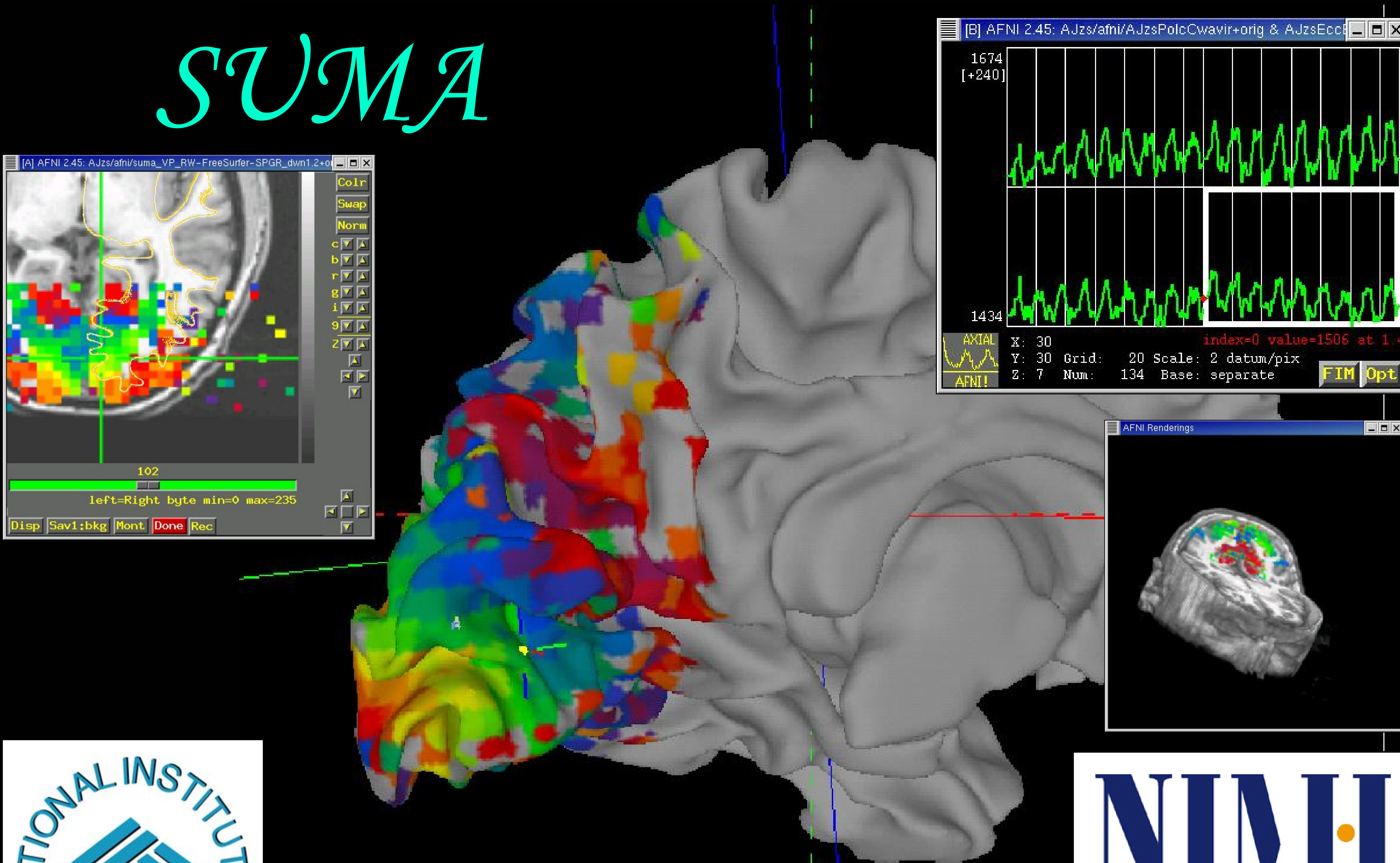
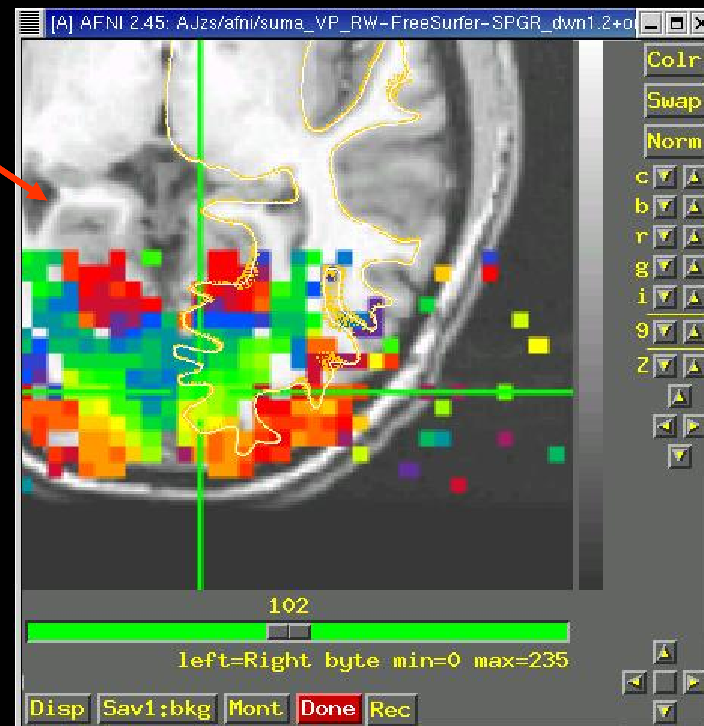
SUMA

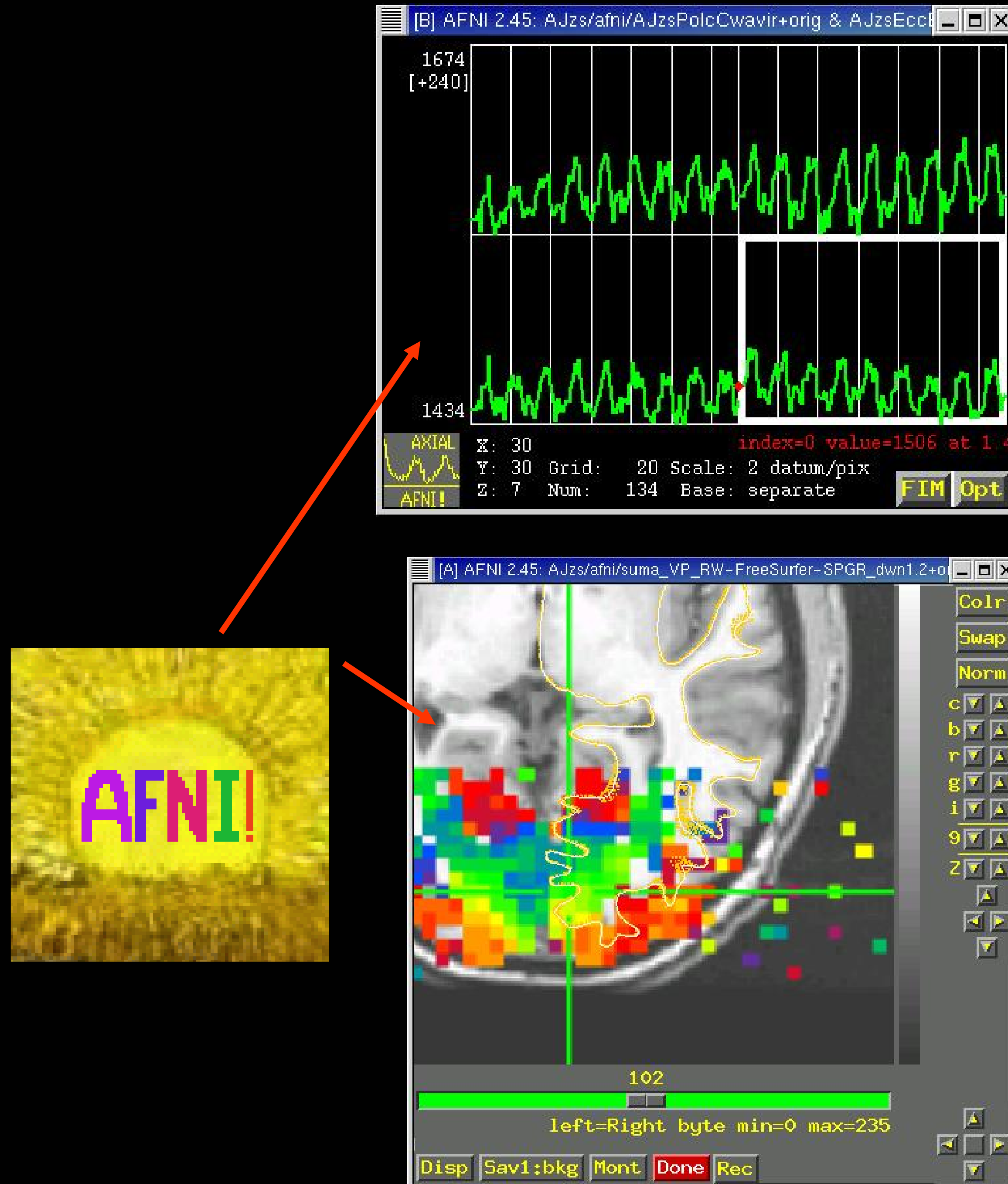


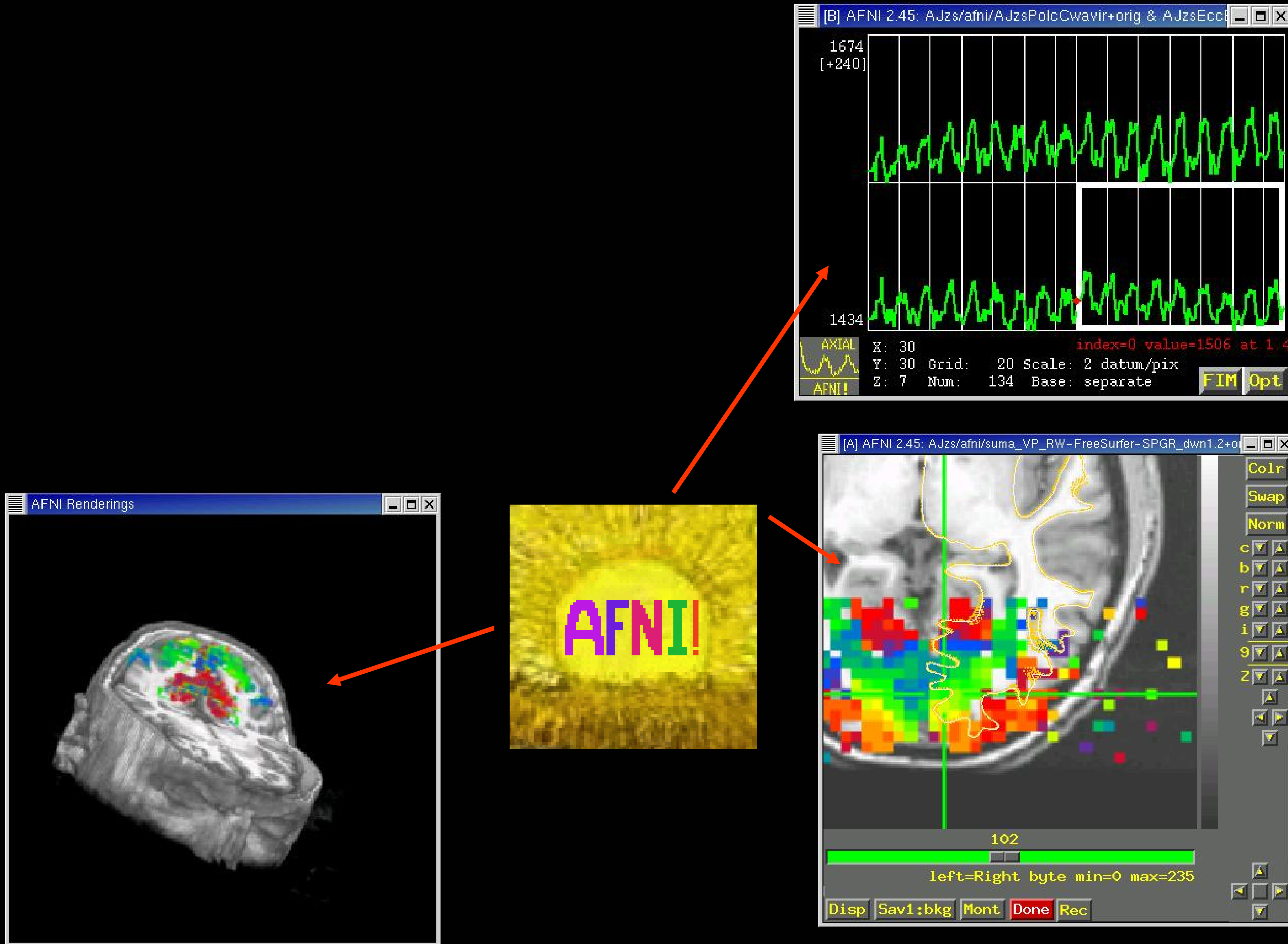
NATIONAL INSTITUTES OF HEALTH

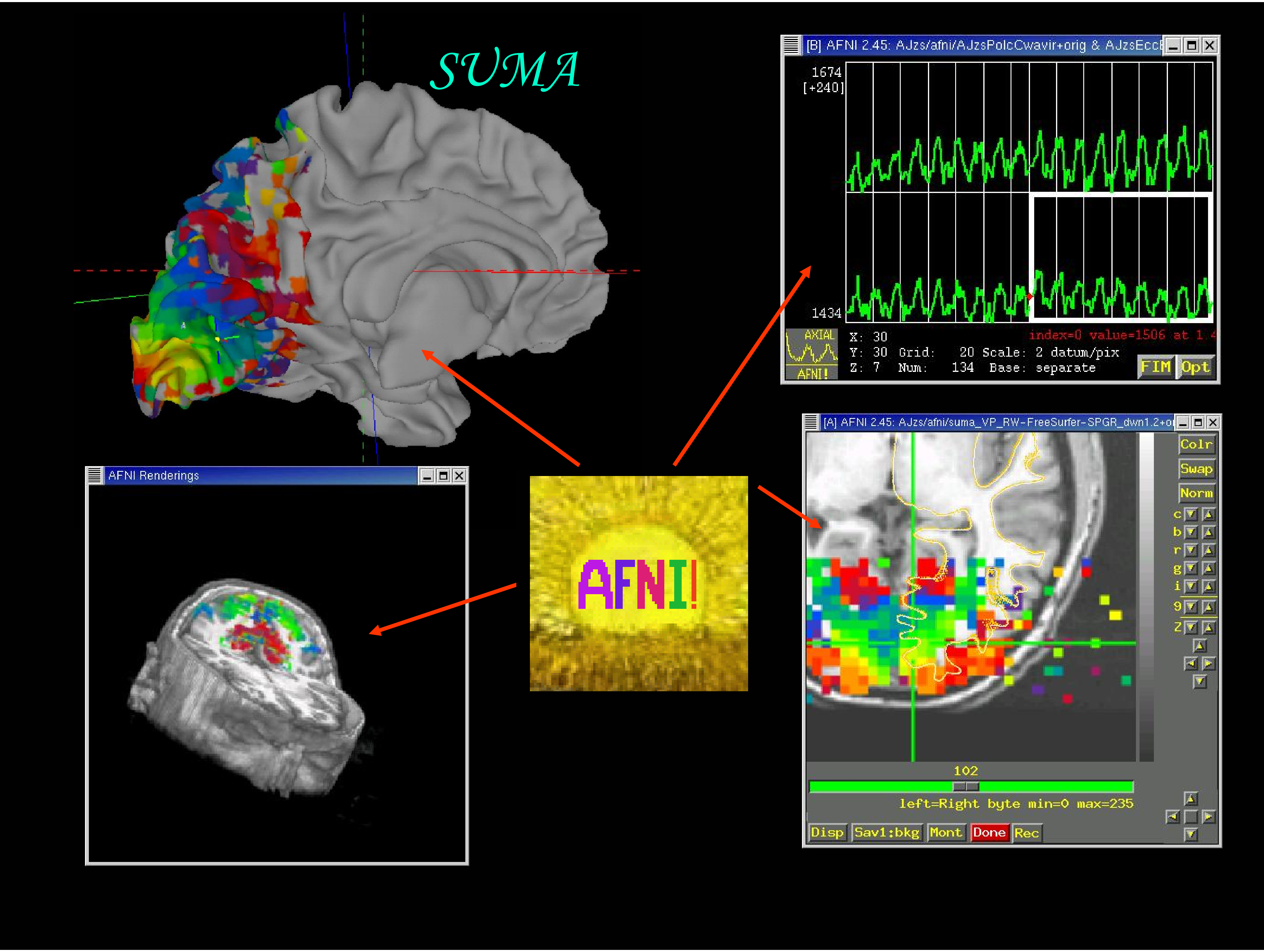
Statistical & Scientific Computing Core

NIMH
National Institute of Mental Health


[A] AFNI 2.45: AJzs/afni/suma_VP_RW-FreeSurfer-SPGR_dwn1.2+o
Colr
Swap
Norm
c
b
r
g
i
9
Z
102
left=Right byte min=0 max=235
Disp
Sav1:bkg
Mont
Done
Rec


[B] AFNI 2.45: AJzs/afni/AJzsPolcCwavir+orig & AJzsEcc
1674
[+240]
1434
AXIAL
AFNI!
X: 30
Y: 30 Grid: 20 Scale: 2 datum/pix
Z: 7 Num: 134 Base: separate
index=0 value=1506 at 1.4
FIM
Opt
AFNI!
[A] AFNI 2.45: AJzs/afni/suma_VP_RW-FreeSurfer-SPGR_dwn1.2+o
Colr
Swap
Norm
c
b
r
g
i
9
z
102
left=Right byte min=0 max=235
Disp
Sav1:bkg
Mont
Done
Rec


[B] AFNI 2.45: AJzs/afni/AJzsPolcCwavir+orig & AJzsEcc
1674
[+240]
1434
AXIAL
AFNI!
index=0 value=1506 at
X: 30
Y: 30 Grid: 20 Scale: 2 datum/pix
Z: 7 Num: 134 Base: separate
FIM
Opt
[A] AFNI 2.45: AJzs/afni/suma_VP_RW-FreeSurfer-SPGR_dwn1.2+o
Colr
Swap
Norm
102
left=Right byte min=0 max=235
Disp
Sav1:bkg
Mont
Done
Rec
AFNI Renderings
AFNI!


SUMA
[B] AFNI 2.45: AJzs/afni/AJzsPolcCwavir+orig & AJzsEcc
1674
[+240]
1434
AXIAL
AFNI!
X: 30
Y: 30 Grid: 20 Scale: 2 datum/pix
Z: 7 Num: 134 Base: separate
index=0 value=1506 at 1.4
FIM
Opt
[A] AFNI 2.45: AJzs/afni/suma_VP_RW-FreeSurfer-SPGR_dwn1.2+o
Colr
Swap
Norm
102
left=Right byte min=0 max=235
Disp
Sav1:bkg
Mont
Done
Rec
AFNI Renderings
AFNI!

SUrface MApping with AFNI

- Surface mapping & viewing program tightly linked to AFNI
- Complements AFNI's slice and volume rendering modes
- Provides a framework for fast and user-customizable surface based analysis
- Supports surface models created by:

  FreeSurfer http://surfer.nmr.mgh.harvard.edu

  SureFit/Caret http://stp.wustl.edu/resources/display.html

  BrainVoyager http://www.brainvoyager.com
- Allows representation of sparsely defined 3D data

System requirements

MESA Library 4.0.1 (an API closely resembling OpenGL)
http://www.mesa3d.org

- Currently Running On:
  - Centrino, 1.4 Ghz / Linux
  - 1Gb RAM
  - nVIDIA GeForce4 graphics card 64 Mb
- Also Runs On:
  - SGI
  - SUN with openGL 1.2 or newer
  - Mac OSX with Xfree86, Motif and Mesa

Installation

- Binaries for SUMA are available from:

  http://afni.nimh.nih.gov/ssc/ziad/SUMA/SUMA_DownloadTable.htm

- On some machines (i.e. Linux), SUMA should be compiled locally for optimal performance.
- To compile SUMA you need to have:
  - Motif (both libraries and header files)
  - Mesa (OpenGL)
  - AFNI's source code distribution
- Detailed installation info for various systems:

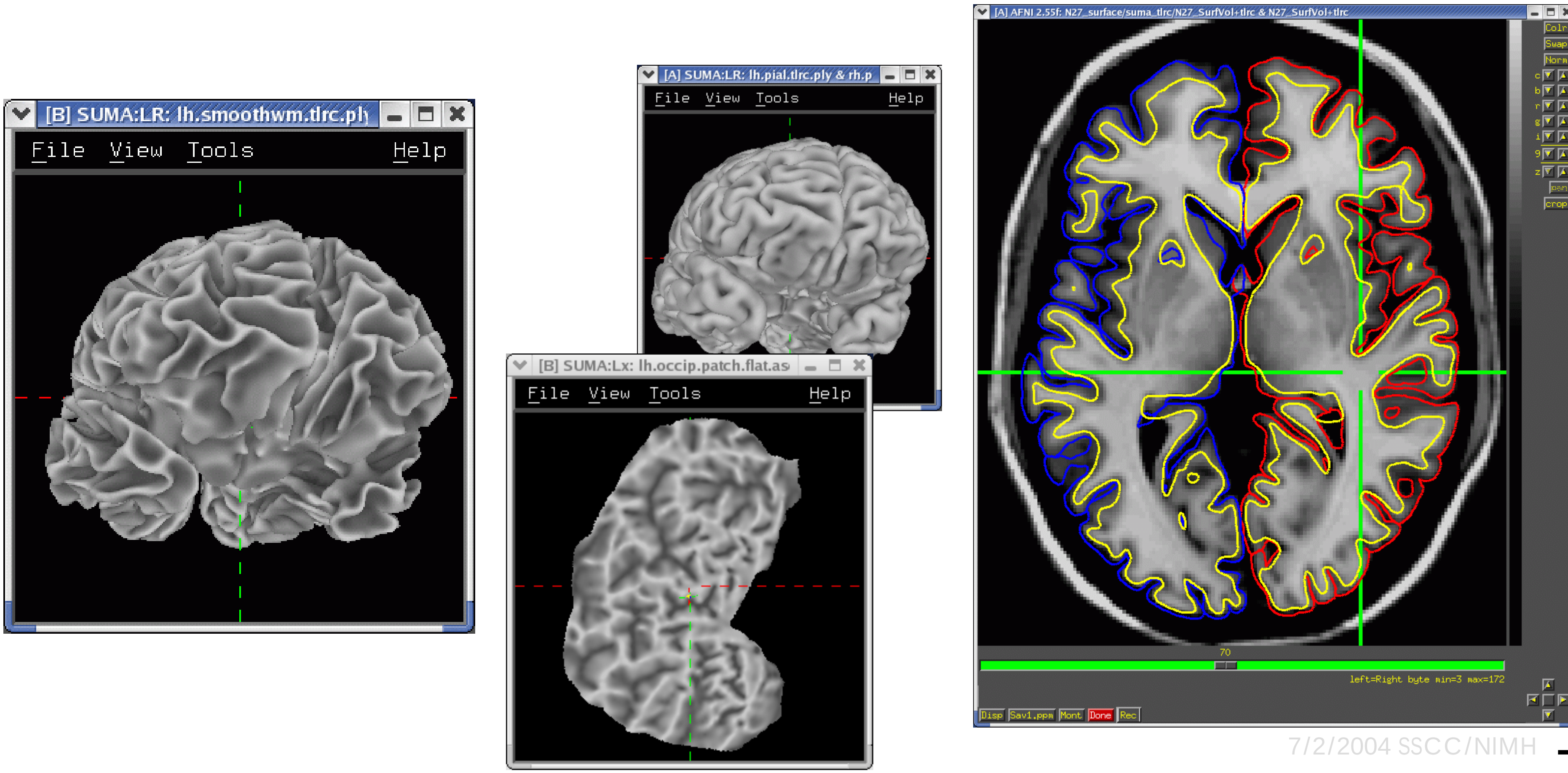
  http://afni.nimh.nih.gov/ssc/ziad/SUMA/SUMA_Installation.htm

Where do surface models come from ?

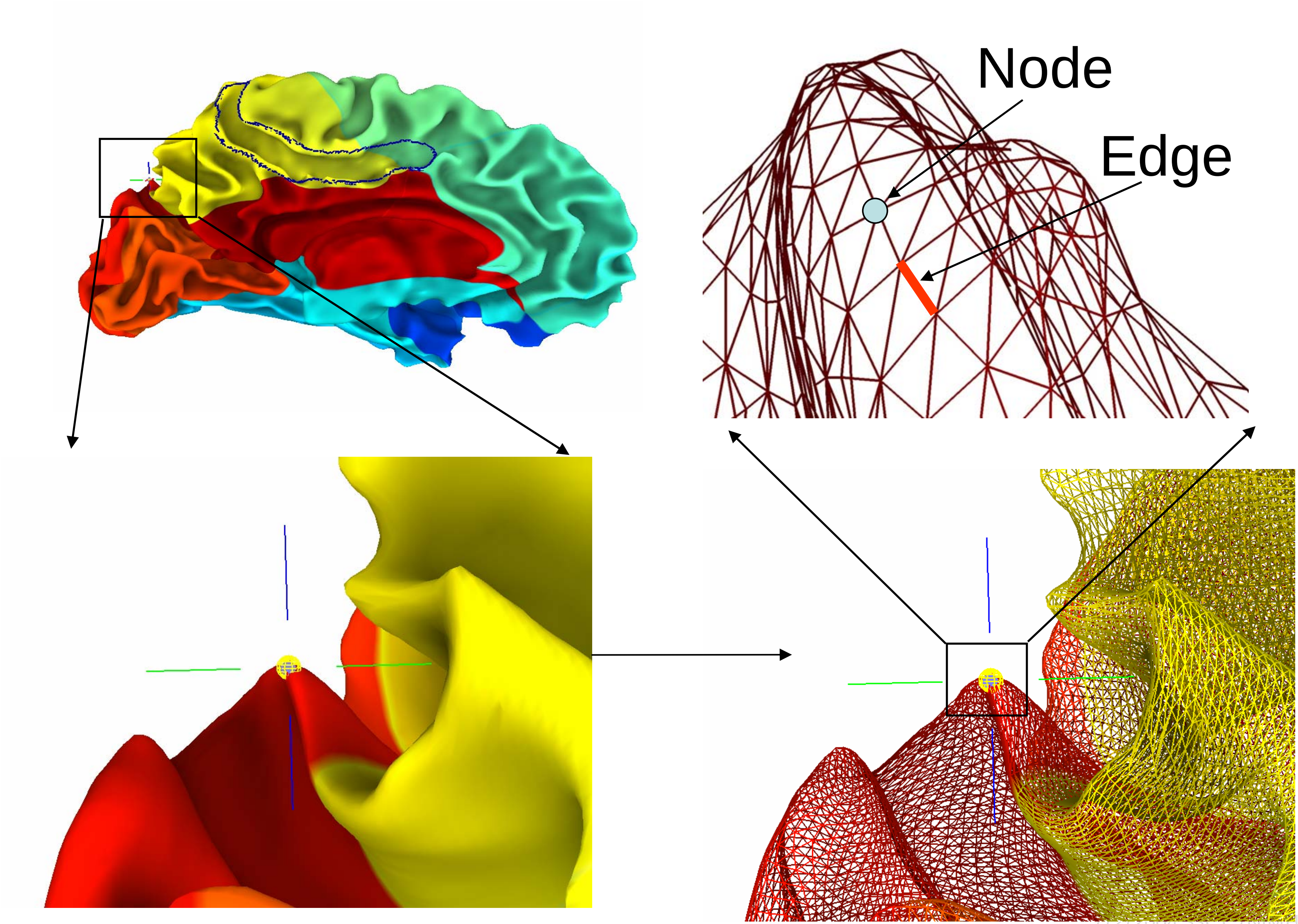
- For surface based analysis:
  - Must create surfaces for individual subjects
  - PreSUMA:
    - Collect, align and average high-quality, high-resolution anatomical data
      - On NIMH's 3T-1, 4 MPRAGE data sets will do
    - Correct image non-uniformity
      - Using AFNI's 3dUniformize or the N3 normalization tool [J.G. Sled et al. 98]
    - Create and correct surfaces
      - Using FreeSurfer, SureFit or BrainVoyager
  - CircumSUMA:
    - Align surface with experimental data
      - Using @SUMA_AlignToExperiment
    - Map experimental volumetric data to surface
      - Using AFNI and SUMA
  - PostSUMA:
    - Fame, Fortune and Fortitude for some, if not all, or none of you

Where do surface models come from ?

- For display (mostly) of Talairach data:
  - Use the Talairach surfaces created from the N27 brain data set using FreeSurfer.
    - Ready to use, no surface creation or alignment needed.

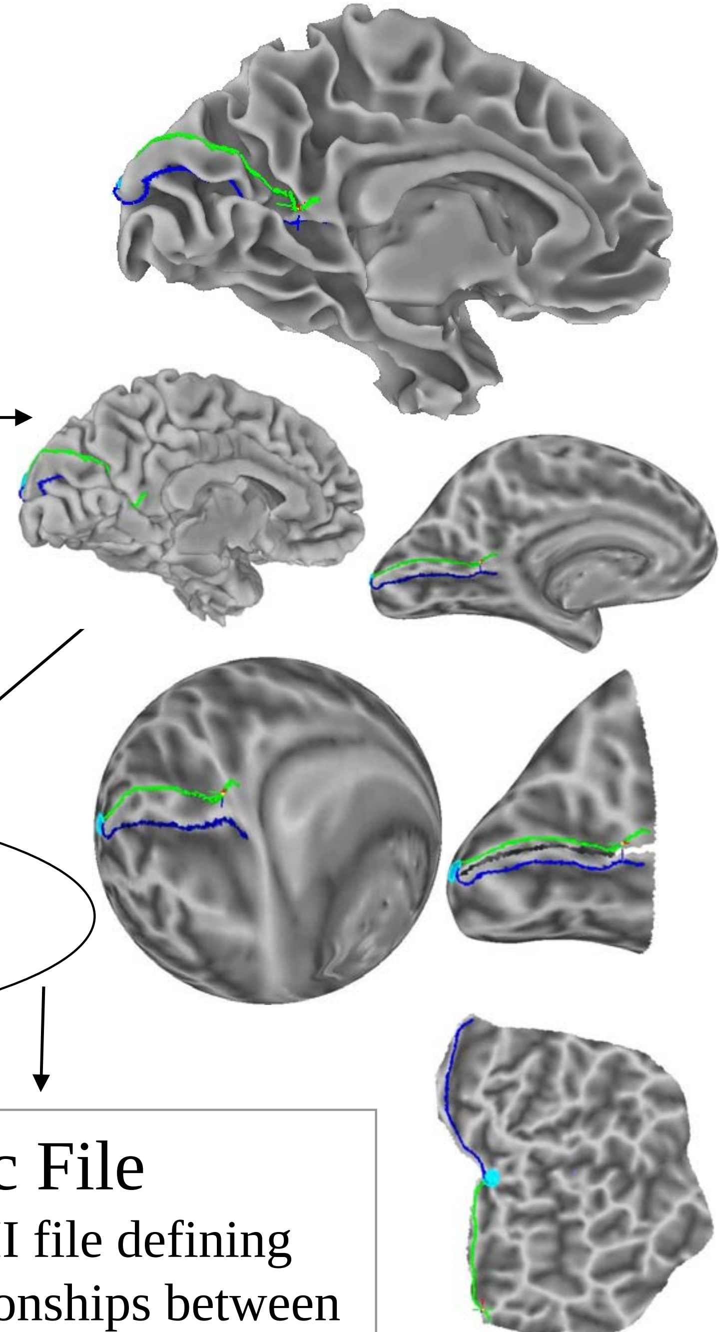


What’s a surface made of?



A: Preparing surface models for SUMA

High-Res. Anatomical MRI data → Create Surface Models (FreeSurfer, SureFit, etc.) →



@SUMA_Make_Spec_FS
@SUMA_Make_Spec_SF

SurfVol
AFNI-format Surface Volume that is aligned with surface models

Spec File
ASCII file defining relationships between different surfaces

A: Preparing surface models for SUMA

- Create the Surface Volume
  - SurfVol is an AFNI data set created from data used to create the surfaces.
- Create the surface specifications (spec) file
  - The Spec file defines the relationships between the different surfaces.
- Both the surface volume and the Spec file are created automatically
  - Using @SUMA_Make_Spec_FS for FreeSurfer surfaces
  - Using @SUMA_Make_Spec_SF for SureFit surfaces
- At this point, surface models should be in excellent alignment with SurfVol
  - Use SUMA to verify alignment
  - Scroll through the volume to make sure surfaces are accurate
    - Check especially for inferior temporal and occipital areas

A: Preparing surface models for SUMA

- Demo for FreeSurfer surfaces:
  - cd suma_demo/SurfData
    - This is where FreeSurfer's directories reside
  - @SUMA_Make_Spec_FS -sid *DemoSubj*
    - Creates the SurfVol from the .COR files
      - DemoSubj_SurfVol+orig
    - Creates ASCII versions of surfaces found in the surf directory
      - lh*.asc and rh*.asc (if rh surfaces are provided)
    - Creates he Spec files for the left and right hemisphere surfaces
      - DemoSubj_lh.spec and DemoSubj_rh.spec (if rh surfaces are provided)
  - cd SUMA
  - afni –niml &
    - launches AFNI to allow the viewing of DemoSubj_lh.spec
    - the –niml option tells AFNI to listen to connections from SUMA
  - suma –spec DemoSubj_lh.spec –sv DemoSubj_SurfVol+orig
    - *or execute the script: ./run_suma*

Check for proper alignment and defects

- With both SUMA and AFNI running
  - Press ‘t’ in the suma window to establish a connection to AFNI and send the mapping reference surface(s).
  - If you open the surface volume in AFNI, you should see the surface overlaid on top of it.
    - **Switch Underlay** to DemoSubj_SurfVol+orig and open an axial view. You will see a trace of the intersection of the surface with the anatomical slices displayed.
      - You could also see boxes representing the nodes that are within +/- 1/2slice from the center of the slice in view.
      - Colors and node box visibility can be changed to suit your fancy from the **Control Surface** button in AFNI.
  - Navigate through the volume in AFNI.
    - make sure you have an excellent alignment between volume and surface
    - make sure surface adequately represents areas of the brain that are difficult to segment
      - occipital cortex
      - inferior frontal and inferior temporal regions

Check for proper alignment and defects

Note: Viewed without the volume underlay, it is extremely difficult to tell if surface models with no topological defects accurately represent the cortical surface.

- The Surface Volume and the surfaces must be in perfect alignment.
  - If you have an improper alignment, it should be addressed here.
    - This should not happen for FreeSurfer and SureFit surfaces created in the standard fashion.
  - Watch for error messages and warnings that come up in the shell as the surfaces are read in. These messages should be screened once since they do not change unless the surface's geometry or topology is changed.

Basic SUMA viewer functions

- Rotating the surface:
  - **Mouse button-1:** keep it down while moving the mouse left to right. This rotates the surface about the screen's Y-axis (dotted green). Let go of button-1.
  - Repeat with up and down motion for rotation about X-axis and motion in various directions for rotations mimicking those of a trackball interface.
  - Also try up/down/left/right arrow keys.
    - Arrow keys rotate by increments specified by the environment variable: SUMA_ArrowRotAngle (degrees).

Basic SUMA viewer functions

- Translating the surface:
  - **Mouse button-2:** keep it down while moving the mouse to translate surface along screen X and Y axes or any combinations of the two.
  - Also try shift+arrow keys.
- Zooming in/out:
  - **Both buttons 1&2 or Shift + button 2:** while pressing buttons, move mouse down or up to zoom in and out, respectively.
  - Also try keyboard buttons 'Z' and 'z' for zooming in and out, respectively.

Basic SUMA viewer functions

- Cardinal views:
  - ctrl + Left/Right: Views along LR axis
  - ctrl + Up/Down: Views along SI axis
  - ctrl + shift + Up/down: Views along AP axis

- Resetting the view point:
  - Press Home to get back to the original vantage point.

- Using momentum feature:
  - Press ‘m’ to toggle momentum on. Click the left mouse button and release the button as you are dragging the mouse.

Basic SUMA viewer functions

- Picking a Node or Facet:
  - **Mouse button 3**: press over a location on a surface to pick the closest facet and node to the location of the pointer.
    - The closest node is highlighted with a blue sphere
    - The closest facet is highlighted with a gray triangle
  - Note the information written to the shell regarding the properties of the picked Node and Facet.
  - When connected to AFNI (after having pressed ‘t’), watch the AFNI crosshair jump to the corresponding location in the volume.
  - Conversely, position the crosshair in AFNI (left click) at a position close to the surface and watch the crosshair relocate in SUMA.
- You can swap button 1 & 3’s functions using the environment variable: SUMA_SwapButtons_1_3

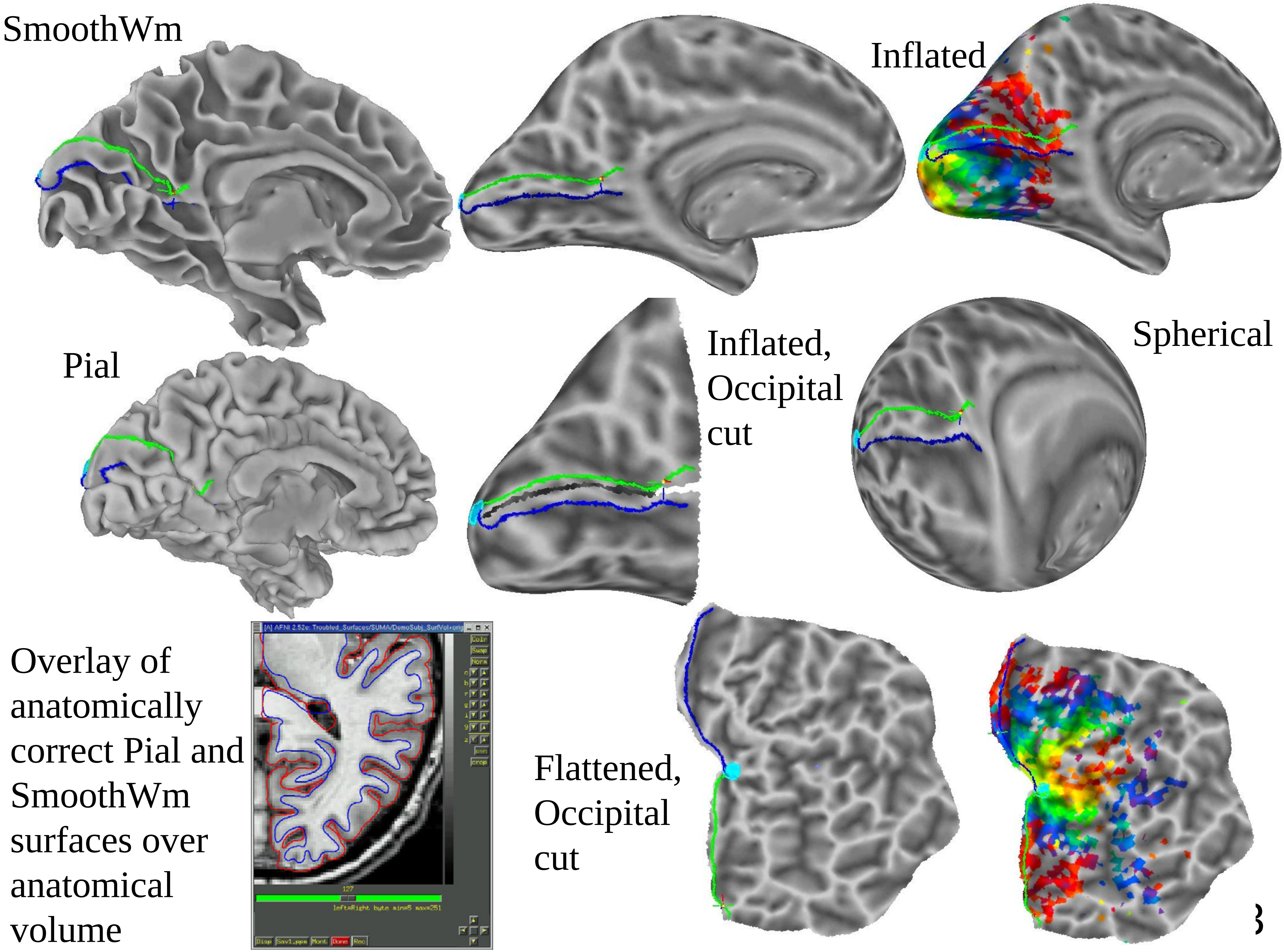
Hands-On

world ←→AFNI ←→ SUMA ←→ world

- AFNI and SUMA are independent programs and communicate using NIML formatted data elements
  - via shared memory if both programs are on the same computer
  - via network sockets otherwise
  - Both AFNI and SUMA can also communicate with other programs
- NIML: NeuroImaging Markup Language developed by Dr. R.W. Cox
  - NIML will be the main format for SUMA’s data storage.
- NIML API library for packing/unpacking data is available and documented.
- Communication protocol allows any independent program to communicate with AFNI.
- Advantages include:
  - Programs execute on separate machines
  - Fast development
  - Screen real-estate
- Blemishes include:
  - Only one AFNI can be listening for connections
  - Only one SUMA can connect to AFNI

Relationships between surface models

- Surface Geometry:
  - refers to the spatial coordinates of the nodes forming the surface model
- Surface Topology:
  - refers to the connectivity between nodes forming the surface model
- Models with different geometry but similar topology are created for each surface model.
  - white/grey, pial, inflated, spherical, flattened, etc.
- Some models' geometries are anatomically correct
  - Pial and/or white matter surfaces can be used for relating to volume data.
  - Inflated, flattened and spherical cannot be directly linked to volume data. The link is done via their corresponding anatomically-correct surfaces.


SmoothWm
Inflated
Pial
Inflated, Occipital cut
Spherical
Overlay of anatomically correct Pial and SmoothWm surfaces over anatomical volume
Flattened, Occipital cut

Sample Spec File

```
# delimits comments
# define the group
	Group = DemoSubj

# define various States
	StateDef = smoothwm
	StateDef = pial
	StateDef = inflated

NewSurface
	SurfaceFormat = ASCII
	SurfaceType = FreeSurfer
	FreeSurferSurface =
lh.smoothwm.asc
	LocalDomainParent = SAME
	SurfaceState = smoothwm
	EmbedDimension = 3

NewSurface
	SurfaceFormat = ASCII
	SurfaceType = FreeSurfer
	FreeSurferSurface = lh.pial.asc
	LocalDomainParent =
lh.smoothwm.asc
	SurfaceState = pial
	EmbedDimension = 3

NewSurface
	SurfaceFormat = ASCII
	SurfaceType = FreeSurfer
	FreeSurferSurface = lh.inflated.asc
	LocalDomainParent = lh.smoothwm.asc
	SurfaceState = inflated
	EmbedDimension = 3

NewSurface
	SurfaceFormat = ASCII
	SurfaceType = FreeSurfer
	FreeSurferSurface = lh.sphere.asc
	LocalDomainParent = lh.smoothwm.asc
	SurfaceState = sphere
	EmbedDimension = 3
```

Details of the Spec file:

- The Spec file contains information about the surfaces that will be viewed.
  - Information is specified in the format: *field* = *value*.
  - The = sign must be preceded and followed by a space character.
  - # delimit comment lines, empty lines and tabs are ignored.
  - In addition to fields, there is also the **NewSurface** tag which is used to announce a new surface.
  - Unrecognized text will cause the program parsing Spec file to complain and exit.

Details of the Spec file:

- The fields are:
  - Group: Usually the Subject's ID. In the current SUMA version, you can only have one group per spec file. All surfaces read by SUMA must belong to a group.
  - FreeSurferSurface: Name of the FreeSurfer surface.
  - SurfaceFormat: ASCII or BINARY
  - SurfaceType: FreeSurfer or SureFit
  - SurfaceState: Surfaces can be in different states such as inflated, flattened, etc. The label of a state is arbitrary and can be defined by the user. The set of available states must be defined with StateDef at the beginning of the Spec file.

Details of the Spec file:

- The fields are:
  - StateDef: Used to define the various states. This must be placed before any of the surfaces are specified.
  - Anatomical: Used to indicate whether surface is anatomically correct (Y) or not (N).
  - LocalDomainParent: Name of a surface whose mesh is shared by other surfaces in the spec file.

    The default for FreeSurfer surfaces is the smoothed gray matter/ white matter boundary. For SureFit it is the fiducial surface. Use SAME when the LocalDomainParent for a surface is the surface itself.

    - At the moment, only surfaces that are a Domain Parent are sent to AFNI. In the very near future, Anatomically Correct surfaces will be sent to AFNI regardless of their Domain Parent.
    - A node-to-node correspondence is maintained across surfaces sharing the same domain parent.
  - EmbedDimension: Embedding Dimension of the surface, 2 for surfaces in the flattened state, 3 for other.

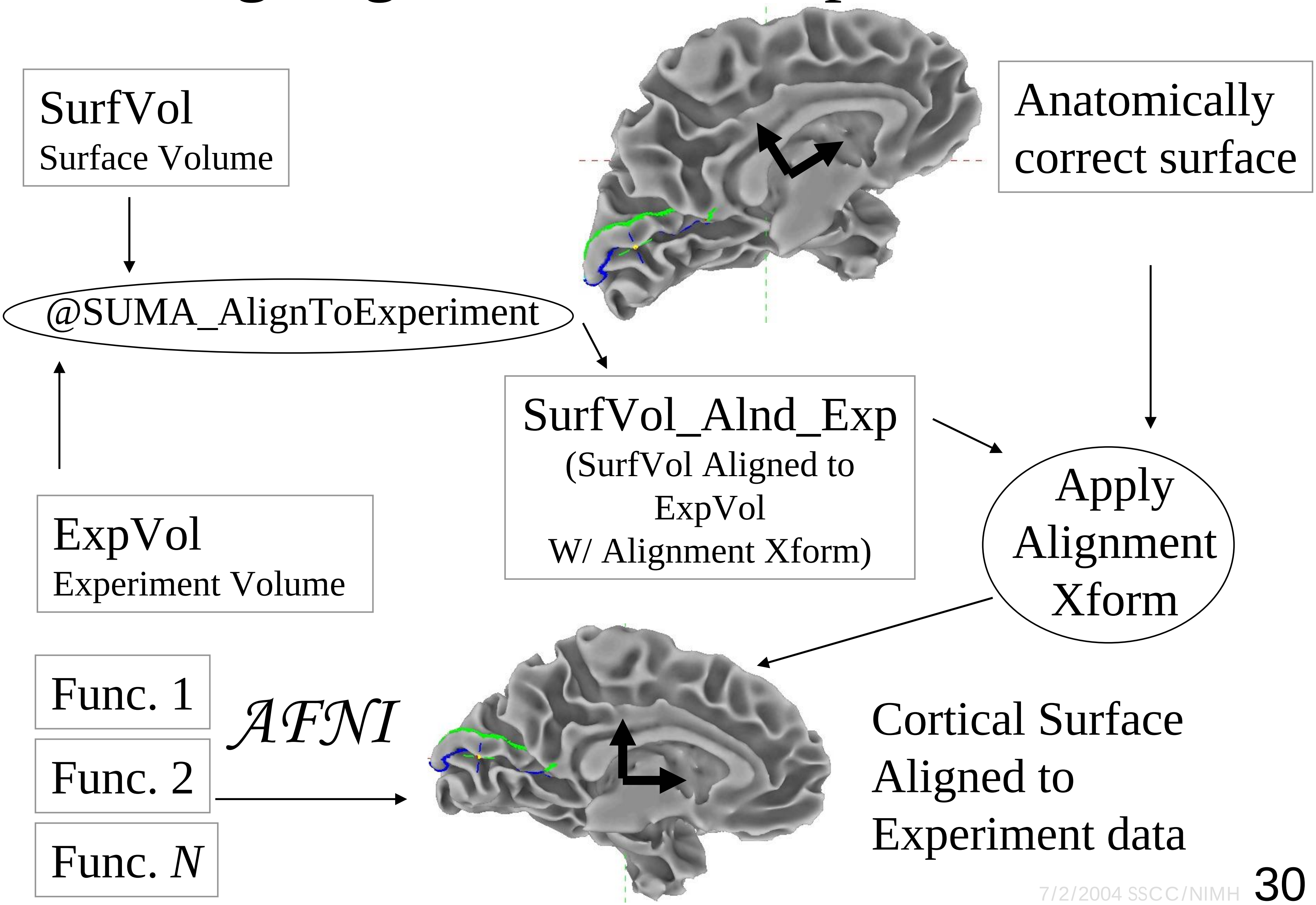
Viewing the group of surfaces

- Switch Viewing States:
  - **'.'** switches to the next viewing state (pial then inflated etc.)
  - **','** switches to the previous viewing state
  - Navigate on any of the surfaces and watch AFNI’s crosshair track surface
  - SPACE toggles between current state and Mapping Reference state
- Viewing multiple states concurrently:
  - ‘ctrl+n’ opens a new SUMA controller (up to 6 allowed, more possible but ridiculous)
  - switch states in any of the viewers
  - all viewers are still connected to AFNI

Viewing the group of surfaces

- Controlling link between viewers:
  - Open SUMA controller with ‘ctrl+u’ or View->SUMA Controller
  - SUMA controller crosshair locking options:
    - ‘-’: no locking
    - ‘i’: node index locking (i.e. topology based)
    - ‘c’: node coordinate locking (i.e. geometry based)
  - SUMA controller view point locking
    - ‘v’: depress toggle button to link view point across viewers.
      - Surface rotation and translation in one viewer is reflected in all linked viewers

B: Aligning Surface w/ Experiment Data



B: Aligning Surface w/ Experiment Data

- Surface Volume is aligned to experiment's anatomical volume with 3dvolreg
  - Brain coverage and image types should be comparable, not necessarily identical
  - If alignment with 3dvolreg is inadequate, try 3dTagalign
- Functional data are assumed to be in register with experiment's anatomical
  - If alignment is poor, try 3dAnatNudge or Nudge Dataset plugin
- Functional data are not interpolated

B: Aligning Surface w/ Experiment Data

- Demo: (close previous SUMA and AFNI sessions)
  - cd suma_demo/afni
    - DemoSubj_spgrax+orig (experiment's high-res. anatomical scan)
    - DemoSubj_EccExpavir+orig & DemoSubj_EccExpavir.DEL+orig (EPI anat. & func.)
  - @SUMA_AlignToExperiment DemoSubj_spgrsa+orig ../SurfData/SUMA/DemoSubj_SurfVol+orig
    - This script will use 3dvolreg to align the experiment's anatomical volume to the Surface Volume.
    - The script will take care of resampling (with 3dresample) the experiment's anatomical volume to match the Surface Volume if need be.
    - The output volume is named with the prefix of the Surface Volume with the suffix _Alnd_Exp (read Aligned to Experiment).
    - *continue next slide…*

B: Aligning Surface w/ Experiment Data

- Demo *(continued)*:
  - afni –niml &
    - We're launching AFNI to make sure that SUMA/DemoSubj_SurfVol_Alnd_Exp+orig and DemoSubj_spgrsa+orig are well aligned.
    - Switch Underlay to DemoSubj_SurfVol_Alnd_Exp+orig
    - Open New AFNI controller (B)
    - Switch Underlay in B to DemoSubj_spgrsa+orig
    - Make sure controllers A and B are locked (XYZ Lock, i.e. NOT IJK Lock). Do this through Define Datamode → Lock menu.
    - *continue next slide…*

B: Aligning Surface w/ Experiment Data

- Demo *(continued)* :
  - Open the same views in both controllers (say Axial).
  - Click in one view and check if crosshair in other view points to a similar location.
  - Note how despite the difference in scan pulse sequence, SNR, and coverage, the alignment is pretty good.

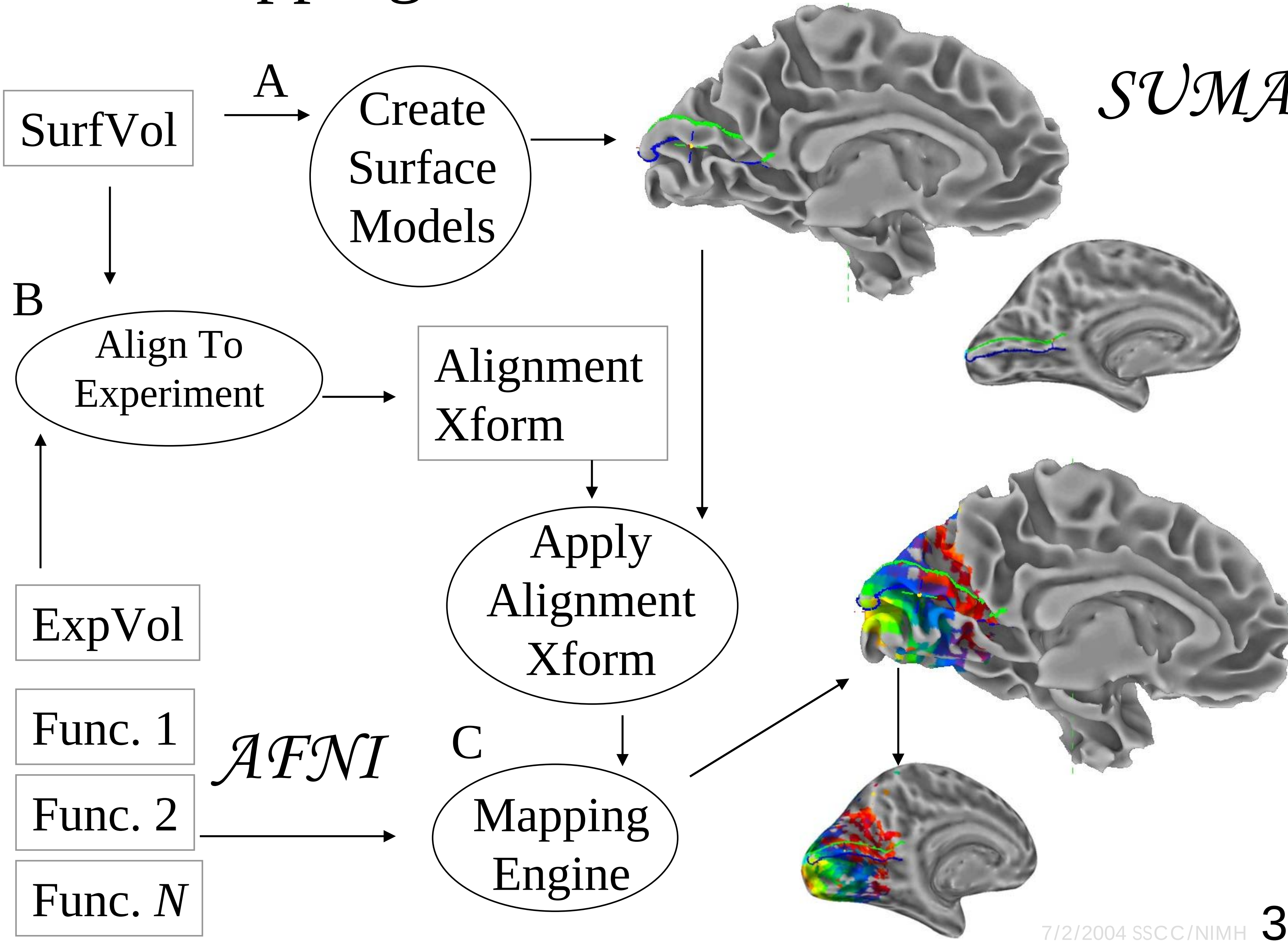
    average of 4 MPRAGE for surface model (~40 min at 3T)

    one SPGR for experiment anatomy (~5 min at 3T)
  - Since DemoSubj_SurfVol_Alnd_Exp+orig is now aligned with the experiment's data and is of a superior quality, you should consider using it as your anatomical underlay.
  - Close controller B.

B: Aligning Surface w/ Experiment Data

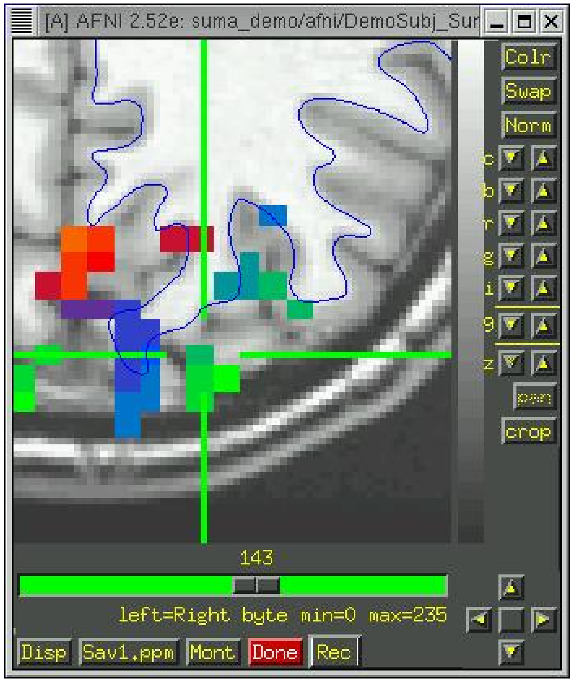
- Demo *(continued)*:
  - suma –spec ../SurfData/SUMA/DemoSubj_lh.spec –sv DemoSubj_SurfVol_Alnd_Exp+orig
    - *or execute the script: ./run_suma*
    - launching SUMA to make sure surface alignment is OK
  - ‘t’ to talk to AFNI
    - You should see a surface overlaid onto DemoSubj_SurfVol_Alnd_Exp+orig data set.
    - Alignment should be proper, otherwise you have a problem.

C: Mapping FMRI Data Onto Surface



C: Mapping FMRI Data Onto Surface

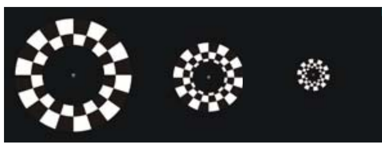
- Interactive mapping is done by AFNI
  - Mapping is done by intersecting Mapping Reference surface (the one sent to AFNI by SUMA) with the overlay data volume.
    - Nodes inside a functional voxel receive that voxel's color
- Demo *(continued)*:
  - Switch Overlay to DemoSubj_EccExpavir.DEL
  - Define Overlay with:
    - Olay: Delay
    - Thr: Corr. Coef.
    - Pos. color mapping
    - #20 color map
    - See Function
      - You should see the function on the surface model in SUMA.
      - The colors are applied to all topologically related surfaces
      - NOTE: Only AFNI controller A sends function back to SUMA
    - Change threshold in AFNI and watch change in SUMA



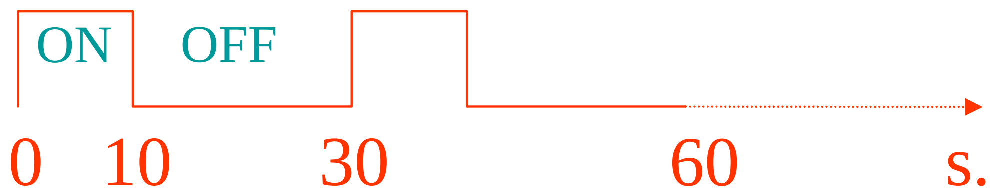
Sample FMRI data: Eccentricity Mapping

- Scan Parameters:
  - EPI: NIH-EPI, TR=2sec, 17 Coronal Slices, 134 samples, 3.75 x 3.75 x 4 mm
  - Anat: SPGR, 0.94 x 0.94 x 1.1, 120 axial slices
- Stimulus Timing:

Eccentricity



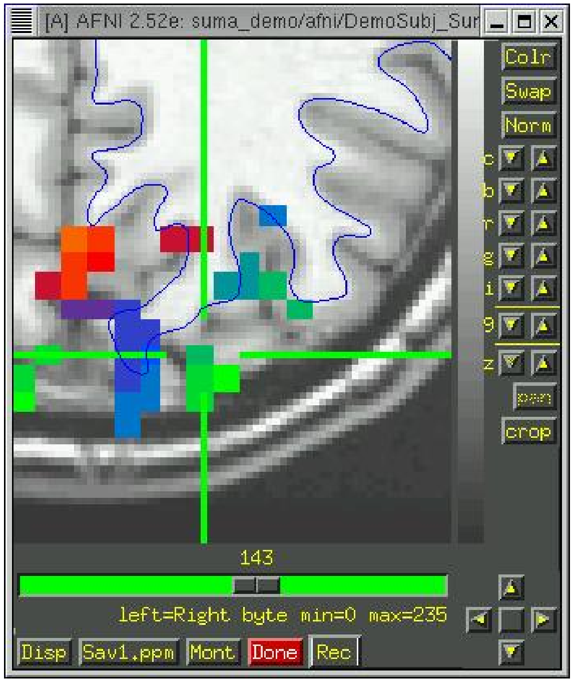
0 10 s.



- Activation delay estimated with 3ddelay
- Demo:
  - Rotate color map in AFNI and watch changes in SUMA
    - note how colors progress along the calcarine sulcus
    - try the dance on inflated and spherical surfaces

C: Mapping FMRI data onto surface

- The problem with intersection mapping
  - Only voxels intersecting the surface are mapped.
  - Can use Pial surface for Mapping Reference instead of SmoothWm.

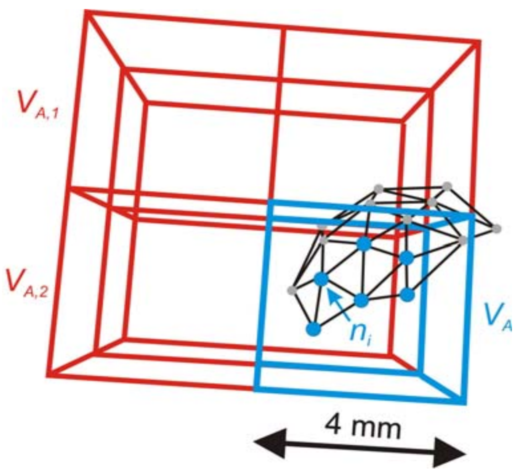


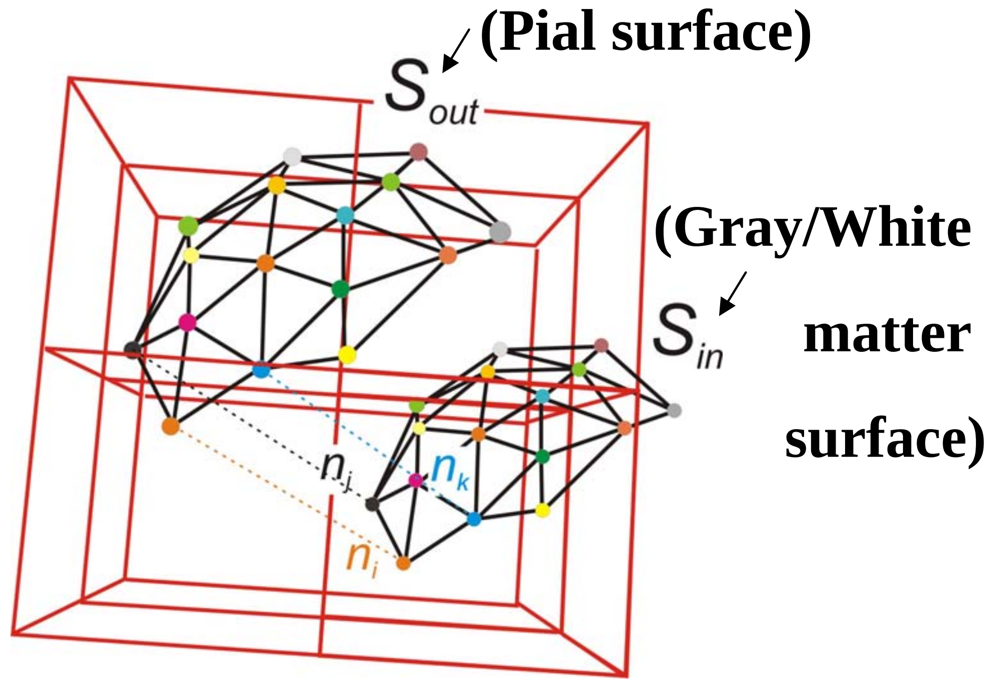
- Other mapping methods

  using 3dVol2Surf by Rick R. Reynolds
  - mapping based on intersection of cortical sheet (all of gray matter) with volume

Mapping options

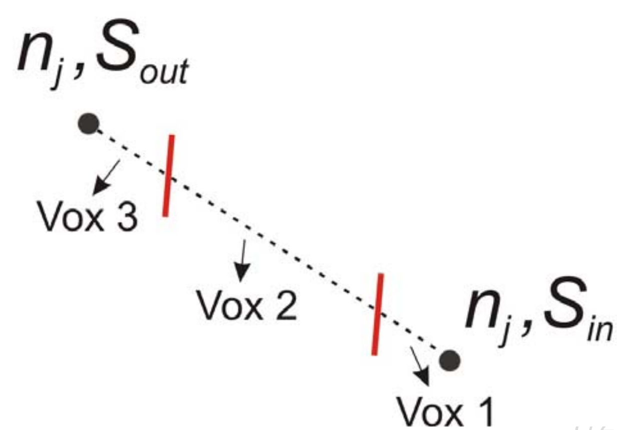
- Surface/volume Intersection
  - One voxel per node





Node segment intersecting 3 voxels

- Shell/volume Intersection
  - Multiple voxels possible per node



Mapping data: Volume → Surface

- Use 3dVol2Surf to map individual subject data onto each surface
  - Example: Mapping functional data onto surface, with thresholding

```
3dVol2Surf
    -spec           ../SurfData/SUMA/DemoSubj_lh.spec            \
    -surf_A         lh.smoothwm.asc                              \
    -surf_B         lh.pial.asc                                  \
    -sv             DemoSubj_SurfVol_Alnd_Exp+orig               \
    -grid_parent    DemoSubj_EccExpavir.DEL+orig                 \
    -cmask          '-a DemoSubj_EccExpavir.DEL+orig[2]          \
    -expr           step(a-0.5)'                                 \
    -map_func       ave                                          \
    -f_steps        10                                           \
    -f_index        nodes                                        \
    -oom_value      -1.0                                         \
    -oob_value      -2.0                                         \
    -out_1D         out_Del.1D.dset
```

by Rick Reynolds

Options for 3dVol2Surf example

- -spec: SUMA spec file containing surface(s) to be used in mapping.
- -surf_A (-surf_B): Surface(s) to be used in the mapping.
  3dVol2Surf uses one or two surfaces for the mapping.
- -sv: Surface Volume used to align surface to data
- -grid_parent: AFNI volume containing data to be mapped. DataVol contains 4 sub-bricks with the last one being the threshold.
- -cmask: Option for masking data in DataVol. Threshold value was 0.5 using the cross correlation coefficients.
- -map_func: Method for handling multiple voxel to one node mapping
- -oom_value -1 : Assign -1 to nodes in inactive voxels
- -oob_value -2: Assign -2 to nodes that fall outside data coverage
- -out_1D: Output data set file

- Use –help option for detailed help (~500 lines)

Results: Volume → Surface

- Demo:
  - Run the script: ./run_3dVol2Surf
  - The output: out_Del.1D.dset is tabulated below:

node	1dindex	i	j	k	vals	v0	v1	v2	v3
0	10080	32	29	2	4	-1	-1	-1	-1
1	10080	32	29	2	4	-1	-1	-1	-1
…	…	…							
21	10081	33	29	2	3	18.15	893.3	0.65	278.9
22	10081	33	29	2	3	18.15	893.3	0.65	278.9
23	10144	32	30	2	4	-1	-1	-1	-1

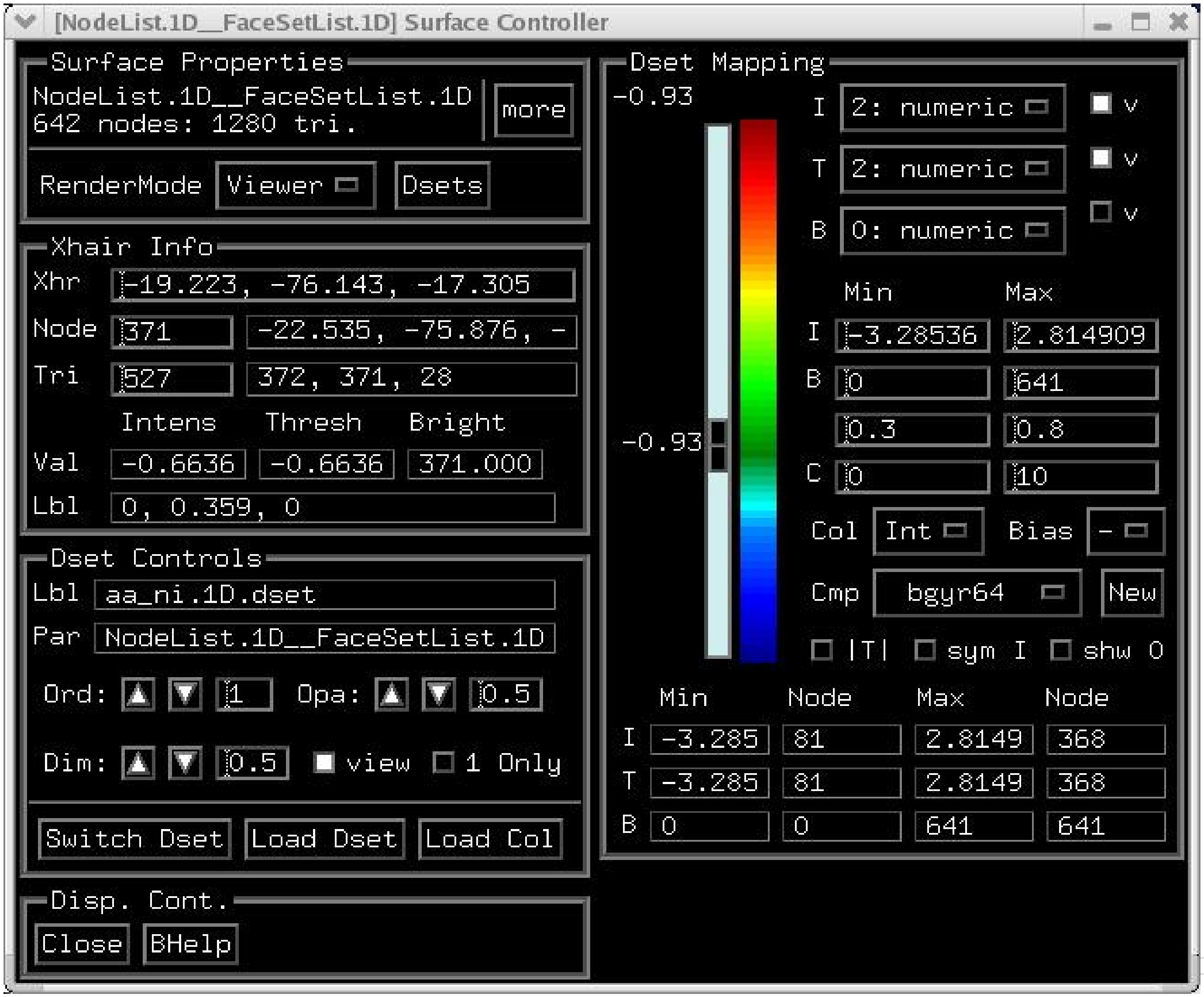
- Why did we insist on having an output for every node, although we have data on a just a few ?
- This file format (1D) is simple to understand however it is not efficient for reading and writing large files.
  - NIML Format that allows for faster I/O will be in use next.
- In some cases, the output file is larger than the limit of 2GB on most current systems. You can get around this limitation by writing the output file in chunks using the options -first_node and -last_node.

Surface-based Datasets (Dsets)

- These data sets form matrices with one column representing the node index (*ni*, a.k.a node id) followed by p values (*i_val. 1 … i_val. p*) associated with each node.
- Those *p* values can potentially be any assortment of parameters, though some dataset formats will be limited.
- In some instances, the node index column may be missing and the node's index is assumed to be equal to the row index.
- Unlike its volumetric counterpart, the domain over which the data are defined is not implicitly defined in the dataset.
  - Such a dataset alone cannot be visualized without specifying its surface domain (location and connectivity of the nodes).
- Dsets can be colorized (mapped to a colormap) offline using ScaleToMap (the olde way) or interactively (the in way) using SUMA's Surface Controller interface.
- A colorized data set is called a "color plane".

Colorizing Results Interactively

- Demo *(continued):*
  - In SUMA, press ‘ctrl+s’ to open Surface Controller.
    - Use View→Surface Controller if you were born after 1981.



Colorizing results interactively

- Demo *(continued):*
  - Press “Load Dset” and read in “out_Del.1D.dset”
    - SUMA will colorize the loaded Dset (create a color plane for Dset) and display it on the top of pre-existing color planes.
    - Set ‘1 Only’ in Dset Controls panel (more later)
    - We begin by describing the right side block “Dset Mapping” which is used to colorize a Dset. Many of the options mimic those in AFNI’s “define OverLay” controls.
    - Many features are not mentioned here. See online docs. and BHelp.
  - From the Dset Mapping block (right side of interface)
    - Select column 6 (6:numeric) for Intensity (I)
      - Values from column 6 will get mapped to the colorbar
    - Select column 8 for Threshold (T)
      - Press ‘v’ button to apply thresholding
      - Use scale to set the threshold. Nodes whose value in col. 8 does not pass the threshold will not get colored
      - Why do we see no difference when threshold is between 0 and 0.5?

Dset Mapping block

- Demo *(continued):*
- **Mapping Parameters Table:**
  - Used for setting the clipping ranges.
  - Clipping is only done for color mapping. Actual data values do not change.
- Col. *Min*:
  - Minimum clip value. Clips values (v) in the Dset less than Minimum (min): if v < min then v = min
- Col. *Max*:
  - Maximum clip value. Clips values (v) in the Dset larger than Maximum (max): if v > max then v = max
- Row *I*
  - Intensity clipping range. Values in the intensity data that are less than Min are colored by the first (bottom) color of the colormap. Values larger than Max are mapped to the top color.
- Left click locks ranges from automatic resetting.
- Right click resets values to full range in data.

Dset Mapping block

- Demo *(continued):*
- **Col:**
  - Switch between color mapping modes.
    - *Int*: Interpolate linearly between colors in colormap
    - *NN* : Use the nearest color in the colormap.
    - Dir: Use intensity values as indices into the colormap. In Dir mode, the intensity clipping range is of no use.
- **Cmp:**
  - Switch between available color maps. If the number of colormaps is too large for the menu button, right click over the 'Cmp' label and a chooser with a slider bar will appear.
  - Alternately, as with many of SUMA's menus, detach the menu by selecting the dashed line at the top of the menu list. Once detached, the menu window can be resized so you can access all elements in very long lists.
  - More help is available via ctrl+h while mouse is over the colormap Bias:

Dset Mapping block

- Demo *(continued):*
- ***The Colormap:***
  - *The colormap is actually a surface in disguise and shares some of the functions of SUMA's viewers:*
    - *Keyboard Controls:*
      - *r: record image of colormap.*
      - *Ctrl+h: this help message*
      - *Z: Zoom in.*

        Maximum zoom in shows 2 colors in the map
      - *z: Zoom out.*

        Minimum zoom in shows all colors in the map
      - *Up/Down arrows: move colormap up/down.*
      - *Home: Reset zoom and translation parameters*
    - *Mouse Controls:*
      - *None yet, some maybe coming.*

Dset Mapping block

- Demo *(continued):*
- **|T|:**
  - Toggle Absolute thresholding.
    - OFF: Mask node color for nodes that have:

      T(n) < Tscale
    - ON: Mask node color for nodes that have:

      | T(n) | < Tscale

  where: Tscale is the value set by the threshold scale.

  T(n) is the value in the selected threshold column (T).
- **sym I:**
  - Toggle Intensity range symmetry about 0.
    - ON : Intensity clipping range is forced to go from -val to val . This allows you to mimic AFNI's ranging mode.
    - OFF: Intensity clipping range can be set to your liking.
- **shw 0:**
  - Toggle color masking of nodes with intensity = 0
    - ON : 0 intensities are mapped to the colormap as any other values.
    - OFF: 0 intensities are masked, a la AFNI

Dset Mapping block

- Demo *(continued):*
- **Data Range Table:**
  - Full range of values in Dset.
    - Right click in “Node” columns to have cross hair jump to that node’s location.

Xhair Info block

- Demo *(continued):*
- **Xhr:**
  - Crosshair coordinates on this controller's surface. Entering new coordinates makes the crosshair jump to that location (like 'ctrl+j').
    - Use 'alt+l' to center the cross hair in your viewer.
- **Node:**
  - Node index of node in focus on this controller's surface. Nodes in focus are highlighted by the blue sphere in the crosshair. Entering a new node's index will put that node in focus and send the crosshair to its location (like 'j').
- **Node Values Table:**
  - Data Values at node in focus
    - Col. *Intens*: Intensity (I) value
    - Col. *Thresh*: Threshold (T) value
    - Col. *Bright*: Brightness modulation (B) value
    - Row. *Val*: Data Values at node in focus
- **Node Label Table:**
  - Row. *Lbl*:
    - Color from the selected Dset at the node in focus. For the moment, only color is displayed. The plan is to display labels of various sorts here.

Dset Controls block

- **Dset Info Table:**
  - Row. *Lbl*: Label of Dset.
  - Row. *Par*: Parent surface of Dset.
- **Ord:**
  - Order of Dset's colorplane. Dset with highest number is on top of the stack. Separate stacks exits for foreground (fg:) and background planes (bg:).
- **Opa:**
  - Opacity of Dset's colorplane. Opaque planes have an opacity of 1, transparent planes have an opacity of 0. Opacities are used when mixing planes within the same stack foreground (fg:) or background(bg:).
  - Opacity values are not applied to the first plane in a group. Consequently, if you have just one plane to work with, opacity value is meaningless.
  - Color mixing can be done in two ways, use F7 to toggle between mixing modes.

Dset Controls block

- **Dim:**
  - Dimming factor to apply to colormap before mapping the intensity (I) data. The colormap, if displayed on the right, is not visibly affected by Dim but the colors mapped onto the surface are.
  - For RGB Dsets (.col files), Dim is applied to the RGB colors directly.
- **view:**
  - View (ON)/Hide Dset node colors.
- **1 Only:**
  - If ON, view only the selected Dset's colors. No mixing of colors in the foreground stack is done.
  - If OFF, mix the color planes in the foreground stack.
  - This option makes it easy to view one Dset's colors at a time without having to worry about color mixing, opacity, and stacking order.
  - Needless to say, options such as 'Ord:' and 'Opa:' in this panel are of little use when this button is ON.

Colorizing results with ScaleToMap:

- This step is **no longer necessary** with SUMA's new interface for colorizing data.
- In the old days, ScaleToMap was the only way for getting node-based results into SUMA
- Execute: source run_ScaleToMap
- ScaleToMap

```
ScaleToMap   -input      out_Del.1D.dset  0  6     \
             -cmap       afni_p20                  \
               -apr      30                        \
             -msk        -3.0 -0.1                 \
             -nomsk_col                            \
             > out_Del.1D.col
```

  - -input : Specify the 1D format input file and the columns containing the node index and the node data
  - -cmap : Specify the colormap to use.
    - Here we are using AFNI's standard colormaps (positive, 20 colors).
    - Could use your own colormaps, see MakeColorMap
  - -apr: AFNI positive range value.
    - Scaling a la AFNI, other scaling methods that do not require 0 → range or –range →range are also available
    - When data are integer valued (such as ROI datasets), you can use a direct mapping where a node with value i gets mapped to the ith color in the map.
  - -msk: Mask data values in the specified range (inclusive)
  - -nomsk_col: Do not color masked nodes (default is a dark gray).
    - You can specify your own mask color
  - The output of ScaleToMap is to stdout, which can be redirected with ' > ' symbol to a text file

Viewing a color file in SUMA

- A color file, such as ScaleToMap's output out_Del.1D.col is of the format:

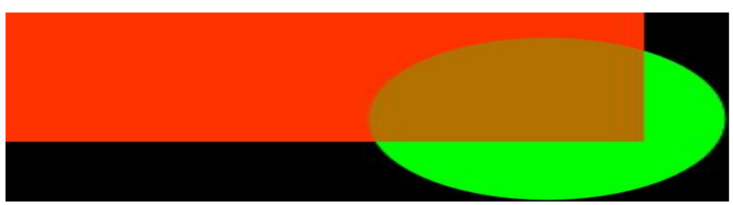
#Node_Index	R	G	B
23	0.3	0.8	0.6
127	0.8	0.91	0.12
….	…	….	…

- You can load it into SUMA with the 'c' key or 'Load Col'
  - The color data is loaded into a new data set (Dset) of the type RGB. No "Dset Mapping" block is available for RGB Dsets.
  - The recently loaded Dset obscures planes below it because it has a default opacity of 1
    - We will discuss color plane opacity and order next.

Color overlay planes

- Colorized Dsets are organized into layered color planes
  - 2 commonly used planes are:
    - Surface Convexity (usually in gray scale)
    - AFNI Function (usually in color)
  - Planes are assigned to two groups
    - Background planes (like Convexity)
    - Foreground planes (like AFNI Function)
  - Many other planes can be added to either group.
- Color planes of the same group are mixed together:
  - Planes are stacked based on their order and opacity.
  - Opacity of 1st plane does not affect color mixing.
  - There are 2 modes for mixing colors. See option F7 in SUMA.

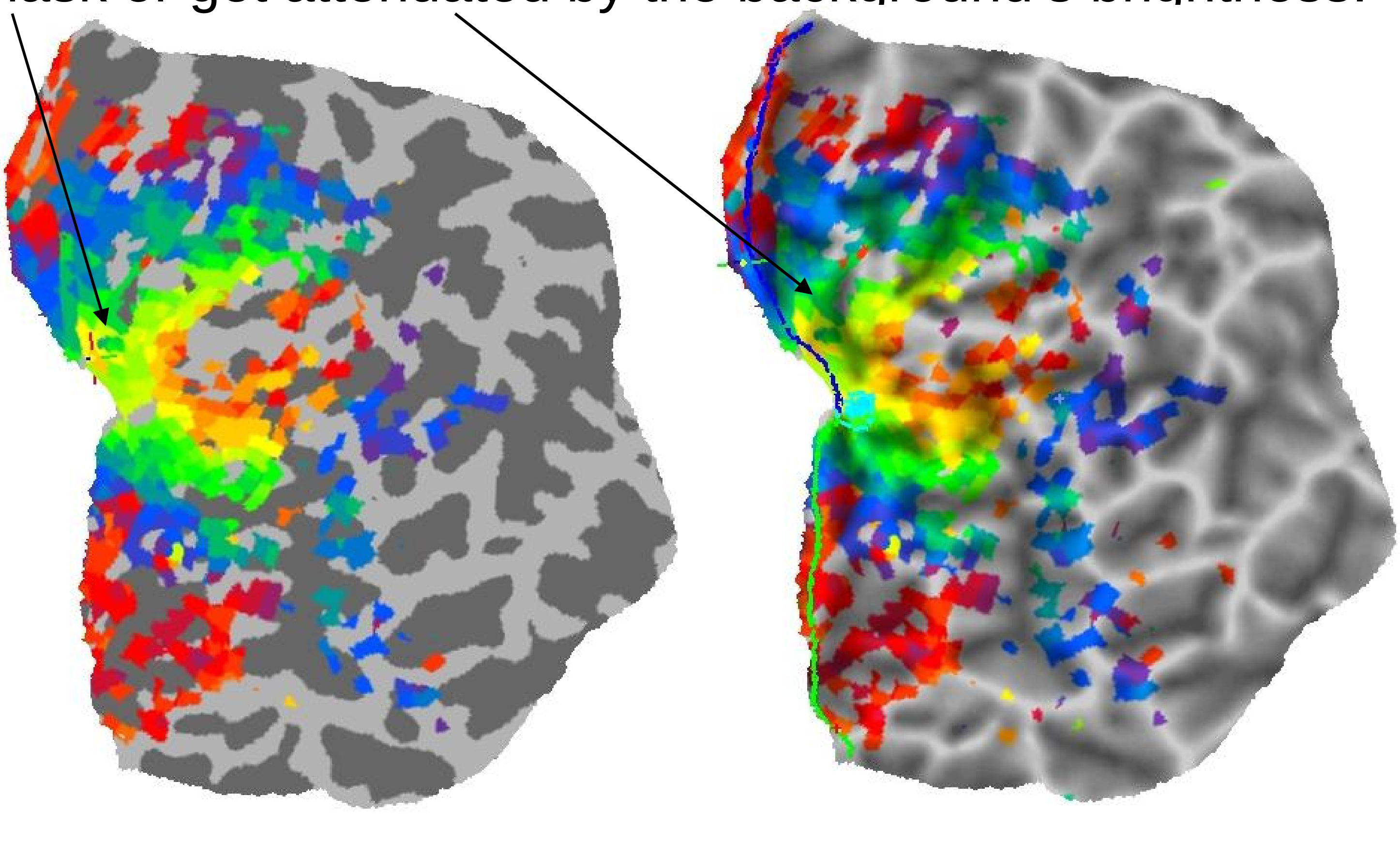
Plane 1: 80% opacity



Plane 2: 30% opacity

Color overlay planes

- Node colors displayed on surface are obtained by:
  - 1st: mixing background planes
  - 2nd: mixing foreground planes
  - 3rd: layering mixed foreground atop mixed background plane
    - When foreground colors overlap background colors, they either mask or get attenuated by the background's brightness.



Layering mixed fore- & background plane

- Demo *(continued):*
  - View the surface in its inflated state with AFNI function
  - Turn foreground plane(s) off by pressing 'f' once
    - Now all you see is the background plane(s)
  - Turn background planes off by pressing 'b' once
    - Now all you see is the No Color color on all nodes
  - Turn foreground plane(s) back on with 'f'
    - Now you have foreground without background
  - Turn background plane(s) back on with 'b'
    - Now you have foreground atop background
    - Notice how you can still see the background underneath the foreground—this is due to the background brightness attenuation of the foreground colors.
  - Toggle background intensity attenuation off and on with 'a' and see the effect on the resultant maps.

Playing with color plane opacity

- Demo *(continue from inflated view with function)*
  - ‘ctrl+s’ or View → Surface Controller to open the surface controller
    - Turn OFF ‘1 Only’
  - Load in color plane lh.1D.col with ‘Load Col’ or ‘c’
    - This is an RGB Dset, color mapping controls are hidden
    - Plane is placed atop of the foreground group
    - Its opacity is 1 so it will obscure the functional data
    - Background attenuation is not affected by plane’s opacity.
      - try turning it on and off again with ‘a’
    - Now lower the opacity of lh.1D.col and watch the colors from the planes below start to show through

Playing with color plane order

- Demo *(continue from inflated view with function)*
  - You could put lh.1D.col below the function
    - Switch Color Plane to get a list of available planes
    - Prefixes fg: and bg: denote plane's group membership
    - Select lh.1D.col and lower its order
    - Select FuncAfni_0 and play with its opacity
    - Note: You can't make a plane change its group membership, yet.
  - You can't delete a loaded color plane yet, but you can hide it.
  - Turn '1 Only' ON if you just want to see the selected plane.
- Test
  - Find a way to flip between the mapping from AFNI and the mapping done with 3dVol2Surf before.
    - Appreciate the differences between the two mappings.

Recording your rendered images

- Using ‘r’ in SUMA to record the current scene.
- Using ‘r’ on the colorbar creates an image of the colorbar.
- Using ‘R’ to record continuously the rendered scene.
- Images are captured by an AFNIesque viewer.
  - Identical consecutive images are rejected
  - Images caused by X expose events are ignored
  - Images can be saved in all ways allowed by AFNI, including MPEG and animated GIF
  - If you let the recorder run continuously with very large images, you might quickly run out of memory
- You can save/load viewer setting used to create figure
  - Use ‘File→Save’ View and ‘File→Load View’

Hands-On

Drawing surface-based ROIs

- Demo
  - ‘Ctrl+d’ or Tools → Draw ROI to open ROI drawing tool
    - When in Draw ROI mode (cursor turns into target circles)
      - The pick (usually third) mouse button is used for drawing
      - Picking is done by combining shift key and pick button
    - If pen mode is selected (cursor turns into a pen)
      - Drawing and picking (with shift) are done with the 1st mouse button and rotations are done with the third button
  - When you draw for the first time, a new “drawn ROI” is created.
    - Note the Parent: field in the ROI frame is now filled

Drawing surface-based ROIs

- Demo
  - Move the mouse to a new location and click again to draw a line from the location of the first click to a new one.
  - Or click while dragging the mouse to create a smoother line.
  - Both drawing methods might fail if you are drawing over very rough terrain.
    - When that happens, you continue from where the drawing stopped.
    - Drawing in 3D looks easy but it isn't.
  - Use Undo and Redo when you get your drawings messed up.
  - To close a loop you can press 'Join' button or double click

Drawing surface-based ROIs

- Demo
  - To fill a closed loop, click inside the loop.
    - Note: If you have patterns that make a figure 8 jealous you might have to do multiple fills.
  - Press Finish when you are done drawing and have set the drawing label and value to your liking.
  - Now you can draw another ROI, and another and another.
  - Drawing can be started/continued on any of the related surfaces
    - When you hit ‘join’, the loop is closed using the surface where the ROI was created
  - Switch ROI to switch between ROIs and delete/modify them.
  - Load to load ROIs from file
  - Save to save ROIs to file
    - 1D/Niml Format of ROI file
    - This/All which ROI to save into the file

Complex ROIs in NIML format

- Delete all the ROIs you have created so far and draw two overlapping ROIs in the occipital cortex.
- Save both ROIs into a NIML format and call the output OccROIs
- The ROIs are saved into a file similar to the one show below:

```
# <Node_ROI
#  ni_type = "SUMA_NIML_ROI_DATUM"
…
#  idcode_str = "XYZ_pyoAQRr-j8KAsyW87xg3Xw"
#  Parent_idcode_str = "XYZ_MH4NN7U51kCKOdJMo9LdnQ"
#  Label = "ROI_1"
#  iLabel = "1"
#  Type = "2"
#  ColPlaneName = "DefROIpl"
#  FillColor = "0.000000 0.000000 0.625000"
#  EdgeColor = "0.000000 0.000000 1.000000"
#  EdgeThickness = "2"
# >
 1 4 40 … 1086 1497 1512 ...  # </Node_ROI>

# <Node_ROI
….
#  Label = "ROI_2"
#  iLabel = "19"
#  Type = "2"
#  ColPlaneName = "DefROIpl"
# >
 1 4 203 3241 ….. # </Node_ROI>
```

Surface ROI → Volume ROI

- ROIs files should be transformed to dataset files (Dset).
  - A dataset file has an equal number of values for each node
  - The way we drew the two ROIs, certain nodes belong to more than one ROI (i.e. they have more than one value)
- The program ROI2dataset is used to change ROIs to datasets
- Execute: source run_ROI2dataset

```
ROI2dataset    -prefix  OccROIs.1D.dset    \
               -of      1D                 \
               -input   OccROIs.niml.roi   \
```

  -prefix: Name of output file

  -of : Output format

  -input : Name of input ROI file

- Note the warning:

  Warning SUMA_ROIv2dataset:
  155/2536 nodes had duplicate entries.
  (ie same node part of more than 1 ROI)
  Duplicate entries were eliminated.

  - At the moment, duplicate entries are ignored, 1st come, 1st adopted. More options could be added if you find necessary.

Surface ROI → Volume ROI

- Output file OccROIs.1D.dset is of the form:

```
124     1
125     1
---
2882    19
---
```

- OccROIs.1D.dset can be transformed into a volume ROI with 3dSurf2Vol
- Execute: source run_3dSurf2Vol by Rick Reynolds

```
3dSurf2Vol  -spec../SurfData/SUMA/DemoSubj_lh.spec        \
            -surf_A       lh.smoothwm.asc                  \
            -surf_B       lh.pial.asc                      \
            -sv           DemoSubj_SurfVol_Alnd_Exp+orig   \
            -grid_parent  DemoSubj_SurfVol_Alnd_Exp+orig   \
            -map_func     max                              \
            -f_steps      10                               \
            -f_p1_mm      -0.5  -f_pn_fr 0.5               \
            -sdata_1D     OccROIs.1D.dset                  \
            -prefix       OccROIs
```

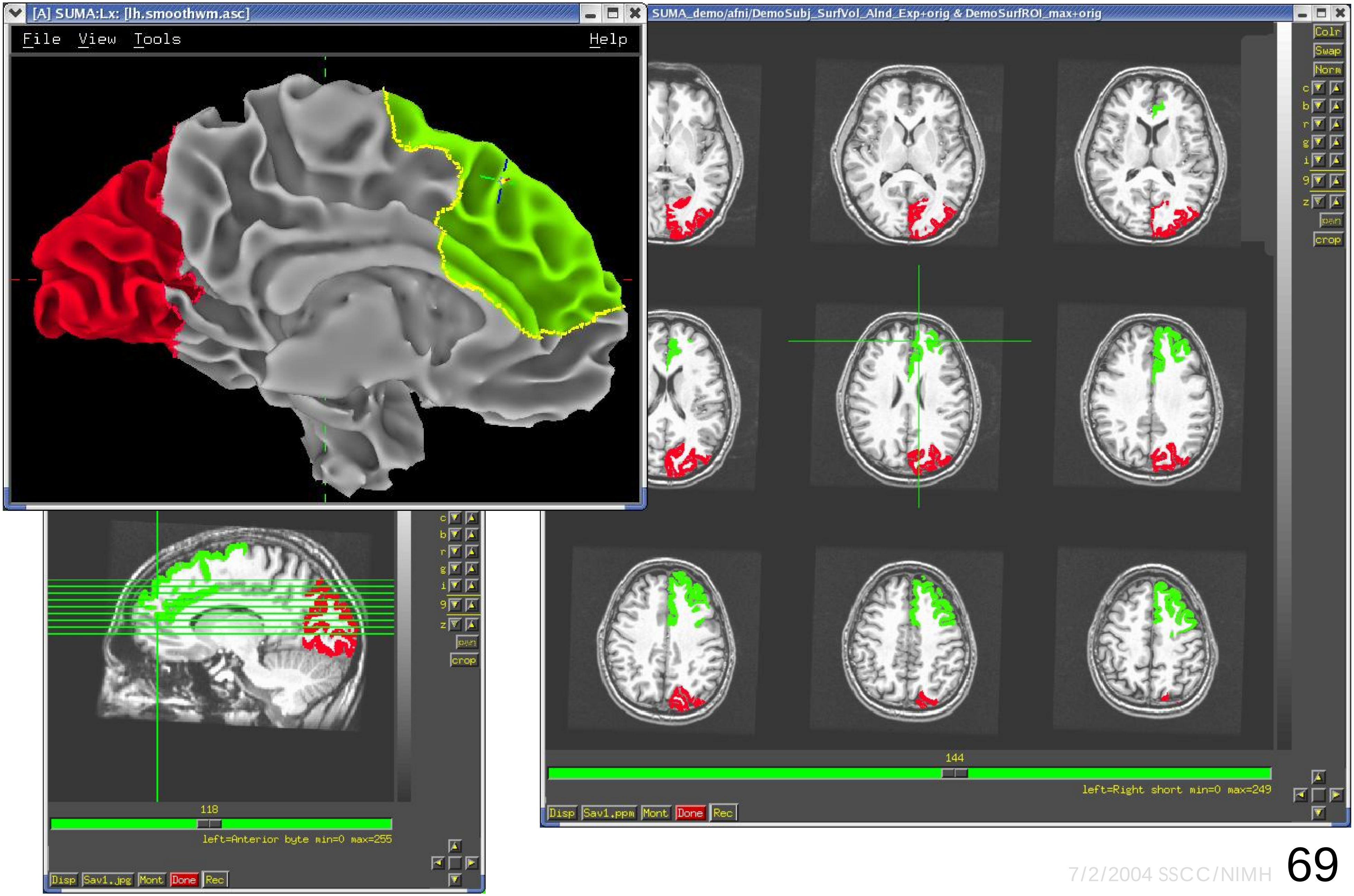
-grid_parent specifies the output volume's geometric properties

-f_p1_mm, -f_pn_fr: Specify extensions on node pair segment, either in mm or fractions of segment length (both used simultaneously here for illustration)

-sdata_1D: Specify surface data file

Surface ROI → Volume ROI

check your own OccROIs+orig in AFNI



Smoothing data on surface models

- Data smoothing (along the surface) is done with SurfSmooth
  - The difficulty in smoothing lies in calculating geodesic distances between node pairs.
  - Smoothing can be performed by solving the diffusion equation on the surface. This can be achieved in a relatively rapid manner with an iterative algorithm (M.K. Chung et al. 2003)
- Demo *(close SUMA and afni)*
  - Execute: source run_SurfSmooth_data
    - The script will use 3dVol2Surf to map the delay data onto the surfaces (uses slightly different options from previous example)
    - SUMA is then launched and the script waits for you to setup the recorder ON
    - Hit enter twice (in the shell) to launch the smoothing program
    - Watch the smoothing progress with each iteration. You can play the whole sequence at the end.

How many iterations ?

- Command line:

```
SurfSmooth  -spec    ../SurfData/SUMA/DemoSubj_lh.spec     \
            -surf_A  lh.smoothwm.asc                       \
            -met LB_FEM                                    \
            -input out_Del2.1D'[6]'                        \
            -Niter 100                                     \
            -fwhm 8                                        \
            -add_index                                     \
            -output out_Del2_smoothed.1D -talk_suma
```

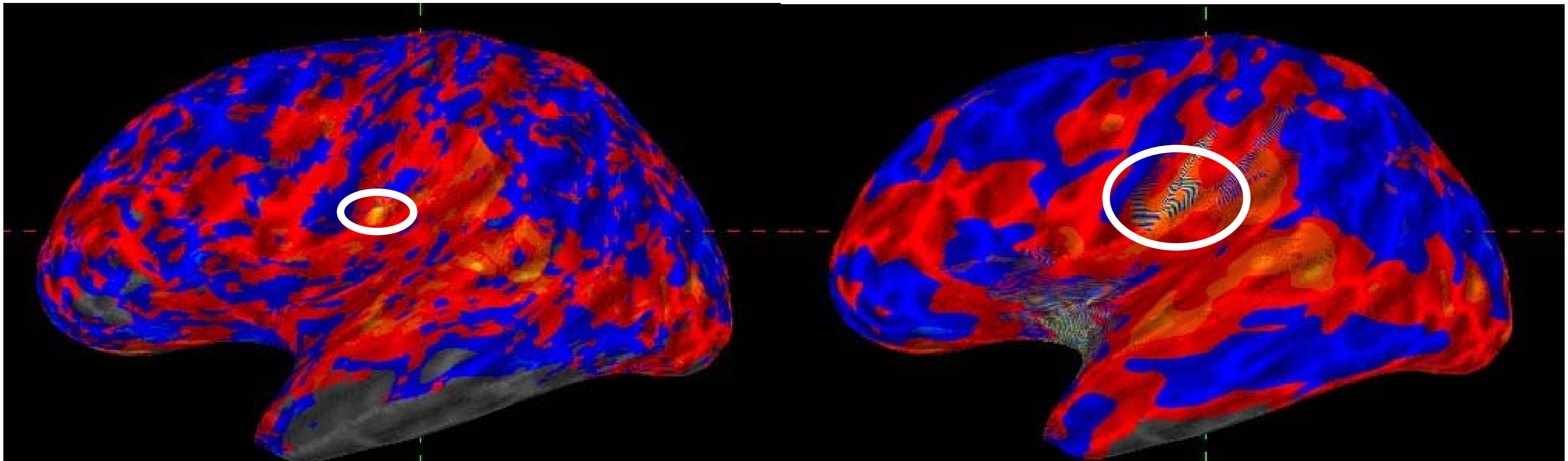
  -met : Specifies the smoothing method.

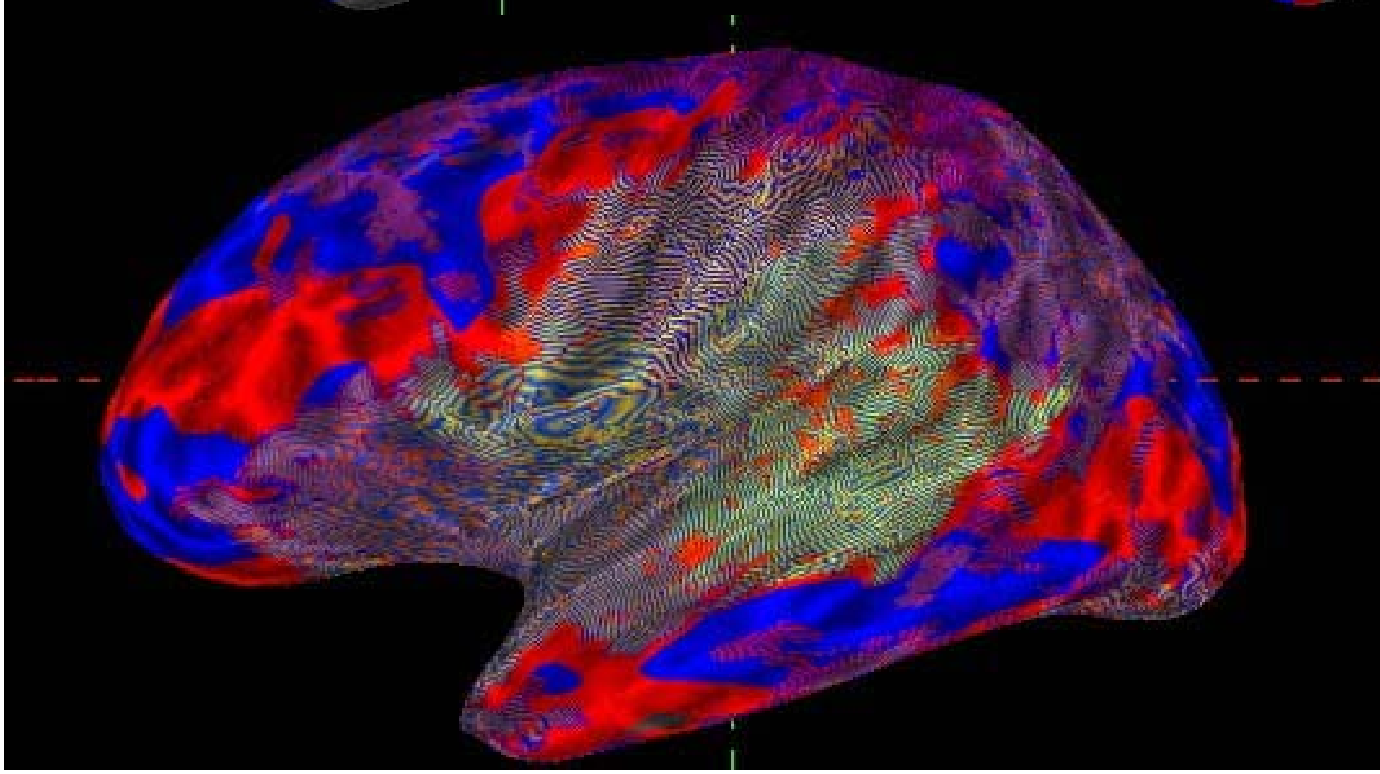
  -Niter: number of iterations

  -fwhm: Full Width Half Max of filter in surface coordinate units (usually mm) of an equivalent Gaussian filter had the surface been flat. With curved surfaces, the equation used to estimate FWHM is an approximation.

- The number of iterations controls the iteration step dt which must satisfy conditions that depend on the internodal distance and the spatial derivatives of the signals being filtered on the surface.
- As a rule of thumb, if increasing Niter does not alter the results then your choice is fine (smoothing has converged).

Smoothing gone bad





Solution:

Increase Niter

Smoothing geometry of surface

- Surface geometry smoothing can also be done with SurfSmooth
  - The difficulty in geometry smoothing is in preventing the surface from shrinking in size.
  - SurfSmooth uses the smoothing algorithm by (Taubin G. 2000)
- Demo *(close SUMA and AFNI)*
  - Execute: source run_SurfSmooth_geom
    - The script will add noise to the smoothwm surface, then filter it
    - SUMA is then launched and the script waits for you to setup
      - Turn off background colors (‘b’)
      - Switch to noisy surface (‘.’)
      - Turn recorder ON
    - Hit enter twice (in the shell) to launch the smoothing program
    - Watch the smoothing progress with each iteration. You can play the whole sequence at the end.

Talairach data display, for panache

- Without creating individualized surfaces, you can display data on a Talairach-ed surface model.
  - The surface models were created with FreeSurfer from the N27 data set. (Holmes, CJ et al. JCAT 1998)
  - Anatomical volume and surfaces were Talairach-ed using AFNI and SUMA, respectively. (Surfaces created by Brenna Argall)
- Demo: (close previous SUMA and AFNI sessions)
  - cd suma_demo/afni_tlrc
    - You should find Anat+tlrc and a functional data set in Talairach space
  - afni –niml &
    - setup function as shown in the earlier slides
  - suma –spec ../SurfData_tlrc/SUMA/lh.tlrc.spec –sv Anat+tlrc
    - start communication with AFNI
    - map function onto surface
    - appreciate the limitation of this approach
    - ignore the limitation and make cool pictures

Auxiliary programs

- To see a list of all programs in SUMA:
  - suma –progs
- 3dSurf2Vol (by R. Reynolds)
  - Maps data in surface domain volumetric domain
- 3dVol2Surf (by R. Reynolds)
  - Maps data in volume domain to surface domain
- CompareSurfaces (by S. Japee)
  - calculates the distance along the normal from one surface to the next
- ConvertSurface
  - Converts surfaces between the different formats that SUMA can read:
    - FreeSurfer
    - SureFit
    - Simple ASCII matrix format
    - PLY format
  - Converts surface coordinates to Talairach space
- CreateIcosahedron (by B. Argall)
  - Creates Icosahedral meshes of varying node counts

Auxiliary programs

- inspec
  - Outputs information about a spec file
- MakeColorMap
  - Creates colormaps for use with AFNI
- MapIcosahedron (by B. Argall)
  - transforms the topology of surface models to standard models without distorting the geometry (tested with FreeSurfer’s spherical mapping).
  - use to check for errors in topology of spherical surfaces
- quickspec
  - Creates a spec file for one or a set of surfaces (quick and dirty)
- ROI2dataset
  - Transforms ROI files into surface data sets

Auxiliary programs

- ScaleToMap
  - Transforms a set of node values to node colors based on chosen color map
- SurfMeasures (by R. Reynolds)
  - Outputs node-based surface measures such as areas, volumes, thickness, surface normals, etc. etc.
- SurfaceMetrics
  - Outputs other information about the surface such as the edge list, curvature, triangle areas (mostly for debugging use)
- SurfSmooth
  - a program for filtering surface data and/or surface geometry
- SurfPatch
  - Creates surface patches from a set of nodes
- SurfQual
  - Locates topological errors in spherical surfaces.

Getting Helped

- suma –help for SUMA's command line usage
- 'ctrl+h' opens a window with help for:
  - SUMA's usage when cursor is in viewer
  - SUMA's colormap usage when cursor is over colormap
- BHelp provides help for most buttons in the GUI interface
- SUMA's web site:
  - http://afni.nimh.nih.gov/ssc/ziad/SUMA/
  - http://afni.nimh.nih.gov/ssc/ziad/SUMA/SUMA_Installation.htm
  - http://afni.nimh.nih.gov/ssc/ziad/SUMA/SUMA_doc.htm
    - You can download a PDF of this one.
- AFNI's Message Board
- E-mail to ziad@nih.gov
- If you can't get help here, please get help somewhere.

Thanks to:

Developers:
- Brenna Argall
- Shruti Japee
- Rick Reynolds

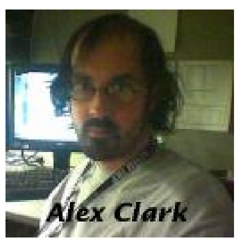
Critics:
- Mike Beauchamp
- Pat Bellgowan

Data and Documentation
- Peggy Christidis

Getting Help Somewhere:
- Richard Doucette

Testing: Samia Saad

SysAdmin:
- Alex Clark



MRI Pulse Sequence and Segmentation
- Hauke Heekeren
- Sean Marrett