basic operation on all the other means and contrasts of interest. It is then possible to combine all the resulting clustered datasets into one big bucket file using the program **3dbucket**.

Also See: Region of Interest Drawing and Usage in AFNI

DUMP data associated with each ROI to a TABLE on the SCREEN

NB: you may not want to dump data from event designs in the same way.

1 open **Afni**
2. click on **Define Datamode**
3. click on **Plugins**
   a. click on **ROI Average**
      1. Choose Source Dataset (e.g. S1-attn-noise+tlrc)
      2. Choose Mask Dataset (e.g., rsmg-mask+tlrc)

3. set subbriks to 0
4. click 1D save
5. give an output name (e.g., S1-attn-noise-rsmg)
6. click To Disk yes
7. click Run & Close
8. repeat for each ROI

NB: want to dump both the mask file and the activation files to determine how many voxels were possible in the activation file (setting the intensity mask prevents 0 value voxels from dumping)

Warp files to standard space

NB: You must refresh the file list before trying to warp functional files to the +tlrc versions. You must have spgr+tlrc open as UnderLay. You may want to use <u>3ddup</u> to duplicate files before altering.

1. open Afni
2. Select anatomy (spgr) and functional files to be rendered
   NB: you must transform them together or they will not match up later
3. push **Define Data mode** button (new panel will open)
   a. click **Warp on demand** buttons for anat and func
   b. change anat resam mode to Li
   c. change func resam mode to Bk
   d. click on write "many" button and highlight all to write out (all will be +tlrc) or click on write "anat" and "func" buttons (does one at a time)
4. exit afni

Make image of # Subjects showing activation per ROI

1. use clustered images with +tlrc extension
2. If this step has not already been done, save clustered images without negative values
   a. afni
   b. plugins
   c. 3d Edit
   d. choose data set
   e. prefix (type in output file name, e.g. S1-cond1-noneg)
   f. options: No Neg
   g. run+keep or run+quit
3. for each subject to be combined in composite:
   3dcalc -prefix *ouput file* -a *input file* -expr "step(a)"
   e.g. **3dcalc -prefix S1-clus2 -a S1-cond1-noneg+tlrc -expr "step(a)"**
4. combine files created in step 2
   3dcalc -prefix *ouput file* -a *input file1* -b *input file2* … -n *input filen* -expr *"*a+b*…n"*
   e.g. **3dcalc -prefix composite -a S1-cond1-noneg+tlrc -b S2-cond1-noneg+tlrc -c S3-cond1-noneg+tlrc -expr "a+b+c"**

An alternative to *3dmerge*, *3dclust* & *whereami*: Output active clusters from the AFNI GUI using plugin '*3dedit'*

3dedit will return a 1D data brick. If you want to retain other sub-bricks, use **3dclust (see part A)**.

NB: You can cluster and remove negative activation (if you want to) in one step using 3dEdit
NB: Input file must contain only one sub-brick.
NB: Unless you select the option to keep thresholded data (scroll down to find it), 3dEdit will remove that information. This may be what you want to do to make masks. However, if you want to threshold the data, you need to keep the threshold information.

1. afni
2. push Define Data Mode button
3. push Plugins button
4. select **3D Edit** plug-in
    a. choose Input [--choose func+orig dataset--] button and select file to cluster
    b. select Prefix box and type in new clustered file name (e.g., S1-attn-noise-clust)
    c. click Options button
        1. change No Neg to true if you want to remove negative BOLD signals
    d. click Threshold button
        1. set Cutoff to .31 (NB: this is 10% of the variance, can set otherwise)
    e. select Cluster button (bottom)
        1. type=keep
        2. calculate Radius(mm)
            1. need > 1* voxel width to connect adjacent voxels
            (e.g. rmm of 4 will connect 2 adjacent 3.44mm voxels)
            2. > 3* voxel width will count any two voxels within that distance (may have gaps between the voxels clustered together)
            (e.g., 4mm so that all 3.44mm voxels must be adjacent to constitute a cluster)
        3. calculate MinVol(ul)
            1. calculate voxel volume (height x width x depth)
            2. calculate minimum volume as # voxels*voxel volume
            (e.g., 177 for a 3 voxel cluster; 295 for a 5 voxel cluster if voxels are 3.44*3.44*5)
    f. push Run+keep or Run+close (top)

NB: it is strongly suggested that you clean up clustered files using Draw Dataset to erase clusters that appear outside the brain or in regions of susceptibility artifact prior to copying them to the mask directory:
Transform ("Warp") files to Talairach space

NB: you must refresh the file list before trying to warp functional files to the +tlrc versions. You must have spgr+tlrc open as UnderLay.
Also note that you may want to use <u>3ddup</u> to duplicate files before altering.

Change directory to appropriate subdirectory.

1. afni
2. Select anatomy (spgr) and functional files to be transformed
    NB: must transform them together or they will not match up later
    You need to select 'Talairach View' button (top middle) to see +tlrc versions
3. push "define data mode" button (new panel will open)
    a. click "warp on demand" buttons for anat and func
    b. change anat resam mode to Li
    c. change func resam mode to Bk
    d. click on write "many" button and highlight all to write out (all will be +tlrc)

or click on write “anat” and “func” buttons (does one at a time)

4. exit afni

• Functional Connectivity Analysis

by Gang Chen Edited by TC

This type of correlation analysis does not consider specific tasks/conditions/stimuli during the analysis.
Notes from TC: *Actually, yes it does.*

The following example of running correlation analysis demonstrates how such analysis is done with AFNI. The steps are built based on discussion on AFNI message board and through personal communications (Thanks especially go to Giuseppe Pagnoni for his contribution and insight).

Notes from TC: *To see graphs of response functions, you must select* ****.irf*** *files as the UNDERLAY.*

(1) Remove the baseline, drifting effect, head motion and any effects of no interest from the original time series. The purpose of this step is to make the seed region's time series immune to any undesirable effects other than the resting state (or task effect). This can be done with program 3dSynthesize (and 3dcalc), but it has to take some output from 3dDeconvolve as input (see note below). More specifically, it needs two inputs: one from option -cbucket in 3dDeconvolve, the other from option -x1D. If you haven't done so in the previous analysis, you may have to re-run 3dDeconvolve with these options. The following command isolates the effects of no interest:

```
3dSynthesize -cbucket CbucketFileFrom3dDeconvolve+orig -matrix x1DFrom3dDeconvolve -select ...
prefix EffectsOfNoInterest
```

NB: see Help menus for 3dDeconvolve and 3dSynthesize

Notes from TC: *You should have already removed baseline, drift, head motion, etc. from the original time series using the pre-processing script, starting on p. 20.*

Then you may want to remove the effects of no interest from the original signal by using 3dcalc:

```
3dcalc -a InputFor3Deconvolve -b EffectsOfNoInterest -expr 'a-b' -prefix CleanData
```

Notes from TC: *This step is optional because 3dDeconvolve has already done the regression.*

(2) Convert the subject's EPI time series data to +tlrc space

```
adwarp -apar anat_subj01+tlrc -dpar 'CleanData_subj01+orig' -dxyz 3 -prefix EPI_subj01
```

where ‘anat_subj01+tlrc’ is an anatomical (spgr) dataset from the same subject.

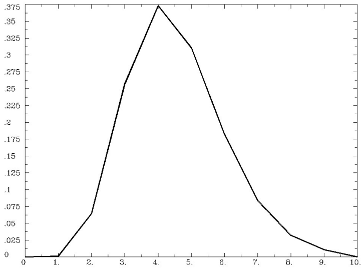
Notes from TC: *You can also use ‘Warp ULay on Demand’ in AFNI.*

(3) Use 3dmaskave (or 3dmaskdump) to extract average time series of the ROI or time series at a voxel with a peak t value (one voxel - (46, -29, 21) - in this case). If no options are given, then all voxels are included.
NB: This might result in a GIGANTIC output file.

```
3dmaskdump -noijk -dbox 46 -29 21 EPI_subj01+tlrc > EPI_subj01.1D
```

Notes from TC: *Here is an example using more familiar names:*

```
3dmaskdump -noijk -dbox -1 8 11 s11_memboth_fa-norm-irf+tlrc > s11_fa_rthalamus_dump.1D
```



Where-am-I tells us that **-1 8 11** is:
within 2mm of Right Thalamus
within 3mm of Right Medial Dorsal Nucleus
within 5mm of Right Anterior Nucleus

Notes:

- x y z are the coordinates copied **as-is** from AFNI; 'Where-am-I?' **corrects** X and Y (changes sign)
- -noijk avoids writing out 3 coordinate index numbers at the beginning to the output file.
- Regarding -dbox, be careful with different coordinate systems. Here (see footnote 1, below) is some detail about the issue. Also check out '3dmaskdump -help'.

The output **s11_fa_rthalamus_dump.1D** is a one-row text file. Convert the one-row time series to one column:

```
1dtranspose s11_fa_rthalamus_dump.1D s11_fa_rthalamus.1D
```

(3.1) Regressors of no interest (optional)

It is recommended that physiological data such as cardiac and respiratory rates be recorded during the scanning and be included as covariates in correlation analysis. If such physiological data are not available, the global signal might be a remedy:

```
3dAutomask -dilate 1 -prefix EPI_subj01_mask EPI_subj01+tlrc
3dmaskave -quiet -mask EPI_subj01_mask+tlrc EPI_subj01+tlrc > subj01_global.1D
```

(4) Correlation analysis (with quadratic fitting for baseline and trend),

```
3dfim+ -input EPI_subj01+tlrc \
 -polort 2 \                          # Quadratic fitting of the baseline  + drift
 -ort_file subj01_global.1D \   # Global signal or physiological regressors
 -ideal_file s11_fa_rthalamus.1D \  # ROI used to seed the correlation
 -out Correlation \
 -bucket s11_fa_corr_rthalamus
```

Alternatively, 3dDeconvolve should be used, especially if some time points need to be censored out or if multiple seeds are involved in the analysis,

```
3dDeconvolve -input EPI_subj01+tlrc \
 -polort 2 \
 -censor ... \
 -num_stimts 2 \
 -stim_file 1 subj01_global.1D -stim_label 1 RegressorOfNoInterest \
 -stim_file 2 s11_fa_rthalamus.1D -stim_label 2 CorrCoef \
 -tout -rout -fitts s11_fa_corr_rthalamus \
 -bucket s11_fa_corr_rthalamus
```

```
[-fitts  fprefix]    fprefix =
prefix of 3D+time output dataset
which will contain the (full
model) time series fit to the
input data
```

NB: 3dDeconvolve can only spill out R^2, not the correlation coefficient R itself. So if you want R, you would have to take square root and find out its sign based on the sign of its corresponding beta value.

```
3dcalc -a s11_fa_corr_rthalamus+tlrc'[SubbrickForR2]' \
-b s11_fa_corr_rthalamus+tlrc'[SubbrickForBeta]' -expr 'ispositive(b)*sqrt(a)-isnegative(b)*sqrt(a)'  \
-prefix s11_fa_R_rthalamus
```

(5) Convert correlation coefficients to Gaussian

Correlation coefficients range from -1 to 1. To be able to run group analysis, Fisher's Z-transformation formula can be used to reduce skewness and make the sampling distribution more normal when sample size is big enough: $z = (1/2) * \ln[(1+r)/(1-r)]$, where z is approximately normally distributed with mean r, and standard error $1/(n-3)^{0.5}$ (n: sample size).

```
3dcalc -a s11_fa_R_rthalamus+tlrc -expr 'log((1+a)/(1-a))/2' -prefix s11_fa_Z_rthalamus
```

(6) Group analysis with Z-scores or regressor coefficients for all subjects:

```
3dttest -prefix Grp_t -base1 0 \
-set2 \
Corr_subj01_Z+tlrc \
Corr_subj02_Z+tlrc \
Corr_subj03_Z+tlrc \
Corr_subj04_Z+tlrc \
Corr_subj05_Z+tlrc \
Corr_subj06_Z+tlrc \
Corr_subj07_Z+tlrc \
Corr_subj08_Z+tlrc \
Corr_subj09_Z+tlrc \
Corr_subj10_Z+tlrc \
Corr_subj11_Z+tlrc \
Corr_subj12_Z+tlrc
```

```
-set2 datasets ... = Specifies the
collection of datasets to put into
the second set.  There must be at
least 2 datasets in each of set1
(if used) and set2.
```

• Context-Dependent Correlation Analysis

by Gang Chen Edited by TC

Typically, subjects perform some tasks, or are under some conditions, or go through some stimuli during the scanning session. Thus, unlike simple correlation analysis, we may want to account for the effect of a specific task/condition/stimulus on the connectivity model in the analysis, and the interaction between this psychological effect (task/condition/stimulus) and the neuronal response (physiological effect) at the seed region. In other words, the integration process at mutliple levels in the brain is dependent on the context of tasks/conditions/stimuli. Other than the seed time course, such a context-dependent correlation analysis, aka psychophysiological interaction or physiophysiological interaction (PPI), requires the insertion of a specific regressor in the model: a second-order regressor of the interaction.

Suppose we are interested in finding out the connectivity of an ROI for a contrast between two conditions, A and B. **MyInput+orig** is the normalized input file of a single subject with which you have already run regression analysis and calculated the contrast (-glt option) with 3dDeconvolve. If there are multiple runs, run steps (1)-(4) below for each run separately. The seed region can be the entire ROI, a single peak voxel or a sphere around the peak voxel. The procedure is the same regardless of the experiment type, event-related or block design.

NB: To see graphs of response functions, you must select ***.irf** files as the UNDERLAY.

Step 1. For each condition:

A. extract the average time series of the ROI

3dmaskave computes a mean for every time point/volume in the input dataset (entire run = 200+ points) that satisfies the criteria in the options list. If no options are given, then all voxels are included. Use an ROI mask to get the average for separate regions of the brain:

EXAMPLE:

3dmaskave –mask ROI+orig MyInput+orig > Seed.1D

(see REGION OF INTEREST (ROI) -BASED ANALYSIS above)

or

B. get the time course of the single voxel whose t-value peaks within the ROI [at (46, -29, 21) in this case]

3dmaskdump writes to an ASCII file the values from the input datasets that satisfy the mask criteria given in the options. If no options are given, then all voxels are included. NB: This might result in a GIGANTIC output file.

3dmaskdump -noijk -dbox 46 -29 21 MyInput+orig > seed.1D

Notes:

-input is a *-irf file after 3dDeconvolve

-coordinates are copied **as-is** from AFNI

-output ‘seed.1D’ is a one-row text file:

```
0 0.000675 0.032008 0.127819 0.18536 0.154309 0.090428 0.04159 0.01602 0.005389 0
```

-noijk avoids writing out 3 coordinate index numbers at the beginning to the output file.

Regarding -dbox, be careful with different coordinate systems. Here (see footnote 1, below) is some detail about the issue. Also check out '3dmaskdump -help'.

Step 2. Remove the trend from the seed time series

EXAMPLE:

3dDetrend -polort ? -prefix seed_row seed.1D

The output from seed_row.1D is a one-row text file:

```
# <AFNI_3D_dataset
#  self_idcode = "XYZ_AFmDIYUn_miwSZMJzFKgWA"
#  ni_type     = "11*float"
#  ni_dimen    = "1,1,1"
#  ni_delta    = "1,1,1"
#  ni_origin   = "0,0,0"
#  ni_axes     = "R-L,A-P,I-S"
# >
 -0.059418 -0.058743 -0.02741 0.068401 0.125942 0.094891 0.03101 -0.017828 -0.043398 -0.054029 -
0.059418
# </AFNI_3D_dataset>
```

Now, convert the one-row time series to one column:

1dtranspose seed_row.1D seed_col.1D

```
    -0.059418
      -0.058743
       -0.02741
       0.068401
       0.125942
       0.094891
        0.03101
      -0.017828
      -0.043398
      -0.054029
      -0.059418
```

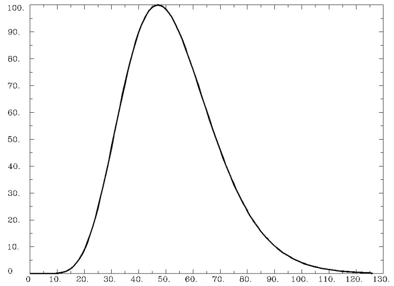
Step 3. Run deconvolution on the seed time series WHY? We already did this!

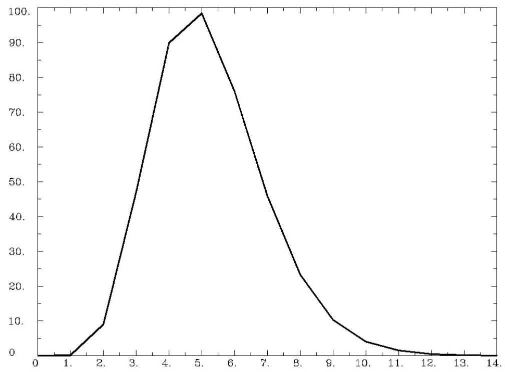
First generate a standard impulse response function:

waver -dt 0.1 -GAM -inline 1@1 > GammaHR.1D

Output of GammaHR.1D using -dt 0.1:

Output of GammaHR1.1D using -dt 1:





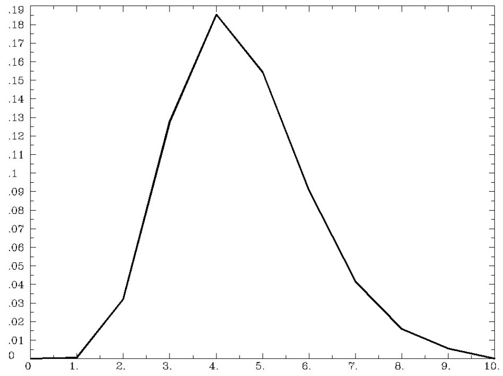
Then run:

```
3dTfitter -RHS seed_col.1D -FALTUNG GammaHR.1D seed_neur 012 -1
```

CAN’T GET THIS TO RUN: ** FATAL ERROR: Unknown argument on command line: '-FALTUNG'

Plot out Seed_Neur.1D and see if it looks reasonable compared to the stimulus presentation in the experiment. You may want to experiment with different penalty functions (check details of option -FALTUNG in 3dTfitter -help).

OUTPUT FROM 3DMASKDUMP LOOKS REASONABLE TOO. WHY NOT JUST USE THIS?



4. Obtain the interaction regressor

First create a 1D file, *AvsBcoding.1D*, with 0's (at those TR's where neither condition A nor B occurred), 1's (at those TR's where condition A occurred), and -1's (at those TR's where condition B occurred). If you only consider one condition A, there are two options: (1) If the baseline condition more or less matches up with condition A, make it a constrast with condition A (coded with 1's) versus baseline (coded with -1's); (2) If you don't believe that there is modulation (interaction) between the seed and the baseline, code condition with 1's and all other time points with 0's.

```
1deval -a Seed_Neur.1D -b AvsBcoding.1D -expr 'a*b' > Inter_neu.1D
```

The interaction is created as

```
waver -GAM -peak 1 –dt # (insert TR time) -input Inter_neu.1D -numout #TRs > Inter_ts.1D
```

THIS SOUNDS LIKE 3dDeconvolve!

5. Concatenate the regressors if there are multilple runs

Run cat separately on *seed_col.1D* (final output from step(2)) and *Inter_ts.1D* (final output from step(4)), and use the 2 concatenated 1D files for the next step.

6. Regression analysis

Basically add two more regressors from step (5) to your original 3dDeconvolve script by using option -stim_file, and add option -rout in 3dDeconvolve since the correlation coefficient for regressor *Inter_ts.1D* will be taken for group analysis. That is, include all of the original regressors (of interest and no interest) plus the two new ones. That way all sources of variation in the data would be properly accounted for.

7. Convert the correction coefficients for interaction to Z scores through Fisher transformation

3dDeconvolve can only output coefficient of determination R^2 (-rout), not correlation coefficient R itself. So we need to take square root of R^2 and find out its sign based on the sign of its corresponding beta value:

3dcalc -a Corr_subj01+orig'[SubbrickForR^2]' *-b* Corr_subj01+orig'[SubbrickForBeta]*'-expr 'ispositive(b)*sqrt(a)-isnegative(b)*sqrt(a)'-prefix* Corr_subj01R

Since correlation coefficients range from -1 to 1, to be able to run group analysis, Fisher's Z transformation formula can be used to reduce skewness and make the sampling distribution more normal when sample size is big enough: z = (1/2) * ln((1+r)/(1-r)), where z is approximately normally distributed with mean r, and standard error $1/(n-3)^{0.5}$ (*n*: sample size).

3dcalc -a Corr_subj01R+orig -expr 'log((1+a)/(1-a))/2' -prefix Corr_subj01_Z

8. Repeat the above steps for all other subjects

9. Convert the brains of Z values to common space (e.g. +tlrc)

10. Run group analysis on the Z values of the interaction effect

--Footnotes:

1) Flipping axis of evil: RAI/DICOM and LPI/SPM
Author: Ziad Saad (---.nimh.nih.gov)
Date: 02-14-06 16:38

There has been some confusion about the coordinates output used by various AFNI programs. I hope this posting and recent modifications to AFNI programs' output will clarify matters for GOOD!

All mm coordinates are expressed in a particular order. The two most common ones are RAI/DICOM or LPI/SPM. The only difference between the two is the sign of the first two (x,y) coordinates. In AFNI's main controller, these coordinates along with their order, are specified in the top left corner. The default is RAI/DICOM.

In AFNI you can set the coordinate order to your liking with the envrionment variable AFNI_ORIENT. Programs (except 'whereami') will report coordinates in the order specified by this environment variable, unless they are specifically instructed to do otherwise. At any rate, the new version of the programs dealing with coordinates, such as whereami, 3dclust and 3dExtrema, have been modified to output a string specifying the coordinate order.

The program 'whereami' will output coordinates in the LPI/SPM order, regardless of the AFNI_ORIENT variable because the report is given in multiple coordinate systems such as Talairach, MNI, and MNI Anatomical. The latter two are almost always in LPI order in the literature so we decided to minimize confusion and report whereami's coordinate in LPI all the time.

cheers,
-ziad

VI. DATA PRESENTATION

A. 3-D Rendering with plugin 'Render Dataset'

You must select 'Accumulate' to save any of your renderings. In the image window, you'll now see more options. Click 'Disp' – this opens up the 'Display Options' window. To save a jpeg select 'Save to .jpg(s)'; select 'Sharpen' – note how that sharpens the image; experiment with the other options (getting the perfect image that you want is up to you!)

B. Visualize 2 or more BINARY activation maps in one image:

1) make binary data file for each condition: examples:
3dcalc –prefix bin-mean-ff –a mean-ff +tlrc –expr "step(a)"
3dcalc –prefix bin-mean-mf –a mean-mf+tlrc –expr "step(a)"

2) multiply second condition by 2:
3dcalc –prefix bin2-mean-mf –a bin-mean-mf+tlrc –expr "2*a"

3) combine the two activation maps:
3dcalc –prefix ff+mf –a bin-mean-ff+tlrc –b bin2-mean-mf+tlrc –expr "a+b"

4) look at color scale in 'Define Overlay'. Note that the Olay range = 0-3.

Set # of color panes in slider window to 4. 1st pane (0-0.24) = 0; 2nd pane (0.25-0.49) = 1 (first condition); 3rd pane (0.5-0.74) = 2 (second condition); 4th pane (0.75-1.00) = 3 (where the two conditions overlap)

Making a MOVIE (.mpg)

Xhairs 'Off'
Set time slider to first frame
Rec > Stay On; scroll through all frames
In Image Recorder window:
RIGHT CLICK on Save:bkg
Select 'Sav:mpeg'; click on 'Sav:mpeg'; type in filename
Select 'Start' and 'End' frames you want in the movie

Editing WAV files with Praat:

1. Read > Read from file...
2. Manipulate > To Manipulation
3. Select minimum pitch = 75 Hz
4. Click Edit – spectrogram will open up
5. Click&Drag over the part you want to edit
6. Click on the bar below the pink selection window – you should hear the original file
7. **Select Pitch > Shift pitch frequencies...**

Display structures from a TT database in AFNI:

Open a view window in AFNI – Rightclick – Select "Atlas Colors"
Scroll through list, select structure, and choose a color to highlight that structure.
Checkout: localimages/stroop/masks

VI. Running the GE 3.0T Scanner

Set-up:

- **Test Audio volume through headphones**
- **Patient Audio Volume = 58-63**
- **Make sure BNC cable from PC is connected to BNC labeled “A” going to scanner**

Step 1: Remove old P-files (either first thing or during first scan)

1) Check to see if old P-files are taking up too much space. If they are more than a few days old, delete them.
2) click on Terminal icon; right-click on blue background
3) select <service tools> then <command window> pops up

```
>cd /usr/g/mrraw
>df –lh [use this to check available space]
> ls
>ls –la
>rm P*
```

Step 2: Subject in scanner

1) push button to turn on marker light
2) advance table so marker is even with notch on head coil (around eyebrows)
3) push ‘set’

Step 3: Setup Scans

Click “Idle” icon (upper left); in ‘Rx Manager’ click <Scan Modes>

1) Make sure that scan mode = “Research”; Auto Table Movement “ON”; everything else “OFF”
   Click <Accept>
2) In “Patient Register” click <New Pt>
3) Insert coded Patient ID (e.g., S1) **Hit Return**
4) “SAR Limits” should be set to “First Level” (default) then click <Accept>
5) Enter patient weight **Hit Return**
6) Under “PATIENT PROTOCOLS” panel click <Other> and select 0.101_r_tom_stroop <Accept>
7) click on first scan (scout or localizer)
8) click <Save Series> then <Download> [note that “State” changes to ACTive in protocol list on left]
9) click <SCAN>

Prescribe scans: remaining scans can be prescribed while first is running

Next scan to prescribe is T1 (fse); click on that scan in the ‘Series Description’ list:
Press <View Edit> then, adjust slices by clicking in view window, then <Save Series>

To run T1 scan (~7min):

***NB:** Inform subject that the table might move slightly to adjust the image
Click on the T1 scan in ‘Series Description’; press <Download> then <SCAN>

Next is Spiral #1 (~13min):

1) Click on Spiral #1. You will then copy the prescription from the T1.
2) Click <View Edit> <Copy Rx> <double-click the previous scan> <Save Series>

To Trigger Scanner from E-prime:

1) Click on the correct scan in ‘Series Description’; press <Download>
2) Tell subject what comes next
3) press <PrepScan> first. A short pre-scan will run – wait for it to finish.
4) Then press <SCAN>. Make sure E-prime is loaded and ready to run and press spacebar to trigger

***NB:** After scan, wait 45 sec for spiral to write before moving it to mrisun

Next is Spiral #2 (~13min):
1) Repeat procedure as for Spiral#1
2) Click <View Edit> <Copy Rx> <double-click the previous scan> <Save Series>
To Trigger Scanner from E-prime:
1) Click on the correct scan in 'Series Description'; press <Download>
2) Tell subject what comes next
3) press <PrepScan> first. A short pre-scan will run – wait for it to finish.
4) Then press <SCAN>. Make sure E-prime is loaded and ready to run and press spacebar to trigger
***NB:** After scan, wait 45 sec for spiral to write before moving it to mrisun

Next is SAG (sagittal SPGR) (~7 min):
Click on SAG, then <View Edit>
Then adjust boxes around each view ; click <Save Series> (can't do <Copy Rx> because this is a different orientation)
To run SPGR scan:
Click on the scan in 'Series Description'; press <Download> then <SCAN>

Step 4: Transfer Files

Click on upper right icon. This opens a 'Browser Window' that lets you see the collected scans and view them.

Transfer Functional "P" files
Open a new command window. [right click the upper right icon – <Service Tools> - <Command Window>]

```
>cd /usr/g/mrraw
>ls
>ls –la (write down P-file code and file size in log book)
>ftp ultimate
        Username: christensen
        Password: tc1111tc
>cd /Stroop (or another experiment directory)
> mkidr S# (subject #)
>cd S#
>prompt [Interactive mode off.]
>mput P*
>ls –la (check to see file sizes are the same)
exit
```

```
ftp mrisun
        Username: plante
        Password: sw8rdf6sh
cd /data4/plante/tom/Stroop
```
(input IS case-sensitive)

While waiting for P-files to transfer...

series 1 (scout) = 1 space 27
series 2 (fse) = 1 space 26
series 5 (spgr) = 1 space 128

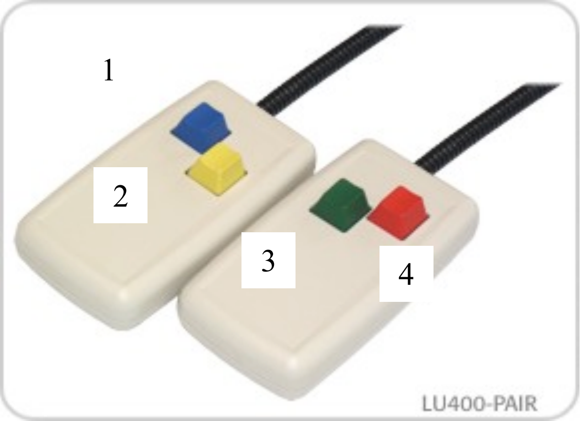
Transfer Structural "e" files:
Open a new command window. [right click the upper right icon – <Service Tools> - <Command Window>]
>type **extract** (follow directions: a=enter exam #, b=enter series # (see box above), c=enter first & last image # separated by a space). This will copy anatomical files to /usr/g/insite/tmp

```
>cd /usr/g/insite/tmp
>ftp ultimate
      Username: christensen
      Password: tc1111tc
>cd /Stroop/S#
>prompt [Interactive mode off.]
>e.g., mput e####* (where #### is the exam number from 'extract')
exit
```

```
ftp mrisun
        Username: plante
        Password: sw8rdf6sh
cd /data4/plante/tom/Stroop/S#
```

When Finished:
Press <End Exam> <Discontinue or Complete>

VII. Fun with E-Prime!



1. open **E-DataAid** and open the ***.edat** file you want to work on
2. click "Tools" then "Arrange Columns" then "<Remove All"
3. press CTRL key, then select all columns you want to view (e.g., SlideTarget.ACC (accuracy); SlideTarget.CRESP (correct resp); SlideTarget.RESP (subject's resp); SoundFile; SlideTarget.RT (time from stim onset to button press)

NB: If the null events do not require a response, E-Prime will count them as mistakes (SlideTarget.ACC=0). Select the <Filter> menu, then "SlideTarget.CRESP" Under <Checklist> click the boxes "2" and "3" (notice there is a "blank" category because E-Prime saw no response given during the null events). It is important to de-select the null responses if you are calculating overall accuracy figures because it will give you many false negatives.

4. Click <Edit> <Add Variable> and type a variable name; then select level as "Trial". Leave variable type as "String"
5. Paste in the column of Corrected Onset Times (p. 19)
6. To select responses for only one category of stimulus:
Click <Tools> <Filter> <SpeakerGender> <Checklist> select whichever subcategory you'd like to see; do the same for <WordGender>. Once you've done this, you will see that now you've "filtered" only the 30 stimuli that correspond to this particular combination of variables.
7. Move the two columns of data to Excel
8. For SlideTarget.RT, convert milliseconds to seconds (=A1/1000)
9. Add the 2 values across each row: this column of 30 numbers for each stimulus will be saved as the text file used in 3dDeconvolve.
10. Repeat for each STIMULUS CATEGORY and for each RUN

Now you can analyze the response stats:

1. Click <Tools> <Analyze> <Load Analysis>
2. Click "ACCxgender.anl" <open> <Run>

Wrong Answers

Right Answers

EXAMPLE OUTPUT:

SlideTarget.RT:Mean, SlideTarget.RT:StdErrS, SlideTarget.ACC:CountNum by SpeakerGender, WordGender, Noise and SlideTarget.ACC

SpeakerGender	WordGender	Noise	Stats	0	1
female	female	no	CountNum SlideTarget.ACC		30.00
female	female	no	Mean SlideTarget.RT		857.63
female	female	no	StdErrS SlideTarget.RT		45.08
female	male	no	CountNum SlideTarget.ACC	2.00	28.00
female	male	no	Mean SlideTarget.RT	1192.50	782.82
female	male	no	StdErrS SlideTarget.RT	183.50	24.63
female	neutral	no	CountNum SlideTarget.ACC		30.00
female	neutral	no	Mean SlideTarget.RT		785.87
female	neutral	no	StdErrS SlideTarget.RT		30.15
male	female	no	CountNum SlideTarget.ACC		30.00
male	female	no	Mean SlideTarget.RT		910.57
male	female	no	StdErrS SlideTarget.RT		46.07
male	male	no	CountNum SlideTarget.ACC	1.00	29.00
male	male	no	Mean SlideTarget.RT	1151.00	849.31
male	male	no	StdErrS SlideTarget.RT	N/A	46.92
male	neutral	no	CountNum SlideTarget.ACC		30.00
male	neutral	no	Mean SlideTarget.RT		833.67
male	neutral	no	StdErrS SlideTarget.RT		47.86
null	null	null	CountNum SlideTarget.ACC	31.00	
null	null	null	Mean SlideTarget.RT	0.00	
null	null	null	StdErrS SlideTarget.RT	0.00	

Accuracy Cou

Mean RT

± SEM

3. Click "Excel Copy..." to move to Excel

APPENDIX – Things that could come in handy someday...

Step IV -
D. De-trend images with '3dTshift' and the '-rlt+' option

***FYI*: If in some experiment you need to include the '–tpattern' option, for this to work correctly your timing file ('offsets.1D' or equivalent) must be converted to SECONDS (don't ask why – I have no idea)! According to an AFNI tech (Rick Reynolds - 9/26/07), if "the slice timing in the dataset is accurate, then you don't need to provide any timing information on the 3dTshift command line." i.e., you don't need the '-tpattern *filename*' option anyway (see help files for more info on 3dTshift).**

E. Register functional images (perform movement correction) with '3dvolreg'

- **Or from the AFNI window (note: I haven't figured out how to limit the correction to just slices 4-213 using the plugin; we could try using 'S1-run1-shift+orig[4-213]' as input):**

Register last run of set.

1. afni
2. [In pop-up window]

a. select Define Data Mode button {to your right}
b. select Plugins button {to the bottom right}
c. select 3D Registration button

1. select 'Choose Dataset' and choose one (e.g. S1-run1-shift+orig)
2. set 'Base Brick' to a timepoint not noted as an outlier (from 'to3d' output)
3. click 'Outputs'; type in an Output name (e.g., S1-run1-reg)
4. type in an output 1D file name (e.g., S1-run1-reg.1D)
5. leave 'dfile' blank
6. click graph=yes
7. select Run+Close or Run+Keep
8. save graph as a ps file

Align remaining runs.

1. afni
2. [In pop-up window]

a. select Define Data Mode button {to your right}
b. select Plugins button {to the bottom right}
c. select 3D Registration button

1. select Choose Data Set and choose one (e.g. S1-run2-shift+orig)
2. set 'Base Brick' to a timepoint not noted as an outlier (from to3d output)
3. type in an Output name (e.g., S1-run2-reg)
4. select Basis button {to your left}
5. select registered version of last run (e.g., S1-run1-reg)
6. type in an Output name (e.g., S1-run1-reg)
7. type in an output 1D file name (e.g., S1-run2-reg.1D)
8. leave 'dfile' blank
9. click graph=yes
10. select Run+Close or Run+Keep
11. save graph as a ps file

G. Smooth (blur) each 3D+time dataset with '3dmerge' (all runs)

```
N.B.: The 3 '-1blur' options just provide different ways of
      specifying the radius used for the blurring function.
```

```
        The relationships among these specifications are
           sigma = 0.57735027 * rms = 0.42466090 * fwhm
        The requisite convolutions are done using FFTs; this is by
        far the slowest operation among the editing options.
```

Note on masking from Joe P:
Author: Joe Paxton (---.client.fas.harvard.edu)
Date: 01-12-08 16:38
The AFNI program 3dAutomask can create a mask given an EPI (or spiral) 3D+time dataset. The command you'll need will be something like this: 3dAutomask -dilate 1 -prefix outputFile inputFiles

Then, assuming you're doing a group analysis, you'll likely want to combine that mask with other such masks in order to create a union mask. To do so, you could use 3dMean and 3dcalc:

```
3dMean -prefix outputFile -sum inputFiles
3dcalc -prefix outputFile -a inputFileFrom3dMean -expr 'ispositive(a)'
```

J. Run deconvolution (GLM) analysis for each subject with AFNI '3dDeconvolve'

Deconvolve the functional image file obtained with a BLOCK DESIGN

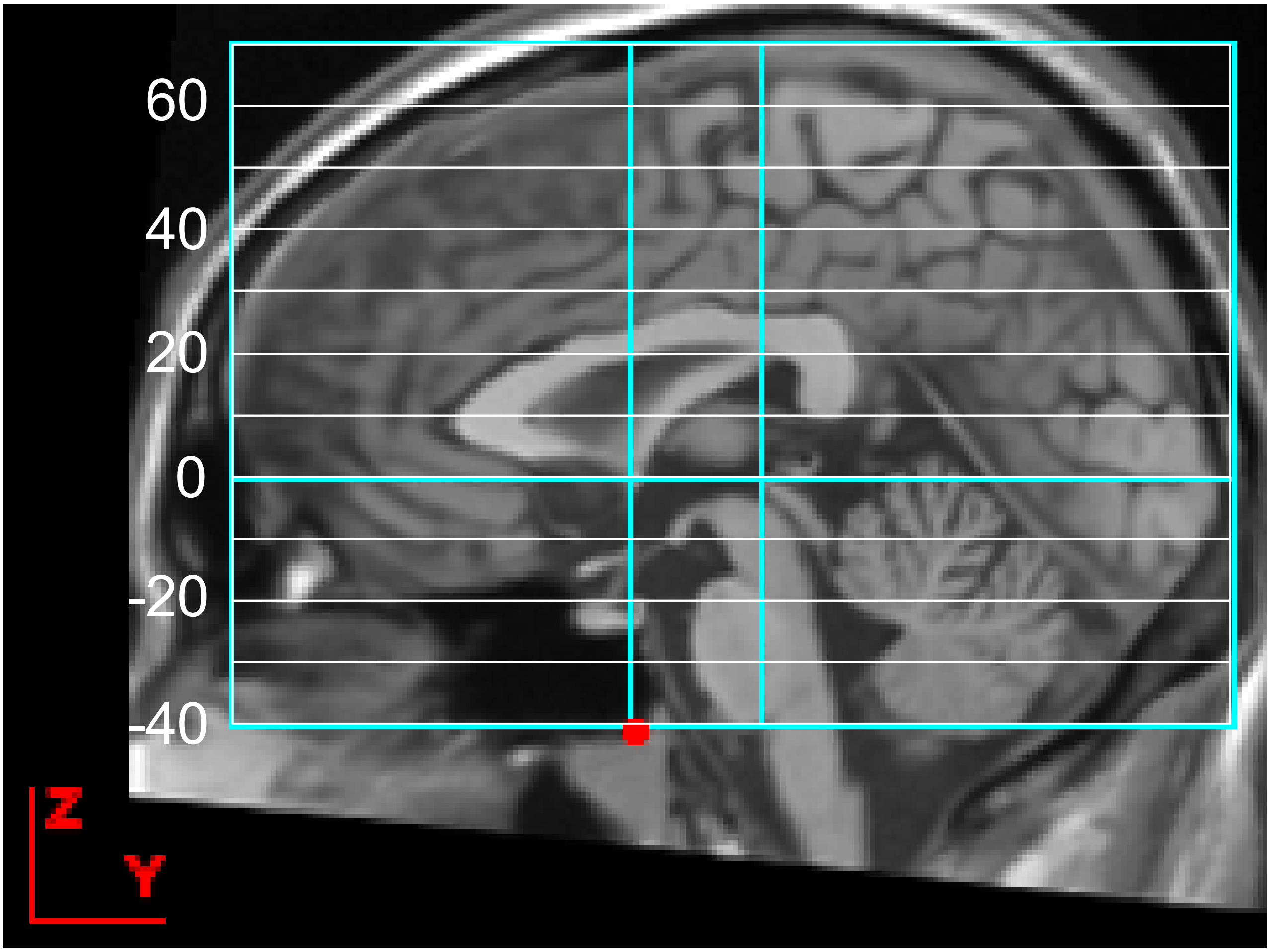
1. In a text editor, type (all in one line or use the '\' limiter at the end of each line)
3dDeconvolve -input *S#-cond-cat+orig* -concat concat.1D -mask *S#-mask+orig* -nlast ### -num_stimts # -stim_file 1 *event1*.1D -stim_file 2 *event2*.1D -stim_maxlag 1 8 -stim_maxlag 2 8 -stim_nptr 1 # -stim_nptr 2 # -tshift -iresp 1 *S#-event1* -iresp 2 S#-*event2* -bucket S1-*condition*-fbuc -rout -fout

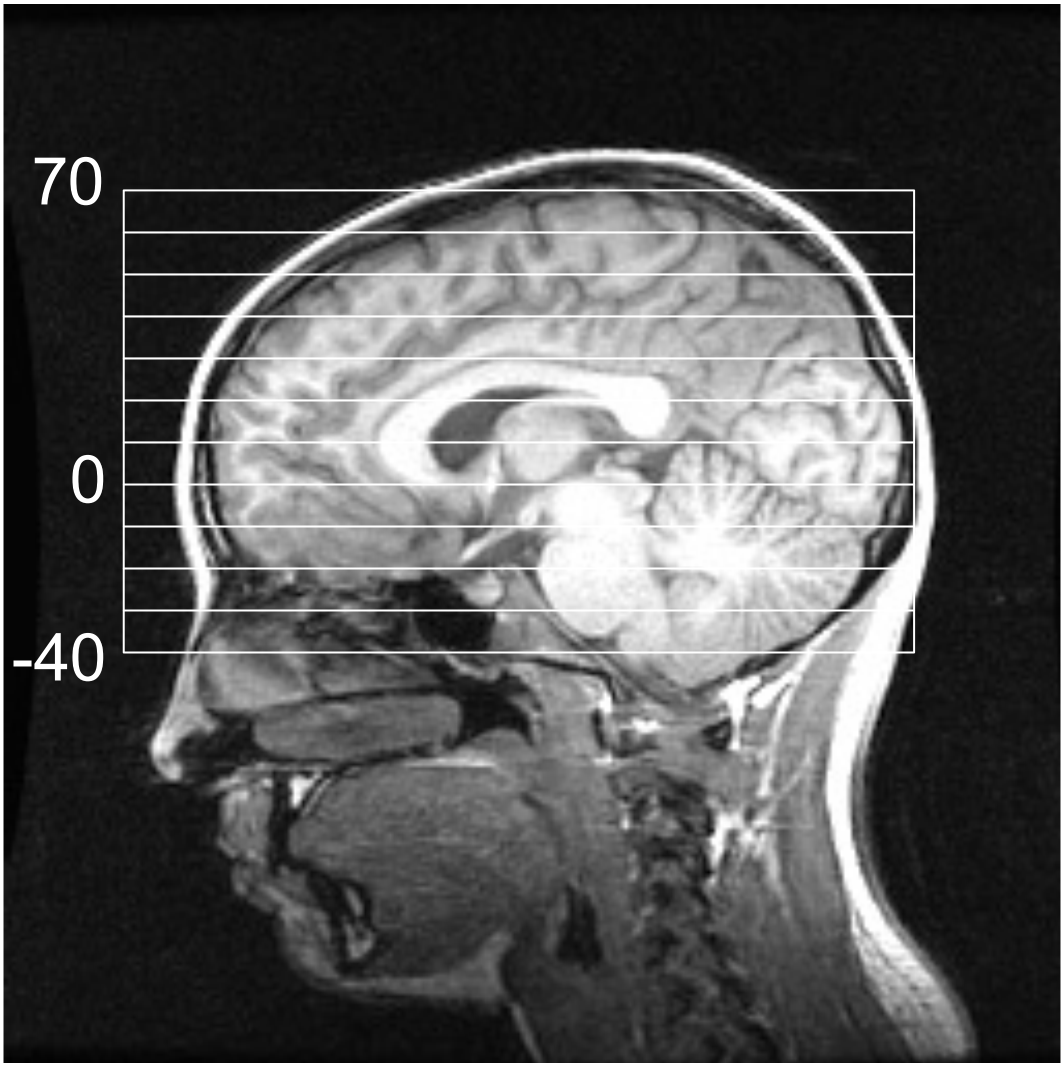
-nlast # = last time point to be analyzed
-num_stimts # = number of event types to be analyzed.
-stim_file 1 = first .1D file corresponding to event 1
-stim_file 2 = second .1D file corresponding to event 2 [repeat for all events]
-stim_maxlag 1 # = the maximum time lag for estimating the BOLD response for event 1
-stim_maxlag 2 # = the maximum time lag for estimating the BOLD response for event 2
-stim_nptr 1 # = how many units are each TR divided into in terms of where stimuli occur for event 1
-stim_nptr 2 # = how many units are each TR divided into in terms of where stimuli occur for event 2
-iresp= file containing the impulse response function (the BOLD response) data
-rout calculates the partial R^2 data
-fout asks for the partial F-stat data per event

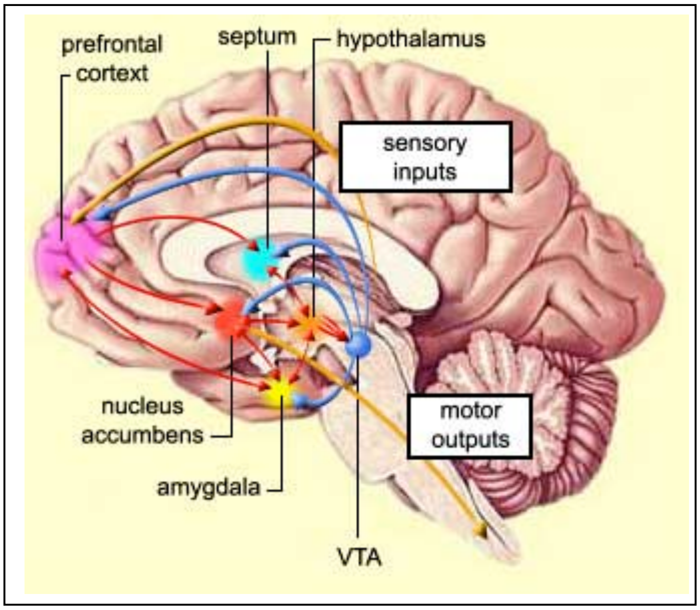
(e.g., **3dDeconvolve -input S1-cond-cat+orig -concat concat.1D -mask S1-mask+orig -nlast 456 -num_stimts 2 -stim_file 1 verbs1.1D -stim_file 2 nouns.1D -stim_maxlag 1 32 -stim_maxlag 2 32 -stim_nptr 1 8 -stim_nptr 2 8 -tshift -iresp 1 S1-verbs -iresp 2 S1-nouns -bucket S1-all_words-fbuc -rout -fout**

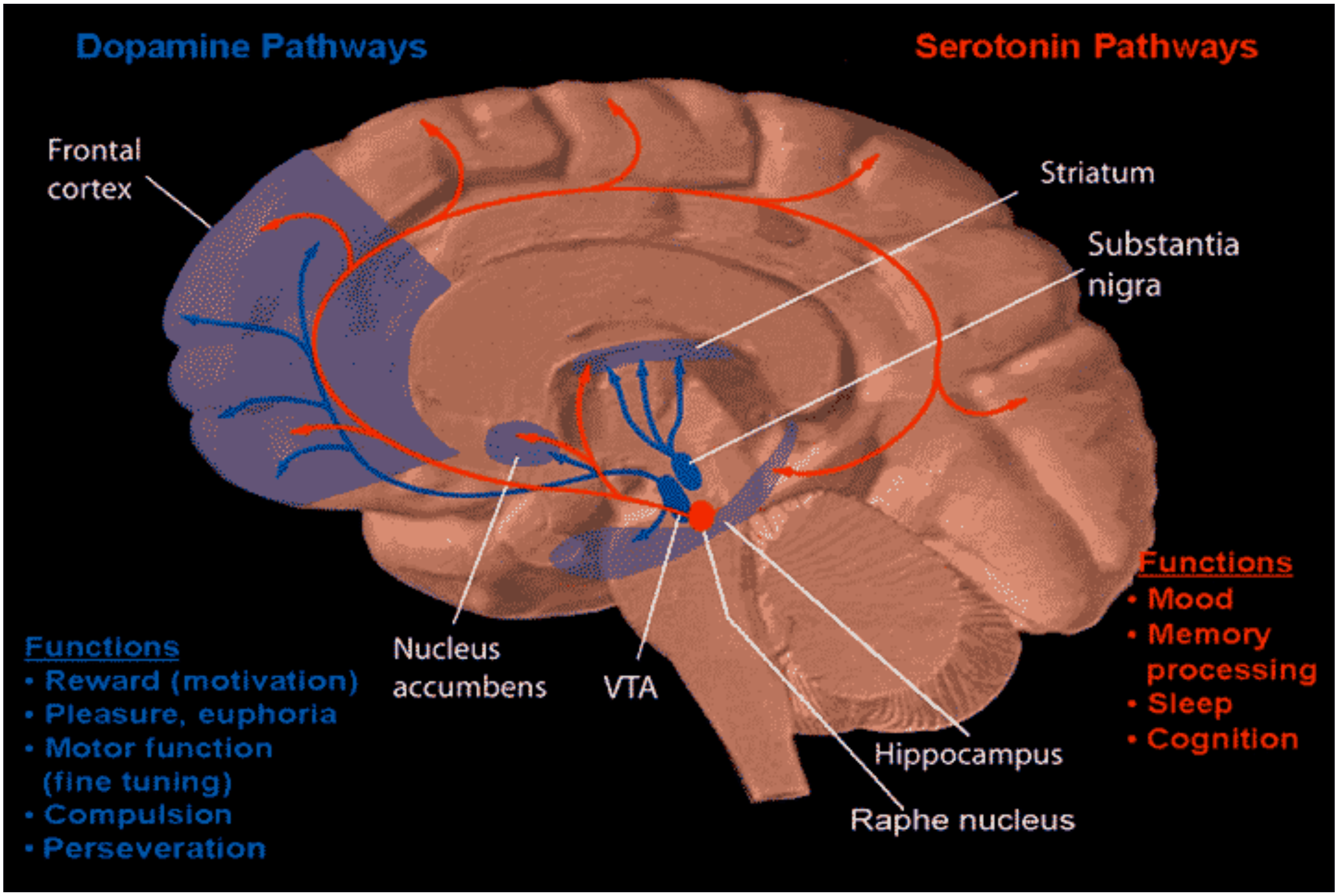
2. Save as Deconvolve.prog in a programs subdirectory
3. Change permissions (chmod 775 *.prog)
4. copy into directory where needed
3. execute by typing ./Deconvolve.prog at the command line

Talairach Grid:




70
0
-40


prefrontal cortext
septum
hypothalamus
sensory inputs
nucleus accumbens
amygdala
VTA
motor outputs



http://www.drugabuse.gov/pubs/teaching/largegifs/slide-2.gif

MAKING FUNCTIONAL IMAGE (FIM) FILES:

FIM FILES ARE USEFUL FOR PILOT DATA OR A QUICK CHECK ON AN INDIVIDUAL FILE. THEY ARE ONLY USEFUL FOR **BLOCK DESIGN** STUDIES.

NOTE: IF THE STUDY USES AN **EVENT DESIGN**, THEN USE **3dDECONVOLVE**. You are advised to read the **3dDeconvolve** manual pages to determine whether additional options would be advisable.

MAKE FIM or FITT FILES
NB: you generally will need two types of activation files: clustered files and fim files. Making other types of files is also explained.

To correlate w/ multiple time series & request multiple forms of output (usual case):

NB: you will need to use the single wave file with multiple columns of wave data in it.
NB: if you want %change, you had to use rlt+ at the 3dTcat stage.
NB: all -out features are optional

3dfim+ -input *inputfile*+orig -ideal_file *ideal_wave.*1D -ort_file *orthoganal.*1D -out ‘FIT Coef’ -out ‘Best Index’ -out ‘% Change’ -out Baseline -bucket *output file* e.g., 3dfim+ -input S1-cond1-dt -ideal cond1.1D -ort S1-reg.1D -out ‘FIT Coef’ -out ‘Best Index’ -out ‘% change’ -out Baseline -bucket S1-cond1-fim

Warp files to standard space

[NB: must exit program to refresh file list before trying to warp functional files to the spgr talairach version. You must have T1 version up to warp.
Note: may want to use 3ddup to duplicate files before altering]

Change directory to preprocessing (prep) subdirectory
Copy all *spgr* files to fim subdirectory
(e.g., cp *spgr* ../fim)

1. afni
2. Select anatomy (spgr) and functional files to be rendered
   NB: must transform them together or they will not match up later
   You may need to select Talairach View button (top middle) to see +tlrc versions
3. push “define data mode” button (new panel will open)
   a. click “warp on demand” buttons for anat and func
   b. change anat resam mode to Li
   c. change func resam mode to Bk
   d. click on write “many” button and highlight all to write out (all will be +tlrc)
      or click on write “anat” and “func” buttons (does one at a time)
4. exit afni

MAKING SIMPLE STIMULUS-RESPONSE "IDEAL WAVE" FILES

NB: Different kinds of wave files are made for block and event designs. Please follow the relevant instructions. Wave files need to be created only once per experiment; after that they are saved and reused. They may be modified for each subject to eliminate bad timepoints if 3dDespike was not used. Choose from the methods below to fit your later analysis purpose. Also note, these WAVE files are simply a string of '0s' and '1s' that mark the time (TR) of each event. In newer versions of AFNI it is possible to instead use the actual times of stimulus events or even response times (see p. 18 - "Make Timing Files").

- **EVENT DESIGNS**

NB: The simplest thing to do is to make one '.1D' file for each event you are going to analyze.

If the events start concurrently with a TR, then use the following:

NB: the timepoint on each line could represent a TR or a fraction of a TR. For example, for spaced acquisition, stimuli may occur at fractions of TRs!

A. Using a text editor, you will be putting one number for each line of text

1. Starting with the first time point
   a) put a 1 on the line if the target event began at that timepoint
   b) put a 0 on the line if no event or an alternate event began at that time point
2. Repeat for each event file needed.

If events DO NOT start concurrently with a TR (e.g., spaced acquisition), use the following:

A. Using a text editor, you will be putting one number for each line of text

1. Type the time in seconds that an event of interest occurs on each line
2. Repeat for each event file as needed.

e.g., 3.2
12.3
19.6…

- **BLOCK DESIGNS**

A. **Make Square Waves**

1. sqwave -on *#on* -off *#off* -onkill *cue#* -offkill *cue#* -length *timepoints* -name *name.1D*

(e.g., sqwave -on 11 -off 11 –onkill 1 -offkill 1 -length 80 -name *name.1D)*

a. *#on* = length of the "on" period (i.e., images per on cycle; the number of 10s in the square wave, e.g., 10 calculated from 80 images, 8 on+off periods=10 images per period).

b. *#off* = number of off cycles (see a. above)

c. *#cue*= number of time points for cue

d. *timepointss* is taken from the log book

e. *name.1D* is the square wave file name (must use *.1D form)

B. Make Hemodynamic Waves

1. waver -dt # -delaytime # -input *file.1D* > *output.1D*

(e.g., waver –dt 3 –delaytime 0.5 –input sent.1D > sent05.1D)

a. -dt is the TR time in seconds

b. -input file is the square wave file made with sqwave command above

c. -delaytime is lag between onset of stimuli and start of hemodynamic response. Do delays of 0-3 seconds in .5 increments

2. Bring up waveform in text editor
    a. under view, click "line status"
    b. edit to correct number of lines (= #timepoints)

NB: Default settings (for the 'Cox special' waveform):

1. –undershoot = .2 sec
2. –restoretime= 2 sec
3. –risetime= 4 sec
4. –falltime= 6 sec
5. –delaytime (0-3second in .5 sec intervals)

C. MAKE A SINGLE WAVE FILE FOR USE WITH 3DFIM+ (a program to calculate the simple cross-correlation of an ideal reference waveform with the measured FMRI time series for each voxel).

NB: wave files must run in consecutive columns within a single file

1. transfer files to a pc
2. cut and paste columns in Excel
3. save as a text file
4. transfer back to workstation

Brain stripping

1. Must have a +tlrc view of the brain made
2. from command line:
    a. 3dIntracranial -prefix *filename* -anat *filename+tlrc*
       e.g., 3dintercranial -prefix S1-bare -anat S1-spgr+tlrc

Make Invisible spots on structural 3d for activations to show through

1. strip brain (see above)
2. check to see how bright the voxels are in the stripped brain
    a. afni
    b. Define DataMode
    c. Plugins
    d. Misc
    e. Anat Info (note: must have stripped brain selected under Switch Anatomy)
       read voxel intensities (at bottom)
3. write down a constant that is a little brighter than brightest voxel intensity
4. create a dummy brain that is a binary version of the activation to be displayed
    a. 3dcalc -prefix foo -a *activation-file+tlrc* -expr “step(a)”
    e.g. 3dcalc -prefix foo -a Total-ap+tlrc -expr “step(a)”
5. multiply dummy by constant (step 3 above) and add result to stripped brain
    a. 3dcalc -prefix *display-anat* -a *stripped-brain+tlrc* -b foo+tlrc -expr “(b**constant*)+a”
    e.g. 3dcalc -prefix Cond1-anat -a S1-stripped+tlrc -b foo+tlrc -expr “(b*200)+a”

3D Render

NB: If the original view is <u>not</u> an axial, will need to axialize images
1. 3daxialize –prefix *filename*-ax *filename*+orig
   (e.g., 3daxialize –prefix NL1-spgr-ax NL1-spgr+orig)

Rendering:
1. afni
2. push “Define Datamode” (new panel will open)
2. push “plugins”
    a. select “Renderer” (new panel will open)
    b. click “choose underlay dataset” and choose one
    c. click “choose overlay dataset” and choose one
    d. click “draw” to do 3d image
    e. “cutouts” allows you to slice off sections
        1. click how many cutouts you want
        2. define them below (will pop up)
        3. use “+or” toggle between “and” “or” slice combos
    f. use “roll”, “pitch”, & “yaw” to adjust angle and tilt of image
    g. “accumulate” accumulates successive draws (need to use to save an image)
    f. change quality of image with “brightness” and “opacity” windows
    g. under “functional overlay” window
        1. click “see overlay”
        2. when “cut out” is on, just see at the current slice

3. “remove cluster” for clean-up

Other Useful Stuff:

3dcalc

3dcalc -verbose -a *filename-dt1+orig* -b *filename-dt2+orig* -expr “(a/b)*10 -prefix *filename-norm*

3ddup

3ddup –prefix *filename filename*+orig
Note: only changes header, need to copy duplicate header + original BRIK together to other subdirectories.

3dinfo

3dinfo *filename+orig*
prints header info for a file

3drefit

3drefit –xorigin # -yorigin # -zorigin # -xdel # -ydel # -zdel #
where –xorigin etc are the values for the corners of the x, y, z dimensions
-xdel etc are the pixel dimensions
type 3drefit for other options

AFNI Manual

September 2008