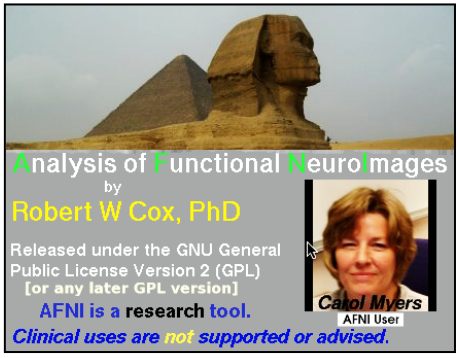
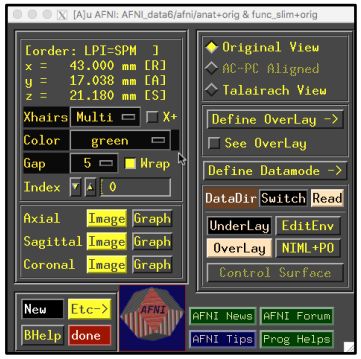
AFNI

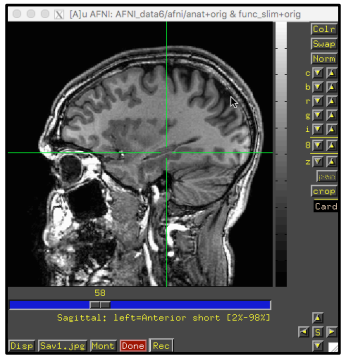
Didactics and Demonstrations

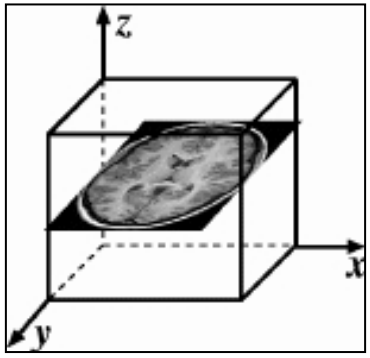
Introduction, Concepts, Principles

March 2020









Web Page to Bookmark

https://afni.nimh.nih.gov/pub/dist/doc/htmldoc/index.html

== https://bit.ly/AFNIstuff

- **Links to AFNI Installation instructions, Documentation, Data, Tips, *et cetera***
- **To use hands-on videos: install AFNI and then the Prep for Bootcamp link**

AFNI = Analysis of Functional NeuroImages

- Created to provide an environment for FMRI data analyses
- AFNI refers to both the program of that name and the entire package of external programs and plugins (several hundred)
- Important principles in the development of **AFNI**:
  - Allow user (you) to stay close to the data and view it in different ways
  - Give users the power to assemble computing pieces in different ways to make customized analyses
    - ***With great power comes great responsibility***
      — **to *understand* the analyses and the tools**
  - "Provide mechanism, not policy"
  - Allow other programmers to add features/components

Principles (and Caveats) We Live By

- Fix significant bugs as soon as possible
  - But, we define "significant"
- Nothing is secret or hidden (**AFNI** is open source)
  - But, possibly not very well documented or advertised
- Release early and often
  - So, All users are beta-testers for life
- Help the user (message board; consulting with NIH users)
  - Until our patience expires
- Try to anticipate users' future needs
  - What we *think* you need may not be what you *really* need

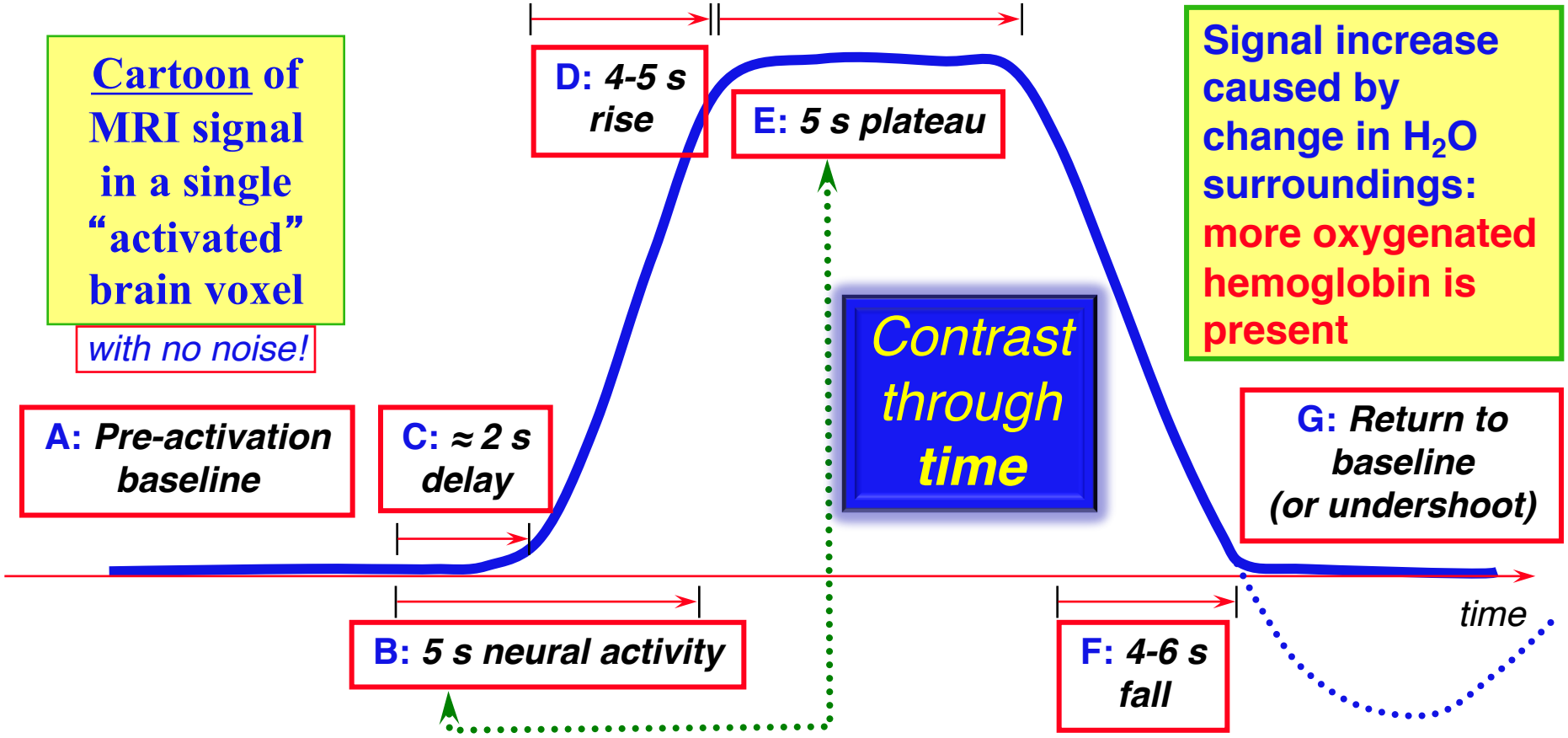
Before We Really Start

- **AFNI** has many programs and each has many options
- Assembling the programs to do something useful *and* good seems confusing (OK, ***is*** confusing) when you start
- To help overcome this problem, we have "super-scripts" that carry out important tasks
  - Each script runs multiple **AFNI** programs
  - **We recommend** using these as the basis for FMRI work
- `afni_proc.py` = Single subject FMRI pre-processing and time series analysis for functional activation
- `align_epi_anat.py` = Image alignment (registration), including anatomical-EPI, anatomical-anatomical, EPI-EPI, and alignment to atlas space (Talairach/MNI)

A

What is Functional MRI?

- 1991 discovery: MRI-measurable signal increases few % *locally* in brain after increases in neuronal activity (Kwong, *et al.*)

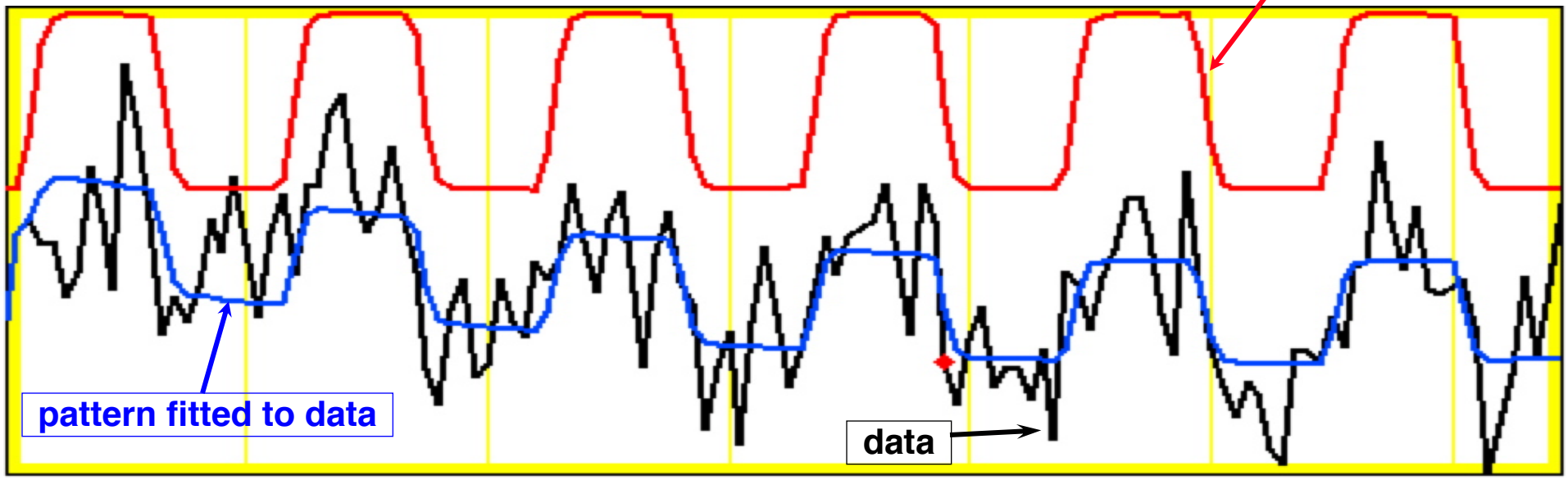


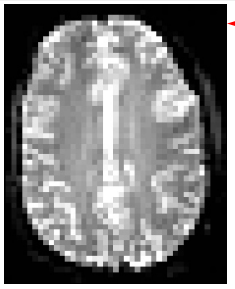
How FMRI Experiments Are Done

- Alternate subject's neural state between 2 (or more) conditions using sensory stimuli, tasks to perform, ...
  - Can only measure relative signals, so must look for ***changes*** in the signal between the conditions
- Acquire lots of MR images repeatedly during this process
- Search for voxels whose NMR signal time series (up-and-down) matches stimulus time series pattern (on-and-off)
  - FMRI data analysis is basically pattern matching ***in time***
- MRI signal changes due to neural activity are small
  - Need 500 or so images in time series (in each slice) ➔ takes 30 min or so to get reliable activation maps
  - Other small effects can corrupt the results ➔ post-process the data to reduce these effects & ***be vigilant***

Sample Data Time Series

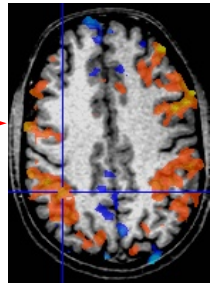
- 64 × 64 matrix (TR=2.5 s; 130 time points per imaging run)
- Somatosensory task: **27 s "on", 27 s "rest"**
- Note that this is ***really*** good data





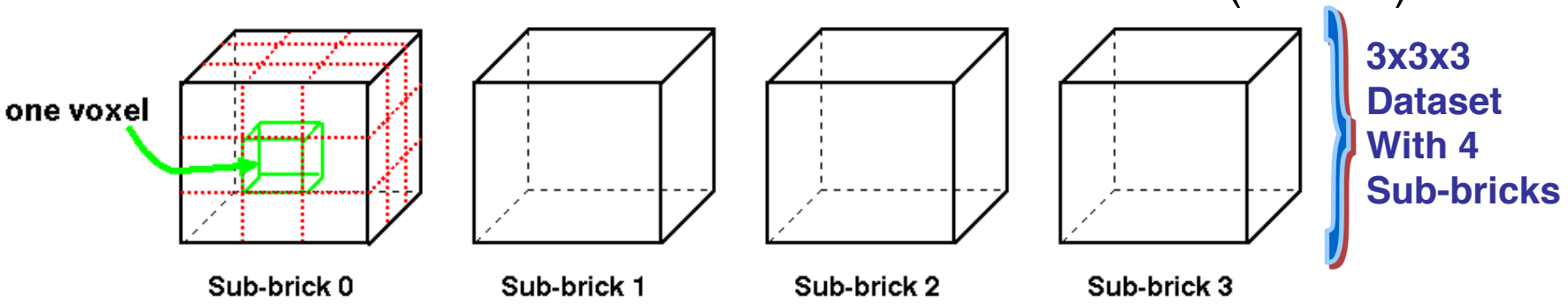
One echo-planar image

One anatomical image, with voxels that match the pattern given a color overlay

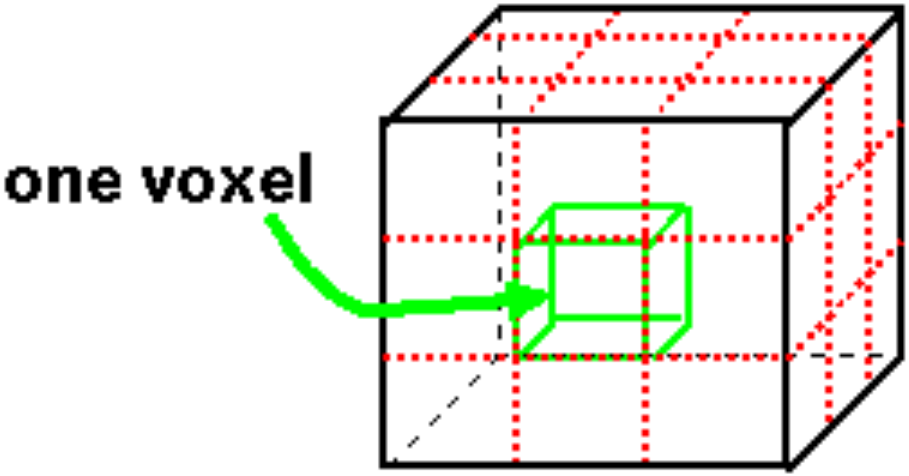


Fundamental **AFNI** Concepts

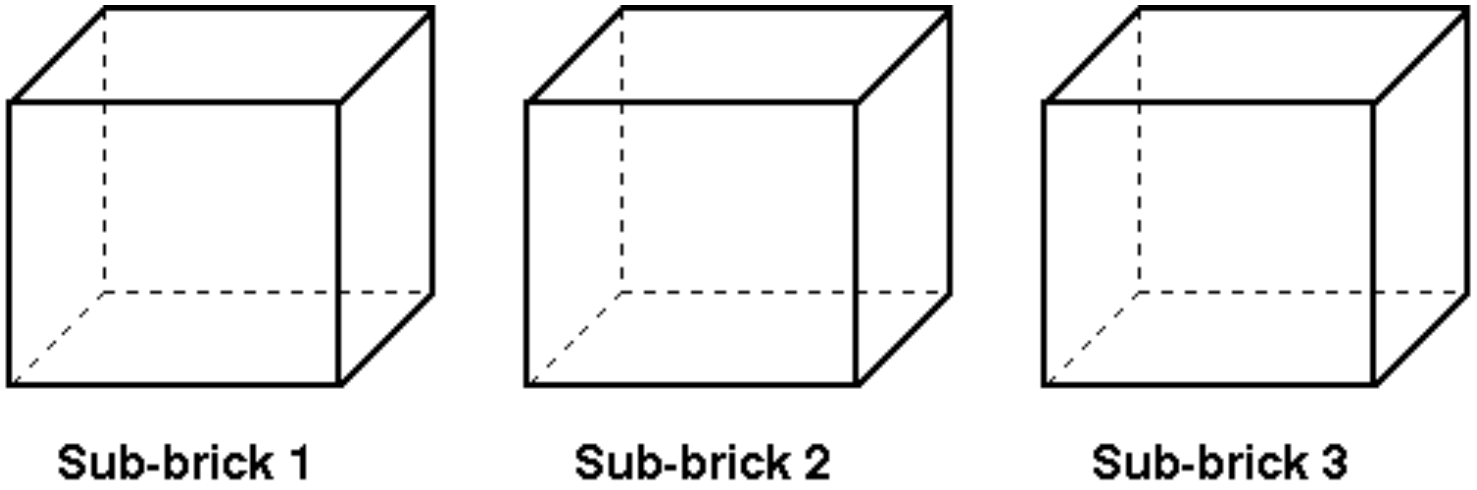
- Basic unit of data in **AFNI** is the dataset ← *Jargon!*
  - A collection of 1 or more 3D arrays of numbers
    - Each entry in array is in a particular spatial location in a 3D grid
    - Image datasets: each array holds collection of slices from scanner
      - Each number is the signal intensity for that particular voxel
    - Derived datasets: each number is computed from other dataset(s)
      - e.g., each voxel value is a *t*-statistic reporting "activation" significance from an FMRI time series dataset, for that voxel
  - Each 3D array in a dataset is called a sub-brick ← *Jargon!*
    - There is one number in each voxel in each sub-brick (volume)



Dataset Cartoon: A Little Bit Bigger



"sub-brick" = one 3D volume inside a multi-volume dataset

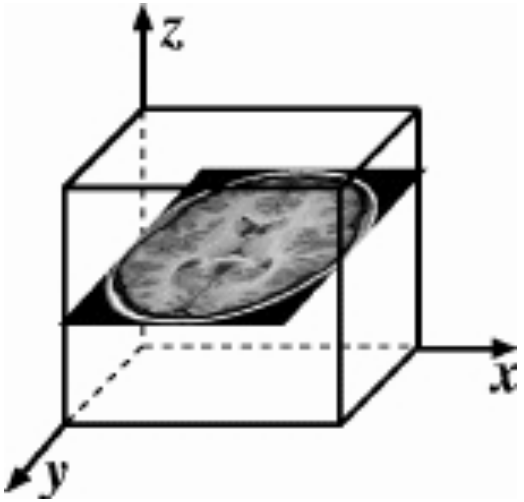


What's in a Dataset: Header Stuff

- Besides the voxel numerical values, a dataset also contains auxiliary information, including (some of which is optional):
  - *xyz* dimensions of each voxel (in mm)
  - Orientation of dataset axes;

    for example, *x*-axis=R-L, *y*-axis=A-P, *z*-axis=I-S

    = axial slices (we call this orientation "RAI")
  - Location of dataset in scanner coordinates
    - Needed to overlay one dataset onto another
    - Important to get right in FMRI, since we deal with many datasets
  - Time (s) between sub-bricks, for 3D+time datasets ← *Jargon!*
    - Such datasets are basic unit of FMRI data (one per imaging run)
  - Statistical parameters associated with each sub-brick
    - e.g., a *t*-statistic sub-brick has degrees-of-freedom parameter stored



AFNI Formatted Dataset Files - 1

- **AFNI formatted** datasets are stored in 2 files
  - The .HEAD file holds all the auxiliary information (ASCII)
  - The .BRIK file holds all the numbers in all the sub-bricks
- Datasets can be in one of 2 coordinate systems ("views")
  - Original data or +orig view = from the scanner
  - "Talairach" or +tlrc view =
    - Dataset has been rescaled to conform to the Talairach-Tournoux atlas dimensions *or* another atlas, such as MNI
    - AKA Stererotaxic coordinates
    - **All** datasets scaled+aligned to some atlas are labeled +tlrc
      - Header file holds name of actual atlas "space" (*e.g.*, "MNI")
      - Alignment can be *linear* or *nonlinear* (`3dQwarp` program)

AFNI Formatted Dataset Files - 2

- **AFNI** dataset filenames consist of 3 parts
  - The user-selected prefix (almost anything)
  - The view (one of +orig, or +tlrc)
  - The suffix (one of .HEAD or .BRIK)
  - **TonyFauci_epi+tlrc.HEAD** & **TonyFauci_epi+tlrc.BRIK**
  - You supply the prefix; the **AFNI** program supplies the rest

- **AFNI** programs can ***read*** datasets stored in several formats
  - ANALYZE (.hdr/.img file pairs); very old nowadays
  - MINC-1 (.mnc); i.e., from mnitools (also very old now)
  - CTF (.mri, .svl) MEG analysis volumes
  - ASCII text (.1D) — text numbers arranged into columns
  - Have conversion programs to write out MINC-1, ANALYZE, ASCII, and NIfTI-1.1 files, if desired

NIfTI Dataset Files

- NIfTI-1 (.nii or .nii.gz) is a standard format that **AFNI**, SPM, FSL, FreeSurfer, BrainVoyager, et al., agreed upon
  - Goal: easier interoperability of tools from various packages
- All data is stored in one file (*cf.* **http://nifti.nimh.nih.gov/**)
  - 352 byte header (extensions allowed; **AFNI uses this feature**)
  - Followed by the image binary numerical values
  - Allows 1D–5D datasets of diverse numerical types
  - .nii.gz suffix means file is compressed (using Unix program gzip)
- **AFNI** reads and writes NIfTI-1 (and NIfTI-2) datasets
  - **To write**: when you give the prefix for the output filename, end it in "**.nii**" or "**.nii.gz**", and **AFNI** programs will automatically write NIfTI-1.1 format instead of .HEAD/.BRIK
  - **To read**: just give the full filename ending in ".nii" or ".nii.gz"

Creating Datasets from DICOM Files

- Program 1: Rick Reynolds' **AFNI** program **Dimon**
  - Was originally created for sending image data directly into **AFNI** for "realtime FMRI"
- Program 2: Chris Rorden's **dcm2niix_afni**
  - Can create a whole collection of datasets
  - Works with more DICOM formats than **Dimon** does
  - *Problem*: Standard NIFTI .nii format can't store complicated slice timings
    - So programs like **dcm2niix_afni** cannot store this information even if the program can find it in the DICOM files
- Solution: use **3drefit** to add the slice timing information to the header (inside **AFNI** extension for NIFTI .nii files)

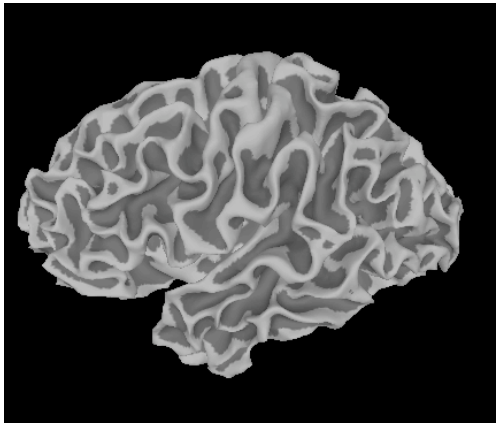
Dataset Directories

- Datasets are stored in directories (AKA “folders”)
  - All the datasets in the same directory, in the same view, are presumed to be aligned in *xyz*-coordinates
    - Voxels with same value of (*x*,*y*,*z*) correspond to same brain location
  - Can **overlay** (in color) any one dataset on top of any other one dataset (**underlay** in grayscale) from same directory
    - Even if voxel sizes and spatial orientations differ
    - Overlay of one dataset upon another is based on *xyz* coordinates
  - Typical **AFNI** contents of a directory are all data derived from a single scanning session for one subject
    - Anatomical reference (T1-weighted SPGR or MP-RAGE volume)
    - 10-20 3D+time datasets from FMRI EPI functional runs
    - Statistical datasets computed from 3D+time datasets
    - Datasets transformed from +orig to +tlrc coordinates

Getting and Installing **AFNI**

- **AFNI** runs on Unix systems: Linux, Sun, Mac OS X
  - Can also run under Windows Subsystem for Linux
    - Requires also installing X11 (Unix graphics display software)
- You can download precompiled binaries from our Website
  - **https://afni.nimh.nih.gov/pub/dist/doc/htmldoc/index.html**
  - Documentation, message board, data, class materials,…
- You can download source code and compile it
  - And from GitHub: **https://github.com/afni/AFNI**
- **AFNI** is updated fairly frequently, so it is important to update occasionally = **@update.afni.binaries**
  - It's hard to help you with outdated versions***!***
  - ***Please check for updates every 6 months (or less)***

SUMA, et alii



- **SUMA** is the **AFNI** surface mapper
  - For displaying surface models of cortex
    - Surfaces from **FreeSurfer** (MGH) *et cetera*
  - Interactively display functional activations mapped from 3D volumes to the cortical surface representation
  - Draw ROIs directly on the cortical surface
    - vs. **AFNI**: ROIs are drawn into the 3D volume
- SUMA is separate from **AFNI**, but can "talk" with **AFNI**
  - Click in **AFNI** or SUMA to change focus point, and the other program jumps to that location at the same time
  - Functional (color) overlay in **AFNI** can be sent to SUMA for simultaneous display
- And much more — stayed tuned for the SUMA talks to come!

That's All for Now