AFNI & FMRI

Introduction, Concepts, Principles

Analysis of Functional NeuroImages

by

Robert W Cox, PhD

Released under the GNU General Public License Version 2 (GPL)
[or any later GPL version]

AFNI is a research tool.

Clinical uses are not supported or advised.

http://afni.nimh.nih.gov/afni

AFNI = Analysis of Functional NeuroImages

- Developed to provide an environment for FMRI data analyses
  - And a platform for development of new software
- **AFNI** refers to both the program of that name and the entire package of external programs and plugins (more than 200)
- Important principles in the development of AFNI:
  - Allow user to stay close to the data and view it in many different ways
  - Give users the power to assemble 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 that can interact with the rest of the package

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
  - 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 will need may not be what you actually end up needing

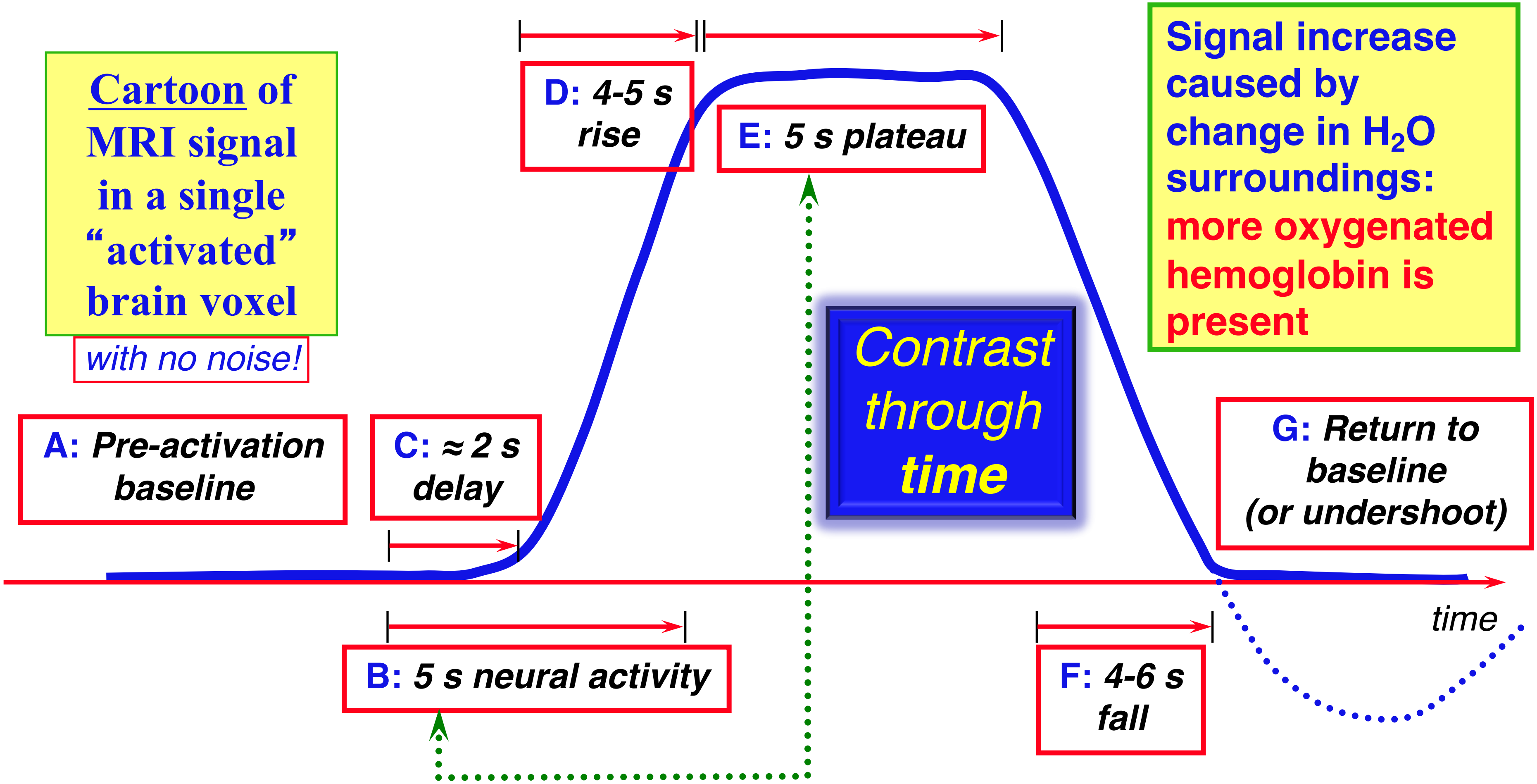
* SSCC NIMH

Before We Really Start

- AFNI has many programs and they have 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
    - When you need help, it will make things simpler for us *and* for you if you are using these scripts
- **`afni_proc.py`** = Single subject FMRI pre-processing and time series analysis for functional activation
  - **`uber_subject.py`** = GUI for **`afni_proc.py`**
- **`align_epi_anat.py`** = Image alignment (registration), including anatomical-EPI, anatomical-anatomical, EPI-EPI, and alignment to atlas space (Talairach/MNI)

What is Functional MRI?

- 1991: Discovery that MRI-measurable signal increases a few % *locally* in the brain subsequent to increases in neuronal activity (Kwong, *et al*.)



Cartoon of MRI signal in a single “activated” brain voxel

with no noise!

D: ***4-5 s rise***

E: ***5 s plateau***

Contrast through time

Signal increase caused by change in H_2O surroundings: more oxygenated hemoglobin is present

A: ***Pre-activation baseline***

C: ***≈ 2 s delay***

G: ***Return to baseline (or undershoot)***

B: ***5 s neural activity***

F: ***4-6 s fall***

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 MR images repeatedly during this process

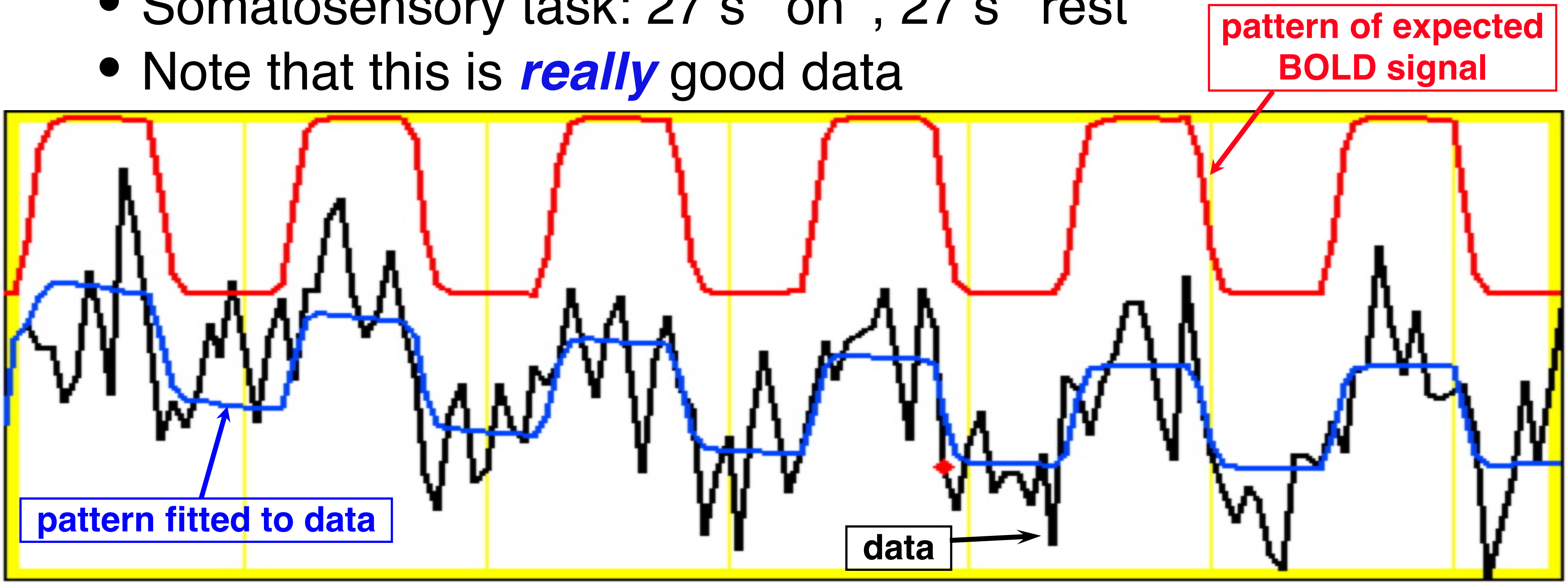
- Search for voxels whose NMR signal time series (up-and-down) matches the stimulus time series pattern (on-and-off)
  - → FMRI data analysis is basically pattern matching ***in time***

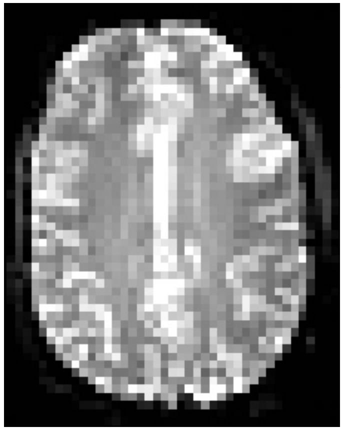
- Signal changes due to neural activity are small
  - Need 500 or so images in time series (in each slice) → takes 30 min or so to get reliable activation maps
    - Usually break image acquisition into shorter "runs" to give the subject and scanner some break time
  - Other small effects can corrupt the results → post-process the data to reduce these effects & ***be vigilant***

- Lengthy computations for image recon and temporal pattern matching → data analysis usually done offline

Sample Data Time Series

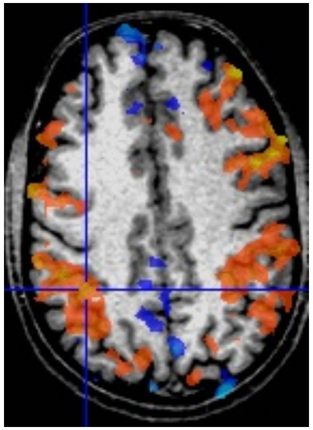
- 64 × 64 matrix (TR=2.5 s; 130 time points per imaging run)
- Somatosensory task: 27 s “on”, 27 s “rest”
- Note that this is ***really*** good data





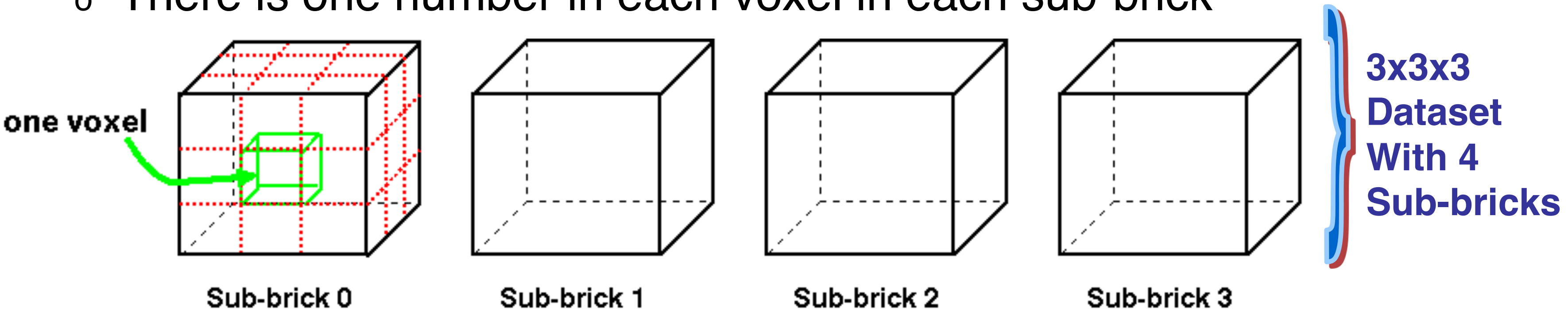
One echo-planar image

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

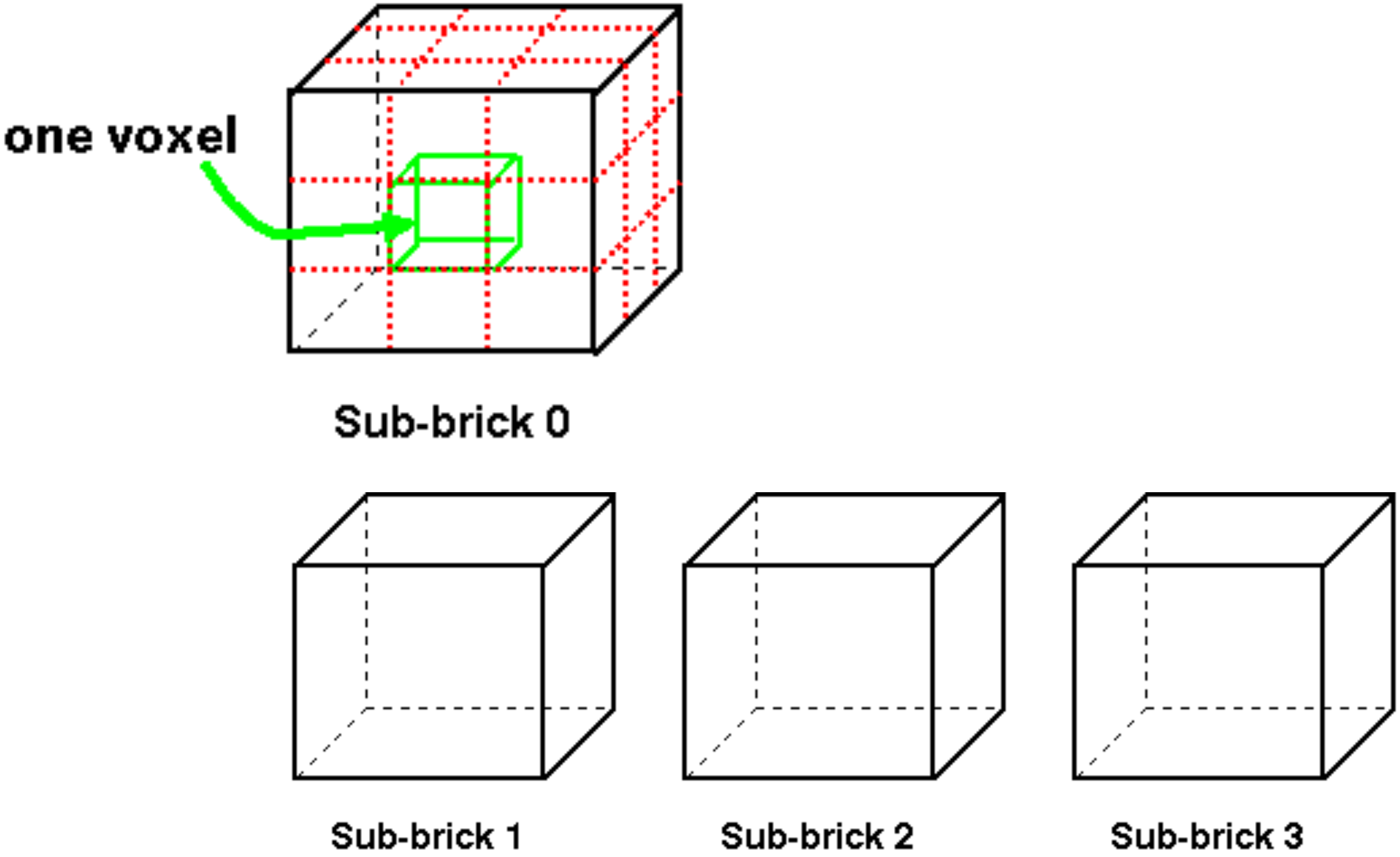


Fundamental AFNI Concepts

- Basic unit of data in AFNI is the dataset ← Jargon!
  - A collection of 1 or more 3D arrays of numbers
    - Each entry in the array is in a particular spatial location in a 3D grid (a voxel = 3D pixel)
    - Image datasets: each array holds a collection of slices from the 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



A Little Bit Bigger

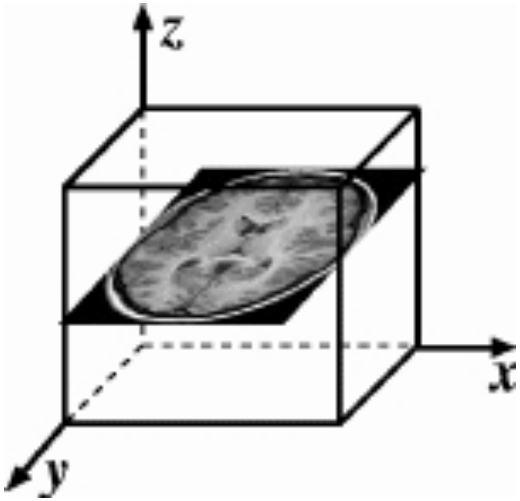


What's in a Dataset: Header Stuff

- 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
    - Very important to get right in FMRI, since we deal with many datasets
  - Time between sub-bricks, for 3D+time datasets ← *Jargon!*
    - Such datasets are the 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
    - e.g., an *F*-statistic sub-brick has 2 DOF parameters stored
  - Et cetera, et cetera, et cetera …



AFNI Dataset Files - 1

- AFNI formatted datasets are stored in 2 files
  - The .HEAD file holds all the auxiliary information
  - The .BRIK file holds all the numbers in all the sub-bricks
- Datasets can be in one of ~~3~~ 2 coordinate systems ("views")
  - Original data or +orig view: from the scanner
  - ~~AC-PC aligned or +acpc view:~~
    - ~~Dataset rotated/shifted so that the anterior commissure and posterior commissure are horizontal (*y*-axis), the AC is at (*x*,*y*,*z*)=(0,0,0), and the hemispheric fissure is vertical (*z*-axis)~~
  - Talairach or +tlrc view:
    - Dataset has also been rescaled to conform to the Talairach-Tournoux atlas dimensions (or another atlas, such as MNI)
    - AKA Talairach or Stererotaxic coordinates
    - **All** datasets scaled+aligned to some atlas are labeled +tlrc
      - Header can contain name of actual atlas "space" (e.g., MNI)
      - Alignment can be *linear* or *nonlinear* (**`3dQwarp`** program)

AFNI Dataset Files - 2

- AFNI dataset filenames consist of 3 parts
  - The user-selected prefix (almost anything)

    Jargon!
  - The view (one of +orig, ~~+acpc~~, or +tlrc)
  - The suffix (one of .HEAD or .BRIK)
  - **QinShiHuangdi+tlrc.HEAD** & **QinShiHuangdi+tlrc.BRIK**
  - When creating a dataset with an AFNI program, you supply the prefix; the program supplies the rest

- AFNI programs can ***read*** datasets stored in several formats
  - ANALYZE (.hdr/.img file pairs); i.e., from SPM, FSL
  - MINC-1 (.mnc); i.e., from mnitools [but not MINC-2]
  - CTF (.mri, .svl) MEG analysis volumes
  - ASCII text (.1D) — numbers arranged into columns
  - Have conversion programs to write out MINC-1, ANALYZE, ASCII, and NIfTI-1.1 files from AFNI datasets, if desired

NIfTI Dataset Files

- NIfTI-1 (.nii or .nii.gz) is a standard format that AFNI, SPM, FSL, BrainVoyager, et al., have agreed upon
  - Adaptation and extension of the old ANALYZE 7.5 format
  - Goal: easier interoperability of tools from various packages
- All data is stored in 1 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 (with Unix program gzip)
- AFNI now 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 all 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" – more about that later
- Program 2: Chris Rorden's **dcm2niix_afni**
  - Can create a whole collection of datasets
  - Works with more DICOM formats than **Dimon** does
  - *Problem*: The standard NIFTI .nii format cannot 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 (in AFNI extension for NIFTI .nii files)

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
  - **http://afni.nimh.nih.gov/afni**
  - Also: documentation, message board, humor, data, class materials, …
- You can download source code and compile it
  - Also from GitHub: **https://github.com/afni/AFNI**
- AFNI is updated fairly frequently, so it is important to update occasionally -- **@update.afni.binaries**
  - We can't help you with outdated versions***!***
  - ***Please check for updates every 6 months (or less)***

AFNI at the NIH Scanners

- AFNI can take 2D or 3D images in “realtime” from an external program and assemble them into 3D+time datasets slice-by-slice
- FMRI Facility scanners at the NIH (GE and Siemens) are set up to start AFNI on a remote Linux computer automatically when EPI acquisition starts, and then the **Dimon** program is used to send images into AFNI as they are reconstructed:
  - For immediate display (images and graphs of time series)
  - **Plus**: graphs of estimated subject head movement
- Goal is to let you see image data as they are acquired, so that if there are any big problems, you can fix them right away
  - Sample problem: someone typed in the imaging field-of-view (FOV) size wrong (240 cm instead of 24 cm), and so got garbage data, ***but only realized this too late*** (after scanning 8 subjects this way) — ***D’oh!***

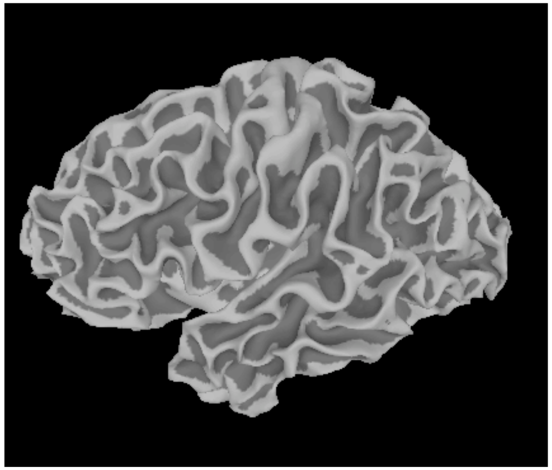
Other Parts of AFNI

- Batch mode programs and scripts
    - Are run by typing commands directly to computer, or by putting commands into a text file (script) and later executing them
- Good points about batch mode
    - Can process new datasets exactly the same as old ones
    - Can link together a sequence of programs to make a customized analysis (a personalized pipeline)
    - Some analyses take a long time (are not interactive)
- Bad points about batch mode
    - Learning curve is "all at once" rather than gradual
    - If you are, like, under age 35, you may not know how to, like, type commands into a computer to make it do things
        - But we don't make you use punched cards or paper tape (yet)

AFNI Batch Programs

- Many many important capabilities in AFNI are **only** available in batch programs
  - A few examples (of more than 100, from trivial to complex)
- **3dDeconvolve** + **3dREMLfit** = multiple *linear* regression on 3D+time datasets; fits each voxel's time series to activation model, tests these fits for significance (**3dNLfim** = nonlinear fitting)
- **3dvolreg** = 3D+time dataset registration, to correct for small subject head movements, and for inter-day head positioning
- **3dANOVA** + **3dLME** = 1-, 2-, 3-, and 4- way ANOVA/LME layouts: combining & contrasting datasets in Talairach space
- **3dcalc** = general purpose voxel-wise calculator (very useful)
- **3dsvm** = SVM multi-voxel pattern analysis program
- **3dresample** = re-orient and/or re-size dataset voxel grid
- **3dSkullStrip** = remove "skull" from anatomical dataset
- **3dDWItoDT** = compute diffusion tensor from DWI (nonlinearly)

SUMA, et alii



- **SUMA** is the AFNI surface mapper
  - For displaying surface models of cortex
    - Surfaces from **FreeSurfer** (MGH) or **Caret** (Wash U) or **BrainVoyager** (Brain Innovation)
  - Can display functional activations mapped from 3D volumes to the cortical surface
  - Can draw ROIs directly on the cortical surface
    - vs. AFNI: ROIs are drawn into the 3D volume
- SUMA is a separate program from AFNI, but can "**t**alk" with AFNI (like a plugout) so that volume & surface viewing are linked
  - 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!

Other Educational Presentations

- ~~How to get images into AFNI or NIfTI format (program **to3d**)~~
- Detailed hands-on with using AFNI for data viewing (fun)
- Signal modeling & analysis: theory & hands-on (**3dDeconvolve et al.**)
- Image registration (**3dvolreg et al.**)
- ~~Volume rendering hands-on (fun level=high)~~
- ~~ROI drawing hands-on (fun level=extreme)~~
- ~~Transformation to Talairach hands-on (fun level=low)~~
- Group analysis: theory and hands-on (**3dANOVA*x*** *and beyond*)
- ~~Experiment design~~
- FMRI analysis from start to end (the “soup to nuts” hands-on)
- SUMA hands-on (fun level=pretty good)
- Surface-based analysis
- Connectivity (resting state, white matter tracts)
- AFNI “Jazzercise” (practice sessions & directed exercises)