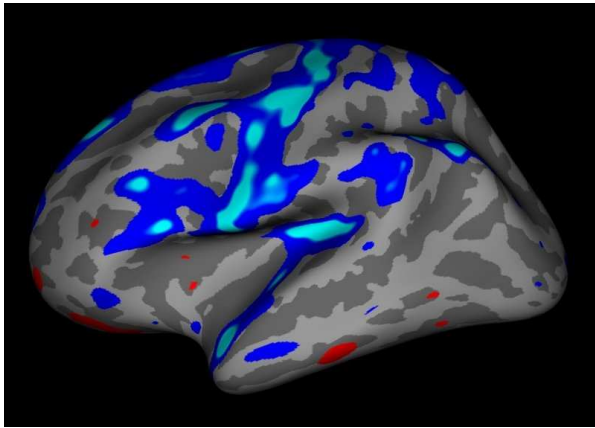# FreeSurfer Tutorial

FreeSurfer is brought to you by the Martinos Center for Biomedical Imaging
(supported by NCRR/P41RR14075), Massachusetts General Hospital, Boston, MA USA

2008-06-01 20:47

FreeSurfer Tutorial

Table of Contents

2008-06-01 20:47

---

top

FreeSurfer Slides

1.

   Introduction to Freesurfer - Bruce Fischl

2.

   Anatomical Analysis with Freesurfer - Doug Greve

3.

   Surface-based Group Analysis - Doug Greve

4.

   Applying FreeSurfer Tools to FSL fMRI Analysis - Doug Greve

FreeSurfer Tutorial

Overview

The FreeSurfer tools deal with two main types of data: volumetric data (volumes of voxels) and surface data (polygons that tile a surface). This tutorial should familiarize you with FreeSurfer's volume and surface processing streams, the recommended workflow to execute these, and many of their component tools. The tutorial also describes some of FreeSurfer's tools for registering volumetric datasets, performing group analysis on morphology data, and integrating FSL Feat output with FreeSurfer (overlaying color coded parametric maps onto the cortical surface and visualizing plotted results). After completing the tutorial, you should be able to:

- perform surface reconstructions;
- generate subcortical segmentations;
- fix errors encountered during the volume or surface processing;
- overlay functional data onto surfaces;
- perform group analysis of structural (e.g. thickness) and functional data.

Course Outline

In the following sessions, you will be shown a variety of command strings. Only those that appear between lines should be copy-and-pasted into the terminal for this tutorial. Those commands appear like this:

```
command arg1 arg2
```

-

  Session 1
  o

     Session 1a - Use the volume and surface viewing tools to observe correctly processed output data.
  o

     Session 1b - See examples of problematic output data, and learn how to fix the problems.
  o

     Extra - Long-form tutorial covering Freesurfer's morphometry and reconstruction tool recon-all.
  o

     Extra - recon-all process-flow reference table.

-

  Session 2
  o

     Session 2a - Learn how to conduct a group analysis using Qdec, and to visualize and inspect the results.
  o

     Session 2b - Applying FreeSurfer tools to FSL fMRI Analysis (FEAT)
  o

     Extra - Group Analysis tutorial using mri_glmfit from the command line, prior to the release of Qdec.
  o

     Extra - Visualization and inspection of group analysis results using tksurfer, prior to release of Qdec.

Additional

-

  Home page: http://surfer.nmr.mgh.harvard.edu
-

  Mailing list: send mail to majordomo@surfer.nmr.mgh.harvard.edu with the following command in the body of your email message: subscribe freesurfer
-

2008-06-01 20:47

Inspection of Freesurfer Output

- To follow this exercise exactly be sure you've downloaded the tutorial data set before you begin. If you choose not to download the data set you can follow these instructions on your own data, but you will have to substitute your own specific paths and subject names. If you are using the tutorial data please set the environmental variable TUTORIAL_DATA to the location that you have downloaded the data to (here, it has been copied to $FREESURFER_HOME/subjects):

```
tcsh
setenv TUTORIAL_DATA $FREESURFER_HOME/subjects/buckner_data/tutorial_subjs
```

- Notice the command to open tcsh. If you are already running the tcsh command shell, then the 'tcsh' command is not necessary.

In this exercise you will visualize and inspect correctly processed output data so that you can become familiar with what the end product should look like. The exercise will step you through visual inspection of a variety of output, but is not necessarily the recommended procedure to take when trying to verify each subject. Some steps are only necessary to check when there are problems. However it is a good idea for new users to become familiar with what the expected output should look like and how to view it.

First you need to set your SUBJECTS_DIR to the appropriate place:

```
setenv SUBJECTS_DIR $TUTORIAL_DATA
cd $SUBJECTS_DIR
```

* this will set your SUBJECTS_DIR to the location where your tutorial data is if you have defined the variable TUTORIAL_DATA as indicated at the top of this tutorial. If you are not using the tutorial data you should set your SUBJECTS_DIR to the directory in which the subject you will use for this tutorial is located.

Viewing Volumes with Tkmedit

The volumes that are output can be loaded into tkmedit, along with surface outlines and the subcortical segmentation. With one command line you can load in the brainmask.mgz and wm.mgz volumes, the rh.white and lh.white surfaces (outlines), and the subcortical segmentation.

```
tkmedit good_output brainmask.mgz lh.white \
        -aux T1.mgz -aux-surface rh.white \
        -segmentation aseg.mgz $FREESURFER_HOME/FreeSurferColorLUT.txt
```

You should see a tkmedit window open up to this:

(use the zoom 🔍+ and move ✛ buttons to match this image)
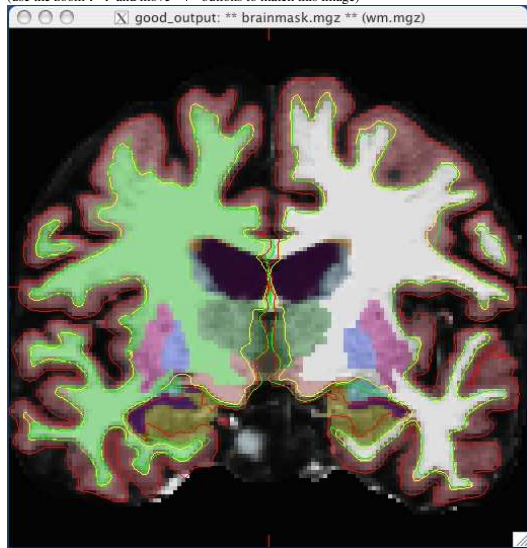
You are currently looking at the brainmask.mgz (loaded as the main volume) with the surfaces displayed and the aseg.mgz (subcortical segmentation) overlayed. The pial (red line), white (yellow line) and orig (green line) surfaces are all shown. You can toggle between the brainmask.mgz (main volume) and the wm.mgz (aux volume) and you can click on and off the aseg.mgz overlay. To become more familar with the buttons in the Tkmedit Toolbox please read the Freesurfer Tools section of this tutorial.

Here are the things you can look at while this is loaded in tkmedit:

- intensity normalization
- skull strip
- wm.mgz volume
- the final surfaces
- subcortical segmentation

For the first run through you might find it easiest to toggle off the aseg with the ⊞ button. Using ctrl-g on the keyboard will have the same effect. You may also find it easier if you toggle off all the surfaces with the 🌀P , 🌀O , and 🌀M buttons.

Intensity Normalization

Scroll through the brainmask volume and notice that the intensity is all uniform. You should not see any very bright or very dark spots. If you click on any voxel that is in the wm you can see that it has been normalized to an intensity of (or very close to) 110. The voxel intensity is shown in the Tkmedit Toolbox. This check and the following, Skull Strip check, can be done simultaneously since they both require you to look at features on the brainmask volume.

Skull Strip

Scroll through the brainmask volume and notice that there is no skull left in your image. Notice also that the cerebellum is still included in the brainmask volume. You should not see any large areas of skull left behind, or any areas of cortex or cerebellum removed from this volume. You should compare the brainmask.mgz volume to the T1.mgz volume that is also loaded to ensure that the skullstrip has worked properly. You can switch between views in a number of ways, ctrl-1 will show the main volume and ctrl-2 will show the auxiliary volume, or you can use the buttons 🔲 and 🔲 . This check and the previous, Intensity Normalization check, can be done simultaneously since they both require you to look at features on the brainmask volume.

White Matter Volume

To check the wm volume you should load it in as a new aux volume. To do this go to **File -> Aux Volume -> Load Aux Volume** and browse to wm.mgz. This will open the white matter volume, which will look like this:
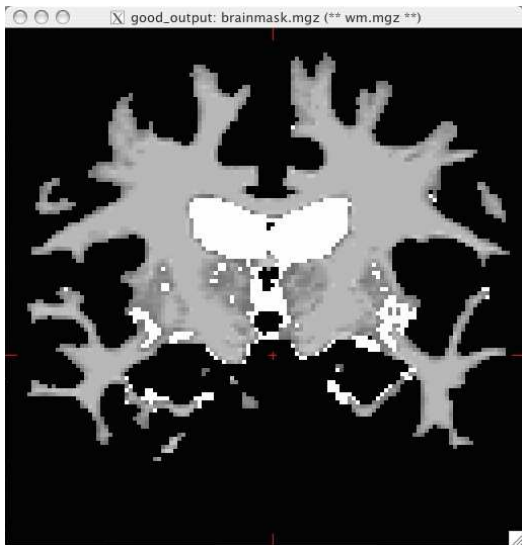
This volume is comprised of all the voxels that freesurfer is calling white-matter, shown in shades of gray. These are the voxels that were normalized to an intensity of, or very close to, 110 as described above. The bright white voxels are voxels that have been added to the volume during the automatic editting of the wm volume. These edits fill the entire ventricle and basal ganglian defect. You can alternate between the wm.mgz volume (⬛ button) and the brainmask.mgz volume (⬛ button) to see how well freesurfer has classified the white matter.

Final Surfaces

Switch back to the main volume by using the ⬛ button (ctrl-1 on the keyboard will also do this), this will show the brainmask.mgz volume. To check your surfaces you will need to toggle them back on with the

---

P for the pial surface, M for the white surface, and o for the orig surface. The surfaces that are overlaid are the pial surface (red line), white surface (yellow line), and orig surface (green line). The orig surface is the "first guess" at the boundary between the white matter and gray matter. After topology fixing and some other steps the white surface is generated. The white surface is the best and final estimation at the boundary between the white matter and gray matter. The white surface and the orig surface will appear nearly identical, but there will be regions where they differ as a result of the topology fixing and smoothing that occurs. The white surface is the surface used in all calculations of thickness so it is important that this surface follows the boundary of the white matter accurately. The pial surface should accurately follow the boundary between the gray matter and the CSF. As you scroll through the slices keep in mind that you are looking at a 2-dimensional rendering of a 3-dimensional image, be sure to look at more than just one view too (i.e., sagittal, coronal and horizontal).

There are regions where the surfaces are not intended to be accurate that you should be aware of. Areas around the hippocampus and amygdala, as well as along the midline cutting plane will often show some inaccuracies. The pial surfaces will not follow the border of the amygdala, instead it will curve inward, mimicking the white surface (see coronal slice 137). Along the midline cut it is possible to see some overlapping of the surfaces from one hemisphere to another.

Subcortical Segmentation

Toggle on the subcortical segmentation with the ⊞ button. This will show the complete segmentation of the subcortical structures. Each structure is labeled with a unique color/number distinction. If you click on a voxel the structures name and number label will be shown in the Tkmedit Toolbox. Scrolling through the slices you will be able to see that everything is labeled, and done so accurately. Sometimes it is easier to see the structures and their boundaries looking in either the sagittal or horizontal view, so be sure to check around in all of them.

Aparc+Aseg segmentation

To load in the aparc+aseg.mgz segmentation you can go to **File --> Load Segmentation**, and browse to the aparc+aseg.mgz, which will look like this:
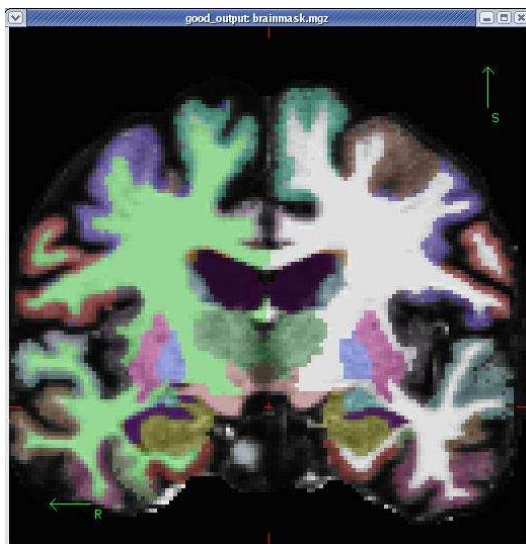
This segmentation shows the same subcortical structures that are labeled in the aseg.mgz, but uses the cortical parcellation labels around the cortex.

Viewing Surfaces with Tksurfer

Now that you've checked out everything in tkmedit you can close it and begin to inspect the surfaces that are output, for this you will use tksurfer. Tksurfer displays one hemisphere at a time. This exercise will go through visualizing things on the left hemisphere only, but everything works the same on the right hemisphere (except in the initial command you should specify rh instead of lh if you want to look at the right hemisphere). To become more familar with the buttons in the Tksurfer Toolbox, please read the Freesurfer Tools section of this tutorial.

Here are the things you can look at with tksurfer:

- pial, white and inflated surface
- sulc and curv curvature files
- thickness files
- cortical parcellation

To open tksurfer with the left hemisphere inflated surface of your subject, use the following command:

```
tksurfer good_output lh inflated
```

You should see a tksurfer window open up to this:
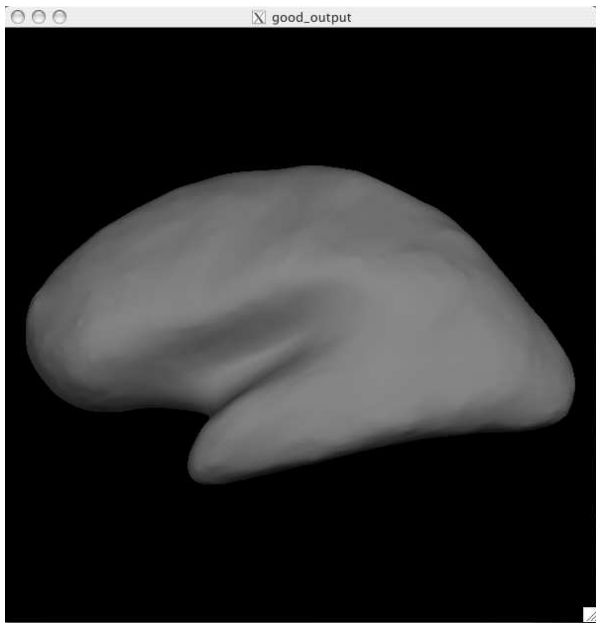
You are currently looking at the inflated surface. The surface can be rotated using the buttons in the navigation toolbar: . Use the redraw button to repaint the image if it gets corrupted by window movement.

Inflated surface

The inflated surface is good to look at when checking to see if you need to make edits to the wm.mgz volume. You'll notice as you inspect this surface that it is smooth and free from holes, bumps and other defects. If you click on the surface, you will see the coordinates of the vertex you clicked on in the Tools window. To clear the marks made on the surface after clicking on it, you can do Ctrl+Shift+middle click on each mark to erase it.

Pial Surface

You can load in other surfaces, and tksurfer will then allow you to switch between them all. The easiest way to do this is to hold down CONTROL and click with the right mouse button on the various surface buttons. To load the Pial Surface hold ctrl and right click the pial surface button a box will pop up where you could browse to the location of the surface you want to load, but it should be already filled in with the path to the lh.pial surface - so you can hit ok. This is what the pial surface will look like:
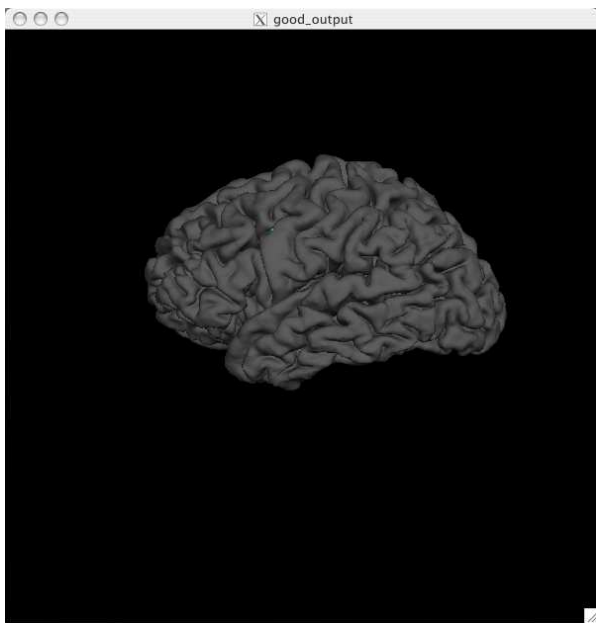
The pial surface is showing you the outer boundary of the gray matter/CSF. This is the same file that was viewed in tkmedit, just represented as a surface image rather than the red outline on the volume. You can inspect this surface by rotating it around as you wish.

White Surface

You can follow the same procedure that you used to load the pial surface for the white surface, except this time hold ctrl and right click the white surface button You can inspect this surface by rotating it

around as you wish. This is what the white surface will look like:
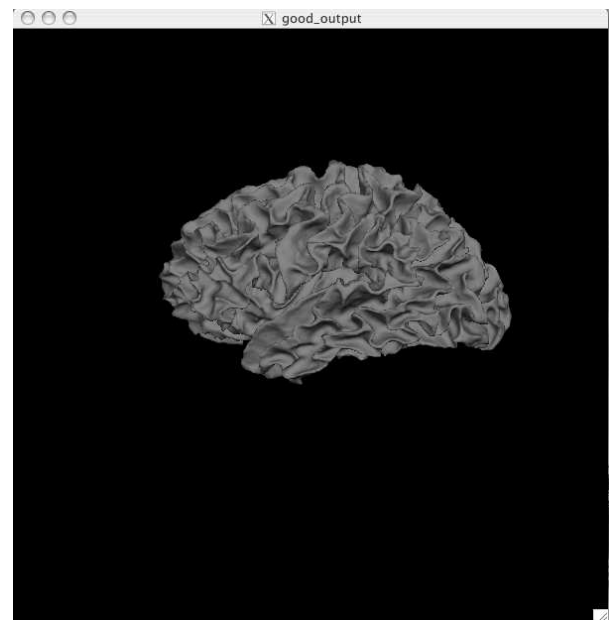
The white surface shows the boundary between the white and the gray matter. Again, this is the same file that was viewed in tkmedit (as the yellow outline) just represented in a 3D manner.

Curv and Sulc Files

Switch back to viewing the inflated surface of the brain by pushing the ⬚ M button. You can now load in the curvature file, lh.curv, by holding ctrl and right clicking on the ⬚ button. The box that pops up will automatically have selected the lh.curv file, so click ok. The curvature file will look like this:
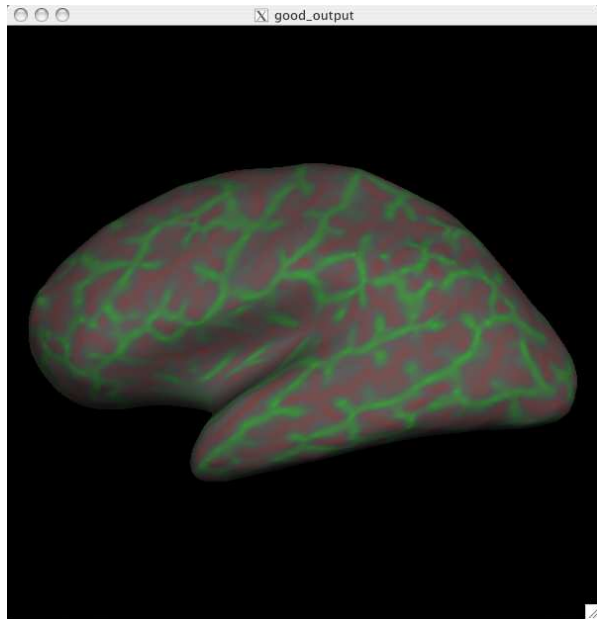
This is showing the slightly smoothed mean curvature. It has the units of 1/mm, and with an outward pointing normal vector field. Negative regions are folded-out and shown in green (gyral), and positive regions are folded-in and shown in red (sulcal)

To view the sulc file you could have changed the dialog box to say lh.sulc, or now you can go to **File -> Curvature -> Load Curvature** and browse to the lh.sulc file (found in $FREESURFER_HOME/subjects/buckner_data/tutorial_subjs/good_output/surf/). Be sure you select the curvature file for the correct hemisphere, if you select the wrong one it will not look right. The lh.sulc file will load a slightly different display of green/red values than the lh.curv. These are showing the sulcal depth and again here the red regions are sulcal and the green regions are gyral. You can view this as you click on the pial and/or white surfaces.

Thickness Maps

To view the thickness maps it is probably best to toggle off the curvature ⬚ . You can load the thickness map, lh.thickness, by holding ctrl and right clicking ⬚ and browse to the lh.thickness file (you may have to go up a directory - it's found in $FREESURFER_HOME/subjects/buckner_data/tutorial_subjs/good_output/surf/). Be sure you select the thickness file for the correct hemisphere, if you select the wrong one it will not look right. The lh.thickness map should open up and look like this (after the threshold is changed to Linear):
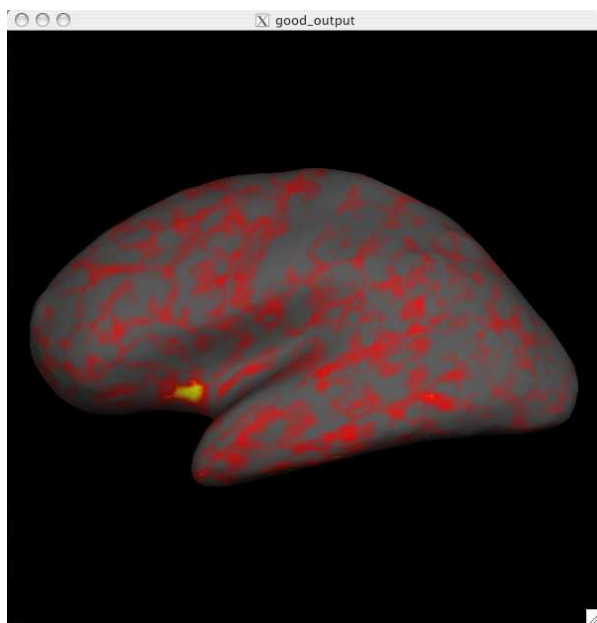
You can adjust the thresholds of the map by going to **View -> Configure -> Overlay** which will open a new window that will allow you to change the thresholds. First click the **Thresholds: Linear** checkbox to make the image match that shown above. Then try some different settings and see how it affects the display on the map (some options to try min of 1.0 and max of 3.0, min of 2 max of 5, etc). After you change the min and max values be sure to hit apply.

Cortical Parcellations

To view the cortical parcellation it is probably best to toggle off the curvature ⬚ and thickness files ⬚ , and to switch to the pial surface, ⬚ P . You can load the parcellation by going to **File -> Label -> Import Annotation** and browsing to the lh.aparc.annot file (found in $FREESURFER_HOME/subjects/buckner_data/tutorial_subjs/good_output/label/). Be sure you select the label file for the correct hemisphere, if you select the wrong one it will not look right. The lh.aparc.annot should open up and like this:

You can inspect the parcellation by rotating the surface to see all sides. You can switch to the inflated view, or turn the labels to outline view, whichever way is most comfortable for you to view the parcellations. The parcellation that is loaded here was created with the Desikan-Killiany atlas. By default there are two parcellations that are made when recon-all is run. The second parcellation, called ?h.aparc.a2005s.annot, is created with the Destrieux atlas. The difference is the number and designation of the areas that are labeled. You can load this second parcellation by first going to **File -> Label -> Delete all Labels**. This will remove the first parcellation. Then you can repeat the steps for loading a parcellation, this time browsing to lh.aparc.a2005s.annot to load the second parcellation.

Using Tkmedit and Tksurfer together

When you are viewing the same subject in tkmedit and tksurfer at the same time (generally done using two separate terminals, one to launch tkmedit and one to launch tksurfer) you can use some tools to switch from one point on the surface (in tksurfer) to the same point in the volume (in tkmedit). To do this, first put your cursor at the point you want in tksurfer. Next, click the save point button in tksurfer. This will save the cursor position. Then, in the tkmedit toolbox window, click the goto saved point button . This will now bring you to the same point, only in the volume. Look for the red plus sign (it is small and can be hard to find, depending on the saved point, but should be near the surface for this example). This technique is very useful when you see something wrong on the surface of a subject (in tksurfer) and you want to see what is happening in the volume in that same place.

2008-06-01 20:47

Troubleshooting your output

*
    To follow this exercise exactly be sure you've downloaded the tutorial data set before you begin. This set of exercises is not possible without the tutorial data set. Please set the environmental variable TUTORIAL_DATA to the location that you have downloaded the data to (here, it has been copied to $FREESURFER_HOME/subjects):

```
tcsh
setenv TUTORIAL_DATA $FREESURFER_HOME/subjects/buckner_data/tutorial_subjs
```

* Notice the command to open tcsh. If you are already running the tcsh command shell, then the 'tcsh' command is not necessary.

This set of exercises will take you through a few examples of problem outputs, asking you to identify the problems and possible methods to fix the problems. Each example will have a before and after picture, as well as an explanation of how to fix the problems seen.

Below is a list of common things that require manual intervention. As you move through this exercise there will be links to pages instructing you how to fix these problems. While you are trying to identify the problems with the subjects listed below you may find it helpful to refer back to the previous exercise, or open your own instance of the subject good_output to compare these subjects to a good example.

* Skull strip
* Edits to the wm volume
* Edits to the brainmask volume
* Adding control points
* Talairach transformation

Subject 1

First make sure you have your SUBJECTS_DIR set to the correct location:

```
tcsh
setenv SUBJECTS_DIR $TUTORIAL_DATA
cd $SUBJECTS_DIR
```

Now you can open the first subject, subject1_before, in tkmedit.

```
tkmedit subject1_before brainmask.mgz \
  lh.white -aux T1.mgz -aux-surface rh.white &
```

This will open the brainmask.mgz volume, the T1.mgz loaded as aux, and the surfaces for both hemispheres.

You can feel free to open other volumes in aux or to load in the aseg.mgz if you want or need to. Take a look at this first subject, inspecting the various outputs that were mentioned in the previous exercise and see if you can indentify what is wrong.

If you are stuck and you need a hint you can click for some help.

When you think you've identified the problem, click here for detailed instructions on how you can fix it. You can also compare this to the finished version, subject1_after.

Subject 2

Now, take a look at the next subject, subject2_before.

```
tkmedit subject2_before brainmask.mgz \
  lh.white -aux T1.mgz -aux-surface rh.white &
```

Again, this will bring up the brainmask.mgz volume, the T1.mgz volume, and the surfaces for both hemispheres.

In your second terminal window, if not already open, open the surfaces in tksurfer:

```
tksurfer subject2_before lh inflated &
tksurfer subject2_before rh inflated &
```

This will open the inflated surfaces for both hemispheres in tksurfer.

You may need to open other volumes in order to see or fix the problem. Take a look through this subject and see if you can identify what is wrong.

If you are stuck and you need a hint you can click for some help.

When you think you've identified the problem, click here for detailed instructions on how you can fix it. You can also compare this to the finished version, subject2_after

Subject 3

Now, take a look at the next subject, subject3_before.

```
tkmedit subject3_before brainmask.mgz \
  lh.white -aux T1.mgz -aux-surface rh.white &
```

Again, this will bring up the brainmask.mgz volume, the T1.mgz volume, and the surfaces for both hemispheres.

In your second terminal window, if not already open, open the surfaces in tksurfer:

```
tksurfer subject3_before lh inflated &
tksurfer subject3_before rh inflated &
```

This will open the inflated surfaces for both hemispheres in tksurfer.

You may need to open other volumes in order to see or fix the problem. Take a look through this subject and see if you can identify what is wrong.

If you are stuck and you need a hint you can click for some help.

When you think you've identified the problem, click here for detailed instructions on how you can fix it.

Subject 4

Now, take a look at the next subject, subject4_before.

```
tkmedit subject4_before brainmask.mgz \
  lh.white -aux T1.mgz -aux-surface rh.white &
```

Again, this will bring up the brainmask.mgz volume, the T1.mgz volume, and the surfaces for both hemispheres.

In your second terminal window, if not already open, open the surfaces in tksurfer:

```
tksurfer subject4_before lh inflated &
tksurfer subject4_before rh inflated &
```

This will open the inflated surfaces for both hemispheres in tksurfer.

You may need to open other volumes in order to see or fix the problem. Take a look through this subject and see if you can identify what is wrong.

If you are stuck and you need a hint you can click for some help.

When you think you've identified the problem, click here for detailed instructions on how you can fix it.

Subject 5

Now, take a look at the next subject, subject5_before.

```
tkmedit subject5_before brainmask.mgz \
  lh.white -aux T1.mgz -aux-surface rh.white &
```

Again, this will bring up the brainmask.mgz volume and the T1.mgz volume and the surfaces for both hemispheres.

In your second terminal window, if not already open, open the surfaces in tksurfer:

```
tksurfer subject5_before lh inflated &
tksurfer subject5_before rh inflated &
```

This will open the inflated surfaces for both hemispheres in tksurfer.

You may need to open other volumes in order to see or fix the problem. Take a look through this subject and see if you can identify what is wrong.

If you are stuck and you need a hint you can click for some help.

When you think you've identified the problem, click here for detailed instructions on how you can fix it.

2008-06-01 20:47

Fixing a bad skull strip

-

    To follow this exercise exactly be sure you've downloaded the tutorial data set before you begin. If you choose not to download the data set you can follow these instructions on your own data, but you will have to substitute your own specific paths and subject names.

Occasionally, the skull stripping step either removes more than just the skull, causing part of the brain to be removed as well, or too little, leaving behind portions of the skull. Both of these problems need to be corrected before continuing to the next step, either by manually editing the volumes or by adjusting input parameters to the skull stripping step, and running the skull strip again until a good result is obtained. Often the sagittal view reveals skull strip failures. Note that the inflated 2D surface is a less reliable gauge of skull strip failure unless large portions of the brain are missing, or lots of skull is retained.

Subject 1 has a poor skull strip, an entire hemisphere of the cerebellum has been stripped away along with the skull. This page will walk you through the process of fixing this particular subject and also offer suggestions for fixing other common skull stripping problems.

If you look at coronal slice 91 for subject1_before you can see that the brainmask.mgz volume (the first picture) is missing the right hemisphere of the cerebellum and that it is present in the T1.mgz volume (the second picture):
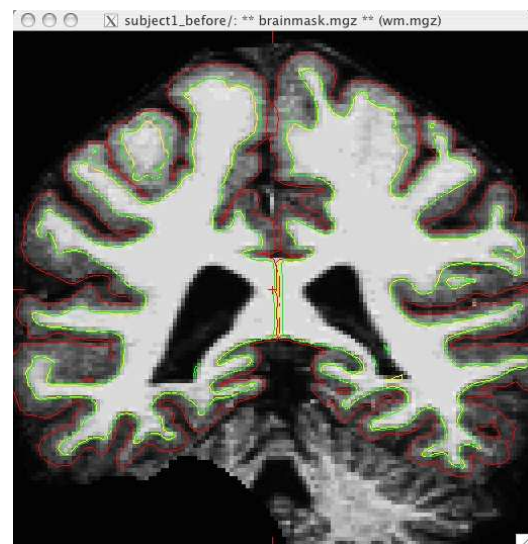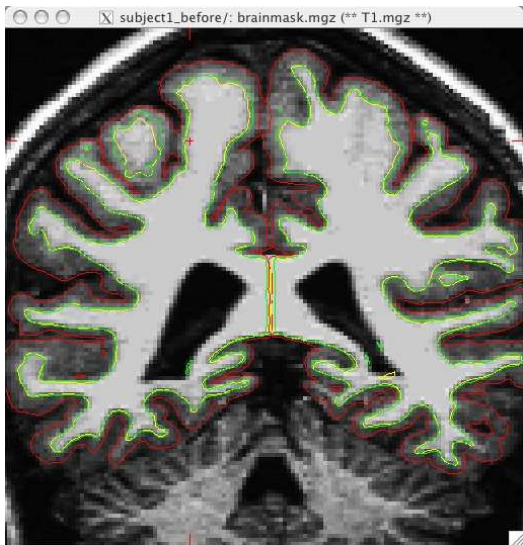
In general there are two ways to fix a volume when there is something missing from the cortex or cerebellum, you can clone the missing pieces in manually or you can adjust the parameters of mri_watershed to do it automatically. For this case, because there is such a lot missing on so many slices you should adjust the parameters of mri_watershed.

Adjusting watershed parameters

The watershed algorithm is used during the skull stripping step to find a boundary between the brain and skull. The mri_watershed program uses a default preflooding height of 25 percent. If we want the algorithm to be more conservative (i.e. if part of the brain has been removed), you will want to make that number larger than 25. If you want the algorithm to be more aggressive (i.e. part of the skull has been left behind), you will want to make the height less than 25. There aren't any hard and fast rules about how to select your height value. You can adjust the preflooding height by passing the following flag to recon-all:

```
recon-all -skullstrip -wsthresh <h> -clean-bm -subjid <subject name>
```

where <h> is replaced with the preflooding height you'd like to use and <subject name> is replaced with your subject. The clean-bm flag is used to instruct recon-all to write over the old brainmask.mgz volume with your new edits. If you do not use this flag your changes will not take effect.

Part of the brain is missing

Now we will take another look at the first volume we looked at, where part of the cerebellum had been removed. You can adjust the watershed threshold by passing the -wsthresh flag to recon-all. In this instance, since too much was removed, we want to raise the watershed threshold so use the command:

```
recon-all -skullstrip -wsthresh 35 -clean-bm -no-wsgcaatlas -subjid subject1_before
```

Take a look at your output volume (brainmask.mgz has been changed) along with the original T1 volume (T1.mgz), and verify the result of the new skull stripping is correct.

```
tkmedit subject1_before brainmask.mgz lh.white -aux T1.mgz -aux-surface rh.white
```

It should look like this:
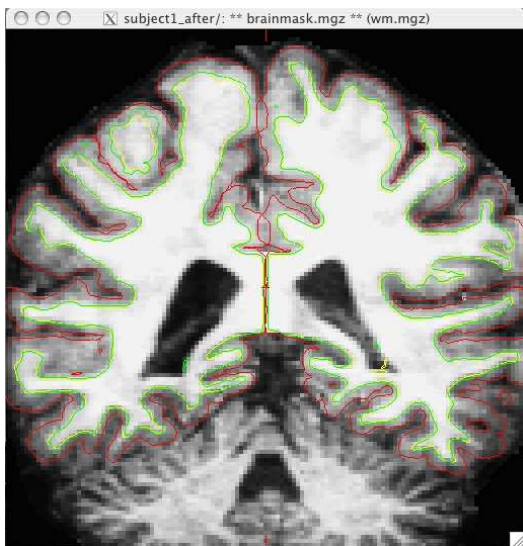
Some skull still remains

Sometimes the skull strip will leave pieces of skull in the brainmask volume. Subject 111 is an extreme example of this, it can be the case that there is just a bit of skull left. Open subject 111 to see what it looks like:

```
tkmedit 111_watershed_before brainmask.mgz
```

and you should see this:

For this example, since there is so much skull remaining, we want to lower the watershed threshold, so you could use the command:

```
recon-all -skullstrip -wsthresh 5 -clean-bm -no-wsgcaatlas -subjid 111_watershed_before
```

Take a look at your output volume (brainmask.mgz has been changed) along with the original T1 volume (T1.mgz), and verify the result of the new skull stripping is correct.

```
tkmedit 111_watershed_before brainmask.mgz lh.white -aux T1.mgz -aux-surface rh.white
```

You can compare this to the brain volume in 111_watershed_after to see that your changes look the same.

Reprocessing the data
If you make changes to the brainmask.mgz volume, you can re-start the recon-all process from this point by using the following command:

```
recon-all -autorecon2 -subjid <subject name>
```

Manual editing

When the skull stripping process has left just a few slices with either missing brain regions or too much skull you can edit these manually using tkmedit.

Part of the brain is missing Sometimes there will be small regions missing from the pons or cerebellum, or from part of the cortex itself. To fix these you'd need to open the output volume from the skull stripping step (brainmask.mgz) and the original T1 volume (T1.mgz) simultaneously.

For this following example we will use a subject with a missing cerebellum to demonstrate the tools, but in reality this is not an example of something you would want to edit by hand.

```
tkmedit 091_watershed_before brainmask.mgz -aux T1.mgz
```

Switch back and forth between the two volumes a few times with Ctrl-1 and Ctrl-2, and use the arrow keys to view the different slices. Notice that a large part of the cerebellum has been stripped away along with the skull around slice 91.

In the tkmedit toolbar, go to:

Tools -> Configure Volume Brush...

Set Mode to "Clone", and Clone Source to "Aux Volume". Click the Close button to close the configuration window. You can also change the size and shape of your brush, to do this go to:

Tools -> Configure Brush Info...

Select a radius and shape that you are comfortable using. Close the configuration menu, and click the "Edit Voxels" button in tkmedit toolbar.

Use Ctrl-1 and Ctrl-2 to cycle between the two volumes. Find a place in the image where the cerebellum is missing in the output volume, then use the middle button on the mouse to paint in cerebellum from the auxiliary volume. Cycle back and forth between the volumes frequently so you know where you are.

Continue painting until the slice is no longer missing part of the brain. Repeat for the other slices in the output volume. Go to **File -> Save Main Volume As...** and save your output.

Some skull still remains
If there are small pieces of skull still remaining on only a few slices you can manually remove the voxels. To do this make sure that the "Edit Voxels" button is still selected. Removing voxels is very similar to painting in voxels, except you use the right mouse button instead of the middle button. Use Ctrl-1 and Ctrl-2 to cycle between the volumes. Find a place in the image where skull remains. Use the right mouse button to delete the voxels. Continue on the other slices until all skull is removed. Save your volume.

Ctrl-z and **Edit -> Undo Last Edit** in tkmedit only allow you to go back one edit. If you need to erase a mistake made when painting in voxels, you can use the right mouse button to delete them again. If you remove too many voxels, you can use the clone tool to paint areas back in from the original T1 volume, similar to painting in voxels in section 2.1.

2008-06-01 20:47

Making Edits to the White Matter

-
  To follow this exercise exactly be sure you've downloaded the tutorial data set before you begin. If you choose not to download the data set you can follow these instructions on your own data, but you will have to substitute your own specific paths and subject names.

Sometimes the white matter is not segmented correctly: sometimes voxels that should be white matter are excluded, and other times voxels that should not be white matter are included in error. Either of these occurences can be fixed with simple manual edits. Below you will find examples of a few of the common problems and how to fix them. For some people it is easiest to see these problems when looking at the inflated surface in tksurfer. Others can pick them out while viewing the volumes in tkmedit. Either method is fine for identifying these problems. However they can only be fixed using tkmedit with the wm.mgz volume.

Geometric inaccuracy due to brain lesion

Subject 2 is an example of white matter being excluded from the wm.mgz volume due to the presence of a brain lesion. This page will walk you through fixing this specific example as well as other common problems with the wm.mgz volume.

First, make sure you have subject 2 loaded in tkmedit:

```
tkmedit subject2_before brainmask.mgz lh.white -aux wm.mgz -aux-surface rh.white
```

If you look at coronal slice 155 you will notice an area as you get to the anterior horn of the lateral ventricle where the white surface (yellow line) does not follow the surface of the brain, but in fact cuts into it. This geometric inaccuracy is caused by a lesion near the lateral ventricle where white matter has been marked as non-white matter. In this case, the autofill routine has already filled the ventricles, but the brain pathology resulted in an incorrect segmentation causing this hole in the surfaces.
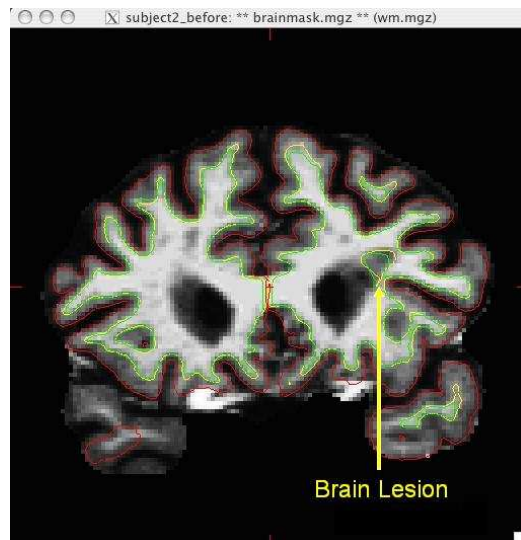
You can see this as a dimple or hole on the inflated surface in tksurfer too (see next image). If not already open, in a second terminal window, open the lh surface in tksurfer:

```
tksurfer subject2_before lh inflated &
```
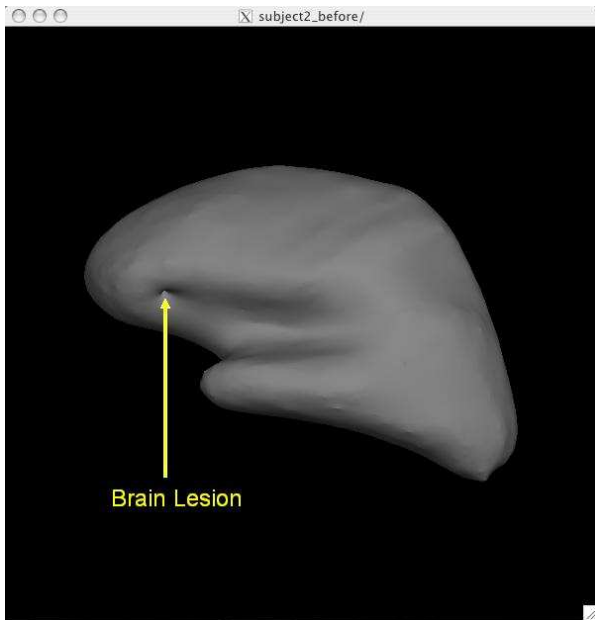
With the wm.mgz volume loaded into tkmedit you can see that this area has been left out of the wm volume completely.
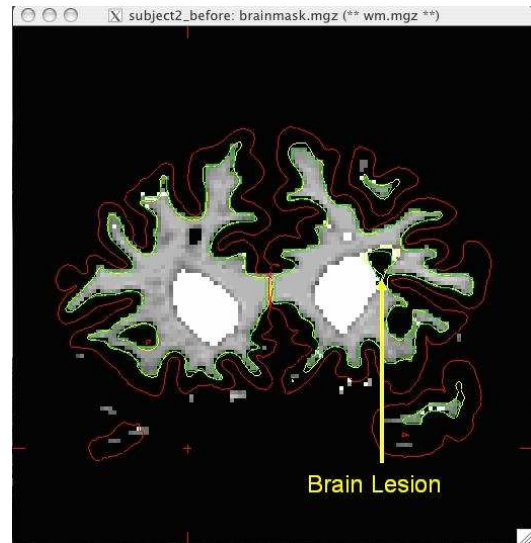
To fix this problem you will need to fill in the missing voxels in the wm.mgz volume. First, it's a good idea to use the arrow keys to scroll through the individual slices in the volume until you get a good idea of where the problem starts and ends. You will want to start filling in voxels when the inaccuracy appears, and keep filling them in slice by slice until the problem is no longer visible. The coronal view and a brush radius of one or two are good settings for painting in voxels. Zoom In and Out either by using the white icons in the tkmedit Tool window or by Ctrl-right mouse button (Zoom In) and Ctrl-left mouse button (Zoom Out) to help you see the inaccuracy.

To begin editing voxels, click on the Edit Voxels Tool button, then go to **Tools --> Configure Brush Info...** Set the *radius* to *2* and *Shape* to be *Circle*. By default tkmedit will edit on the main volume loaded, if the wm volume is loaded as your aux volume you will also need to select *Aux volume* as the *Target*. Use the middle button on your mouse to begin painting in the voxels. If you fill in too many voxels, the right mouse button acts as an eraser. Start with slice 155, when you are done filling the region,

your slice should look like this:
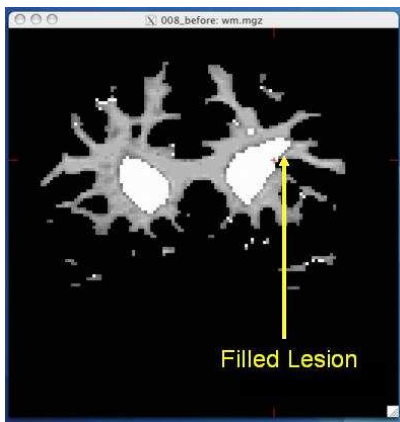
Go to the next slice, and fill in this slice in the same way. You will notice that the hole is not completely enclosed. To see where the boundary should be, switch back and forth between the brainmask.mgz and wm.mgz volume with Ctrl-1 and Ctrl-2. Use the brainmask.mgz volume as a guide, and make the boundary of the voxels that you paint in correspond as closely as possible to the boundaries suggested in the brain volume. You will need to continue filling in slices in this fashion until the region is completely filled in. If you would like to compare your edits to ones that have been done for you you can open a second tkmedit with the subject2_after wm.mgz volume, and for each slice check your results with the corresponding slice in the subject2_after volume. The subject2_after volume has the region filled in both the right and left hemispheres. At any time, you can save the changes you've made to the wm volume by selecting 'Save Volume' in tkmedit's 'File' menu, and clicking 'OK'. You can verify your results by viewing the changes made to the subject2_after wm volume

Tips:

- Sometimes regions may appear enclosed in some slices (i.e. appear as holes), open in subsequent slices (i.e. no longer appear as holes), then enclosed again as you scroll forward. The rule of thumb when editing these is to keep filling until you reach the slice where they finally open up and are no longer enclosed.
-

When filling in regions that are in both hemispheres, you may opt to fill them in both hemispheres at the same time, when you become more comfortable with editing inaccuracies. To load the white surface for the other hemisphere you will need to go to **File --> Aux Surface --> Load Aux Main Surface** and select the other hemisphere's white surface (i.e., rh.white). This will load the rh.white surface as the Aux surface and you can now follow the surfaces for both hemispheres at the same time.
- When editing geometric inaccuracies in the future, you may find it easier to edit the first slice and last slice first, thereby 'capping' the slices that need filling. Then you just need to fill all the slices between the endcaps.
- Sometimes there is variation between a subject's left and right hemispheres, so that in a particular slice one hemisphere's region will finally 'open up', but the other hemisphere's region is still enclosed. In such cases, continue to fill the enclosed region only, even though the other is open. This will address the topological problem.
- Problems like the one shown here, due to brain lesions, should always be fixed with edits to the wm.mgz volume. Do not use control points to try and automatically adjust the intensity in these areas.

After you have saved all of your edits, you could recreate the final surfaces with the command:

```
recon-all -autorecon2-wm -subjid subject2_before
```

This will take a long time to run, so there is no need for you to run it now.

Non-white matter classified as white matter

Sometimes non-white matter (i.e., skull) is included as white matter. To load in the example use this command:

```
tkmedit 111_manual_edits_before wm.mgz lh.white -aux brainmask.mgz -aux-surface rh.white
```

Scroll through the brainmask.mgz volume slices and as you get towards the frontal lobe you will see a group of voxels included as white matter. They are surrounded by the white surface (yellow line). This group lies outside of the pial surface (red line) and is clearly not white matter. In order to correct this inaccuracy it is necessary to delete the voxels which are not part of the white matter.

misclassified as white matter
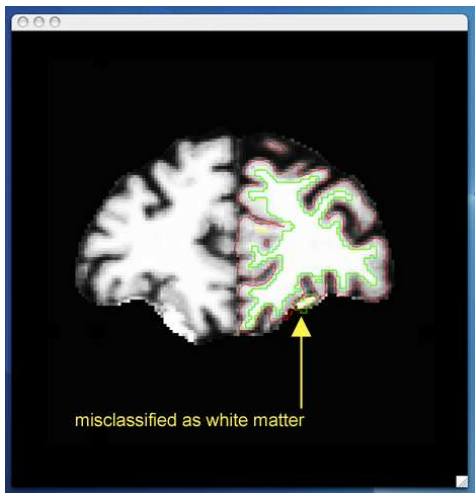
41
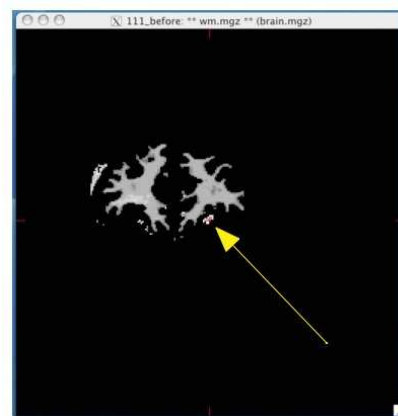
Using Ctrl-1 and Ctrl-2 to switch back and forth between the brain and wm volumes, you can clearly see that this region should not be labeled as white matter. To delete these voxels, click the Edit Voxels Tool

button in the tkmedit toolbar, and put your mouse cursor over the voxels you want to delete. Delete them with the right mouse button. This is what slice 179 should look like after you have deleted the errant voxels:

42
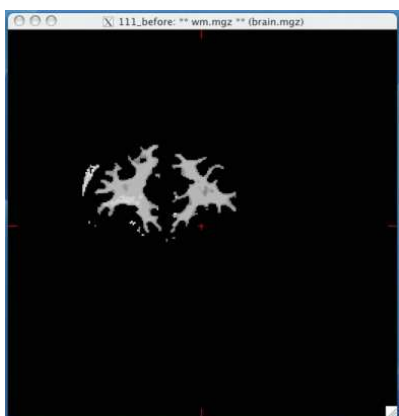
Go through the slices with the up and down arrows, continuing to delete voxels until the inaccuracy is no longer there. (It spans approximately 30 slices.) You can check your results by looking at the 111_manual_edits_after wm.mgz volume in tkmedit. Note that in this volume, only a few of the voxels have been removed from each slice, instead of removing the entire inaccuracy. This is because you only need to remove as many voxels as is necessary for the segmentation to classify this region as non-white matter. Until you are more comfortable with editing inaccuracies, go ahead and remove all of the voxels. More practice will give you a better feel for how many voxels need to be removed.

Once you have made all the edits to the wm.mgz volume you would regenerate the surfaces using the following command:

```
recon-all -autorecon2-wm -autorecon3 -subjid 111_manual_edits_before
```

This step will take a long time and there is no need to run it for the tutorial purposes.

Lateral Ventricle

The automatic edits made to to wm volume will fill the lateral ventricles. On occasion they are not filled entirely. Continue looking at 111_manual_edits_before in tkmedit, with the lh.white surface loaded on top. As you scroll to the posterior portions of the brainmask.mgz you will see the white surface (yellow line) start to follow the edge of the ventricle (look around coronal slice 53).

43

When you toggle between wm and the brain volume, you can see that this portion of the ventricle should be filled in. This inaccuracy can be corrected in the same fashion as the inaccuracies above, using the Edit Voxels tool to fill in the missing voxels.

44
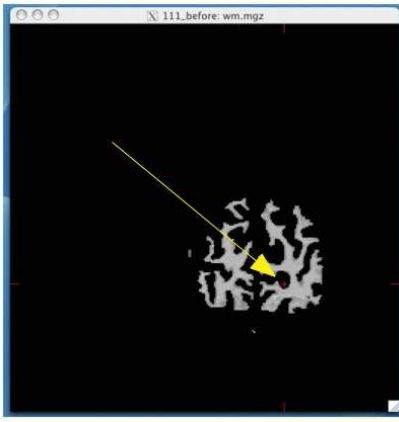
When you are finished filling in slice 53, it should look like this:
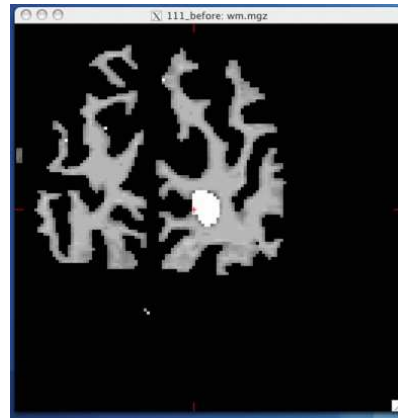
Continue filling in this region until the hole disappears. The geometric inaccuracy is present from slices 50 through 168. Once you have fully edited the wm.mgz volume you would regenerate the surfaces with the following command:

```
recon-all -autorecon2-wm -autorecon3 -subjid 111_manual_edits_before
```

This step will take a long time and there is no need to run it for the tutorial purposes.

2008-06-01 20:47

Correcting Pial Surfaces

-
  To follow this exercise exactly be sure you've downloaded the tutorial data set before you begin. If you choose not to download the data set you can follow these instructions on your own data, but you will have to substitute your own specific paths and subject names.

The pial surface is created by expanding the white matter surface so that it closely follows the gray-CSF intensity gradient as found in the brainmask.mgz volume. Once an accurate white surface is created then you can work on correcting the pial surface if needed. The pial surface boundary and white matter surface boundary should not cross. After the pial surface has been generated, it's a good idea to visually check it for defects that may have been created during automatic topology fixing. To check the pial surface, it may be loaded into tkmedit and viewed along with the brainmask.mgz volume. If the surface appears not to follow the gray-CSF boundary in the volume, edits may be required.

Editing the Volume

Subject 3 is an example of the pial surface including non-cortex within the boundaries. This page will take you through fixing this and other similar problems with the pial surface including non-cortex material.

First, make sure you have subject3_before loaded in tkmedit:

```
tkmedit subject3_before brainmask.mgz lh.white -aux-surface rh.white
```

Use the arrow keys to go through the volume slice by slice, and view the pial surface (red line) and white matter surface (yellow line). Notice the bright diagonal line in slice 161 that has caused the pial surface to expand past the actual pial boundary. This is the result of a bad segmentation incorporating a piece of the dura within the pial surface.
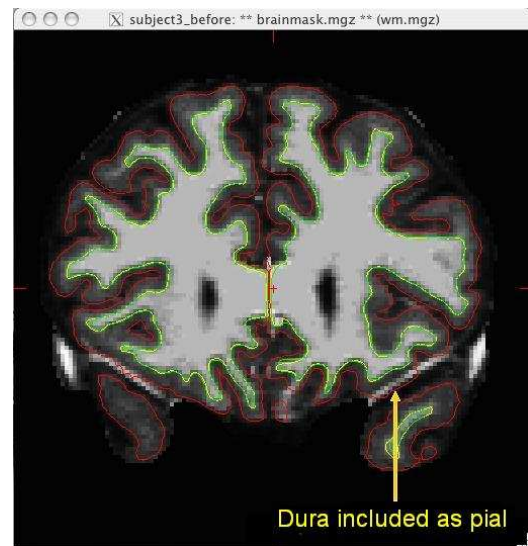
To fix this type of error you can simply edit away the offending voxels from the brainmask.mgz volume. To do this you will need to select the edit voxels tool and set the brush to a size and shape comfortable for you. A circle brush of radius 2 works well for this edit. In the tkmedit toolbar, go to **Tools -> Configure Brush Info...** Set Radius to 2, and Shape to "Circle". Close the configuration menu, and click the "Edit Voxels" button in tkmedit toolbar.

Find a place in the image where the dura is causing errors in the segmentation. Use the right mouse button to delete the voxels. It is not necessary to completely remove the dura to get an adequate pial surface, but it is good to do so until you are more familiar with manual editing. When you are finished removing the bright diagonal line in slice 161, it should look like this:
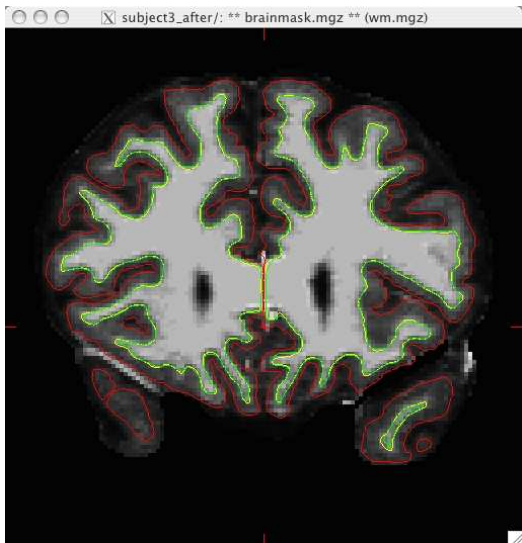
Continue on the other slices until the dura is removed.

Ctrl-z and **Edit -> Undo Last Edit** in tkmedit only allow you to go back one edit. If you remove too many voxels, you can use the clone tool to paint areas back in from the original T1 volume, similar to painting in voxels in section 1.1.

You can check your result by viewing the brain.mgz volume in the 108_after directory.

Regenerating the Surface

When you are finished editing the voxels, you will need to regenerate the surfaces. Since the white matter hasn't been changed, you don't need to resegment the volume. You can regenerate the pial surface with:

```
recon-all -autorecon2-pial -autorecon3 -subjid subject3_before
```

This step will take a long time and there is no need to run it for the tutorial purposes.

2008-06-01 20:47

Using Control Points to Fix Intensity Normalization

-
  To follow this exercise exactly be sure you've downloaded the tutorial data set before you begin. If you choose not to download the data set you can follow these instructions on your own data, but you will have to substitute your own specific paths and subject names.

Sometimes the intensity normalization step will fail because it cannot determine the proper intensity for white matter. The result is an erroneous white matter segmentation. A control point is a manually selected location in the volume that the user feels sure is inside the white matter boundary, and subsequently should be normalized to an intensity of 110. TkMedit displays the intensity of any voxel your cursor is on as a "value" in the Cursor and Mouse sections of its Tools interface.

Manually Selecting Control Points

Subject 4 is an example of a subject that needs some control points in order to ensure that the voxels are normalized correctly and then included in the wm.mgz volume. This page will walk you through setting control points to fix this type of problem.

First, make sure you have subject4_before loaded into tkmedit:

```
tkmedit subject4_before brainmask.mgz lh.white -aux T1.mgz -aux-surface rh.white
```

Note: If you are using your own subject data, where surfaces are not yet available (ie. the -autorecon2 stage has not run yet), then don't include the surfaces in the tkmedit command:

```
tkmedit <my_subj> brainmask.mgz -aux T1.mgz
```

Scroll through this subject and find the location where the white matter is being excluded from the surface. In subject 4 this happens around coronal slice 149.
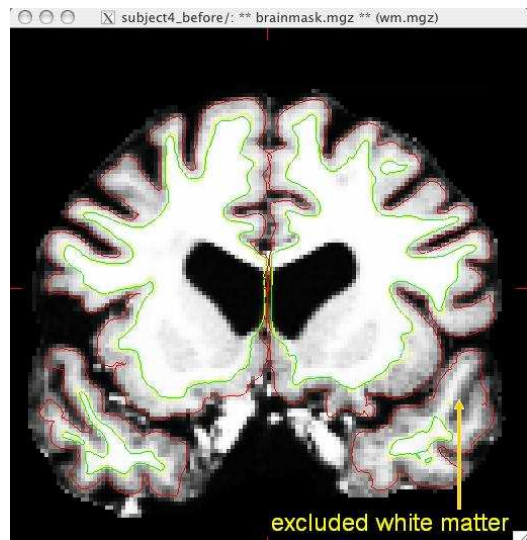
To add control points you will first need to select the Edit Control Points tool. Middle-mouse-button clicking will create a control point; right-button clicking will delete a control point. As you select control points, they will appear as small green crosshairs. Select a few control points around your trouble areas, space them out throughout the brain and on different slices. You want to pick points in a region where the wm intensity is lower than it should be (that is, having a voxel value less than 110).

General tips for adding control points:

- Control points should only be added in regions that are definitely white matter (i.e., not in the cortex, cerebellum, brainstem, or outside of the skull).
- Control points should also only be added in regions where voxel intensity is not 110. A control point in a region that is already normalized to 110 will be useless.
-

Control points should NOT be used to try and normalize a brain lesion to 110. Such defects should be fixed with white matter edits
- Control points can help recover thin wm strands that are dark by putting some at the base of the strand.
- Control points are also useful in areas of very bright intensity.
- Start off with a few control points spread out in your trouble area. You may need to add more. With experience you will be able to determine how many are appropriate, given your specific subject.

Here is an example of one slice with the control points added. Note that there are other control points spread out through other slices as well.
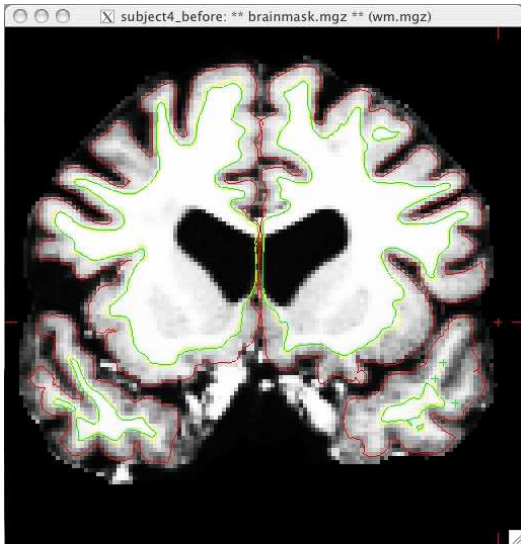
After adding the control points, go to **File -> Save Control Points**; this will create a file called <subject name>/tmp/control.dat. Using the added control points the subject should now look like this:
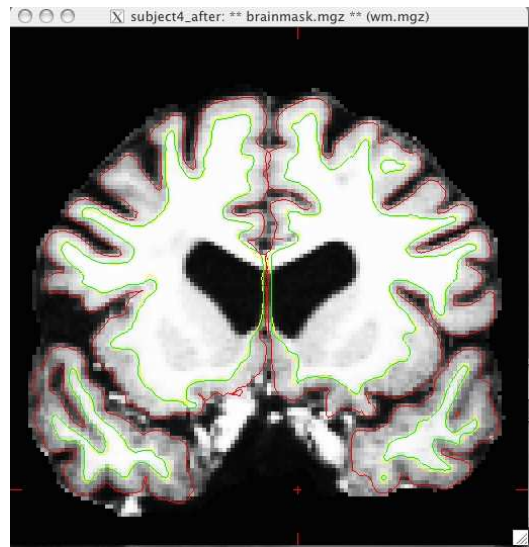
Once your control points are saved you can rerun recon-all as follows:

```
recon-all -autorecon2-cp -autorecon3 -subjid subject4_before
```

This step will take a long time so there is no need to run it for the purposes of this tutorial.

2008-06-01 20:47

Fixing Bad Output From the Talairach Registration

-

To follow this exercise exactly be sure you've downloaded the tutorial data set before you begin. If you choose not to download the data set you can follow these instructions on your own data, but you will have to substitute your own specific paths and subject names.

FreeSurfer computes a linear Talairach transform contained in a 3x4 matrix in a file called talairach.xfm and is located in the <subject name>/mri/transforms directory. Under some circumstances, the alignment can fail the automatic failure detection scheme (-tal-check flag, which can be disabled via -notal-check).

The best way to check the transform is by loading it visually. This can be done directly from tkmedit (File --> Load Transform), however new users sometimes feel more comfortable viewing the transform in tkregister2, where it is possible to view the transform on top of the talairach subject. When viewing in tkmedit you can expect some distortion and stretching from the position of your subject - especially a sagittal tilt, but you want to avoid any changes in orientation (i.e., the coronal view of your subject should be the coronal view of your talairach transform), severe changes in positioning in the window (i.e., your transformation should not show up in the top left corner of the view, while your subject is nicely centered), and rotations or twisting. If any of these are seen, you will need to correct your transform.

Subject 5 is an example of a bad talairach. The surface cutting planes are far off center but the problem actually originates from the bad talairach transform. To view this talairach in tkmedit first be sure you have the subject open:

```
tkmedit subject5_before brainmask.mgz
```

If the surfaces are also loaded into tkmedit you may find it easier if you toggle them off for this part.

In the tkmedit toolbar, go to **File -> Transforms -> Load Transform for Main Volume...** Click browse, select the file talairach.xfm and click 'OK'. In the coronal view, the transform has resulted in an extremely distorted brain.
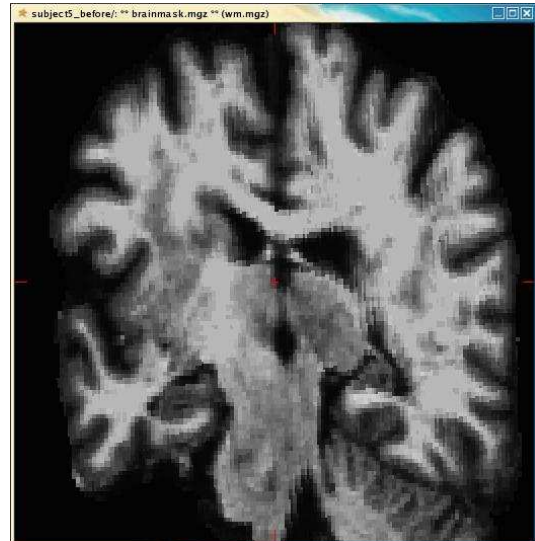
To further view and edit this transform we will use tkregister2. To open this you can close the tkmedit window and open tkregister2 with the command:

```
tkregister2 --mgz --s subject5_before --fstal --surf orig
```

Note: If your subject surfaces are not yet available, then exclude the `--surf orig`flag from tkregister2.

Brief descriptions on how to operate tkregister2 are given below. More documentation is also available by running "tkregister2 --help".

You will see the subject's volume as the TARGET volume and your Talairach volume as a fuzzy MOVEABLE volume. The green lines are the orig surface from the subject. This will be the same in both the TARGET and the MOVEABLE. It can be turned on and off by clicking in the image window and hitting the 's' key.. Find the fmov: box in the tkregister toolbar and make sure it is set to 1.0.

The goal is to stretch, translate, and rotate your MOVEABLE volume so that the two brains look as similar as possible, at least along the key anatomical points (anterior/posterior commissures, the temporal lobes in the coronal plane, and the midline cut).

Use Ctrl-1 and Ctrl-2 to switch between the two volumes (or hit the COMPARE button). You will want to do this frequently to check your progress. Click the SAGITTAL button to switch to a sagittal view, and go to slice 128 by using the slider directly below the SAGITTAL button. In this view you have a good view of the corpus callosum in both the moveable and target volumes.
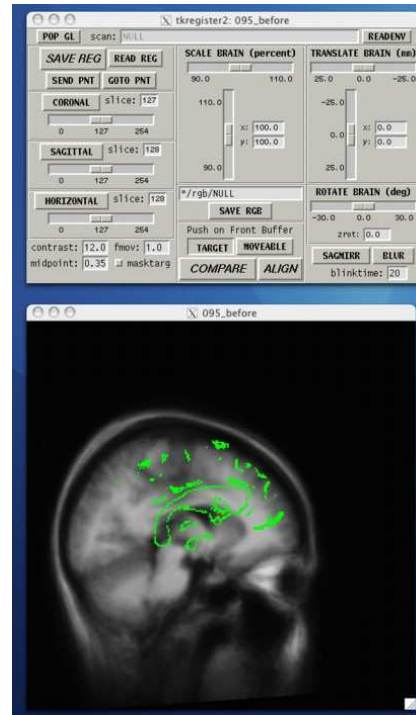
To rotate the moveable volume, use the ROTATE BRAIN slider. You will want to move the slider only a couple of degrees at a time until you achieve the desired effect. You will notice a small red cross icon near the middle of the viewing window. This is the center of rotation. You can change the location of the center of rotation by left-clicking in the viewing window with your mouse.
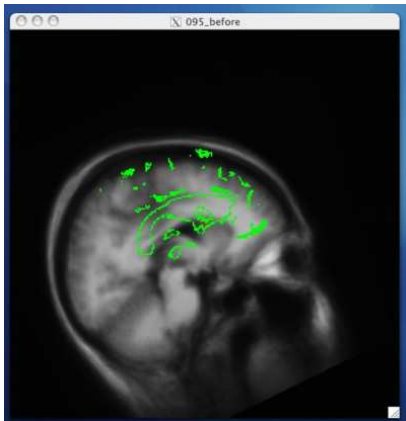
For translation, there are two sliders: one to move the volume left and right, and one to move the volume up and down. Next, translate the moveable brain upwards by using the TRANSLATE BRAIN vertical slider. Again, you only want to move the volume a couple millimeters at a time. You can move the volume left and right in the same way.
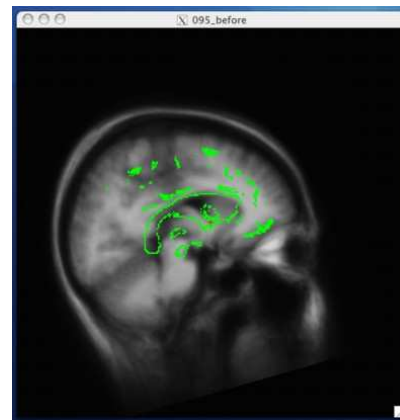
Once you have the corpus callosum aligned as well as possible in the sagittal plane, click the HORIZONTAL button to get a horizontal view, and use the slider directly below the HORIZONTAL button to go to slice 128. Use Ctrl-1 and Ctrl-2 to switch between the two volumes. Use the ROTATE BRAIN and TRANSLATE BRAIN buttons as before to align the midlines of both volumes.
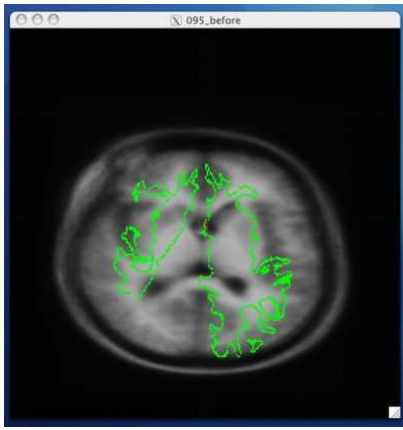
Once you are done aligning the brains in the horizontal view, switch back to slice 128 in the sagittal view. Fine tune your rotation and translation again until the corpus callosum is once again aligned in both volumes.

61
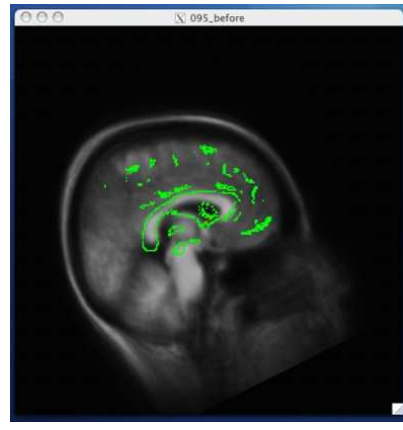
Click the CORONAL button, and go to slice 128. Align the midlines of the brains again in the same way.

62

Continue this way, switching frequently between the HORIZONTAL, SAGITTAL, and CORONAL views, and align the visible brain structures as much as possible in all of the slices. Use the SCALE BRAIN button as needed to scale the brain in the X and Y direction. Keep in mind that you are working in 3D, not in 2D, so any changes made in one view will affect the other views as well.

63

64
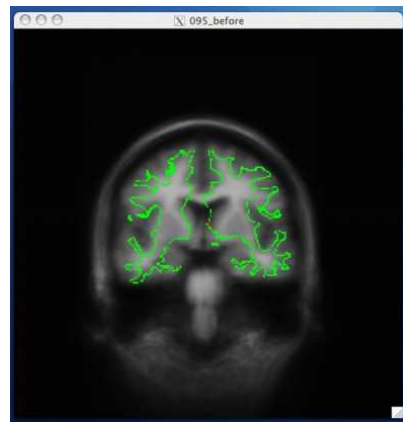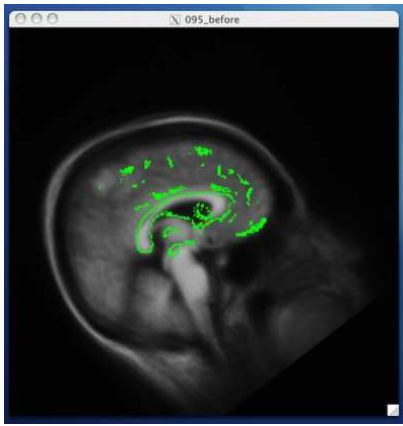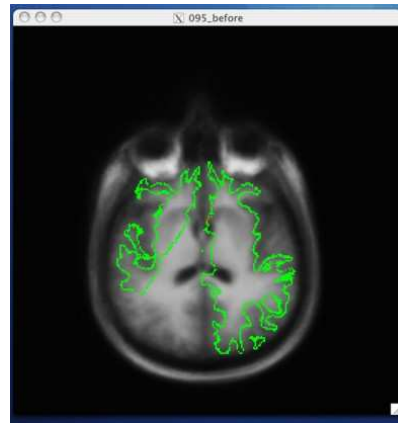
When you are satisfied with your registration result, click the SAVE REG button in the tkregister2 toolbar. You can close tkregister2 by clicking on the red close button in the upper left hand corner of the viewing window, or by typing Ctrl-C in in the same terminal window as the new matrix.

Because the talairach transformation affects everything in the stream it is necessary to rerun the whole process using your new talairach, which is a default. You can do this with the command:

```
recon-all -all -subjid subject5_before
```

This step will take a long time to run, it is not necessary to run this step for the purposes of this tutorial. You can compare your new talairach to the one in subject5_after.

2008-06-01 20:47

Automatically Fixing the Talairach Transform

When you have a bad Talairach transform, you should first attempt to fix it automatically before resorting to a manual registration.

Gray Scale Variations

If you suspect that the bad transform might have been caused by a gray scale problem, first try running mri_convert with "T1" (white matter set to have 110 only) and then "brain" like this (run these commands in the <subject name>/mri directory):

```
mri_convert T1.mgz T1.mnc
mritotal -protocol icbm T1.mnc T1.xfm
```

Check the transform using tkmedit or by looking at the 'Final objective function value' output as the last line of the mritotal command (a number bigger than 0.1 is suspect). If the transform is still bad, try:

```
mri_convert brain.mgz brain.mnc
mritotal -protocol icbm brain.mnc brain.xfm
```

Again, check the transform using tkmedit or noting the final objective function value. A reminder: to check a transform, load the volume into tkmedit, and in the tkmedit toolbar, go to:

File -> Transforms -> Load Transform for Main Volume...

Click browse, select the file transforms/talairach.xfm and click 'OK' to select the original transform, or select one of the new transforms, such as T1.xfm or brain.xfm (being sure to select **File -> Transforms -> Unload Transform for Main Volume** if one was loaded prior).

If you find a good transform using the step above, replace mri/transforms/talairach.xfm with the good one (either using 'T1.xfm' or 'brain.xfm', whichever produced better results). You must use the name talairach.xfm.

If you would like to experiment with these commands, sample data that is known to be quite badly aligned is found in the sample data set directory: buckner_data/tutorial_subjs/095_talairach_before/mri/orig.

Bright Neck Region

If you suspect that the bad transform is the result of a bright neck region, edit the volume to erase the neck and run mritotal on the edited volume. This can be accomplished through simple volume editing: set all the regions to zero where y is greater than a certain value (around 170). This generally works better than using the brain volume. Check your transform using tkmedit.

2008-06-01 20:47

Hints

The following are some hints to help you identify the problem with the 5 troubleshooting subjects. First, be sure you have your SUBJECTS_DIR set correctly:

```
setenv SUBJECTS_DIR $FREESURFER_HOME/subjects/buckner_data/tutorial_subjs
```

Subject 1

The trouble with this subject has occurred in the skull stripping step. Check the brainmask.mgz volume carefully, comparing it to the T1.mgz volume (loaded in aux) to make sure that the skull has been completely stripped away, leaving behind the complete cortex and the cerebellum.

Subject 2

The trouble with this subject has occurred during the white matter segmentation step. Check the surfaces to find the spot that does not match the actual boundaries. If you load the wm.mgz volume as the aux volume you will see a hole in this area. If you are looking in tksurfer you will see a hole or dimple in the inflated surface. You can use the "save point" "goto point" commands to find this spot in the volume.

Subject 3

The trouble with this subject has occurred while making the final surfaces. Check the final surfaces (pial and white surfaces) to make sure that they follow the actual boundaries and do not include anything that should not be included.

Subject 4

The trouble with this subject has occured during the intensity normalization. Check the white surface and be sure it is including all of the white matter as it should. If you find an area that is not included as white matter be sure to check to see that the intensity is at (or very close to) 110.

Subject 5

The trouble with this subject is a little tricky to figure out. The midline cutting planes are not actually on the midline, but the real root of this problem is in one of the very initial steps of recon-all. The talairach transform is bad and will need to be fixed.

2008-06-01 20:47

FreeSurfer Tutorial: Morphometry and Reconstruction (90 minutes of exercises)

Contents

1.0 The recon-all command and alternatives

The recon-all script is used to process raw scan data, segment the white matter, generate surfaces from the segmented data, and output spherical or flattened representations of the surfaces. Recon-all may be used to execute all or part of the volume and surface processing pipelines. Alternatively, scripts exist to execute chunks of each processing pipeline, and individual commands may be run to execute a single processing step. FreeSurfer makes all of these approaches available, and the user may choose the one(s) which are most comfortable on a case by case basis.

Tkmedit and tksurfer are programs that are used to visually inspect the data at key points during the reconstruction process. Tkmedit provides an interface to view and edit voxels on 2D scan slices, and tksurfer is an interface to view the 3D generated surfaces. The reconstruction steps are not immune to failure, so it is necessary to inspect the output as the reconstruction proceeds. The reconstruction steps can fail for many reasons including differing anatomy between individuals and scan quality. In this section's exercises, some of the more common failure modes and ways to correct them will be shown. Both the volume and surface data processing paths in FreeSurfer will be described; for some stages, before-and-after images will illustrate the way data is modified at each step. But first, the manner in which data are prepared for processing in FreeSurfer will be described.

2.0 Volume and Surface processing: recommended workflow using recon-all

Recon-all is a script that is capable of running any single step in the anatomical processing stream, as well as running sets of steps with a single simple command-line interface. When this script is used you will also be generating many log files so that you can track your progress, troubleshoot failures and repeat exact commands if needed, these files are not created if steps are run outside the recon-all script. To reveal the flow of this processing, a more in-depth look is given below. For clarity, the pipeline is presented below in two logical chunks: the volume processing pipeline and the surface processing pipeline. Prior to beginning the recon stream, you must import your data and create a subject directory. This is done **once**,

---

using the following command:

```
recon-all -i <path-to-first-structural> [-i <path-to-second-structural>] -s <subjid>
```

where <path-to-structural> is, in the case of DICOM input, the first file in a collection of slices composing the structural scan. If a second structural is available, that can be included (shown above as optional in the brackets []). The <subjid> is the name for this subject. This command creates a directory named <subjid> in your $SUBJECTS_DIR directory, with empty subdirectories, except for the <subid>/mri/orig directory, which has the file 001.mgz, corresponding to the first structural, and 002.mgz, if a second structural was specified.

Once this 'import' command is issued, it does not need to be run again. You now proceed to the recon stream processing.

Recon-all can be used to execute the entire processing pathway, using the command:

```
recon-all -autorecon-all -s <subjid>
```

If you know that your data are not prone to failures it is perfectly acceptable, and recommended, to run the entire process at once and check all output at the end, in the manner presented above. If it is your first time going through with your set of data, if your data are prone to failures, or if you want to be sure that you do not waste processing time, it is possible to break this total process into 3 smaller pieces, allowing you to check for and correct errors at a few key points.

The logic of the workflow will become more evident as the volume and surface processing streams are described, and as the points at which troubleshooting may be required become clear. The overall workflow for manual checking of intermediate steps is listed below:

2.1 Workflow

1.

```
recon-all -subjid <subject name> -autorecon1
```
2. stop to check for problems with intensity normalization, talairach transformation and skull stripping
3.

```
recon-all -subjid <subject name> -autorecon2
```
4. stop to check final surfaces and make appropriate edits
5. if:
   - the WM volume was edited:
     ```
     recon-all -subjid <subject name> -autorecon2-wm
     ```
   - control points were added:
     ```
     recon-all -subjid <subject name> -autorecon2-cp
     ```

---

the BRAIN volume was edited:
```
recon-all -subjid <subject name> -autorecon2-pial
```
6.

```
recon-all -subjid <subject name> -autorecon3
```

The individual steps in this workflow will be described in context within the volume and surface processing pipeline descriptions and troubleshooting exercises that follow. You can find further instructions on the recommended and other FreeSurferWorkFlows under the workflow section of the wiki. For a listing of the individual flags used with recon-all see ReconAllDevTable and OtherUsefulFlags.

3.0 Volume Processing: a detailed look

3.1 Creating the SUBJECTS_DIR for FreeSurfer processing
FreeSurfer requires you to set the SUBJECTS_DIR variable to a directory path that contains (or will contain) the subjects you wish to process. Each individual subject will have its own sub-directory, within the defined SUBJECTS_DIR, that will contain all the output of the cortical reconstruction. When you are in the directory you wish to work from you can set the SUBJECTS_DIR variable using this command:

```
setenv SUBJECTS_DIR ${PWD}
```

The individual sub-directories are created by the recon-all script when it is first called, and will be named with the subject ID you provide on the command line. The *bert* id will be used to reference this particular distributed data set in subsequent FreeSurfer commands. In the text below we refer to this id as <subject name>, which should be replaced with the actual subject name you choose for your particular subject when you run these commands. The directory from which this command is run should correspond to the ${SUBJECTS_DIR} environment variable.

The following sections cover the conversion of data sets to file formats recognized by FreeSurfer, motion correction (aligning the multiple datasets acquired for each subject to the same template) and the averaging of the motion-corrected multiple acquisitions.

3.2 Data conversion
Before volume processing steps can begin, the raw data from the scan must be converted into a format recognized by FreeSurfer and placed into a particular directory structure so that each volume can be found by FreeSurfer. The output of the data conversion step is a set of volumes, found here in mgz format:

```
${SUBJECTS_DIR}/<subject name>/mri/orig/*.mgz
```

The process of converting data from one format to another is described below.

3.2.1 Converting data to mgz format

Recon-all will begin by converting DICOM, or other native scanner format, to the mgz format as its first step. It calls the mri_convert program to convert the data. The recon-all command to convert the data is:

```
recon-all -i <in volume> -s <subject name>
```

where <*in volume*> is the file in each acquisition directory that should be used as the template (usually the first file for each volume) and <*subject name*> is the name you want to give this particular subject. The mri_convert command will find the other images that are part of the same volume, and convert them into a single file in mgz format which contains the entire volume.

-

Exercise A. Convert a DICOM volume into mgz format

3.2.2 Multiple acquisitions
If multiple acquisitions exist for a subject, you can specify them all in the same command to be converted into the subject's mri/orig directory. For example, if the subject bert had three structural acquisitions to be used for the reconstruction, you would run the following command:

```
recon-all -i <in volume 1> -i <in volume 2> -i <in volume 3> -s bert
```

when this is finished you will find 3 mgz files, one for each acquisition:
$SUBJECTS_DIR/bert/mri/orig/001.mgz
$SUBJECTS_DIR/bert/mri/orig/002.mgz
$SUBJECTS_DIR/bert/mri/orig/003.mgz

3.2.3 Motion correction and averaging
If multiple acquisitions are available for a single subject, these volumes are spatially registered and averaged together into a single, more accurate representation. In this processing step, multiple scans from each subject are registered using the first scan as the template, and a single averaged, motion corrected volume for each subject is generated as output. Recon-all will look for three-digit zero-padded mgz files in the ${SUBJECTS_DIR}/<subject name>/mri/orig/ directory and motion correct them as the next step in the volume processing pipeline. To motion correct and average multiple acquisitions for a single subject without continuing on to the rest of the recon-all process, the recon-all script can be used in the following way:

```
recon-all -s <subject name> -motioncor
```

This will create ${SUBJECTS_DIR}/<subject name>/mri/orig.mgz as the corrected output volume, and orig.mgz will automatically be *conformed*, meaning that the volume is 256^3, with each voxel being $1mm^3$ and represented by an unsigned char. Note that the mgz format can handle most voxel representations (e.g., int, short, float, double, etc...). Recon-all calls a FreeSurfer tool called mri_motion_correct.fsl, which relies on FLIRT, from the FSL toolset (http://www.fmrib.ox.ac.uk/fsl/flirt/).

3.3 Intensity Correction, Normalization, and Skull Stripping
The next few steps of volume processing for each subject begin with the output of motion correction, the ORIG volume (orig.mgz). Several intensity normalization steps are next, along with a transformation to Talairach space. The intensity corrected T1 volume is fed into an mri_watershed which strips out the skull and any remaining background noise and generates the BRAINMASK volume. This can be considered the end of the first chunk of processing and everything, from conversion to skull stripping, can be accomplished using the following command:

```
recon-all -i <in volume 1> -i <in volume 2> -i <in volume 3> -autorecon1 -s bert
```

If you stop at this point to check your output for potential problems you will want to pay attention to the normalization, Talairach transformation and skull stripping steps. Errors here may require some troubleshooting, as described in the exercises below. If you've run **-autorecon-all** you may notice inaccuracies in your output that are results of errors from these first steps.

3.4 Subcortical Segmentation

After creation of the BRAINMASK volume (brainmask.mgz, which has the skull stripped from it), the subcortical processing and segmentation occurs, yielding an automatic labeling of subcortical structures in the ASEG volume (aseg.mgz). Note that this is the longest stage in the processing, and can take upwards of 15 hours. It occurs in six different steps, and will output the aseg.mgz and a corresponding statistics file that contains the volume of all labeled subcortical structures as well as some other critical measures. This stats file can be found in $SUBJECTS_DIR/<subjid>/stats/aseg.stats. To run ONLY the subcortical segmentation you can use the command:

```
recon-all -subcortseg -s bert
```

The subcortical segmentation is run as part of **-autorecon-all** and also as the first part of **-autorecon2**, the description of which continues below. This ASEG volume is used in subsequent volume processing steps. If you are working with a population for which the aseg tools will not be effective, e.g., monkey data and newborn data, use the flag **-noaseg**. This will alert recon-all to not only skip this lengthy step, but also to use commands in following steps that do not require the presence of the ASEG volume.

3.5 WM Segmentation and Filling

In this last portion of the volume processing pipeline, the input volume is normalized and segmented to generate volume containing only white matter (wm.seg.mgz). Then labels from the ASEG are used to fill in the ventricles and other regions of the brain that tend to cause problems during subsequent automatic topology correction, generating the WM volume (wm.mgz). Finally mri_fill cuts the hemispheres from each other and from the brain stem, and creates a binary mask (FILLED volume - filled.mgz) that distinguishes the two hemispheres for use in the surface processing pipeline. At this point, the volume pipeline is finished and everything else happens *per hemisphere* in the surface processing pipeline which is described below.

- Click to see images of output before and after each step

4.0 Volume Processing: troubleshooting bad output

Keep in mind that these preprocessing steps prepare the volumetric data for generation of surfaces. It is possible that along the way problems may occur, which will affect the final surfaces and later processes as well. If you ran recon-all the whole way through before checking the output, you may see a problem with your final surface, which would likely be due to something that happened during the volume processing stream. Troubleshooting some of the most common problems is explored in the following exercises.

4.1 Spatial Normalization (Talairach)

Since data sets from different subjects will vary greatly due to individual anatomical differences and acquisition parameters, preprocessing involves mapping each data set into a standard morphological space. The spatial normalization procedure computes the translations, rotations, and scales needed to bring each subject's volume into Talairach space, a standard morphological space in which the Anterior Commissure is the origin, and x,y,z dimensions refer to the left-right, posterior-anterior and ventral-dorsal positions. Note that the volume itself is NOT resampled into Talairach space. Only the transformation is computed. The MNI toolset is also used in this processing step (see: Collins, D. L., P. Neelin, et al. (1994).Data in Standardized Talairach Space. *Journal of Computer Assisted Tomography* 18(2): 292-205.) The following exercise examines how the transformation is performed and how to correct problems with the output of this automatic preprocessing step.

-

    Exercise B. Troubleshooting: fixing bad output from the Talairach registration

4.2 Intensity Normalization
The intensity normalization procedure removes variations in intensity that result from magnetic susceptibility artifacts and RF-field inhomogeneities. The output files written by the normalization procedure are:

images: ${SUBJECTS_DIR}/<subject name>/mri/T1.mgz

-

    Exercise C. Troubleshooting: using control points for intensity normalization

4.3 Skull Stripping
In this step, the skull is automatically stripped off the 3D anatomical data set by using a hybrid method that combines watershed algorithms and deformable surface models. The method first inflates a stiff deformable spherical template of the brain from a starting position in the white matter. The surface expands outward from this starting position, until it settles exactly at the outer edge of the brain. The outer brain surface is then used to strip the skull and other non-brain tissue from the T1 volume.

The output files written by this procedure are:

shrinksurface: ${SUBJECTS_DIR}/<subject name>/bem/brain.tri
images: ${SUBJECTS_DIR}/<subject name>/mri/brain.mgz

-

    Exercise D. Troubleshooting: fixing bad output from skull stripping

4.4 Segmentation
The white matter segmentation uses intensity information as well as geometric information to segment the white matter voxels. White matter segmentation is done by the mri_segment command, and its output files are:

images: ${SUBJECTS_DIR}/<subject name>/mri/wm.mgz

The usage for the segment_subject program is:

```
recon-all -subjid <subject name> -segmentation
```

-

    Exercise E. View each of the volumes produced after the volume processing steps with tkmedit

4.5 Fill

The fill step cuts the hemispheres from each other and the brainstem from the brain. It can happen that the cutting planes are not set in the right place automatically. In these instances you can specify seed points, to indicate where the cuts should be made. To do this, follow the directions in the Troubleshooting Guide

5.0 Surface Processing Pipeline: a detailed look

5.1 Creating Final Surfaces
After data have passed through the volume processing stream, surface processing can begin. Below, the surface processing stream is described first; afterward, some specific exercises are presented that illustrate the correction of problems in the WM (white matter) volume and BRAIN volume (including setting control points) that sometimes cause the automatic topology fixer to generate geometric inaccuracies in the final surfaces.

The FILLED volume output from the volume processing stream is used in surface creation; the entire surface processing stream is run twice, once for each hemisphere. In this stream, the FILLED volume is first tessellated to create the orig surface. The orig surface is smoothed and inflated. Next, the topology correction is automatically run once. In the automatic topology correction steps, the inflated surface is transformed into spherical coordinates, corrected, and then smoothed and inflated again. Afterwards, the final surfaces are created. Visual checking of the final surfaces is necessary to check for geometric defects that may be present in the white and pial surfaces. The exercises below will examine several problems you may encounter that lead to such defects. These may require manual editing of control points, WM (white matter) or BRAIN volumes. Everything from the beginning of the subcortical segmentation through the final smoothing and inflation of the final surfaces can be run by using the command:

```
recon-all -autorecon2 -s bert
```

After correct white and pial surfaces are generated, the inflated surface is transformed into spherical coordinates, registered to the spherical atlas, and labels of cortical structures are automatically generated. Two versions of the surface parcellation are run, each using a different atlas. An output file is generated for each parcellation containing measurements of average thickness, surface area and more for each labeled area. This portion of the stream can be run with the command:

```
recon-all -autorecon3 -s bert
```

Also, a flattened surface may optionally be generated from the inflated surface. This surface can be loaded into tksurfer, where cuts along which the surface will be pulled apart and flattened are manually specified and subsequently used the mris_flatten script.

5.2 Troubleshooting Surface Problems

After you've run all the steps involved in generating the final surfaces there may still be inaccuracies or defects that need to be fixed to obtain accurate final surfaces. There are a few things that you can do to alter the final surfaces, those include: further edits to the wm.mgz volume, edits to the brainmask.mgz volume, addition of control points.

- If the WM volume requires editing to correct a defect, the processing stream can be re-entered using the command:

```
recon-all -autorecon2-wm -s bert
```

- If more control points were added, use:

```
recon-all -autorecon2-cp -s bert
```

- If the pial surface was not correct and the BRAINMASK volume had to be edited, re-enter the processing stream using:

```
recon-all -subjid <subject name> -autorecon2-pial
```

6.0 Surfaces: refining surface topology and creating final surfaces

6.1 What are topological defects?

In order to generate a homeomorphic (continuous, invertible) mapping from a subject's cortical model into spherical coordinates, the model must have the same topology as the target, in this case a sphere. The importance of this is that it guarantees that every point in the cortex is associated with one and only one set of spherical coordinates, and that every spherical coordinate maps to exactly one location in the cortex.

A topological "defect" is therefore a region of a cortical model in which the topology is not spherical. This can be visualized by imagining the cortex is a rubber sheet and attempting to deform it onto a sphere. If such a deformation can be found in which every point is mapped to exactly one spherical point, and every spherical point is mapped by exactly one cortical point, then the two surfaces are by definition topologically equivalent (this is in fact part of how we automatically correct the topology). More specifically, there are only two types of topological defects: holes (a perforation in the surface) and handles (an incorrect connection that overlaps the surface, e.g. between two banks of a sulcus). These are actually topologically equivalent, but require different corrections: the hole must be filled and the handle must be cut. Note that filling the handle would result in a surface with the correct topology as well, but one that no longer accurately followed the true cortical surface, which we call regions of "geometric inaccuracy". In this type of case, manually correcting the topological defect will correct the geometric accuracy of the final surface.

6.2 Automatic topology defect correction

As mentioned, the automated topology fixer takes care of removing topological defects from the surface. Automatic topology fixing is run by default, once the first inflated surface has been created. The time complexity of the topology correction goes as the square of the convex hull of the largest defect, so it can take quite long if there are large defects, and be quite rapid otherwise.

To run the topology fixer:

```
recon-all -fix -s bert
```

The topology fixer goes through the following steps and outputs the following files:

Quick sphere RH/LH surface: topologically defective right and left hemisphere surfaces are inflated to spheres

The output files written by this step are:

surface ${SUBJECTS_DIR}/<subject name>/surf/rh.qsphere

surface ${SUBJECTS_DIR}/<subject name>/surf/lh.qsphere

Fix topology RH/LH surface: automatic topology fixing of the right and left hemispheres

The output files written by this step are:

surface ${SUBJECTS_DIR}/<subject name>/surf/rh.orig

curv ${SUBJECTS_DIR}/<subject name>/surf/rh.defect_status

curv ${SUBJECTS_DIR}/<subject name>/surf/rh.defect_labels

surface ${SUBJECTS_DIR}/<subject name>/surf/lh.orig

curv ${SUBJECTS_DIR}/<subject name>/surf/lh.dect_status

curv ${SUBJECTS_DIR}/<subject name>/surf/lh.defect_labels

Resmooth RH/LH white matter: smooths rh.orig and lh.orig surfaces

The output files written by this step are:

surface ${SUBJECTS_DIR}/<subject name>/surf/rh.smoothwm

curv ${SUBJECTS_DIR}/<subject name>/surf/rh.curv

curv ${SUBJECTS_DIR}/<subject name>/surf/rh.sulc

curv ${SUBJECTS_DIR}/<subject name>/surf/rh.area

surface ${SUBJECTS_DIR}/<subject name>/surf/lh.smoothwm

curv ${SUBJECTS_DIR}/<subject name>/surf/lh.curv

curv ${SUBJECTS_DIR}/<subject name>/surf/lh.sulc

curv ${SUBJECTS_DIR}/<subject name>/surf/lh.area

Reinflate RH/LH white matter: inflates rh.orig and lh.orig surface

The output files written by this step are:

surface ${SUBJECTS_DIR}/<subject name>/surf/rh.inflated

surface ${SUBJECTS_DIR}/<subject name>/surf/lh.inflated

6.3 Final surfaces

This step creates the final left and right hemisphere cortical surfaces. The surfaces representing the gray/white boundary are called lh.white and rh.white, and the surfaces representing the gray/CSF boundary are called lh.pial and rh.pial. The white and pial surfaces are used to estimate the cortical thickness at all locations on the cortical surface. The thickness estimates are stored in curv files called lh.thickness and rh.thickness. The usage is:

```
recon-all -subjid <subject name> -make_final_surfaces
```

The output files written by this step are:

surface ${SUBJECTS_DIR}/<subject name>/surf/rh.white

surface ${SUBJECTS_DIR}/<subject name>/surf/rh.pial

curv ${SUBJECTS_DIR}/<subject name>/surf/rh.thickness

surface ${SUBJECTS_DIR}/<subject name>/surf/lh.white

surface ${SUBJECTS_DIR}/<subject name>/surf/lh.pial

curv ${SUBJECTS_DIR}/<subject name>/surf/lh.thickness

Right hemisphere white surface:
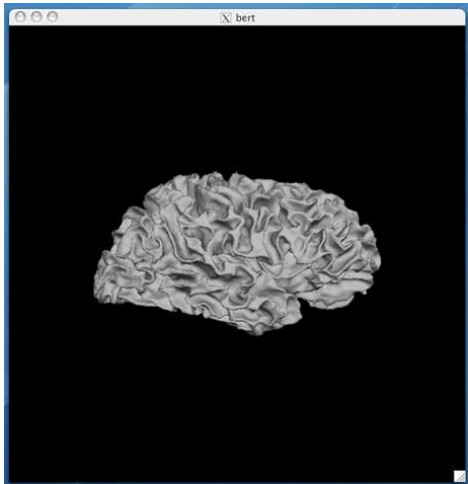
Right hemisphere pial surface:
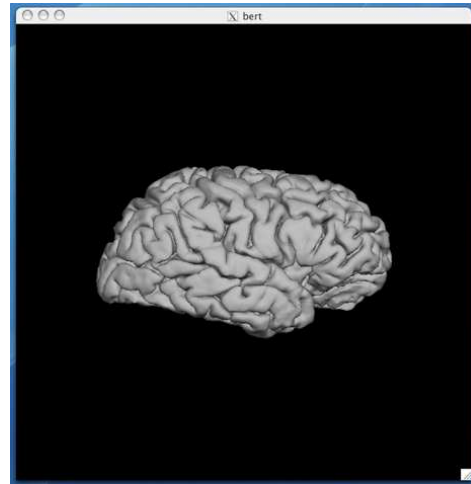
After passing through the topology fixer, it is guaranteed that there will be no topological defects in the surface.

6.4 What are Geometric Inaccuracies?

Inaccuracies in the gray/white boundary can occur for a number of reasons.

Of the three most common reasons for intervention, the first is that regions in which the original surface was topologically incorrect may have been automatically fixed in a topologically correct, but geometrically inaccurate manner. The resulting defect would show that the orig surface no longer follows the white matter segmentation. The problem must be corrected manually, by either filling holes (e.g., temporal lobe editing example) or erasing handles.

The second common case is when structural pathology results in incorrect segmentation, such as in the example below, where damaged white matter appears too dark and the voxels are therefore assigned non-white values, and the surface goes deep into the white matter. In these cases the segmentation must be manually corrected around the pathology.

The third common case is due to magnetic susceptibility artifacts, which, because of local compression of the image, cause voxels in regions that are not actually white matter to appear as bright as (or brighter than) the white matter. In these cases, the incorrectly segmented voxels must be manually erased (e.g., frontal lobe example below).

Other factors (such as MR artifacts, subject motion, etc.) can also lead to geometric defects that require manual intervention, but the above are most commonly encountered.

6.5 Correcting Geometric Inaccuracies in the Surfaces

After topology correction and surface deformation, the resulting gray-white (?h.white) and gray-CSF (?h.pial) surfaces should be overlaid on the brain volume and checked for geometric accuracy by visual inspection. Make sure that the surfaces follow the true boundaries of the top and bottom of the gray matter. In cases where they do not (e.g., due to damaged white matter that gets incorrectly segmented), there are several possible manual interventions:

1. The WM volume may be edited in order to fix incorrect segmentations. See this page for an explanation on how to fix errors of this type.

2. Control points may be placed to make the white matter more uniform. This can be effective, for example, when thin gyri are lost due to bias fields that make them darker than most of the white matter (the default white matter is 110, so if white matter voxels are darker than 90-95 they may be marked as non-white). In these cases, selecting some control points at the base of a gyrus can brighten the entire strand and recover its white matter voxels when the segmentation is rerun. Note that a control point can have an effect on a region around it due to interpolation of the bias correction field. Note also that the bias correction at a control point is estimated as 110 divided by the voxel value, so only control points placed at voxels less than 110 will increase the intensity of the white matter in a region. See this page for explanation on how to fix errors of this type.

3. The pial surface can be incorrect due to blood vessels and dura. In these cases, if the region is of interest, the brain volume can be edited to remove these structures, and the pial surface can be regenerated. See this page for an explanation on how to fix errors of this type.

- Exercise F. Troubleshooting: recognizing and fixing inaccuracies in the white matter surface
- Exercise G. Troubleshooting: correcting the pial surface

7.0 Surfaces: spherical and flattened surfaces, and cortical parcellation

7.1 Spherical Morphometry
This process creates the spherical left and right hemisphere cortical surfaces and then registers them with an average spherical cortical surface representation. The usage is:

```
recon-all -subjid <subject name> -morph
```

The morph_subject program goes through the following steps:

- **Sphere Right/Left Hemisphere Surfaces:** inflates right/left hemispheres to spheres and minimizes metric distortion

The output files written by this procedure are:

surface ${SUBJECTS_DIR}/<subject name>/surf/rh.sphere

surface ${SUBJECTS_DIR}/<subject name>/surf/lh.sphere

- **Register Right/Left Hemisphere Surfaces:** registers right/left spherical surfaces with surface-based atlas

The output files written by this procedure are:

surface ${SUBJECTS_DIR}/<subject name>/surf/rh.sphere.reg

surface ${SUBJECTS_DIR}/<subject name>/surf/lh.sphere.reg

Right hemisphere mapped to a spherical surface overlaid with rh.curv curvature information:
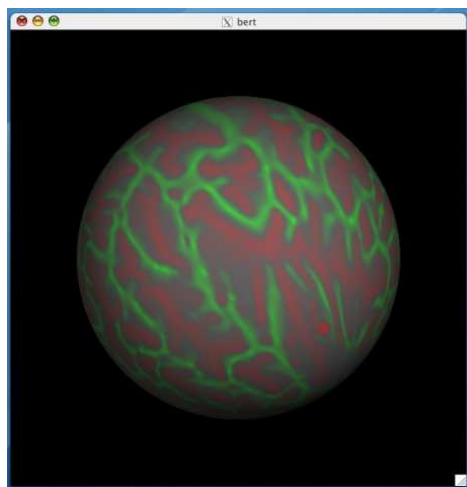
7.2 Automatic Cortical Parcellation
Spherical surfaces are registered with FreeSurfer's spherical atlas, to permit both group analysis and automatic cortical parcellation. As illustrated in the surface processing pipeline diagram, the spherical surface is used as input to mris_register and mris_ca_label, which generate lh.aparc.annot or rh.aparc.annot as output, shown below:
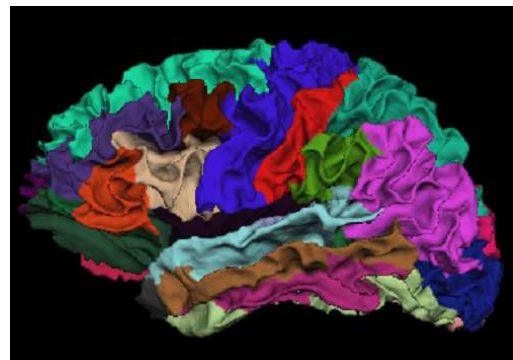
To load this surface in tksurfer, select **File -> label -> import annotation** and choose lh.aparc.annot or rh.aparc.annot from the list.

A table of values (e.g., volume, surface area, average cortical thickness) for each cortical region in the annotations is computed with mris_anatomical_stats and output to:

ascii ${SUBJECTS_DIR}/<subject name>/stats/rh.aparc.stats

ascii ${SUBJECTS_DIR}/<subject name>/stats/lh.aparc.stats

7.3 Image Flattening
The cutting and flattening process is optional and provides the user with flattened images of the whole brain or select parts (e.g., occipital lobe) of the brain. Relaxation cuts are made manually using tksurfer before flattening the surface.

The output files written by this procedure are:

surface ${SUBJECTS_DIR}/<subject name>/surf/rh.*.patch.flat

surface ${SUBJECTS_DIR}/<subject name>/surf/lh.*.patch.flat

-

Exercise H. View the final surfaces with tkmedit and tksurfer

8.0 Troubleshooting

Troubleshooting Guide

2008-06-01 20:47

-

To follow this exercise exactly be sure you've downloaded the tutorial data set before you begin. If you choose not to download the data set you can follow these instructions on your own data, but you will have to substitute your own specific paths and subject names.

Data Conversion

In this exercise, an image from a scanner, in DICOM format, will be converted into mgz format, using recon-all.

You first want to make sure you are working in the appropriate directory and that you have set your SUBJECTS_DIR variable correctly. You can do this by entering the following directory:

```
cd $FREESURFER_HOME/subjects/buckner_data/tutorial_subjs
```

Set the current directory to be the default subjects directory using this command:

```
setenv SUBJECTS_DIR ${PWD}
```

Two acquisitions of an anonymized volume have been provided for this tutorial with the recognizable face characteristics removed, in the directory buckner_data/tutorial_subjs/014-anon and buckner_data/tutorial_subjs/015-anon. Using the recon-all script you can specify both the acquisitions to be converted to mgz format as well as the name you wish to give your subject in the following format:

recon-all -i <in volume 1> -i <in volume 2> -s <subject name>

where <in volume 1> is the first file that appears in the first acquisition directory, and <in volume 2> is the first file that appears in the second acquisition directory. For this example we will use *anon* as our <subject name>. To convert the two *anon* acquisitions use the command:

```
recon-all -i 014-anon/001.dcm -i 015-anon/001.dcm -s anon
```

The output file list can be viewed with the following command:

```
ls ${SUBJECTS_DIR}/anon/mri/orig
```

which will show:

```
001.mgz  002.mgz
```

View the output of the data conversion with tkmedit:

```
tkmedit -f ${SUBJECTS_DIR}/anon/mri/orig/001.mgz
```

The full path to the converted volume is specified with the -f option because a file is being specified explicitly. If tkmedit is called without the -f, it will assume that the data is in ${SUBJECTS_DIR}/<subject name>/mri, where <subject name> is the first argument on the tkmedit command line (anon in the example below).

You can motion correct your two acquisitions using the -motioncor option of recon-all like this:

```
recon-all -motioncor -s anon
```

Once motion correction has been performed using 001.mgz and 002.mgz, the volume ${SUBJECTS_DIR}/anon/mri/orig.mgz has been produced, tkmedit will be able to find that volume using the subject directory and the volume name:

```
tkmedit anon orig.mgz
```

2008-06-01 20:47

Image before and after comparisons (volume processing pipeline)

Volume processing pipeline
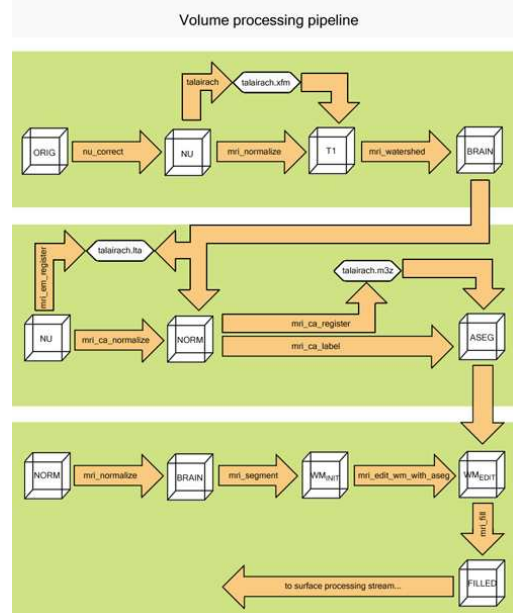
1.0 N3 intensity normalization: nu_correct_subject <subject name>
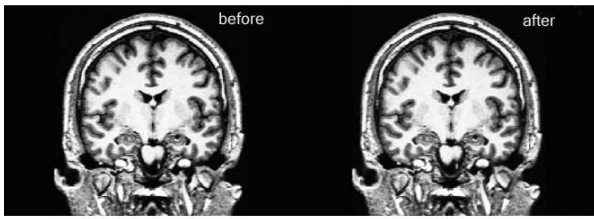
2.0 Cortical intensity normalization: mri_normalize orig T1
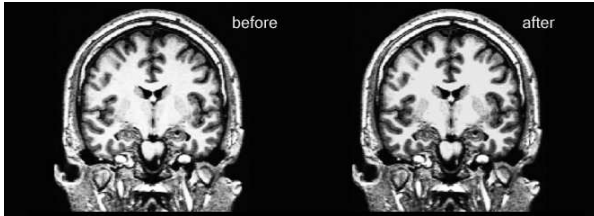
3.0 Cortical Talairach: talairach <subject name>
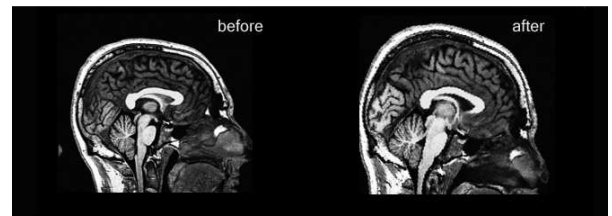
4.0 Cortical skull stripping: mri_watershed T1 brain
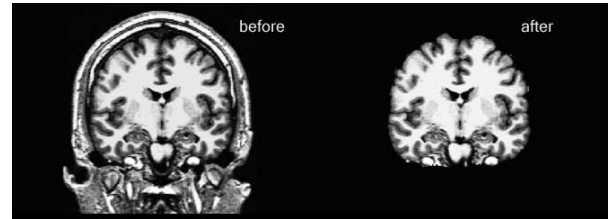
5.0 Subcortical intensity normalization: mri_ca_normalize -mask brain nu $GCA transforms/talairach.lta norm
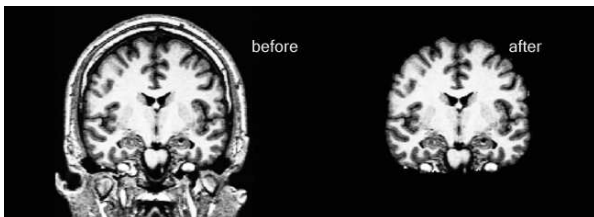
6.0 Subcortical segmentation: mri_ca_label norm transforms/talairach.m3d $GCA aseg
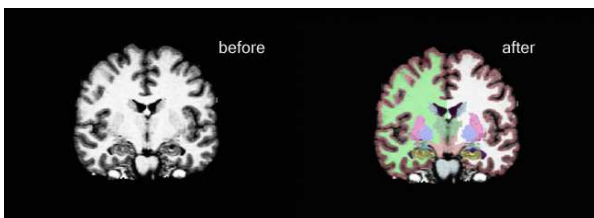
7.0 Subcortical brain intensity normalization: mri_normalize norm brain
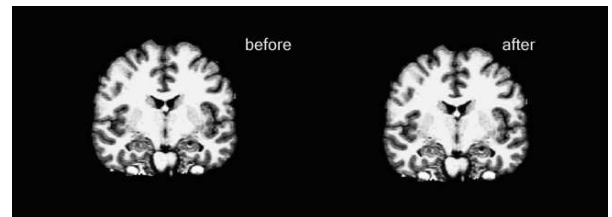
8.0 Subcortical white matter segmentation: mri_segment brain wm
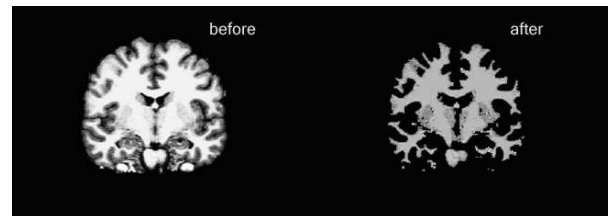
9.0 Auto segmentation Editing: mri_edit_wm_with_aseg wm aseg wm
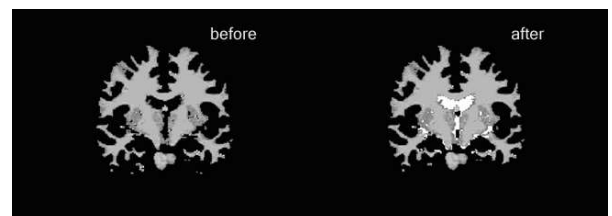
10.0 Filling: mri_fill wm filled

-

To follow this exercise exactly be sure you've downloaded the tutorial data set before you begin. If you choose not to download the data set you can follow these instructions on your own data, but you will have to substitute your own specific paths and subject names.

Visually check the cortical reconstruction output with tkmedit

For each step so far, look at the output generated for the subject 'bert'. As you are performing a cortical reconstruction, it is a good idea to visually examine your data for errors or defects that may have been generated in the reconstruction steps. When using tkmedit, scroll through the volumes with the up and down arrow keys to visually inspect each slice. For a reminder of which inputs and outputs are taken by each step in the reconstruction, refer to this guide First, set your subjects directory environment variable:

```
cd $FREESURFER_HOME/subjects/buckner_data/tutorial_subjs
setenv SUBJECTS_DIR ${PWD}
```

Data conversion
Before preprocessing is done, data must be converted from native scanner format to mgz format. The output of this step is the bert/mri/orig.mgz. To open it in tkmedit:

```
tkmedit bert orig.mgz
```

Intensity normalization
The intensity normalization step takes the bert/mri/orig volume as input and outputs the bert/mri/T1 volume. Look at these volumes simultaneously to see the difference from the intensity normalization step. Use Ctrl-1 and Ctrl-2 to switch between the volumes.

```
tkmedit bert T1.mgz -aux orig.mgz
```

Skull stripping
The skull stripping step takes the bert/mri/T1 volume as input, strips off the skull, and outputs the bert/mri/brainmask volume. Look at the T1 and brainmask volumes simultaneously with tkmedit. Use Ctrl-1 and Ctrl-2 to switch between the volumes.

```
tkmedit bert brainmask.mgz -aux T1.mgz
```

Segmentation
The segmentation step takes the /bert/mri/brainmask volume as input, removes the gray matter and CSF, and outputs the /bert/mri/wm volume. Look at the brainmask and wm volumes simultaneously with tkmedit. Use Ctrl-1 and Ctrl-2 to switch between the volumes.

```
tkmedit wm.mgz -aux brainmask.mgz
```

-

To follow this exercise exactly be sure you've downloaded the tutorial data set before you begin. If you choose not to download the data set you can follow these instructions on your own data, but you will have to substitute your own specific paths and subject names.

Visually check the cortical reconstruction output with tkmedit and tksurfer

For each step so far, look at the output generated for the subject 'bert'. Each time you perform a cortical reconstruction, it is a good idea to visually examine the final surfaces generated after automatic topology fixing for errors or defects that may have been generated in the reconstruction steps. First, set your subjects directory environment variable:

```
cd $FREESURFER_HOME/subjects/buckner_data/tutorial_subjs
setenv SUBJECTS_DIR ${PWD}
```

Generating cutting planes and filling
This step takes the /bert/mri/wm.mgz segmented white matter volume as input, and outputs the bert/mri/filled.mgz volume. To view this volume in tkmedit:

```
tkmedit bert filled.mgz
```

Tessellate
This step takes the /bert/mri/filled.mgz volume as input, and outputs the surfaces bert/surf/lh.orig and /bert/surf/rh.orig for the left and right hemispheres, respectively. To view these surfaces in tksurfer:

```
tksurfer bert lh orig
tksurfer bert rh orig
```

Smoothing and Inflation
The smoothing and inflation step takes the bert/surf/lh.orig and /bert/surf/rh.orig surfaces as input, and outputs the following surfaces in the bert/surf directory: lh.smoothwm, rh.smoothwm, lh.curv, rh.curv, lh.sulc, rh.sulc, lh.inflated, and rh.inflated. To view the smoothed and inflated surfaces in tksurfer:

```
tksurfer bert lh smoothwm
tksurfer bert rh smoothwm
tksurfer bert lh inflated
tksurfer bert rh inflated
```

To view the curv files, load the lh inflated surface into tksurfer again. Go to **File -> Curvature -> Load Curvature...**, click **Browse**, and select the file lh.curv. You will see a red and green pattern overlaid on the inflated surface that represents the curvature of the white matter at that location. Repeat with the right hemisphere inflated and curv files.

Generate final surfaces
The final surfaces are generated by taking the bert/surf/lh.orig and bert/surf/rh.orig surfaces as inputs, and outputs the following surfaces in the bert/surf directory: lh.white, rh.white, lh.pial, rh.pial, lh.thickness, rh.thickness. As mentioned previously, the white and pial surfaces are particularly important to check for defects before continuing through the surface processing pipeline. To view the surfaces in tksurfer:

```
tksurfer bert lh white
tksurfer bert rh white
tksurfer bert lh pial
tksurfer bert rh pial
```

Generate spherical surfaces
The spherical surface is generated by taking the bert/surf/lh.inflated and bert/surf/rh.inflated surfaces as inputs and outputting the surfaces bert/surf/lh.sphere and bert/surf/rh.sphere. To view these files in tksurfer:

```
tksurfer bert lh sphere
tksurfer bert rh sphere
```

To make these spheres more interesting, view the curv files overlaid on the spherical surfaces. Load the lh sphere surface into tksurfer again. Go to **File -> Curvature -> Load Curvature...**, click **Browse**, and select the file lh.curv. You will see a red and green pattern overlaid on the spherical surface that represents the curvature of the white matter at that location. Repeat with the right hemisphere sphere and curv files.

FreeSurfer Tutorial: Process Flow

This table shows the recon-all steps for the current **dev** version of FreeSurfer. See ReconAllStableTablev4 to see a process flow for the latest **stable** version of FreeSurfer.

Click here to see this information presented in a block diagram format and here for a process v. files table.

See also the OtherUsefulFlags for other recon-all options.

recon-all step	Individual Flag	Input	Command Line	Output
	-i <invol1> -i <invol2>	invol1.dcm	mri_convert invol1.dcm orig/001.mgz	orig/001.mgz
		invol2.dcm	mri_convert invol2.dcm orig/002.mgz	orig/002.mgz
	-motioncor	orig/001.mgz	mri_motion_correct.fsl -o rawavg.mgz -wild orig/001.mgz orig/002.mgz	rawavg.mgz
		orig/002.mgz		
		rawavg.mgz	mri_convert rawavg.mgz orig.mgz --conform	orig.mgz
		orig.mgz	mri_add_xform_to_header -c transforms/talairach.xfm orig.mgz orig.mgz	orig.mgz
	-nuintensitycor	orig.mgz	mri_nu_correct.mni -i orig.mgz --o nu.mgz --n 2	nu.mgz
recon-all -autorecon1 -subjid <subjid>	-talairach	talairach_avi	talairach_avi --i nu.mgz --xfm transforms/talairach.auto.xfm	transforms/talairach.auto.xfm
		transforms/talairach.auto.xfm	cp transforms/talairach.auto.xfm transforms/talairach.xfm	transforms/talairach.xfm
		transforms/talairach.xfm	talairach_afd -T 0.005 -xfm transforms/talairach.xfm	transforms/talairach.xfm
			awk -f $FREESURFER_HOME/bin/extract_talairach_avi_QA.awk transforms/talairach_avi.log	transforms/talairach_avi.log
	-normalization	nu.mgz	mri_normalize -g 1 nu.mgz T1.mgz	T1.mgz
		nu.mgz	mri_em_register -skull nu.mgz	transforms/talairach_with_skull.lta
	-skullstrip		$FREESURFER_HOME/average/RB_all_withskull_2007-08-08.gca transforms/talairach_with_skull.lta	
		T1.mgz	mri_watershed -T1 -brain_atlas	brainmask.auto.mgz
			$FREESURFER_HOME/average/RB_all_withskull_2007-08-08.gca transforms/talairach_with_skull.lta T1.mgz brainmask.auto.mgz	
		brainmask.auto.mgz	cp brainmask.auto.mgz brainmask.mgz	brainmask.mgz

97

(continuation of process flow table, page 98)

recon-all -autorecon2 -subjid <subjid>	-gcareg	brainmask.mgz	mri_em_register -mask brainmask.mgz nu.mgz $FREESURFER_HOME/average/RB_all_2007-08-08.gca transforms/talairach.lta	transforms/talairach.lta
	-canorm	brainmask.mgz	mri_ca_normalize -mask brainmask.mgz nu.mgz $FREESURFER_HOME/average/RB_all_2007-08-08.gca transforms/talairach.lta norm.mgz	norm.mgz
	-careg	brainmask.mgz transforms/talairach.lta norm.mgz	mri_ca_register -align-after -nobigventricles -mask brainmask.mgz -T transforms/talairach.lta norm.mgz $FREESURFER_HOME/average/RB_all_2007-08-08.gca transforms/talairach.m3z	transforms/talairach.m3z
	-careginv	transforms/talairach.m3z	mri_ca_register -invert-and-save transforms/talairach.m3z	transforms/talairach.m3z.inv.x.mgz transforms/talairach.m3z.inv.y.mgz transforms/talairach.m3z.inv.z.mgz
	-rmneck	nu.mgz transforms/talairach.m3z	mri_remove_neck -radius 25 nu.mgz transforms/talairach.m3z $FREESURFER_HOME/average/RB_all_2007-08-08.gca nu_noneck.mgz	nu_noneck.mgz
	-skull-lta	transforms/talairach.lta nu_noneck.mgz	mri_em_register -skull -t transforms/talairach.lta nu_noneck.mgz $FREESURFER_HOME/average/RB_all_withskull_2007-08-08.gca transforms/talairach_with_skull.lta	transforms/talairach_with_skull.lta
	-calabel	norm.mgz	mri_ca_label -align -nobigventricles norm.mgz transforms/talairach.m3z	aseg.auto_noCCseg.mgz
		aseg.auto_noCCseg.mgz	mri_cc -aseg aseg.auto_noCCseg.mgz -o aseg.auto.mgz -subjid	aseg.auto.mgz
		aseg.auto.mgz	cp aseg.auto.mgz aseg.mgz	aseg.mgz
	-normalization2	brainmask.mgz norm.mgz	mri_normalize -aseg aseg.mgz -mask brainmask.mgz norm.mgz brain.mgz	brain.mgz
	-maskbfs	brain.mgz	mri_mask -T 5 brain.mgz brainmask.mgz brain.finalsurfs.mgz	brain.finalsurfs.mgz
	-segmentation	brain.mgz	mri_segment brain.mgz wm.seg.mgz	wm.seg.mgz
		wm.seg.mgz aseg.mgz	mri_edit_wm_with_aseg wm.seg.mgz brain.mgz aseg.mgz wm.asegedit.mgz	wm.asegedit.mgz
		wm.asegedit.mgz norm.mgz	mri_pretess wm.asegedit.mgz wm norm.mgz wm.mgz	wm.mgz
	-fill	aseg.auto_noCCseg.mgz norm.mgz	mri_fill -a ./scripts/ponscc.cut.log -xform transforms/talairach.lta -segmentation aseg.auto_noCCseg.mgz wm.mgz filled.mgz	filled.mgz ./scripts/ponscc.cut.log
recon-all -autorecon2 -subjid <subjid>	-tessellate	filled.mgz norm.mgz	mri_pretess filled.mgz 255 norm.mgz filled-pretess255.mgz	filled-pretess255.mgz
		filled-pretess255.mgz	mri_tessellate filled-pretess255.mgz 255 lh.orig.nofix	lh.orig.nofix
		filled.mgz norm.mgz	mri_pretess filled.mgz 127 norm.mgz filled-pretess127.mgz	filled-pretess127.mgz
		filled-pretess127.mgz	mri_tessellate filled-pretess127.mgz 127 rh.orig.nofix	rh.orig.nofix
		?h.orig.nofix	mris_extract_main_component ?h.orig.nofix ?h.orig.nofix	?h.orig.nofix
			rm -f filled-pretess255.mgz filled-pretess127.mgz	
	-smooth1	?h.orig.nofix	mris_smooth -nw ?h.orig.nofix ?h.smoothwm.nofix	?h.smoothwm.nofix
	-inflate1	?h.smoothwm.nofix	mris_inflate -no-save-sulc ?h.smoothwm.nofix ?h.inflated.nofix	?h.inflated.nofix
	-qsphere	?h.inflated.nofix	mris_sphere -q ?h.inflated.nofix ?h.qsphere.nofix	?h.qsphere.nofix
		?h.orig.nofix	cp ?h.orig.nofix ?h.orig	?h.orig
	-fix	?h.inflated.nofix ?h.qsphere.nofix	mris_fix_topology -mgz -sphere qsphere.nofix -ga <subjid> ?h	?h.orig
		?h.orig	mris_euler_number ?h.orig	?h.orig
		?h.orig	mris_remove_intersection ?h.orig ?h.orig	?h.orig
			rm ?h.inflated	
	-finalsurfs	brain.finalsurfs.mgz ?h.orig	mris_make_surfaces -noaparc -mgz -T1 brain.finalsurfs <subjid> ?h	?h.white ?h.pial ?h.thickness ?h.curv ?h.area ?h.cortex.label
	-surfvolume	?h.area ?h.area.pial	mris_calc -o ?h.area.mid ?h.area add ?h.area.pial	?h.area.mid
		?h.area.mid	mris_calc -o ?h.area.mid ?h.area.mid div 2	?h.area.mid
		?h.area.mid ?h.volume	mris_calc -o ?h.volume ?h.area.mid mul ?h.thickness	?h.volume
	-smooth2	?h.white	mris_smooth -n 3 -nw ?h.white ?h.smoothwm	?h.smoothwm
	-inflate2	?h.smoothwm	mris_inflate ?h.smoothwm ?h.inflated	?h.inflated ?h.sulc
		?h.inflated	mris_curvature -thresh .999 -n -a 5 -w -distances 10 10 ?h.inflated	?h.inflated.H ?h.inflated.K
	-segstats	aseg.mgz	mri_segstats --seg mri/aseg.mgz --sum stats/aseg.stats --pv mri/norm.mgz --excludeid 0 --brain-vol-from-seg --brainmask mri/brainmask.mgz --in mri/norm.mgz --in-intensity-name norm --in-intensity-units MR --etiv --surf-wm-vol --ctab $FREESURFER_HOME/ASegStatsLUT.txt --subject <subjid>	stats/aseg.stats
	-cortribbon	orig.mgz ?h.white ?h.pial	mris_volmask --label_left_white 2 --label_left_ribbon 3 --label_right_white 41 --label_right_ribbon 42 --save_ribbon --save_distance <subjid>	?h.ribbon.mgz

98

(continuation of process flow table, page 99)

recon-all -autorecon2 -subjid <subjid>	-sphere	?h.inflated	mris_sphere ?h.inflated ?h.sphere	?h.sphere
	-surfreg	?h.sphere	mris_register -curv ?h.sphere $FREESURFER_HOME/average/?h.average.curvature.filled.buckner40.tif ?h.sphere.reg	?h.sphere.reg
	-jacobian_white	?h.white ?h.sphere.reg	mris_jacobian ?h.white ?h.sphere.reg ?h.jacobian_white	?h.jacobian_white
	-avgcurv	?h.sphere.reg	mrisp_paint -a 5 $FREESURFER_HOME/average/?h.average.curvature.filled.buckner40.tif ?h.sphere.reg ?h.avg_curv	?h.avg_curv
	-cortparc	aseg.mgz ?h.sphere.reg	mris_ca_label -aseg mri/aseg.mgz <subjid> ?h ?h.sphere.reg $FREESURFER_HOME/average/?h.curvature.buckner40.filled.desikan_killiany.2007-06-20gcs ?h.aparc.annot	label/?h.aparc.annot
	-parcstats	label/?h.aparc.annot	mris_anatomical_stats -mgz -f stats/?h.aparc.stats -b -a label/?h.aparc.annot -c label/aparc.annot.ctab <subjid> ?h	stats/?h.aparc.stats label/aparc.annot.ctab
	-cortparc2	aseg.mgz ?h.sphere.reg	mris_ca_label -aseg mri/aseg.mgz <subjid> ?h ?h.sphere.reg $FREESURFER_HOME/average/?h.atlas2005_simple.gcs label/?h.aparc.a2005s.annot	label/?h.aparc.a2005s.annot
recon-all -autorecon3 -subjid <subjid>	-parcstats2	label/?h.aparc.a2005s.annot	mris_anatomical_stats -mgz -f stats/?h.aparc.a2005s.stats -b -a label/?h.aparc.a2005s.annot -c label/aparc.annot.a2005s.ctab <subjid> ?h	stats/?h.aparc.a2005s.stats stats/aparc.annot.a2005s.ctab
	-aparc2aseg	aseg.mgz label/?h.aparc.annot	mri_aparc2aseg --s <subjid> --ribbon	aparc+aseg.mgz
		aseg.mgz label/?h.aparc.a2005s.annot	mri_aparc2aseg --s <subjid> --ribbon --a2005s	aparc.a2005s+aseg.mgz
	-wmparc	aparc+aseg.mgz	mri_aparc2aseg --s <subjid> --labelwm --hypo-as-wm --rip-unknown --ribbon mri/wmparc.mgz --ctxseg aparc+aseg.mgz	wmparc.mgz
		wmparc.mgz	mri_segstats --seg mri/wmparc.mgz --sum stats/wmparc.stats --pv mri/norm.mgz --excludeid 0 --brain-vol-from-seg --brainmask mri/brainmask.mgz --in mri/norm.mgz --in-intensity-name norm --in-intensity-units MR --etiv --subject <subjid> --surf-wm-vol --ctab $FREESURFER_HOME/FreeSurferColorLUT.txt	wmparc.stats

Differences from Stablev4.0.2

- mri_cc uses aseg.mgz whereas in v4.0.2, mri_cc uses aseg.auto.mgz.
  - For those using v4.0.2, this means that if edits are made to the aseg.mgz, you must copy the aseg.mgz to aseg.auto.mgz before running -ccseg to generate a new cc segmentation based on your edits.

2008-06-01 20:49

99

FreeSurfer Tutorial: Surface Group Analysis with QDEC

-

To follow this exercise exactly be sure you've downloaded the tutorial data set before you begin. If you choose not to download the data set you can follow these instructions on your own data, but you will have to substitute your own specific paths and subject names. If you are using the tutorial data please set the environmental variable TUTORIAL_DATA to the location that you have downloaded the data to:

```
tcsh
setenv TUTORIAL_DATA $FREESURFER_HOME/subjects/buckner_data/tutorial_subjs
```

- Notice the command to open tcsh. If you are already running the tcsh command shell, then the 'tcsh' command is not necessary.

In this tutorial, you will learn how to perform statistical analysis of group surface-based data, including:

-

Preprocessing the group data

Constructing a qdec.table.dat file of subject demographics
-

Using QDEC to design and execute your analysis
- Interacting with the Qdec display
- Creating Regions of Interest (ROIs) for further analysis and a final check of your data

Assuming that all surface reconstruction has been completed for all subjects in the study, FreeSurfer's QDEC utility can be used to perform inter-subject/group averaging and inference on the cortical surface. QDEC permits statistical inferences to be made about effects of interest in relation to error variance. The mri_glmfit command is used to model the data as a linear combination of effects related to variables of interest, confounds and errors. QDEC also allows for certain permutation testing and other means for correcting for multiple comparisons. For group analysis, this technique fits a general linear model (GLM) at each surface vertex to explain the data from all subjects in the study. In this section, a brief overview of linear modeling is presented and can be skipped if this material is already familiar. Other software packages have similar types of programs (e.g., FSL's GFEAT).

100

Introduction

QDEC

QDEC is a single-binary application included in the FreeSurfer distribution. QDEC is an acronym for Query, Design, Estimate, Contrast. It is intended to aid researchers in performing inter-subject / group averaging and inference on the morphometry data (cortical surface and volume) produced by the FreeSurfer processing stream. Qdec is a GUI front-end to a 'statistics engine' (the mri_glmfit binary, included in FreeSurfer, currently fills this role) intended to:

1. select the subjects meeting the criteria under study
2. generate the necessary input to the stats engine, which, for mri_glmfit, includes:
   1.
      a **Design** matrix (called **X** in the GLM equation) containing the explanatory variables,
   2.
      a parameter **Estimate** matrix (called **A** in the GLM equation), and
   3.
      the **Contrast** vector(s)
3. generate and optionally display the output data and/or images

Linear Modeling overview

Linear modeling describes the observed data as a linear combination of explanatory factors plus noise, and determines how well that description explains the data being analyzed. In order to understand how to perform group analysis in FreeSurfer, you need to understand the general linear model (GLM) and how to construct a GLM in matrix notation. You can click here for a review of this material. The notation we use here is:$y=X*beta$, where **y** is the vector observed data (e.g., thicknesses for each subject at a vertex), **X** is the known design matrix (e.g., gender, age), and **beta** is the vector of unknown parameter estimates (PEs). The interpretation of the PEs will depend upon how **X** is constructed. For example, they could be interpreted as a slope indicating the change of thickness with age. The analysis/estimation is then the process of computing **beta** given the data **y** and the design matrix **X**. A Null Hypothesis (H0) is constructed with a contrast matrix **C**. Inferences are drawn by testing whether the value **gamma=Cb** is zero.

Preprocess Group Data

set SUBJECTS_DIR

If you are using the tutorial data you will need to set your SUBJECTS_DIR to the directory with the group analysis data in it. If you have installed your tutorial data to your $FREESURFER_HOME the command to set your SUBJECTS_DIR is:

```
setenv SUBJECTS_DIR $TUTORIAL_DATA/group_analysis_tutorial
cd $SUBJECTS_DIR
```

---

* this will set your SUBJECTS_DIR to the location where your tutorial data is located, in the directory where the group analysis subjects are, if you have defined the variable TUTORIAL_DATA as indicated at the top of this tutorial. If you are not using the tutorial data you should set your SUBJECTS_DIR to the directory in which the subjects you will use for this tutorial are located.

recon-all

Prior to using the qdec application, your group subject data must be processed by the standard FreeSurfer processing stream, via the recon-all script. A freesurfer tutorial is available. This processing stream supplies the surfaces and morphometry data on each subject. The data in the tutorial set has been processed for you.

pre-smoothed fsaverage surfaces

Qdec needs each subject to have pre-computed smoothed data for the target surface (fsaverage is the default) for each measure (thickness, sulc, area, curv, etc.). Your SUBJECTS_DIR should contain either a link or a copy of the 'fsaverage' subject found in your $FREESURFER_HOME/subjects directory. Presmoothing the data onto the target surface is not part of the normal recon processing stream, but you can easily create this data with recon-all, using the command:

```
recon-all -s <subjid> -qcache
```

For the purposes of this tutorial, the -qcache command has been run for all of the subjects. The -qcache flag will run numerous back-to-back mris_preproc processes on your machine, so be prepared for it to run for about an hour. The help text of recon-all -help contains a section on other -qcache otpions.

qdec.table.dat

The primary input to qdec is a text file, named qdec.table.dat, containing the subject IDs, and discrete and continuous factors, in table format. This is essentially a table of demographics for your subject including all the variables and factors that you wish to consider. Here is an example file:

```
fsid            gender  age  diagnosis    Left-Cerebral-White-Matter
011121_vc8048   Female  70   Demented     202291
021121_62313-2  Female  71   Demented     210188
010607_vc7017   Female  73   Nondemented  170653
021121_vc10557  Male    75   Demented     142029
020718_62545    Male    76   Demented     186087
020322_vc8817   Male    77   Nondemented  149810
```

For each discrete factor, there should exist a file named <factor>.levels which lists all possible levels. For example, accompanying the example qdec.table.dat file must be a file named 'gender.levels' containing these lines:

```
Female
Male
```

and there must be a file named 'diagnosis.levels' containing these lines:

```
Demented
Nondemented
```

Of course you may have different discrete factor names and levels (or even no discrete factors, in which case all column data are assumed to be continous factors). Continuous factors do not need a <factor>.levels file to define them.

For organizational purposes it is best to make a directory called qdec within your $SUBJECTS_DIR. You can save the qdec.table.dat and <factor>.levels files in there. When Qdec runs it will also save your analyses to this directory. A qdec subdirectory, with a qdec.table.dat has been made for you. Here is a sample of what that looks like:

```
fsid   gender  age   Left-Hippocampus     Right-Hippocampus
004    Male    72    4250                 4452
008    Female  88    2569                 3310
017    Male    25    4163                 3938
021    Male    22    4034                 4144
032    Male    79    3459                 3443
039    Female  69    3865                 4013
```

This file contains 40 subject IDs, their gender, age, left hippocampus volume, and right hippocampus volume. To accompany this file you will need to create a <factor>.levels text file to define the discrete factor names. You can do this using the text editor of your choice (emacs, vi, pico...). For this tutorial, the qdec.table.dat file already exists for you.

Optionally, your qdec.table.dat file can explicitly specify the SUBJECTS_DIR, by including it in the first non-commented line:

```
# This is a comment-line in the qdec.table.dat file.
# This explicitly specifies the SUBJECTS_DIR:
SUBJECTS_DIR /my/path/to/subject/data
.
.
.
```

fsaverage

For display purposes, you will need to have an average subject included in your SUBJECTS_DIR. Freesurfer's fsaverage, made in MNI305 space, will do fine:

```
cd $SUBJECTS_DIR
if (! -e fsaverage) ln -s $FREESURFER_HOME/subjects/fsaverage
```

This will add a copy of fsaverage into your SUBJECTS_DIR.
For the purposes of this tutorial, fsaverage has already been linked to your $SUBJECTS_DIR

If you wish to make your own average subject from your set you can do so using make_average_subject.

Usage

To start qdec, from your $SUBJECTS_DIR, simply type qdec:

```
setenv SUBJECTS_DIR $TUTORIAL_DATA/group_analysis_tutorial
cd $SUBJECTS_DIR

qdec &
```

It may take a few seconds for Qdec to open. The ampersand directs the terminal to run this process in the background, so you may see your command prompt return before Qdec opens.

Subjects

When Qdec opens you are looking at the Subjects tab. The first thing you will need to do is to load your qdec.table.dat file. Click **File -> Load Data Table**, or you can use the ▦ button, and traverse to your subjects directory and select the qdec.table.dat file that you created. When you click Open, it should load your file, the contents scrolling by in the terminal window. If the data are loaded correctly, you should see in the terminal window a summary, like this example:

```
Number of subjects:   40
Number of factors:    188 (1 discrete, 187 continuous)
Number of classes:    2
Number of regressors: 376
```
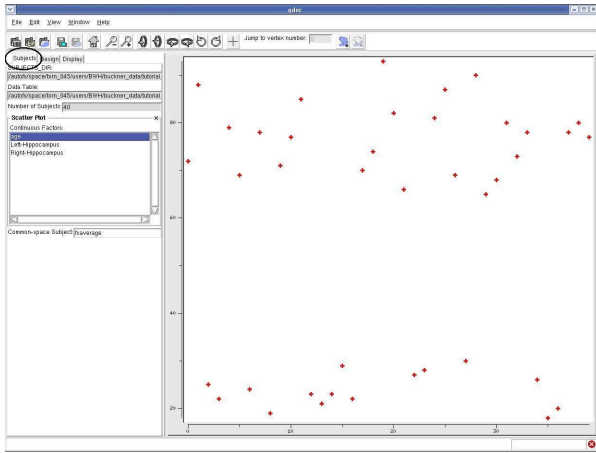
Your continuous (age, Left-Hippocampus, and Right-Hippocampus) factors should appear in a list under **Scatter Plot** on the control panel. If you choose a factor from this list a scatter plot of your data will appear in the window. The x-axis has the subject number (taken from the order the subjects are listed in the qdec.table.dat), and the y-axis has the value of the variable you've selected. In the example shown you can see a plot of the ages of all 40 subjects. You can use this to visually check your data for outliers. In QDEC if you roll your cursor over one of the points on the plot you can find out which subject it is, the ID will be shown in the lower left corner of the QDEC interface.

Design

When you click over to the Design tab your discrete (gender) and continuous (age, Left-Hippocampus, and Right-Hippocampus) factors should appear.
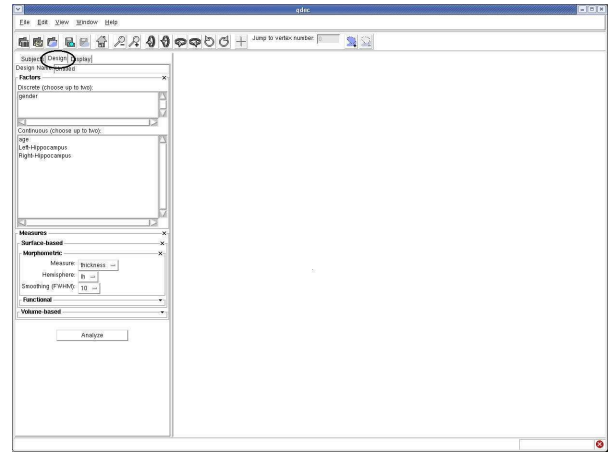
You can select up to four factors in the Design tab to regress against. For the tutorial data, you could select 'gender' and 'age' and 'Left-Hippocampus' and 'Right-Hippocampus', or any combination of those. For simplicity in this example, choose only 'age' leaving the 'Measure', 'Hemisphere' and 'Smoothing' at their defaults (thickness, lh and 10mm). Before you click the 'Analyze' button you will want to name your Design, something like 'LH-Thickness-Age-sm10', and enter that into the "Design Name" text entry box at the top of the window. Now click the 'Analyze' button and the stats will begin processing, executing the mri_glmfit executable. Upon clicking 'Analyze', the terminal will display the output of this processing. Also, progress information is shown in the bottom bar of the QDEC application.
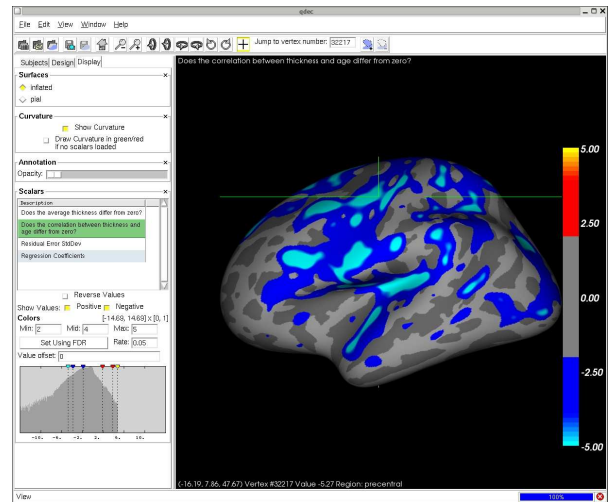
Display

Once the analysis is complete (taking up to several minutes for a large subject set), you can click the Display tab and the fsaverage inflated surface will appear in the display window. You will see a list of questions summarizing the various analyses that were completed.
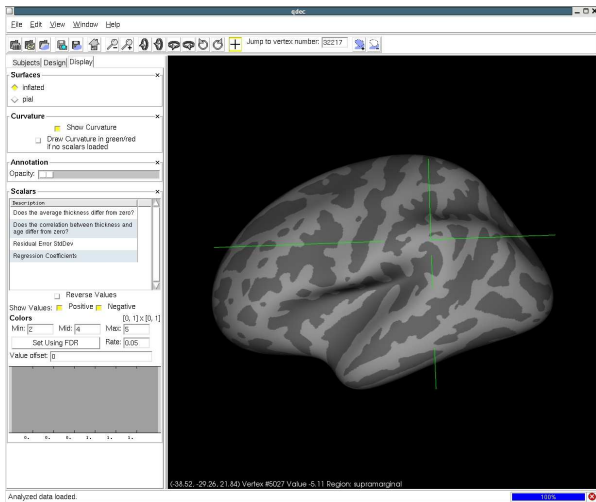
You can click on one of these questions to load the results. If you click on **Does the correlation between thickness and age differ from zero** it will display the statistically significant regions where age and thickness are correlated. Here is an example display:

Notice the green cross-hairs that indicate the vertex you have currently selected. You can change vertices and display a plot of the data for a particular vertex by left-clicking on a point while holding-down the Ctrl key. If you'd like to turn off the cursor display you can use the ⊞ button. Here is an example plot that corresponds to the shown selected vertex:
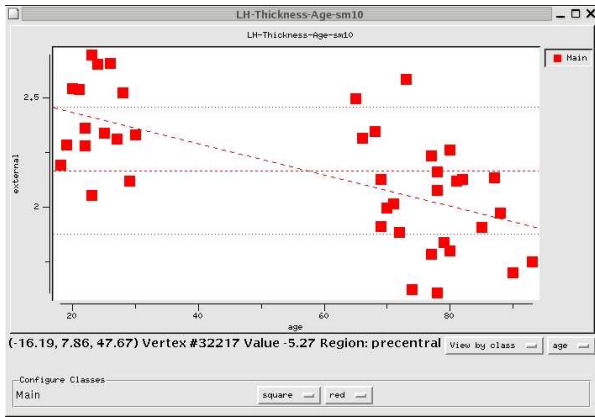
The plot shows your measure on the y-axis (vertical) - in this case, cortical thickness - and the variable on the x-axis (horizontal) - in this case, age. Each data point on the plot is representative of an individual subject, denoting their age and cortical thickness at the vertex you have selected. For this example at this vertex, we can see that the cortex is thinning with age. The information at the bottom of the both the plot window and the QDEC window shows that this vertex has surface coordinates (-16.19, 7.86, 47.67) and is Vertex# 32217. The significance value is -5.27 and it is in the precentral region. The significance in this display is a -log(10)p value, and not a straight p value.

Interacting with your data

Rotating, Panning and Zoom You can rotate the display, hold-down the left mouse button and move the mouse. Holding-down the middle button while moving the mouse will move the display in the window. Holding-down the right mouse button while moving will zoom the display.

There are buttons at the top of the Qdec display that will rotate and zoom as well:

 - rotates up 90 degrees

 - rotates down 90 degrees

 - rotates right 90 degrees

 - rotates left 90 degrees

 - rotates counterclockwise 90 degrees

 - rotates clockwise 90 degrees

 - zoom out

 - zoom in

If you get it rotated too far, the home button will reset it.

Parcellation Display The cortical parcellation is loaded into Qdec upon opening. On the Display tab you can adjust the annotation opacity. There may be a slight delay while the display updates, be patient!
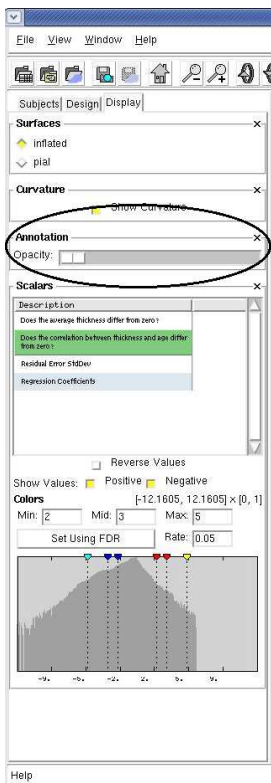
Sliding the button to the right will begin to show the parcellation annotation underneath the overlay. You can bring the opacity to a level that is useful in your interaction with the data. When you have selected a point, which is accomplished by holding down the ctrl key and left-clicking the mouse, the information at the bottom of the window will tell you what region, or parcellation unit, the point is found.

Significance Thresholds

You can also adjust the threshold levels for the overlay on the Display tab.
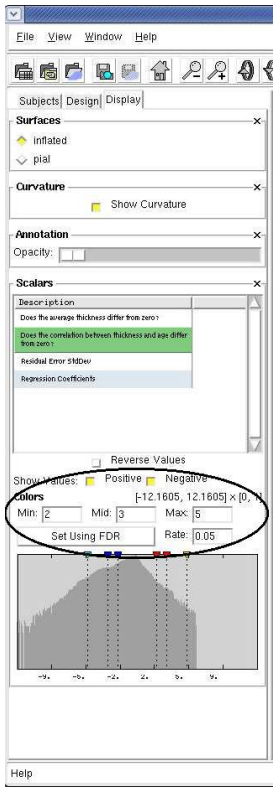
When setting a color scale, you're interested in two things: the threshold (i.e., the value below which the voxel will be transparent - Min), and the saturation point (i.e., the value beyond which the color will not change - Max). In QDEC you can also specify the point where the color will reach the midpoint, with Mid on the control panel. The meaning of these thresholds depends upon the nature of the data you have loaded as the overlay. The map you are currently viewing is -log10(p), where p is the significance, so a Min of 2 will display all vertices with p<.01 and a Max of 5 will show vertices of p<.00001 as the same color. You can lower the threshold to 1.3, 2, 3, to show all vertices with p<.05. You could raise the threshold to 4, 5, 6 to show all vertices with p<.0001.

Variations on Design

With QDEC it is easy to design and run a variety of different analyses. For the first example we looked simply at age and thickness in our subjects. Click back to the Design tab and select gender, to add it to the design. You will want to change the name of the design, call it 'LH-Thickness-Age-Gender-sm10', and click Analyze. When the analysis is done running, click the Display tab and see that there are additional questions in the list summarizing the various analyses that were completed. Among the questions displayed now are **Does the Thickness--age correlation differ between male and female?** and **Does the average thickness differ between male and female?** Click on one of these questions to display the statistically significant regions where the age and thickness correlation are different in men and women, or the average thickness is different in men and women (respectively). Similarly, you can add in one of the other continous variables - hippocampal volume - and run that design.

You can change your design even more, if you click back to the Design tab, you can change your measure from thickness to something else - area, area.pial, volume, sulc, curv, and jacobian_white are your choices. You can also change your level of smoothing - 0, 5, 10, 15, 20, and 25 are your choices. And you can perform any of these on the left (lh) or right (rh) hemispheres. Take a few minutes to select a new design to run, remember to call it something new before you hit Analyze so that the directory of results can be saved.

Define a Region of Interest

FreeSurfer has the ability to compute statistics averaged over a defined region of interest (ROI), which is another popular way to test statistical hypotheses and a good way to check your data. To define a label that marks a region of interest (ROI) on the surface hold down shift then left click and drag to draw your ROI. When drawing your ROI, draw slowly, allowing the display to catch up with you if necessary. There is no need to worry about closing the ROI precisely, when you are done and release the mouse button QDEC will automatically close the ROI for you. You should then see a green outline of the ROI you drew, like this:
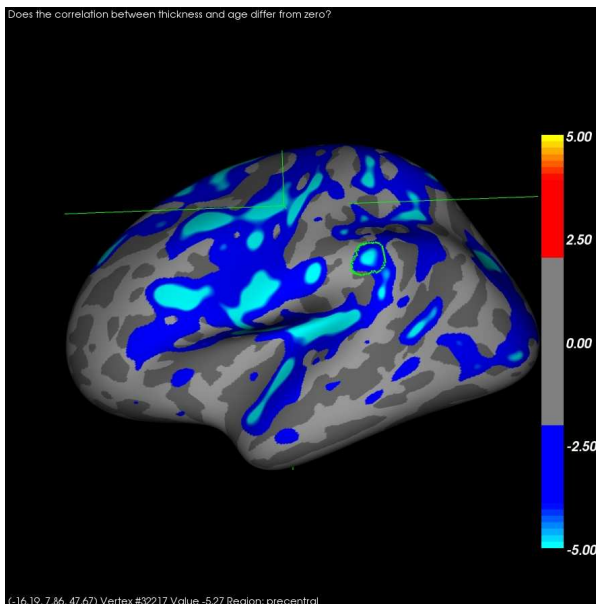
You can then select the *add the selection to the ROI button* , and your label should now be filled in with purple, like this:
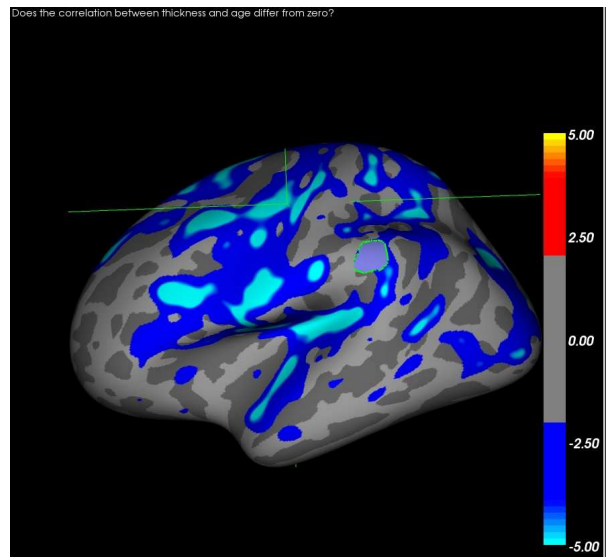
If you do not add your label to the ROI and you start to draw again, QDEC will erase your first label and begin a second. If you have added something to the ROI and want to remove it you can use the *remove selection from ROI button* . When you are finished you can save your label by selecting **File --> Save Label** or clicking the save label button. A dialog box will pop up, and you can choose the location and name to save your label. For this example you can call your label **lh.supramarg.label**, since it is a label of the supramarginal gyrus, and click **Save**.

It may then be useful to map this label to all of the individual subjects in your group study, to either extract statistical values from this region or to visualize the area on each subject to check the integrity of your results. You can do this automatically by selecting **File --> Map Label to Subjects...**, a dialog box

will pop up asking for the label name, you can enter everything before the *.label* of the name, so for this example enter **lh.supramarg**, and click **Ok**. This will use mri_label2label to map this label from your average surface onto all the subjects in your study. When it is complete each subject will have a file, lh.supramarg.label in the label subdirectory. This label can be loaded onto each subject's surface in tksurfer, or opened on the volume in tkmedit.

mris_anatomical_stats You can use mris_anatomical_stats to get a set of statistics on each individual label you've created. The command to run this on the label lh.supramarg.label that you generated for subject 004 is:

```
cd $SUBJECTS_DIR
mris_anatomical_stats -l lh.supramarg.label \
  -t lh.thickness -b -f 004/stats/lh.supramarg.stats 004 lh
```

This will output a stats file to 004/stats/lh.supramarg.stats, which looks like:

```
# Table of FreeSurfer cortical parcellation anatomical statistics
#
# CreationTime 2007/08/16-20:33:33-GMT
# generating_program mris_anatomical_stats
# cvs_version $Id: mris_anatomical_stats.c,v 1.54 2007/08/02 17:37:13 nicks Exp $
# mrisurf.c-cvs_version $Id: mrisurf.c,v 1.557 2007/08/14 01:28:23 fischl Exp $
# cmdline mris_anatomical_stats -l lh.supramarg.label -t lh.thickness -b -f 004/stats/lh.supramarg.stats 004 lh
# sysname  Linux
# hostname minerva
# machine  x86_64
# user     nicks
#
# SUBJECTS_DIR /autofs/space/birn_045/users/BWH/buckner_data/tutorial_subjs/group_analysis_tutorial
# anatomy_type surface
# subjectname 004
# hemi lh
# AnnotationFile lh.supramarg.label
# AnnotationFileTimeStamp 2007/08/16 16:21:56
# Measure Cortex, NumVert, Number of Vertices, 135485, unitless
# Measure Cortex, SurfArea, Surface Area,  96627.3, mm^2
# NTableCols 10
# TableCol  1 ColHeader StructName
# TableCol  1 FieldName Structure Name
# TableCol  1 Units     NA
# TableCol  2 ColHeader NumVert
# TableCol  2 FieldName Number of Vertices
# TableCol  2 Units     unitless
# TableCol  3 ColHeader SurfArea
# TableCol  3 FieldName Surface Area
# TableCol  3 Units     mm^2
# TableCol  4 ColHeader GrayVol
# TableCol  4 FieldName Gray Matter Volume
# TableCol  4 Units     mm^3
# TableCol  5 ColHeader ThickAvg
# TableCol  5 FieldName Average Thickness
# TableCol  5 Units     mm
# TableCol  6 ColHeader ThickStd
# TableCol  6 FieldName Thickness StdDev
# TableCol  6 Units     mm
# TableCol  7 ColHeader MeanCurv
# TableCol  7 FieldName Integrated Rectified Mean Curvature
# TableCol  7 Units     mm^-1
# TableCol  8 ColHeader GausCurv
# TableCol  8 FieldName Integrated Rectified Gaussian Curvature
# TableCol  8 Units     mm^-2
# TableCol  9 ColHeader  FoldInd
# TableCol  9 FieldName  Folding Index
```

```
# TableCol  9 Units     unitless
# TableCol 10 ColHeader CurvInd
# TableCol 10 FieldName Intrinsic Curvature Index
# TableCol 10 Units     unitless
# ColHeaders StructName NumVert SurfArea GrayVol ThickAvg ThickStd MeanCurv GausCurv FoldInd CurvInd
lh.supramarg.label   741   557   1492   2.271   0.646   0.191   0.099   27   3.2
```

This gives the number of vertices, surface area, gray matter volume, average thickness and st. deviation, mean curvature, gaussian curvature, folding index, and curvature index for this region only. You can run this same command on all your subjects to generate these statistics. You could then use aparcstats2table to generate one space delimited table of all these measures for your subjects.

In addition, it might be a good idea to visualize this label on each subject, to be sure there is not a defect in the surface causing your result. You can look at your label in tkmedit on the volume with the command:

```
tkmedit 004 brainmask.mgz lh.white -aux-surface rh.white -label lh.supramarg.label
```

If you scroll towards the posterior end of the brain, you will find your label
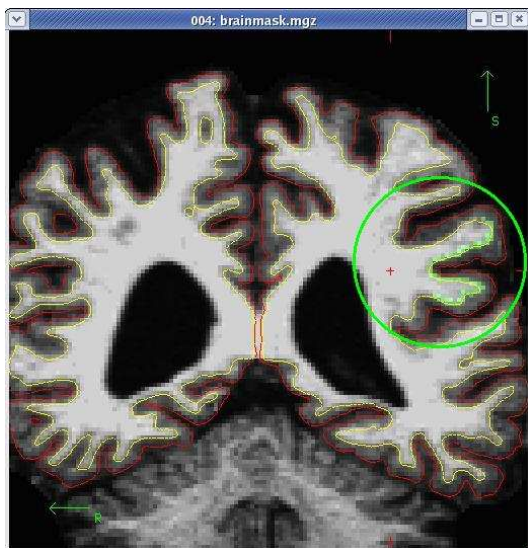
Volumetric Group Analysis

During the normal FreeSurfer processing stream, via the recon-all script, (a freesurfer tutorial is available.) some statistical output files are generated. They are kept in each subjects stats/ subdirectory, and are a result of the subcortical segmentation, aseg, and the cortical parcellation, aparc. These tables include information on each labeled region for the individual subject.

aseg.stats

The statistical output from the subcortical segmentation, called aseg.stats, is a regular text file and will contain the volumes of specific structures.

At the head of the text file there will be information about the command that was run, the version used, the user who ran it and a time stamp. Following this there is information about the volume of the entire brain:

```
# Title Segmentation Statistics
#
# generating_program mri_segstats
# cvs_version $Id: mri_segstats.c,v 1.11.2.5 2006/04/13 18:57:07 nicks Exp $
# cmdline mri_segstats --seg mri/aseg.mgz --sum stats/aseg.stats --pv mri/norm.mgz --ctab-default
--excluded 0 --brain-vol-from-seg --brainmask mri/brainmask.mgz --in mri/norm.mgz --in-intensity-name norm
--in-intensity-units MR --etiv --subject 004
# sysname  Linux
# hostname node0350
# machine  x86_64
# user     PG-user
# anatomy_type volume
#
# SUBJECTS_DIR /buckner_data/group_study
# subjectname 004
# BrainMaskFile  mri/brainmask.mgz
# BrainMaskFileTimeStamp 2006/24/03 13:47:46
# Measure BrainMask, BrainMaskNVox, Number of Brain Mask Voxels, 1744896, unitless
# Measure BrainMask, BrainMaskVol, Brain Mask Volume, 1744896.000000, mm^3
# Measure BrainSeg, BrainSegNVox, Number of Brain Segmentation Voxels, 1255291, unitless
# Measure BrainSeg, BrainSegVol, Brain Segmentation Volume, 1255291.000000, mm^3# Measure IntraCranialVol, ICV, Intracranial Volume, 1679242.759627, mm^3
# SegVolFile mri/aseg.mgz
# SegVolFileTimeStamp 2006/24/03 21:52:14
# ColorTable /space/freesurfer/centos4.0_x86_64/stable/FreeSurferColorLUT.txt
# ColorTableTimeStamp 2006/22/03 05:48:11
# InVolFile  mri/norm.mgz
# InVolFileTimeStamp  2006/24/03 13:55:14
# InVolFrame 0
# PVVolFile  mri/norm.mgz
# PVVolFileTimeStamp 2006/24/03 13:55:16
# ExcludeSegId 0
# VoxelVolume_mm3 1
```

This shows the number of voxels in the brainmask (BrainMaskNVox), the volume of the brainmask (BrainMaskVol), the number of voxels in the brainseg (BrainSegNVox), the volume of the brainseg (BrainSegVol), and the intracranial volume (ICV). This part of the file also tells us that the brainmask.mgz volume is being used as BrainMask (BrainMaskFile mri/brainmask.mgz) and the aseg.mgz segmentation is being used as the SegVol (SegVolFile mri/aseg.mgz). The number of voxels and the volumes should be the same for this subject, since this part of the file also tells us that the voxel volume is 1 mm^3 (VoxelVolume_mm3 1) - and volume is measured in mm^3.

The next section of this file defines the column headers, field name, and units for the rest of the table:

```
# TableCol  1 ColHeader Index
# TableCol  1 FieldName Index
# TableCol  1 Units     NA
# TableCol  2 ColHeader SegId
# TableCol  2 FieldName Segmentation Id
# TableCol  2 Units     NA
# TableCol  3 ColHeader NVoxels
# TableCol  3 FieldName Number of Voxels
# TableCol  3 Units     unitless
# TableCol  4 ColHeader Volume_mm3
# TableCol  4 FieldName Volume
# TableCol  4 Units     mm^3
# TableCol  5 ColHeader StructName
# TableCol  5 FieldName Structure Name
# TableCol  5 Units     NA
# TableCol  6 ColHeader normMean
# TableCol  6 FieldName Intensity normMean
# TableCol  6 Units     MR
# TableCol  7 ColHeader normStdDev
# TableCol  7 FieldName Itensity normStdDev
# TableCol  7 Units     MR
# TableCol  8 ColHeader normMin
```

```
# TableCol  8 FieldName Intensity normMin
# TableCol  8 Units        MR
# TableCol  9 ColHeader normMax
# TableCol  9 FieldName Intensity normMax
# TableCol  9 Units        MR
# TableCol 10 ColHeader normRange
# TableCol 10 FieldName Intensity normRange
# TableCol 10 Units        MR
# NRows 403
# NTableCols 10
```

We can expect to see the *Segmentation Id*, *Number of Voxels*, *Volume*, *Structure Name*, *Intensity normMean*, *Itensity normStdDev*, *Intensity normMin*, *Intensity normMax*, and *Intensity normRange* for each entry in the table.

The remainder of the table shows this information for all the structures that are labeled in the aseg:

```
# ColHeaders  Index SegId NVoxels Volume_mm3 StructName normMean normStdDev normMin normMax normRange
  2    2   237201   237201.0  Left-Cerebral-White-Matter   107.5612   11.3261   31.0000   188.0000   157.0000
  3    3   249096   249096.0  Left-Cerebral-Cortex          69.8956   11.0623    0.0000   139.0000   139.0000
  4    4    31329    31329.0  Left-Lateral-Ventricle        23.0385   11.1648    7.0000    91.0000    84.0000
  5    5     1735     1735.0  Left-Inf-Lat-Vent             42.2160   15.5492   14.0000    94.0000    80.0000
  7    7    13767    13767.0  Left-Cerebellum-White-Matter  87.6124    7.8224   43.0000   116.0000    73.0000
  8    8    48245    48245.0  Left-Cerebellum-Cortex        60.1777    9.4993   25.0000    94.0000    69.0000
 10   10     7025     7025.0  Left-Thalamus-Proper          89.5336   11.9082   19.0000   126.0000   107.0000
 11   11     5252     5252.0  Left-Caudate                  77.3650   11.3959   45.0000   105.0000    60.0000
 12   12     7993     7993.0  Left-Putamen                  81.3400    9.7069   28.0000   115.0000    87.0000
 13   13     2144     2144.0  Left-Pallidum                 97.6942   11.7513   36.0000   121.0000    85.0000
 .
 .
 .
```

You can use the data in this table to perform group stats on the volumes of certain structures that may be of interest to your study. There is a way to combine this data, for your entire group, into one table that will be easily read into a spreadsheet program, by using asegstats2table. You can do this for the tutorial set of subjects with this sample command line:

```
asegstats2table --subjects 004 008 017 021 032 039 040 045 049 067 \
073 074 080 084 091 092 093 095 097 099 102 103 106 108 111 114 123 \
124 128 129 130 131 133 136 138 140 141 144 145 149 \
--meas vol --t asegstats.txt
```

This will combine the volumes from all of your subjects asegs.stats files into one table, asegstats.tct. This table can now be imported into any spreadsheet program for statistical analysis.

aparc.stats

The statistical output from the cortical parcellation, called lh.aparc.stats and rh.aparc.stats, is a regular text file and will contain the thickness of specific structures.

At the head of the text file there will be information about the command that was run, the version used, the user who ran it and a time stamp. Following this there is information about the volume of the entire brain:

```
# Table of FreeSurfer cortical parcellation anatomical statistics
#
# CreationTime 2006/04/25-09:31:20-GMT
# generating_program mris_anatomical_stats
# cvs_version $Id: mris_anatomical_stats.c,v 1.35.2.1 2006/04/21 19:45:19 nicks Exp $
# mrisurf.c-cvs_version $Id: mrisurf.c,v 1.441.2.3 2006/04/12 02:03:02 nicks Exp $
# cmdline mris_anatomical_stats -mgz -f ../stats/lh.aparc.stats -b -a ../label/l h.aparc.annot -c ../stats/aparc.annot.ctab 004 lh
# sysname  Linux
# hostname node0350
# machine  x86_64
# user     FS-user
#
# SUBJECTS_DIR /buckner_data/group_study
# anatomy_type surface
# subjectname 004
# hemi lh
# AnnotationFile ../label/lh.aparc.annot
# AnnotationFileTimeStamp 2006/25/03 05:31:10
# TotalWhiteMatterVolume 634178 mm^3
# Measure Cortex, NumVert, Number of Vertices, 150889, unitless
# Measure Cortex, SurfArea, Surface Area, 102409, mm^2
```

This shows the total white matter volume (TotalWhiteMatterVolume), the number of vertices in the cortex (NumVert), and the surface area of the cortex (SurfArea). This part of the file tells us that the lh.aparc.annot is being used as the annotation file (AnnotationFile ../label/lh.aparc.annot).

The next section of this file defines the column headers, field name, and units for the rest of the table:

```
# NTableCols 10
# TableCol  1 ColHeader StructName
# TableCol  1 FieldName Structure Name
# TableCol  1 Units        NA
# TableCol  2 ColHeader NumVert
# TableCol  2 FieldName Number of Vertices
# TableCol  2 Units        unitless
# TableCol  3 ColHeader SurfArea
# TableCol  3 FieldName Surface Area
# TableCol  3 Units        mm^2
# TableCol  4 ColHeader GrayVol
# TableCol  4 FieldName Gray Matter Volume
# TableCol  4 Units        mm^3
# TableCol  5 ColHeader ThickAvg
# TableCol  5 FieldName Average Thickness
# TableCol  5 Units        mm
# TableCol  6 ColHeader ThickStd
# TableCol  6 FieldName Thickness StdDev
# TableCol  6 Units        mm
# TableCol  7 ColHeader MeanCurv
# TableCol  7 FieldName Integrated Rectified Mean Curvature
# TableCol  7 Units        mm^-1
# TableCol  8 ColHeader GausCurv
# TableCol  8 FieldName Integrated Rectified Gaussian Curvature
# TableCol  8 Units        mm^-2
# TableCol  9 ColHeader  FoldInd
# TableCol  9 FieldName  Folding Index
# TableCol  9 Units        unitless
# TableCol 10 ColHeader CurvInd
# TableCol 10 FieldName Intrinsic Curvature Index
# TableCol 10 Units        unitless
```

We can expect to see the *Structure Name*, *Number of Vertices*, *Surface Area*, *Gray Matter Volume*, *Average Thickness*, *Thickness StDev*, *Integrated Rectified Mean Curvature*, *Integrated Rectified Gaussian Curvature*, *Folding Index* and *Intrinsic Curvature Index* for each entry in the table.

The remainder of the table shows this information for all the structures that are labeled in the aseg:

```
# ColHeaders StructName NumVert SurfArea GrayVol ThickAvg ThickStd MeanCurv GausCurv FoldInd CurvInd
unknown                15085  10384  21630  2.006  1.096   0.123   0.038  168.313  26.766
bankssts                1126    770   1563  2.132  0.462   0.103   0.024    7.056   1.004
caudalanteriorcingulate  931    636   2125  2.721  0.675   0.127   0.031   15.801   0.991
caudalmiddlefrontal     3577   2403   6575  2.447  0.535   0.120   0.028   34.901   3.674
corpuscallosum          1035    680   1215  2.123  0.902   0.136   0.023   17.731   0.799
cuneus                  2966   1958   3769  1.740  0.473   0.140   0.033   37.320   3.630
entorhinal               683    419   1620  2.819  0.632   0.090   0.021    5.349   0.429
fusiform                6622   4607  11486  2.190  0.634   0.130   0.032   77.276   7.884
 .
 .
 .
```

You can use the data in this table to perform group stats on the thickness, area, volume, etc. of certain structures that may be of interest to your study. There is a way to combine this data, for your entire group, into one table that will be easily read into a spreadsheet program, by using aparcstats2table. You can do this for the tutorial set of subjects with this sample command line:

```
aparcstats2table --subjects 004 008 017 021 032 039 040 045 049 067 \
073 074 080 084 091 092 093 095 097 099 102 103 106 108 111 114 123 \
124 128 129 130 131 133 136 138 140 141 144 145 149 \
--hemi lh --meas area --t aparcstatsAREA.txt
```

This will combine the areas from all of your subjects lh.aparc.stats files into one table, aparcstatsAREA.txt. This table can now be imported into any spreadsheet program for statistical analysis. You can change your command to run on the other hemisphere, or on a different measure (volume, thickness).

2008-06-01 20:47

TUTORIAL LOCATION HAS MOVED

Good news for those of you who are confused by FSGD files, and mri_glmfit commands: There is a new group analysis tool in Freesurfer, Qdec. To accompany this tool there is also a new group analysis tutorial.

Please see the new group analysis tutorial, using Qdec. Click here

FreeSurfer Tutorial: Group Analysis

- To follow this exercise exactly be sure you've downloaded the tutorial data set before you begin. If you choose not to download the data set you can follow these instructions on your own data, but you will have to substitute your own specific paths and subject names.

In this tutorial, you will learn how to perform statistical analysis of group surface-based data, including:

- Making an average subject from your set of subjects
-
  Constructing a FreeSurfer Group Descriptor File (FSGD)
-
  Preprocessing the group data
-
  Constructing the design matrix
-
  Constructing contrast matrices to test hypotheses
-
  Correcting for multiple comparisons

Assuming that all surface reconstruction has been completed for all subjects in the study, FreeSurfer's mri_glmfit command can be used to perform inter-subject/group averaging and inference on the cortical surface. Mri_glmfit models the data as a linear combination of effects related to variables of interest, confounds and errors, and permits statistical inferences to be made about effects of interest in relation to error variance. It also allows for certain permutation testing and other means for correcting for mutliple comparisons. For group analysis, this technique fits a general linear model (GLM) at each surface vertex to explain the data from all subjects in the study. In this section, a brief overview of linear modeling is presented and mri_glmfit is described for estimating a linear model and testing hypotheses. The modeling overview can be skipped if this material is already familiar. Other software packages have similar types of programs (e.g., FSL's GFEAT).

```
# AnnotationFile ../label/lh.aparc.annot
```

1.0 Preparing for Group Analysis

For group analysis, you can create an average subject from all the participants in the study. This average will be used as the target subject upon which the results of your group analysis can be output and viewed. To create this average, use *make_average_subject*. One has already been created for the later exercise, so there is no need to execute this sample command:

```
make_average_subject --subjects <subj1> <subj2> ...
```

The default behavior of this script is to create a subject in the $SUBJECTS_DIR named 'average' using each subjects talairach.xfm transform. This behavior can be modified on the command line. You can specify --out your_named_average to change the name of the average subject and --xform talairach.lta (or talairach.m3z) to specify the use of one of the other transforms.

The average subject is created using the processed volumes and surfaces from the set of subjects you specify following the --subjects flag. The make_average_subject command executes both the make_average_volume and make_average_surface subscripts for you.

The distributed example of an average subject can be found in
$FREESURFER_HOME/subjects/buckner_data/tutorial_subjs/group_analysis_tutorial called fsaverage

This average subject was created using the volumes and surfaces of subjects in:

$FREESURFER_HOME/subjects/buckner_data/group_study

(Note: the data in the group_study directory is not required to complete the tutorial. However, it is available for download if you wish to pursue your own group analysis.)

2.0 Linear Modeling overview

Linear modeling describes the observed data as a linear combination of explanatory factors plus noise, and determines how well that description explains the data being analyzed. In order to understand how to perform group analysis in FreeSurfer, you need to understand the general linear model (GLM) and how to construct a GLM in matrix notation. You can click here for a review of this material. The notation we use here is:**y=X*beta**, where **y** is the vector observed data (e.g., thicknesses for each subject at a vertex), **X** is the known design matrix (e.g., gender, age), and **beta** is the vector of unknown parameter estimates (PEs). The interpretation of the PEs will depend upon how **X** is constructed. For example, they could be interpeted as a slope indicating the change of thickness with age. The analysis/estimation is then the process of computing **beta** given the data **y** and the design matrix **X**. A Null Hypothesis (H0) is constructed with a contract matrix **C**. Inferences are drawn by testing whether the value **gamma=Cb** is zero.

3.0 Using mri_glmfit for estimating the linear model and hypothesis testing

As stated earlier, mri_glmfit performs inter-subject/group averaging and inference on the surface by fitting a linear model at each vertex. The model consists of subject parameters (**e.g.,** age, gender, etc). The model is the same across all vertices, though the fit will probably be different at each vertex. To specify the model, a design matrix that represents the GLM must be created.

Create an FSGD file

The FreeSurfer Group Descriptor File (FSGDF) provides a way to describe a group of subjects and their accompanying data. This can include class membership and other continuous variables, for example gender or age. When it exists, the FSGDF is used by mri_glmfit, tksurfer and tkmedit. The FSGDF is more than just a way to specify the design matrix. It also keeps track of group membership and covariate definitions. This information is then used to help visualize the results. This is not possible when only a design matrix is used.

The following study variables correspond to all the subjects found in $FREESURFER_HOME/subjects/buckner_data/tutorial_subjs/group_analysis_tutorial/ and are presented in a spreadsheet. You'll need to compose an FSGD file with the appropriate variables in order to specify a design matrix that can be used to examine the relationship between a subject's age and cortical thickness. You can read more about the rules for creating an FSGD file here. To get started, open the text editor of your choice and add individual tags by following the general example and rules described here. In general, it's a good idea to name the file something intuitive, such as my_gender_age_fsgd.txt. To create your FSGD file you first need to change to the directory you'd like to create the file in:

```
cd $FREESURFER_HOME/subjects/buckner_data/tutorial_subjs/group_analysis_tutorial/stats
```

	A	B	C
1	Subject	Age	Gender
2	4	72	M
3	8	88	F
4	17	25	M
5	21	22	M
6	32	79	M
7	39	69	F
8	40	24	F
9	45	78	F
10	49	19	M
11	67	71	M
12	73	77	M
13	74	85	M
14	80	23	M
15	84	21	M
16	91	23	F
17	92	29	M
18	93	22	F
19	95	70	F
20	97	74	M
21	99	93	F
22	102	82	F
23	103	66	F
24	106	27	M
25	108	28	M
26	111	81	F
27	114	87	F
28	123	69	F
29	124	30	F
30	128	90	M
31	129	65	F
32	130	68	M
33	131	80	F
34	133	73	F
35	136	78	F
36	138	26	F
37	139	75	M
38	140	18	F
39	141	20	F
40	144	78	M
41	145	80	M
42	149	77	M

Upon completion, save the FSGD file, which can be viewed in the xterm by typing:

```
cat my_gender_age_fsgd.txt
```

in the directory where it has been saved.

A correct FSGD file is presented here for comparison. It is needed for a later exercise, so create it now if you have not already done so.

Creating a Design Matrix

The FSGDF is specified in the command-line for mri_glmfit with the option
--fsgd fname <gd2mtx> where *gd2mtx* is the method by which the group description is converted into a design matrix. Legal values for *gd2mtx* are:

- *doss* (different offset, same slope): this will create a design matrix in which each class has its own offset but forces all classes to have the same slope.

- *dods* (different offset, different slope): this value models each class with its own offset and slope (default).

- *none*: this value is used if neither of the previous models work for your particular analysis. Using this value requires that you specify the design matrix manually.

If you do not specify one of the above methods, dods will be used by default.

Note: It is not necessary to run mri_glmfit now to create the design matrix, as mri_glmfit will create it for you later in this exercise.

Please click here for further explanation of the design matrix and how it is used with the FSGD file.

Preprocessing steps

**These are sample commands that you should run for the group analysis. In interest of time these steps have already been run for you, and the output is found in
$FREESURFER_HOME/subjects/buckner_data/tutorial_subjs/group_analysis_tutorial/stats**

Once an FSGD file is set up and the average subject is made the preprocessing steps can be followed. The first step will use mris_preproc to assemble your data into a single file in the common surface space, *average* for this example (which is the average that has been made for this particular study). In this step you will have to specify your FSGD file, *gender_age.txt* here, your target subject, *average* here, the hemisphere and measure you are using. You will also name the output file - it's a good idea to use a naming convention that will make it obvious what comparison you are working with. Before running the command you want to be sure your SUBJECTS_DIR is set appropriately and you are in the glm directory you wish to run the analyses in:

```
setenv SUBJECTS_DIR $FREESURFER_HOME/subjects/buckner_data/tutorial_subjs/group_analysis_tutorial
cd $FREESURFER_HOME/subjects/buckner_data/tutorial_subjs/group_analysis_tutorial/stats
```

now you can run the mris_preproc command:

```
mris_preproc --fsgd gender_age.txt \
  --target fsaverage \
  --hemi lh \
  --meas thickness \
  --out lh.gender_age.thickness.mgh
```

The next step is to do surface smoothing. Smooth input with a Gaussian kernel with the given full-width/half-maximum (fwhm) specified in mm. For all examples to follow we will use a fwhm = 10mm. To do this mri_surf2surf will be used along with the output from mris_preproc and your average subject.

```
mri_surf2surf --hemi lh \
  --s fsaverage \
  --sval lh.gender_age.thickness.mgh \
  --fwhm 10 \
  --tval lh.gender_age.thickness.10.mgh
```

You can do the surface smoothing as part of the first step with mris_preproc, but if you do it afterwards as a separate step you can smooth to many different levels without having to rebuild the data each time.

mri_glmfit

As stated earlier, these are sample commands that you should run for the group analysis. In interest of time these steps have already been run for you, and the output is found in $FREESURFER_HOME/subjects/buckner_data/tutorial_subjs/group_analysis_tutorial/stats

mri_glmfit performs the general linear model (GLM) analysis on the volume or the surface. Options include simulation for correction for multiple comparisons, weighted LMS, variance smoothing, PCA/SVD analysis of residuals, per-voxel design matrices, and 'self' regressors. This program performs both the estimation and inference. The framework for testing specific hypotheses is specified in the form of a contrast vector. For each comparison you want to run you will need to design a separate contrast vector. For information on how to set up a contrast vector please click here.

To use this in your mri_glmfit command create a file that contains just the one line of your contrast vector using the text editor of your choice. Again, using an appropriate naming convention is a good idea. Make sure this contrast vector has been created for you and is called **age.mat**:

```
cd $FREESURFER_HOME/subjects/buckner_data/tutorial_subjs/group_analysis_tutorial/glm
cat age.mat
```

Create age.mat if it does not exist. It should contain this line:

```
0 0 1
```

As an additional exercise, try constructing the contrast vector for testing the same data for difference between males and females independent of age. Click here for the answer.

mri_glmfit will take the output from your smoothing step above, your fsgd file, your average subject and the contrast vector as inputs. You will also have to specify a glm directory name, and in this directory all of the outputs will be saved. It is a good idea to use a descriptive name, so you can easily recognize which outputs are in which directory.

```
mri_glmfit --y lh.gender_age.thickness.10.mgh \
  --fsgd gender_age.txt doss\
  --glmdir lh.gender_age.glmdir \
  --surf average lh \
  --C age.mat
```

The flag *--surf* is used to specify that the input has a surface geometry from the hemisphere of the given FreeSurfer subject. If --surf is not specified, then mri_glmfit will assume that the data are volume-based and use the geometry as specified in the header to make spatial calculations.

When this command is finished you will have an lh.gender_age.glmdir. There will be a number of output files in this directory, as well as two other directories. If you did an *ls* in your glmdir there will be:

```
ar1.mgh   eres.mgh   mri_glmfit.log  rstd.mgh  age/  y.fsgd
beta.mgh  fsgd.X.mat  rvar.mgh  Xg.dat
```

the outputs are as follows:

-
    ar1.mgh - spatial AR1 coefficients
    beta.mgh - all regression coefficients
    eres.mgh - residual error
    fsgd.X.mat - final design matrix in matlab format
    mri_glmfit.log - execution parameters
    rstd.mgh - residual error stddev (just sqrt of rvar)
    rvar.mgh - residual error variance
    Xg.dat - design matrix in text format
    y.fsgd - fsgd file

There will be a subdirectory for each contrast that you specify. The name of the directory will be that of the contrast matrix file (without the .mat extension). The **age** directory will have the following files:

```
C.dat  F.mgh  gamma.mgh  sig.mgh
```

The outputs are as follows:

-
    C.dat - copy of contrast matrix
    F.mgh - F-ratio
    gamma.mgh - contrast
    sig.mgh - significance from F-test (actually -log10(p))

4.0 Using mri_glmfit to correct for multiple comparisons

One method for correcting for multiple comparisons is to perform simulations under the null hypothesis and see how often the value of a statistic from the 'true' analysis is exceeded. This frequency is then interpreted as a p-value which has been corrected for multiple comparisons. This is especially useful with surface-based data as traditional random field theory is harder to implement. This simulator is roughly based on FSLs permuation simulator (randomise) and AFNIs null-z simulator (AlphaSim). Note that FreeSurfer also offers False Discovery Rate (FDR) correction in tkmedit and tksurfer.

The estimation, simulation, and correction are done in three distinct phases:

1. Estimation: run the analysis on your data without simulation. Note: at this point you can view your results with FDR thresholding in tksurfer. FDR is often conservative relative to cluster-based thresholding.
2. Simulation: run the simulator with the same parameters as the estimation to get the Cluster Simulation Data (CSD).
3. Clustering: run mri_surfcluster, passing it the CSD from the simulator and the output of the estimation. These programs will print out clusters along with their p-values.

The estimation step has been described above, using mri_glmfit with no simulation. The simulation step is run using mri_glmfit again, adding in a simulation flag and parameters. If a design is non-orthogonal the permutation simulation can not be run, instead a simple monte carlo simulation can be run. The clustering step is run with mri_surfcluster.

4.1 Simulations

The simulator synthesizes data under the null hypothesis, analyzes, thresholds, clusters, and then keeps track of the number of times clusters of a given size were encounted. The simulation is invoked by calling mri_glmfit and specifying a simulation type and it's associated parameters, with the flag **--sim** which is to be followed by 4 parameters:

```
--sim nulltype nsim thresh csdbasename
```

The first parameter the *nulltype*, which is the method of generating the null data to be tested. Valid options are:

-
    (1) perm - perumation, randomly permute rows of X (cf FSL randomise)
    (2) mc-full - full Monte Carlo simulation, replace input with white gaussian noise, smooth, and analyze.
    (3) mc-z - z Monte Carlo simulation. Does not actually do analysis, just assume the output is z-distributed (cf ANFI AlphaSim)

Which one should you use? Permutation makes the fewest assumptions and is fastest, but it requires that the design matrix be orthogonal (e.g., you cannot have continuous variables such as age or IQ). If you cannot use permutation, then a mc-z will be fastest, but the z assumption requires that you have many subjects (on the order of 80). Otherwise, you must run the full MC simulation, which can take quite a

while. For both MC simulations, you must supply the smoothness of your data as Full-Width/Half-Max (FWHM) of the residual. This can be obtained from the ResidualFWHM value in y.fsgd in the glm output directory.

The next parameter is *nsim* which corresponds to the number of simulations to run. If you want to make inferences at the .01 level then you'll need about 10000 iterations. You can run multiple simulations in parallel, if you have multiple processors, to cut down on processing time.

The next parameter is *thresh* which corresponds to your threshold and is specified as a -log10(pvalue). Eg, for a p-value threshold of .01, use thresh=2.

The last parameter is *csdbasename* which corresponds to the base name of the file which will store the Cluster Simulation Data (CSD). Each contrast will get it's own file. When running multiple simulations in parallel be sure to use a unique *csdbasename* for each run.

Here's a sample command to run the mc-full simulation with 10000 iterations and a p-value threshold of .01:

```
mri_glmfit --y lh.gender_age.thickness.10.mgh \
  --glmdir lh.gender_age.glmdir \
  --fsgd gender_age.txt doss \
  --surf fsaverage lh \
  --fwhm 14.517 --C age.mat \
  --sim mc-full 10000 2 lh.gender_age.glmdir/csd1
```

this will create lh.gender_age.glmdir/csd1-age.csd

If you want to split this into multiple runs you could use the following two commands:

```
mri_glmfit --y lh.gender_age.thickness.10.mgh \
  --fsgd gender_age.txt doss\
  --fwhm 14.517 \
  --glmdir lh.gender_age.glmdir \
  --surf fsaverage lh \
  --C age.mat \
  --sim mc-full 5000 2 lh.gender_age.glmdir/csd1
mri_glmfit --y lh.gender_age.thickness.10.mgh \
  --fsgd gender_age.txt doss\
  --fwhm 14.517 \
  --glmdir lh.gender_age.glmdir \
  --surf average lh \
  --C age.mat \
  --sim mc-full 5000 2 lh.gender_age.glmdir/csd2
```

which will generate csd1-age.csd and csd2-age.csd

4.2 Clustering

Using the outputs from the estimation step and the simulations, mri_surfcluster (or mri_volcluster) will create two outputs: the summary file with a table of the clusters it found, and an output surface map of the clusters wth the cluster-wise p-value. The sample mri_surfcluster command is:

```
mri_surfcluster --in lh.gender_age.glmdir/age/sig.mgh \
  --csd lh.gender_age.glmdir/csd1-age.csd \
  --csd lh.gender_age.glmdir/csd2-age.csd \
  --sum lh.gender_age.glmdir/age/sig.cluster.sum \
  --cwsig lh.gender_age.glmdir/age/sig.cluster.mgh
```

you can pass all the CSD files that were created through this command by adding as many --csd as you need.

The surfcluster summary file, sig.cluster.sum, will look like this:

```
# ClusterNo  Max    VtxMax  Size(mm^2)  TalX   TalY   TalZ    CWP     CWPLow   CWPHi   NVtxs
     1      1.850   134208   391.63    -28.8   34.0   -8.0  0.00100  0.00060  0.00140   542
     2      1.439    18669     0.56    -10.9    9.1  -13.2  0.99960  0.99930  0.99980     1
     3      1.366   148132    13.84     -7.4   11.8  -14.1  0.97440  0.97240  0.97640    25
```

CWP stands for *cluster-wise probability* which is the probability after correction for multiple comparisons. The CWP column is the nomial p-value. CWPLow and CWPHi are the 90% confidence intervals on the p-value. Each cluster gets its own p-value, which depends upon its size. NVtxs is the number of vertexes in the cluster. The output surface map, sig.cluster.mgh, will be a map of these clusters with their CWP. This can be viewed with:

```
tksurfer fsaverage lh inflated \
  -overlay lh.gender_age.glmdir/age/sig.cluster.mgh
```

2008-06-01 20:47

Linear modeling describes the observed data as a linear combination of explanatory factors plus noise, and determines how well that description explains the data being analyzed.

For group morphometric analysis, the observed data is comprised of a set of surface measures (such as cortical thickness) at each vertex in a surface model, for each subject in the group. This data can be organized as a set of vectors, each associated with a different vertex in the surface model, and containing a surface measurement for every subject in the group at the corresponding vertex.

First, a linear model must be designed to include all explanatory variables (EVs) that may account for each vector's values. A simple linear model is given by $y=a*x+b+e$, where the observed data y is a one-dimensional vector of surface measures -- one measurement per subject at a vertex; x is a one-dimensional vector containing a variable, such as age, describing each subject; a is the parameter estimate (PE) for x, for instance the value that a subject's age must be multiplied by to fit the data in y; b is a constant, and in this example, would correspond to the baseline measurement present in the data; and e is the error in the model fitting. If an additional explanatory variable is added to explain the observed data, the model would be given as $y=a1*x1+a2*x2+b+e$, containing two different model waveforms, $a1*x1$ and $a2*x2$, corresponding to two variables, such as age and gender, describing all subjects in the study.

2.1 Estimation overview
Once the model is specified, an estimation step follows, in which the model is fit to each vector's vector separately; no interactions between vertices are taken into account in the examples presented here. This step generates the estimate of the "goodness of fit" of each of the explanatory variables to each vector of surface measurements. Thus if a particular vertex responds strongly to the explanatory variable $x1$, a large value for $a1$ will be produced by model-fitting; if the data appears unrelated to $x2$ then $a2$ will have a very small value.

This kind of linear modeling is commonly expressed in matrix notation, where the the matrix X contains all the explanatory variables (designed effects and confounds) in the model, and the matrix A contains all the PEs. The matrix X is also commonly called the *design matrix* and it can be user-specified in FreeSurfer in the form of an FSGD (FreeSurfer Group Descriptor) file, as the exercises below illustrate. Each column of X corresponds to a different explanatory variable (also called a regressor or a covariate). As typically formulated and solved, the estimation step produces a set of estimates of the PEs, which in turn are used in hypothesis testing.

2.2 Inference overview
Estimates of the PEs can be converted into statistical parametric maps, which are commonly visualized as a color-coded surface overlay. The overlay assigns each vertex a value based on the likelihood that the null hypothesis is false at that vertex. A linear combination of estimates of PEs is used to encode the particular hypothesis of interest. This encoding is accomplished with a user-specified ''contrast vector'', which assigns a contrast weight to each column of the design matrix. A simple example of a contrast vector that tests the null hypothesis for the explanatory variable associated with the first design matrix column would be[1 0 0 0...]. To compute this particular contrast at each vertex, the PE value associated with the first design matrix column at that vertex is divided by the error in its estimate, yielding a t-value. The t-value provides a good measure of confidence in the estimate of the PE value, and can be converted into a probability (P) or Z statistic at that vertex via a standard statistical transformation. T, P and Z values all convey the same information about how significantly the observed data is related to a given explanatory variable.

A t-value map can be produced for each explanatory variable of interest. Each map indicates how strongly vertices on the surface are related to one explanatory variable. Parameter estimates can also be compared to see if one explanatory variable is more strongly related to the data than another. To encode this kind of hypothesis, one PE is subtracted from another using a "contrast" vector such as [1 -1 0 0 ...], a combined standard error is computed, and a new t-map is generated. In a similar fashion, to test for a more complicated collection of effects, a matrix of contrast weights can be specified. A more rigorous description of single and multiple linear regression and GLM, types of analyses, estimation and hypothesis testing is available at http://www.statsoft.com/textbook/stglm.html.

2008-06-01 20:47

Create an FSGD file

The FSGDF format uses tags to identify information, as shown below (this is just for example, and not something that can be made from the tutorial data):

```
Example of a legal file:
------------------------ cut here -----------------
GroupDescriptorFile 1
Title MyTitle
Class Class1 plus blue
Class Class2 circle green
Variables          Age   Weight   IQ
Input subjid1 Class1  10    100   1000
Input subjid2 Class2  20    200   2000
#Input subjid3 Class2  20    200   2000
DefaultVariable Age
------------------------ cut here -----------------
```

Notes:

-
  The first line of the file must be "GroupDescriptorFile 1".

-
  Title is not necessary. This will be used for display.

-
  Class only needs the class name, the next two items, if present, will be used in the display.

-
  The third *Input* tag will be treated as a comment, due to the # at the beginning of the line.

-
  DefaultVariable is the default variable for display.

General rules:

- Tags are NOT case sensitive (for instance, 'Class' and 'CLASS' are the same tag).
- Labels are case sensitive (for instance CLASS1 and Class1 are NOT the same label).
- When multiple items appear on a line, they can be separated by any white space (i.e., blank or tab).
- Any line where # appears as the first non-white space character is treated as a comment (ignored).

- The *Variables* line should appear before the first Input line.
- All Class lines should appear before the first Input line.
- Variable label replications are not allowed.
- Class label replications are not allowed.

- Subject Id replications are not allowed.
- If a class label is not used, a warning is printed out.
-

  DefaultVariable must be a member of the Variable list.
- No error is generated if a tag does not match.
- Empty lines are OK.
- A class label can optionally be followed by a class marker.
-

  A class marker can optionally be followed by a class color.

A correct FSGD file from the given design

```
GroupDescriptorFile 1
Title Thickness vs Age and gender
Class Male
Class Female
Variables Age
Input 004 Male 72
Input 008 Female 88
Input 017 Male 25
Input 021 Male 22
Input 032 Male 79
Input 039 Female 69
Input 040 Female 24
Input 045 Female 78
Input 049 Male 19
Input 067 Male 71
Input 073 Male 77
Input 074 Male 85
Input 080 Male 23
Input 084 Male 21
Input 091 Female 23
Input 092 Male 29
Input 093 Female 22
Input 095 Female 70
Input 097 Male 74
Input 099 Female 93
Input 102 Female 82
Input 103 Female 66
Input 106 Male 27
Input 108 Male 28
Input 111 Female 81
Input 114 Female 87
Input 123 Female 69
Input 124 Female 30
Input 128 Male 90
Input 129 Female 65
Input 130 Male 68
Input 131 Female 80
Input 133 Female 73
Input 136 Female 78
Input 138 Female 26
Input 140 Female 18
Input 141 Female 20
Input 144 Male 78
Input 145 Male 80
Input 149 Male 77
```

Design Matrices with DOSS and DODS

The design matrix is created from the class and variable information in the FSGD file and from the type of design specified when running mri_glmfit (*i.e.*, DODS: different offset different slope; or DOSS: different offset same slope). The design matrix will consist of regressors for intercepts and regressors for slopes. Each class will have an intercept regressor. The intercept regressor is a vector with a 1 for each subject that is a member of the class and 0 otherwise. The slope regressors are handled differently depending upon whether DODS or DOSS is used. For DODS, each class will have a slope regressor for each variable. Like the intercept regressor, the slope regressor for a class will be 0 for subjects not in the class. For subjects in the class, the slope regressor will be the value of the variable. Each variable will have its own set of regressors. For DOSS, each variable will have a single regressor which will be independent of class. This regressor will just be a vector of the variable values listed in the FSGD file. For DODS, the total number of regressors will be Nc*(Nv+1), where Nc is the number of classes and Nv is the number of variables. The first Nc regressors will be the intercepts for each class. The next Nc regressors will be the slopes for each class for the first variable, etc. For DOSS, the total number of regressors will be Nc+Nv. The first Nc regressors will be the intercepts for each class. The next Nv regressors will be the slopes for each variable.

An example FSGDF and design matrices for DODS and DOSS
This similar FSGD file has two classes (Nc=2) and three variables (Nv=3):

```
------------------------ cut here ------------------
 GroupDescriptorFile 1
 Class Class1
 CLASS Class2
 Variables        Age   Weight   IQ
 Input subjid1a Class1   10    100    1000
 Input subjid1b Class1   15    150    1500
 Input subjid2a Class2   20    200    2000
 Input subjid2b Class2   25    250    2500
 DefaultVariable Age
------------------------ cut here ------------------
```

For DODS, the number of regressors is Nc*(Nv+1)=8, and the design matrix would be:

```
1  0  10   0   100   0   1000    0
1  0  15   0   150   0   1500    0
0  1  0   20    0   200    0    2000
0  1  0   25    0   250    0    2500
```

For DOSS, the number of regressors is Nc+Nv=5, and the design matrix would be:

```
1  0  10  100  1000
1  0  15  150  1500
0  1  20  200  2000
0  1  25  250  2500
```

The design matrix is created for you by mri_glmfit, using your FSGD file and your choice of either DOSS or DODS. You will not have to create this yourself.

```
GroupDescriptorFile 1
Title Thickness vs Age and gender
Class Male
Variables Age
```

Run mris_glm to perform estimation

In order to perform estimation you will first need to change to the tutorial data directory and setup SUBJECTS_DIR:

```
cd $FREESURFER_HOME/subjects/buckner_data/tutorial_subjs/group_analysis_tutorial
setenv SUBJECTS_DIR ${PWD}
```

Next, create a 'stats' directory (or some other appropriate name) and copy your FSGD file into it:

```
mkdir stats
cp my_gender_age_fsgd.txt stats/
cd stats
```

Assuming the FSGD file is properly created, mris_glm needs to be configured on the command line in order to perform estimation. Examples of the command lines for estimating the left hemisphere and then the right hemisphere are given below and a brief explanation follows:

```
# For the left hemisphere
mris_glm --surfmeas thickness \
        --hemi lh \
        --trgsubj average \
        --fsgd ./my_gender_age_fsgd.txt doss \
        --beta ./beta_doss-thickness-100lh.mgz \
        --var ./var_doss-thickness-100lh.mgz \
        --y ./y_doss-thickness-100lh_000.mgz \
        --nsmooth 100

# For the right hemisphere
mris_glm --surfmeas thickness \
        --hemi rh \
        --trgsubj average \
        --fsgd ./my_gender_age_fsgd.txt doss \
        --beta ./beta_doss-thickness-100rh.mgz \
        --var ./var_doss-thickness-100rh.mgz \
        --y ./y_doss-thickness-100rh_000.mgz \
        --nsmooth 100
```

In this example, mris_glm will read the thickness maps for each of the subjects, smooth them with 100 iterations of nearest-neighbor smoothing, resample the maps to the common surface space defined by the average subject, convert the FSGD file my_gender_age_fsgd.txt into a design matrix by using doss, and save the regression coefficients (--beta,), noise variance (--var) and the preprocessed data (i.e., the input data after resampling and smoothing) (--y). Note that the preprocessed data can be used as input to other group analysis packages.

Upon completion, check to see that mris_glm wrote the following files in the directory from which it was run by typing:

```
ls -l beta_doss-thickness-100lh_000.mgz
ls -l var_doss-thickness-100lh.mgz
ls -l y_doss-thickness-100lh_000.mgz

ls -l beta_doss-thickness-100rh_000.mgz
ls -l var_doss-thickness-100rh.mgz
ls -l y_doss-thickness-100rh_000.mgz
```

in the "stats" directory. The files called beta_doss-thickness-100?h_000.mgz contain the regression weights; files called var_doss-thickness-100?h.mgz contain the noise variance; and files y_doss-thickness-100?h_000.mgz contain the input data after resampling and smoothing, for both hemispheres.

2008-06-01 20:47

•

To follow this exercise exactly be sure you've downloaded the tutorial data set before you begin. If you choose not to download the data set you can follow these instructions on your own data, but you will have to substitute your own specific paths and subject names.

Specify contrast vectors to test hypotheses

Contrast vectors are given to mri_glmfit to specify the comparison you want to look at. They will be used in conjunction with the design matrix that was generated from your FSGD file. For instance, a contrast vector such as [1 0 0 0 ...] is used to examine the strength of the observed effect from the EV in the first design matrix column. Another contrast vector, [1 -1 0 0 ...], is used to compare the effects between the first two EVs in the design matrix. You can specify your contrast vector as a separate file, which will be read in by mri_glmfit, and used to test your hypotheses. For the example described here there were two classes (male, female) and one variable (age). The 3 regressors in the design matrix, when using DOSS, will be:

1. Intercept of Male
2. Intercept of Female
3. Age Slope

Your contrast vector can be set up, using these regressors, to make any comparisons you'd want. If you wanted to compare the thickness differences between Males and Females, while regressing out age you could use a contrast vector [1 -1 0]. If you wanted to make a direct comparison between thickness and age, regardless of gender, you could use the contrast vector [0 0 1].

If you are using DODS there will be 4 regressors in the design matrix:

1. Intercept of Male
2. Intercept of Female
3. Age Slope (male)
4. Age Slope (female)

Your contrast vector can be set up, using these regressors, to make any comparisons you'd want. If you wanted to compare the thickness differences between Males and Females, while regressing out age you could use a contrast vector [1 -1 0 0]. If you wanted to make a direct comparison between thickness and age between class, you could use the contrast vector [0 0 1 -1].

2008-06-01 20:47

Contrast vector solution

A contrast vector for testing the dataset for a difference between males and females with age regressed out would be:

```
[1 -1 0]
```

2008-06-01 20:47

Group Analysis command lines

- To follow this exercise exactly be sure you've downloaded the tutorial data set before you begin. If you choose not to download the data set you can follow these instructions on your own data, but you will have to substitute your own specific paths and subject names.

In order to compute the contrasts you will first need to change to the tutorial data directory and setup SUBJECTS_DIR:

```
cd $FREESURFER_HOME/subjects/buckner_data/tutorial_subjs/group_analysis_tutorial
setenv SUBJECTS_DIR ${PWD}
```

Change into the directory you ran the estimation step in, most likely called 'stats':

```
cd stats
```

Upon successful completion of the make_average_subject command you should have an average subject in your SUBJECTS_DIR. After successfully completing Exercise A. and Exercise B. you should have an FSGD file called my_gender_age_fsgd.txt and a contrast file called age.mat, both in your SUBJECTS_DIR/stats directory. Confirm that those files exist:

```
ls my_gender_age_fsgd.txt age.mat
```

If my_gender_age_fsgd.txt does not exist, complete Exercise A. If age.mat does not exist, complete Exercise B., or just create a file called age.mat containing this line:

```
0 0 1
```

The following are sample commands, that can be used with the data, to complete a group analysis:

```
# For the left hemisphere

mris_preproc --fsgd my_gender_age_fsgd.txt \
  --target average \
  --hemi lh \
  --meas thickness \
  --out lh.my_gender_age.thickness.mgh
```

Once this completes you should see the file lh.my_gender_age.thickness.mgh. The next step, the smoothing step, will use this as input:

```
mri_surf2surf --hemi lh \
  --s average \
  --sval lh.my_gender_age.thickness.mgh \
  --fwhm 10 \
  --tval lh.my_gender_age.thickness.10.mgh
```

Once this is complete you should see the file lh.my_gender_age.thickness.10.mgh. The next step will test the hypothesis you've set up using your contrast vector:

```
mri_glmfit --y lh.my_gender_age.thickness.10.mgh \
  --fsgd my_gender_age_fsgd.txt doss \
  --glmdir lh.my_gender_age.glmdir \
  --pca \
  --surf average lh \
  --C age.mat
```

You should have two new directories in FREESURFER_HOME/subjects/buckner_data/tutorial_subjs/group_analysis_tutorial/stats - lh.my_gender_age.glmdir and rh.my_gender_age.glmdir. If you do:

```
ls lh.my_gender_age.glmdir
```

you should see this:

```
age/      beta.mgh  fsgd.X.mat    pca-eres/  rvar.mgh  y.fsgd
ar1.mgh   eres.mgh  mri_glmfit.log  rstd.mgh   Xg.dat
```

and if you do:

```
ls lh.my_gender_age.glmdir/age
```

you should see this:

```
C.dat  F.mgh  gamma.mgh  maxvox.dat  sig.mgh
```
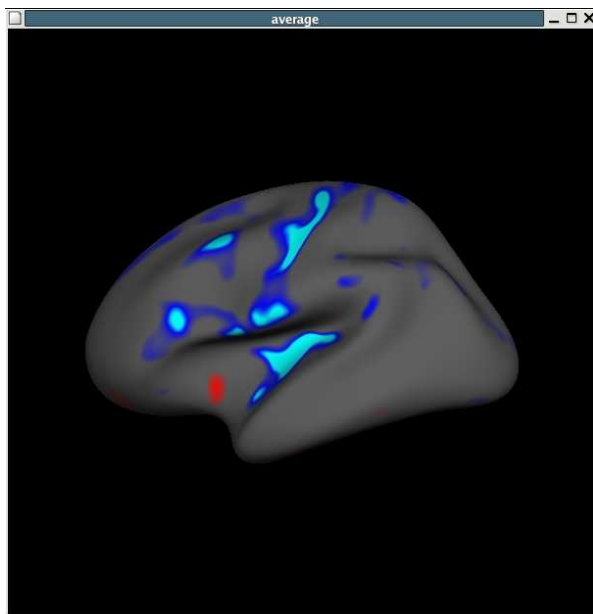
You can view your results by opening the average subject in tksurfer:

```
tksurfer average lh inflated
```

and loading in your overlay, **File -> Load Overlay** and browse to stats/lh.my_gender_age.glmdir/age/sig.mgh.

It should look like this:

If you would like to also view the scatter plots associated with this you can do so by **File -> Load Group Descriptor File** and browse to stats/lh.my_gender_age.glmdir/y.fsgd. Once that file is loaded, a window titled 'Thickness' will appear. Once that appears, then click on any point on the surface, and the thickness data for that surface vertex will appear in the Thickness plot window. The surface areas in blue indicating cortical thinning with age, as shown in the plot.

The commands are the same for the right hemisphere, replacing every **lh** with an **rh**.

2008-06-01 20:47

TUTORIAL LOCATION HAS MOVED

Good news for those of you who are confused by FSGD files, and mri_glmfit commands: There is a new group analysis tool in Freesurfer, Qdec. To accompany this tool there is also a new group analysis tutorial.

Please see the new group analysis tutorial, using Qdec. Click here

Visualization and Inspection of Group Analysis Results

-
  To follow this exercise exactly be sure you've downloaded the tutorial data set before you begin. If you choose not to download the data set you can follow these instructions on your own data, but you will have to substitute your own specific paths and subject names.

The statistical parametric map created from the group analysis can be loaded into tksurfer for visualizing as an overlay on the average surface, and inspecting either vertex-by-vertex, or within a defined region of interest (ROI). In particular, tksurfer is able to configure the colormap, to threshold the statistical parametric map and to plot correlation between classes or variables for both a single vertex and ROI. In these exercises, tksurfer will be used to visualize and inspect results in a number of ways.

Visualizing and plotting

Using the precomputed average surface, the following exercise shows how to load the statistical parametric map as an overlay in tksurfer, how to configure the colormap and threshold the parametric map, and how to plot correlation dynamically while interactively clicking surface vertices. The example below demonstrates how to view the pre-computed result from the left hemisphere (lh) mapping. To view the right hemisphere (rh) result, replace **lh** with **rh** in the commands and file names below.

In this example, the statistical parametric map will be overlayed onto the average surface. Before you begin you should be sure your SUBJECTS_DIR is set appropriately:

```
setenv SUBJECTS_DIR $FREESURFER_HOME/subjects/buckner_data/tutorial_subjs/group_analysis_tutorial
cd $SUBJECTS_DIR
```

To load the average surface into tksurfer, run the following commands:

```
tksurfer average lh inflated
```

Tksurfer will display a dark grey inflated surface in its display window. You will probably find it easier to have a curvature file loaded when you look at the statistical maps. To load in the `lh.sulc` file hit ctrl

and right click the curvature button and type `lh.sulc` into the dialogue box. Then load the lh map as an overlay: from the tools window, load in file `stats/lh.gender_age.glmdir/age/sig.mgh`

---

in the stats directory by hitting ctrl and right clicking the overlay button . Check the button that says "Calculate Identity Matrix". The display window should look something like the image shown below. The surface and overlay can be viewed from different angles in the Display Window by clicking tksurfer's orientation icons.
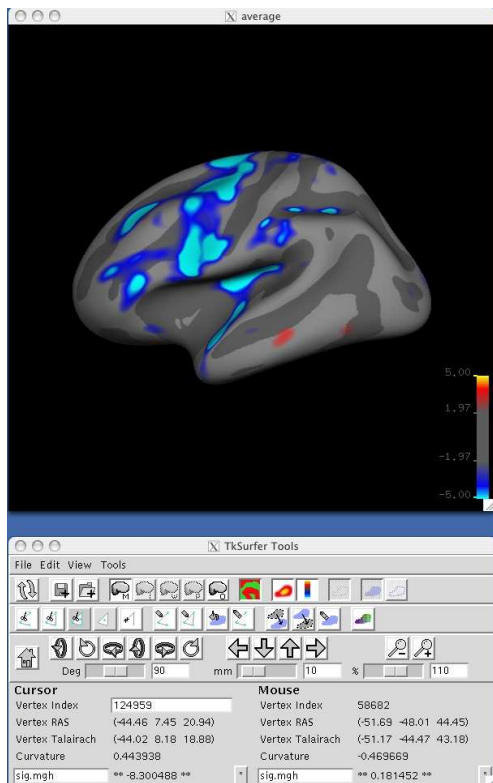
Next, load FSGD file `lh.gender_age.glmdir/y.fsgd` in the same directory by selecting **File -> Load Group Descriptor File**. At first, an empty plot will be displayed in a separate window. To generate a plot for a particular vertex, select a point by clicking on the parametric map at that vertex and the plot window will be updated accordingly. Rolling the mouse over data points on the plot will display the corresponding subject ID. Take some time to plot and inspect results for different vertices, both in significant regions of the parametric map and elsewhere. The plot window should look something like the image below:
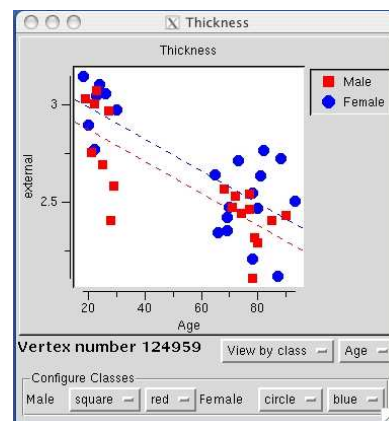
Thresholding and Setting the Color Scale

The parametric maps's colormap can also be configured for display by selecting **View -> Configure -> Overlay**. This action brings up a new GUI panel which displays a histogram of the overlay's values, and allows values for min, max, mid and slope to be set as shown below. Experiment with these settings to see how they affect the display.

When setting a color scale, you're basically interested in two things: the threshold (ie, the value below which the voxel will be transparent), and the saturation point (ie, the value beyond which the color will not change). In tkmedit/surfer, there is an extra degree of freedom which allows the color scale to be divided into two parts, each with a different rate of color change. This extra flexibility is more of a curse than a benefit as most people are only interested in having a uniform change in color across the color scale.There

are three color scale parameters in tkmedit/tksurfer:

fthresh - the value below which no color is shown ("Min" in the control panel)
fmid - the value at which the color gets to its midpoint ("Mid" in the control panel)
fslope - the slope of the colorbar above fmid ("Slope" in the control panel)

The color scale will saturate at fmid + 1/fslope. fthresh is the threshold mentioned above. To assure that the color scale will be uniform, set:

fmid = (saturation + threshold)/2
fslope = 2/(saturation - threshold)

Note that this forces the slope between fthresh and fmid to be the same as between fmid and the saturation level.

The meaning of these thresholds depends upon the nature of the data you have loaded as the overlay. Eg, the map you are currently viewing is -log10(p), where p is the significance, so a threshold of 2 will display all vertices with p<.01.
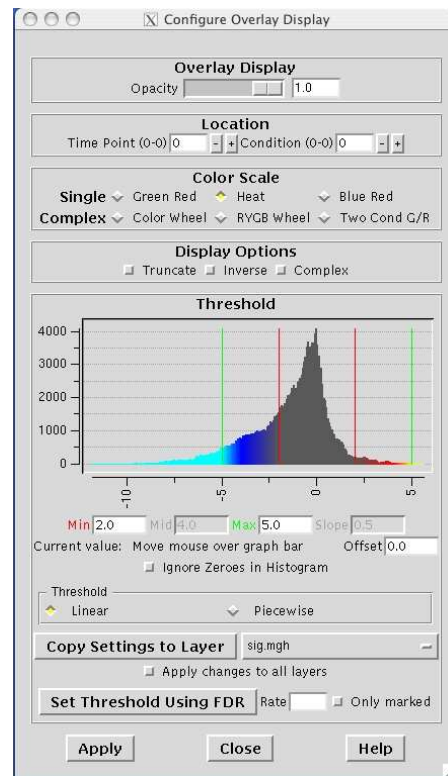
False Discovery Rate Thresholding

False Discovery Rate (FDR) is a method used to set the vertex-wise (or voxel-wise) threshold (ie, fthresh above) based on the risk of False Discovery that the user is willing to accept. The False Discover Rate is the number of vertices falsely declared active as a proportion of the number of TRULY ACTIVE vertices. Thresholding based on FDR is usually less consertative than False Positive Rate (FPR) methods such as Bonferroni correction. The Bonferroni FPR is the number of vertices falsely declared active as a proportion of the TOTAL number of vertices (active or not). The FDR threshold is based on the data and so can change from data set to data set. For more information on FDR see : Thresholding of Statistical Maps in Functional Neuroimaging Using the False Discovery Rate. Genovese, Lazar, Nichols. NeuroImage 15:870-878, 2002.

FDR is built directly into the tksurfer interface. To use it, choose your desired FDR, enter it into the the FDR entry box and hit "Set Threshold Using FDR". tksurfer will compute the threshold needed to realize this FDR. See how the Min threshold changes. Hit "Apply" to change the threshold applied to the map. Note: the values in the map MUST be -log10(p-value). This is the form of the output from mris_glm.

Defining a region of interest (ROI)

Tksurfer has the ability to compute and display statistics averaged over a defined region of interest (ROI), which is another popular way to test statistical hypotheses. The following exercise shows how to define, fill, select and plot statistics for an ROI. To define a label that marks a region of interest (ROI) on the surface, first right click in the graphics window to clear any marked vertices. To outline the region that the label should be drawn, make a series of left clicks on the surface. Then click the "Make Closed Fill Boundary" icon in the tools window . This should connect the points you clicked, drawing a yellow or red line between them. If it seems to hang for a while, press Ctrl-c to cancel the line and try again. The ROI should look like the one shown outlined in red in the image below:
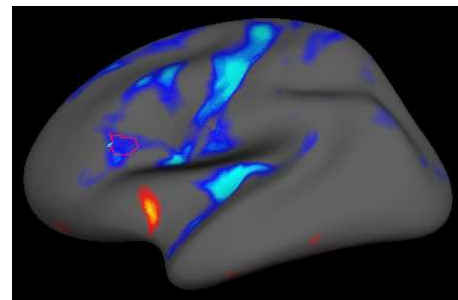
To fill the label, click once in the middle of the polygon and then click on the "Custom Fill" icon . This action will raise a dialog box with several fill options. Select the one that says "Up to and including paths" and then click the fill button. As a result, the ROI should appear like this:
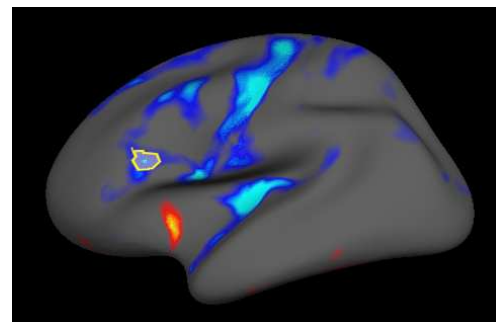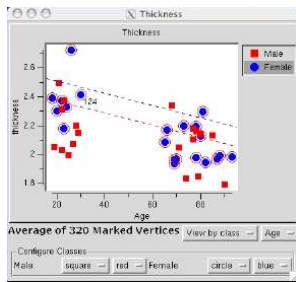

In order to plot statistics averaged over this new ROI, first select the label using **Tools -> Labels -> Mark Selected Label**. Tksurfer indicates that the ROI is selected with the label's background color turns to white as shown below:
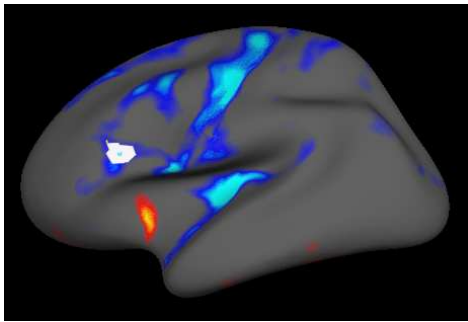
Now select **Tools -> Group -> Graph Marked Vertices Avg** to update the plot with the data averaged over the selected ROI:

Finally, save an image of this plot to a TIFF file for including in a publication or presentation: select **Tools -> Group -> Save Plot to Postscript File**. Also save the plot data to a table in a tab-delimited format for analysis using other software packages, select **Tools -> Group -> Save Plotted Data to Table**.

Mapping an ROI from one subject to another

After defining an ROI(<labelname>) on a subject(<source>), that ROI can then be mapped to another subject(<target>) with the command:

```
cd $SUBJECTS_DIR
mri_label2label --srcsubject <source> --srclabel <source>/label/<labelname> --trgsubject
 <target> --trglabel <target>/label/<labelname> --regmethod surface  --hemi <hemi>
```

2008-06-01 20:47

---

top

FreeSurfer Tutorial: Applying FreeSurfer Tools to FSL fMRI Analysis (FEAT)

The purpose of these series of exercises is to give you some familiarity with integrating FreeSurfer and FSL's functional analysis. The main challenge in the integration is getting a subject's anatomical data properly registered with their functional data. Once registered, you can use FreeSurfer display tools to visualize your FSL functional activation maps on the subject's anatomical volume and on the surface. You can also convert to the common surface space for group analysis with mri_glmfit or FSL's flame or randomise.

For structural data, we will use ${SUBJECTS_DIR}/bert and ${SUBJECTS_DIR}/fsaverage. For functional data, we will use fbert1.nii.gz and fbert2.nii.gz. The functional data set consists of two runs, 85 volumes each, 64 x 64 x 35 voxels, with size 3.4375 x 3.4375 x 4.0 mm^3. TR = 3 sec. The experiment a periodic block design with 15 sec ON blocks of simultaneous finger tapping, flashing checker board, and auditory tone. The OFF blocks are rest periods. The paradigm starts with an OFF block. The FEAT functional analysis has aleady been done, and the runs have been combined with GFEAT (Fixed-effects, one-sample group mean). This data was smoothed in the volume at 5mm, but when preparing for a group surface-based analysis, we recommend that you smooth on the surface prior to group analysis and not smooth in the volume at all (or less than the voxel size) You can download the tutorial data from here.

Throughout the tutorial, it will be assumed that you are in the fbert-feat directory. If you list the directory (ls), you will see:

```
fbert1.nii.gz -- run 1 raw functional data
fbert1.design.fsf -- run 1 FEAT design
fbert1.feat -- run 1 FEAT directory design
fbert2.nii.gz -- run 2 raw functional data
fbert2.design.fsf -- run 1 FEAT design
fbert2.feat -- run 2 FEAT directory design
fbert.gfeat -- runs 1 and 2 combined in FFx model
run-fsfeat-tut -- script to run through all the non-interactive components
```

You will see some other files/directories there too, but these are the most important.

1.0 Registration

The registration process computes a matrix that maps the FEAT example_func to the subject's anatomical using FLIRT. This matrix can then be used in later steps to display functional maps on the anatomical volume and the surface.

-
  Exercise A. Registering FSL Feat output to the anatomical

2.0 Overlaying onto Same-Subject Anatomical

The statistical maps from Feat may be overlaid onto the subject's anatomical volume, the surface derived from the anatomical volume, or the FSL's standard volume. All these options are described in the following exercise.

-
  Exercise B. Overlaying FSL Feat statistical maps

3.0 Surface-based Group Functional Analysis

The statistical maps from Feat may be overlaid onto the subject's anatomical volume, the surface derived from the anatomical volume, or the FSL's standard volume. All these options are described in the following exercise.

-
  Exercise C. Surface-based Group Analysis

4.0 Mapping automatic segmentations

FreeSurfer automatically generates cortical and subcortical segmentations from the subject's anatomical data. These segmentations can be mapped into the functional space for performing region of interest (ROI) analysis. Then, the segmentation for a particular structure can be extracted to create a binary mask. Go through the following exercise for details.

-
  Exercise D. Mapping automatic segmentations to the functional space

2008-06-01 20:47

- To follow this exercise exactly be sure you've downloaded the tutorial data set before you begin. If you choose not to download the data set you can follow these instructions on your own data, but you will have to substitute your own specific paths and subject names.

Registering FSL Feat output to the anatomical

The registration is a multi-step process. Each step is described in detail below. To begin the exercises, first enter the following:

```
tcsh
setenv SUBJECTS_DIR $FREESURFER_HOME/subjects/buckner_data/tutorial_subjs
cd $SUBJECTS_DIR/fbert-feat
ln -s $FREESURFER_HOME/subjects/fsaverage
ln -s $FREESURFER_HOME/subjects/bert
```

The subjects 'fsaverage' and 'bert' may already exist in the subjects directory, so ignore any warnings that might be issued that say 'File exists'.

1.1 Automatic registration

The program that performs the registration is called reg-feat2anat (which uses FLIRT to do the actual computations). Information about how to run this program and what this program is doing can be obtained with:

```
reg-feat2anat --help
```

Now, run the registration for bert's first fMRI data:

```
reg-feat2anat --feat fbert1.feat --subject bert
```

Verify that this created directory fbert1.feat/reg/freesurfer. There are several files with matrices in them. The most important one is anat2exf.register.dat.

1.2 Manual Checking/Editing the registration

You should ALWAYS visually check your registration. This is done with tkregister2 through reg-feat2anat. This program brings up a GUI which allows you to inspect and interactively edit the registration matrix. tkregister2 is a complicated with lots of features. You can get a full help on it with:
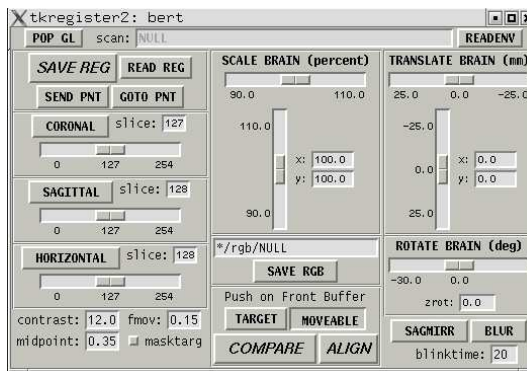
```
tkregister2 --help
```

To check the registration computed in the previous step, run:

```
reg-feat2anat --feat fbert1.feat --manual
```

You will see the following GUI window:

You will also see an anatomical similar to the one on the left below. Pressing the "COMPARE" button will allow you to flip between the anatomical (left) and functional (middle). The green line is the orig surface. It is the same in both the anatomical and functional. The right is the same image as the middle without the surface.
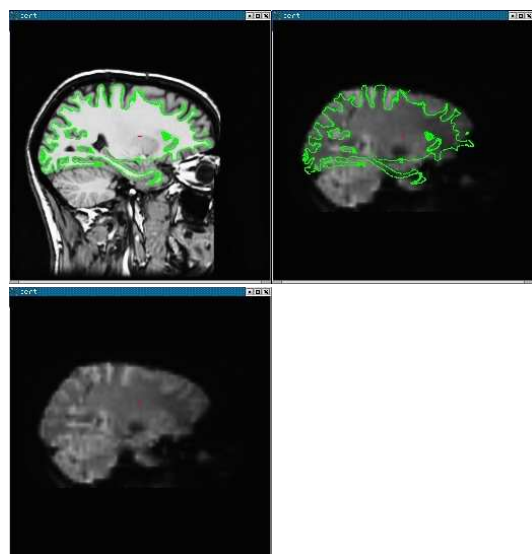
1.3 Navigating through tkregister2

tkregister2 is controlled through the interface and through keypress commands. When using a keypress command, the image window must have control of the cursor; this can be accomplished by clicking in the image window. Here are some useful commands.

A. To switch between the functional and anatomical, press the "COMPARE" button.
B. To change orientation, press the CORONAL, SAGITTAL, or HORIZONTAL button.
C. To change slice, operate the slider.
D. To change the functional brightness, change the value in the "fmov" entry box. Or you can use the 'i' keypress command to turn on automatic intensity normalization.
E. To toggle the surface on and off, use the 's' keypress command.

You can get more information on keypress commands and general usage with:

```
tkregister2 --help.
```

1.4 Evaluating the registration

The first step in evaluation is to make sure that there was no catastrophic failure, e.g., the functional is rotated by 90 degrees with respect to the anatomical, or is shifted by a large amount. Such failures are usually caused by failures in the registration to standard space (see section on troubleshooting).

The second step is to make sure there was not a left-right flip. This is tricky because the brain is very symmetrical. However, the cortical folding patterns often have subtle asymmetries. In general, you cannot see the folding patterns in the functional, but you can see the CSF which will also follow the surface (see the images above). Make sure that asymmetries in the (green) surface are present in the CSF.

The final step is to check the alignment. This is done by making sure that green surface follows the contour of the CSF.

1.5 Pitfalls

There are several ways in which the registration may be good but can appear bad. First, don't evaluate the registration in areas known to have B0 distortion (e.g., orbital frontal, medial temporal lobe). Second, be careful using the ventricles for alignment. The anterior portion is often susceptible to B0 distortion. Third, if it looks like the surface cuts through a CSF fold, look at adjacent slices to see if it lines up better. This can happen because functional voxels are so big. Finally, be careful trying to use what looks like the edge of the brain in the functional. For example, in the middle image above, it looks like the surface extends beyond the top of the brain. However, if we adjust the brightness (set fmov to 1; see image below), one can clearly see that the surfaces are actually in brain.

1.6 Editing the registration

If for some reason you do not believe that the automatic registration is sufficiently close, you can edit the registration manually. tkregister allows you to translate, rotate, and scale the brain using the sliders on the right side of the GUI. All the modifications happen in the viewing plane. The rotations are about the red cross. Note: you should not need to adjust the scale (ie, stretch the brain) because both brains belong to the same person. This is equivalent to contraining the registration to be rigid (ie, 6 DOF). When done, press the "SAVE REG" button. Note that if you modify the registration, don't re-run reg-feat2anat as that will overwrite your edits. As an exercise, try translating the functional. First, view the functional and then move the "TRANSLATE" slider. You will see the functional brain translate (the green surface will be fixed). DON'T save the registration (if you do, then just re-run reg-feat2anat).You can get more information on editing the registration with: tkregister2 --help.

1.7 Check FEAT registration to Standard Space

This registration tool can be used to check and adjust the FEAT registration to standard space with:

```
reg-feat2anat --feat fbert1.feat --manxfm func2std
```

2008-06-01 20:47

Overlaying FSL Feat statistical maps

- To follow this exercise exactly be sure you've downloaded the tutorial data set before you begin. If you choose not to download the data set you can follow these instructions on your own data, but you will have to substitute your own specific paths and subject names.

In this exercise, the data set of subject **bert** is used for demonstration. Type the following prior to beginning the exercises:

```
tcsh
setenv SUBJECTS_DIR $FREESURFER_HOME/subjects/buckner_data/tutorial_subjs
cd $SUBJECTS_DIR/fbert-feat
ln -s $FREESURFER_HOME/subjects/fsaverage
ln -s $FREESURFER_HOME/subjects/bert
```

The subjects 'fsaverage' and 'bert' may already exist in the subjects directory, so ignore any warnings that might be issued that say 'File exists'.

1.0 Overlaying the statistical map onto the bert's orig volume

Use the following command to display the zmap (zstat1.nii.gz) from the first run overlaid onto the bert's orig volume. It will display the automatic segmentation, and will also set the threshold at z = 2.3:

```
 tkmedit bert orig.mgz lh.white -aux brain.mgz \
   -overlay ./fbert1.feat/stats/zstat1.nii.gz \
   -overlay-reg ./fbert1.feat/reg/freesurfer/anat2exf.register.dat \
   -segmentation aparc+aseg.mgz  -fthresh 2.3 -fmax 4.3
```

You should see the image below:
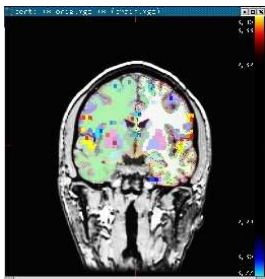
When you click or mouse over a voxel, the cortical or subcortical structure that that voxel belongs to will be displayed in the control panel. You can view any of the volumes in the stats dir in this way as well as the clustered maps in the feat directory.

2.0 View statistical maps on bert's surface

To view any of the statistical maps on bert's surface, close the tkmedit GUI (or open a new terminal window) and run:

```
tksurfer bert lh inflated \
  -overlay ./fbert1.feat/stats/zstat1.nii.gz \
  -overlay-reg ./fbert1.feat/reg/freesurfer/anat2exf.register.dat \
  -fthresh 2.3 -fmid 3.3 -fslope 1 -annot aparc.annot
```

Change the cortical parcellation to outline mode with View->LabelStyle->Outline. You should see the image below:
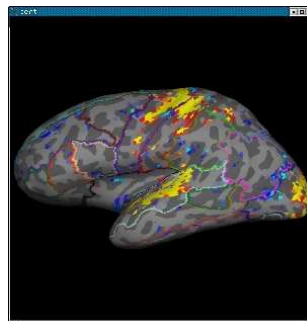
When you click or mouse over a vertex, the control panel will display the name of the cortical structure. You can view any of the volumes in the stats dir in this way as well as the clustered maps in the feat directory. You can also run the tkmedit and tksurfer commands above in separate shells and use the Save-Point/Goto-Point functionality to navigate through the volume and surface.

3.0 Displaying Same-Subject, Cross-Run GFEAT Results

Typically, one collects more than one run/series of functtional data for each subject. The individual runs are analyzed separately, then combined in standard space with GFEAT using a fixed-effects model. Since the data are no longer in the subject's native functional space, a different registration matrix is needed to map the GFEAT results to the individual. Each run of reg-feat2anat will create a reg/freesurfer/anat2std.register.dat. Any one of these can be used to map the GFEAT data to the subject's anatomy.

First, verify that the registration is good with:

```
tkregister2 --mov fbert.gfeat/mean_func.nii.gz --surf \
  --reg fbert1.feat/reg/freesurfer/anat2std.register.dat
```

mean_func.nii.gz is the mean of the example_func's in standard space.Note: if there is a problem with this registration, you need to repair the registrations for each individual run.

Now show gfeat results on anatomical volume:

```
tkmedit bert orig.mgz -seg aparc+aseg.mgz \
  -ov fbert.gfeat/cope1.feat/stats/zstat1.nii.gz \
  -ovreg fbert1.feat/reg/freesurfer/anat2std.register.dat \
  -fthresh 2.3 -fmax 4.3
```

Here we've used the anat2std.register.dat from the first run.

Now show gfeat results on the surface:

```
tksurfer bert lh inflated -annot aparc.annot \
  -ov fbert.gfeat/cope1.feat/stats/zstat1.nii.gz \
  -ovreg fbert1.feat/reg/freesurfer/anat2std.register.dat \
  -fthresh 2.3 -fmid 3.3 -fslope 1
```

2008-06-01 20:47

Mapping automatic segmentations to the functional space

-

  To follow this exercise exactly be sure you've downloaded the tutorial data set before you begin. If you choose not to download the data set you can follow these instructions on your own data, but you will have to substitute your own specific paths and subject names.

Type the following prior to beginning the exercises:

```
tcsh
setenv SUBJECTS_DIR $FREESURFER_HOME/subjects/buckner_data/tutorial_subjs
cd $SUBJECTS_DIR/fbert-feat
ln -s $FREESURFER_HOME/subjects/fsaverage
ln -s $FREESURFER_HOME/subjects/bert
```

The subjects 'fsaverage' and 'bert' may already exist in the subjects directory, so ignore any warnings that might be issued that say 'File exists'.

1.0 Mapping the segmentations

The cortical and subcortical segmentations automatically generated by freesurfer can be mapped into the functional space, which can be useful for doing region of interest (ROI) analysis. This can be done with aseg2feat:

```
aseg2feat --feat fbert1.feat --aseg aparc+aseg
```

This command will create fbert1.feat/reg/freesurfer/aparc+aseg.nii.gz. These are segmentations, meaning that each voxel has an integer value that corresponds to a particular structure. The mapping from structure number to name is given in ${FREESURFER_HOME}/FreeSurferColorLUT.txt.

2.0 Creating binary masks

The segmentation for a particular structure can be extracted to create a binary mask (i.e., a volume where the voxel value is 1 if it is in the structure and 0 otherwise). To make a binary mask of the left putamen, which has been assigned label 12 (see ${FREESURFER_HOME}/FreeSurferColorLUT.txt), use the following command:

```
fslmaths ./fbert1.feat/reg/freesurfer/aparc+aseg.nii.gz \
    -thr 12 -uthr 12 \
    ./fbert1.feat/reg/freesurfer/lh.putamen.nii.gz
```

To view this binary mask on the anatomical:

```
tkmedit bert orig.mgz -aux brain.mgz \
    -overlay ./fbert1.feat/reg/freesurfer/lh.putamen.nii.gz \
    -overlay-reg ./fbert1.feat/reg/freesurfer/anat2exf.register.dat \
    -fthresh 0.5 -segmentation aparc+aseg.mgz
```

You should see the image below:
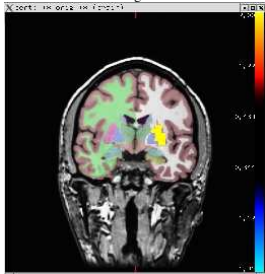
3.0 Creating ROI summaries

Once you have the segmentation mapped to the subject's native functional space, you can create summaries of the functional activation. Eg,:

```
mri_segstats --seg fbert1.feat/reg/freesurfer/aparc+aseg.nii.gz \
  --nonempty --ctab-default \
  --in fbert1.feat/stats/cope1.nii.gz --sum fbert1.segstats.dat
```

This will create fbert1.segstats.dat which will be a text file with a table of data. Each row will be a segmentation. The columns will contain various measures, including the number of functional voxels and the mean, stddev, min, max, and range of the cope over each ROI.Sample.

2008-06-01 20:47

Exercise Overview

Below are links to the tutorial exercises, extracted from the tutorial's supporting text. To follow these exercises exactly be sure you've downloaded the tutorial data set before you begin. If you choose not to download the data set you can follow these instructions on your own data, but you will have to substitute your own paths and subject names.

1.0 Morphometry and Reconstruction

Exercise A: Convert a DICOM volume into mgz format

Exercise B: Fixing bad output from the Talairach registration

Exercise C: Using control points for intensity normalization

Exercise D: Fixing bad output from skull stripping

Exercise E: View each of the volumes produced after the preprocessing steps with tkmedit

Exercise F: Recognizing and fixing inaccuracies in the white matter surfaces

Exercise G: Correcting Pial Surfaces

Exercise H: View the final surfaces with tkmedit and tksurfer

2.0 Group Analysis

Exercise A. Create an FSGD file using the table above

Exercise B. Specify contrast vectors to test hypotheses

Exercise C. Use mris_glm to compute the contrast

3.0 Visualization and Inspection of Group Analysis Results

Exercise A. Visualizing and plotting

4.0 Working with FSL Feat Output and FreeSurfer

Exercise A. Registering FSL Feat output to the anatomical

Exercise B. Overlaying FSL Feat statistical maps

Exercise C. Mapping automatic segmentations to the functional space

5.0 Working with SPM Output and FreeSurfer

5.1 Functional Registration

Exercise A. Experimenting with manual registration

Exercise B. Automatic registration using spmregister and verifying results with qmedit

5.2 Functional Overlay

Exercise A. Visualizing the parametric map overlayed onto structural data

Exercise B. Resampling the parametric map onto a surface

Exercise C. Visualizing and navigating results

2008-06-01 20:47

FreeSurfer Tutorial: FreeSurfer Tools

Tkmedit and tksurfer are two important tools in the FreeSurfer package.

Tkmedit displays volumetric anatomical data and allows the user to navigate through that data and view it from different orientations. Tkmedit also displays other data types such as functional data and surfaces as overlays onto this anatomical data.

Tksurfer displays surface data and allows the user to navigate through that data and view it from different orientations. Tksurfer also displays other data types such as functional data and curvature as overlays onto this surface data.

Tkmedit and tksurfer are described in detail below.

1.0 Tkmedit

- Interface
- General usage
- Working with data
- Quick reference

2.0 Tksurfer

- Interface
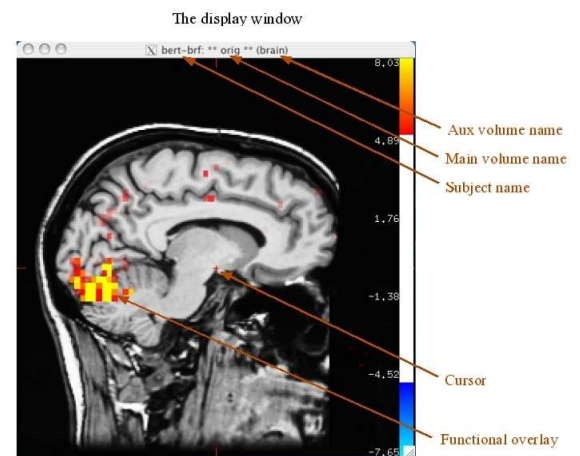- General usage
- Working with data
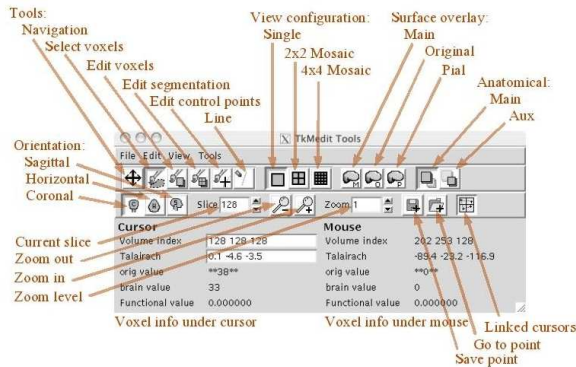- Quick reference

---

2008-06-01 20:47

Tkmedit interface

The display window

The tools window

Tools:
Navigation
Select voxels
Edit voxels
Edit segmentation
Edit control points
Line

Orientation:
Sagittal
Horizontal
Coronal

Current slice
Zoom out
Zoom in
Zoom level

View configuration:
Single
2x2 Mosaic
4x4 Mosaic

Surface overlay:
Main
Original
Pial

Anatomical:
Main
Aux

Linked cursors
Go to point
Save point

Voxel info under cursor

Voxel info under mouse

2008-06-01 20:47

General usage

1.0 Quick start

Tkmedit is started by one of two methods: using the tkmedit-sess script, or invoking it from the command line with the tkmedit command. Depending on which method you use, you may see different types of data loaded, but all methods require an anatomical data set to be loaded. Other types of data can be loaded from the File menu.

To explore your data, you can move the mouse over the picture of the volume (the Display Window) and see information about the area under the mouse in the window on the bottom (the Tools Window). You can change the view with the orientation buttons (the ones shaped like cross-sections of the head labeled 'C,' 'H,' and 'S,' which stand for Coronal, Horizontal and Sagittal) and the slice number field in the Navigation Toolbar. To save pictures of your data, go to the Tools menu and select the Save RGB item near the bottom.

2.0 Viewing area

Tkmedit consists of two windows. The top window, called the Display Window, shows the currently displayed slice. Multiple panes can show multiple orientations and configurations. The bottom window, called the Tools Window, contains the controls and feedback information. Here is the interface.

Information about the currently displayed data can be obtained by moving the mouse over the Display window or by clicking to set the cursor. As the mouse is moved in the display window, the mouseover section of the Tools window will display information about the voxel (or volume element) under the tip of the mouse arrow. Alternatively, most tools set the cursor upon clicking with the left button. The Cursor section of the Tools window displays information about the voxel under the cursor. By default, the cursor is drawn as a red cross. Guidelines are drawn on the sides of the Display Window to help locate the cursor.

It is possible to display multiple orientations of data simultaneously in the Display Window. This is done by splitting the window up into multiple viewing panes, each with a different slice, orientation, or other viewing characteristics. The cursor can be linked between so that a click in one pane to set the cursor will update the cursor in all the other panes. There are three built-in configurations: Single View, Multiple Views, and Mosaic View. These can be activated by clicking on the appropriate icon in the Navigation Toolbar (　Single, 　Multiple, 　Mosaic) or by choosing the appropriate item in the View menu.

In the latter two views, there is more than one pane in the Display Window. A green border is drawn around the active pane. To make another pane active, click in it to set the cursor. The mouseover section of the Tools Window will only display information about the voxel under the mouse if it is in the active pane.

2.1 Single view

This is the default view. It is the simplest and fastest view for tkmedit to render.

2.2 Multiple views

This viewing mode divides the window into four panes. Each of the three orientations is shown, with one orientation duplicated. This configuration is optimal for viewing all planes around a single point. By default, the cursor is linked so that setting it in one view moves the current slice in all other views to the cursor's location in that view's orientation.

2.3 Mosaic view

When mosaic view is first activated, the volume is sliced up in the orientation of the last active pane. Sixteen panes are created, each showing a different slice in the same orientation. Note that this mode is slow on most machines and is not optimal for 'live' navigation. It is mainly provided as a way to set up a view to be printed or saved to an image file.

3.0 Navigation

3.1 Changing slices

The current slice can be changed with the controls on the right side of the slice label in the Navigation Toolbar, with the mouse button 2 with the Navigation Tool (⊕), or with the arrow keys. The current orientation can be changed by clicking on the appropriate button in the Navigation Toolbar (　 coronal, 　 horizontal, or 　 sagittal) or by pressing the 'X,' 'Y,' and 'Z' keys.

With the Navigation Tool active, clicking with button 2 in the top half of the screen increases the slice number by 1, and clicking in the bottom half decreases it by 1. Clicking in dragging will continuously change the slice number; dragging up will increase the slice number, dragging down will decrease it.

3.2 Zooming

The view in any pane may be zoomed in to magnify the size of a voxel. The current zoom level is displayed both in the Navigation Toolbar and on the Navigation Toolbar. The zoom level may be changed in three ways:

1. By holding down the control key and clicking with button 1 to set the cursor and zoom in, or button 3 to zoom out. This works with any tool active.
2. By clicking on the zoom buttons in the toolbar or the zoom slider in the display area, zooming in or out around the cursor but not set the cursor.
3. By activating the Navigation Tool and clicking with button 3. Clicking in the top half of the Display window zooms in, and clicking in the bottom half zooms out. Clicking and dragging up continuously zooms in, and dragging down zooms out. This operation does not change the cursor.

3.3 Panning

While zoomed in, you will not be able to see the entire slice. To pan around in the current slice, you may hold down the control key and click with button 2 to set the cursor and re-center, or activate the Navigation Tool and click and drag with button 1 Using that Navigation Tool will not change the cursor.

As you can see, navigation is primarily down in a modal fashion with the Navigation Tool or at any time with the control-click combinations, although the Navigation Tool is more intuitive and also allows slice changing. You will probably benefit from using both methods.

4.0 Cursor

The cursor serves two main purposes. It acts as a central focus point when multiple viewing panes are open, and it selects a voxel whose information will be displayed in the Tools Window. It also serves as a kind of bookmark for sharing cursors with tksurfer, as a centroid for rotation, selecting a voxel to be graphed in the Time Course Window, and other various functions, explained later.

4.1 Linked cursors

When the viewing area is split into multiple panes, each panel may have a separate cursor location. If cursor linking is active, setting the cursor in one panel will set the cursor in all panes. Cursor linking can be activated with the 　 button on the Main Toolbar.

For example, in the Multiple Views configuration, setting the cursor in the coronal pane will change the slice in the horizontal and sagittal planes. So the cursor becomes a way of navigating through the volume using the three orientations.

Depending on the viewing configuration, with Linked Cursors active, other attributes are linked as well, including zooming and display flags. In Mosaic View mode, only the in-plane cursor coordinates will be linked, so the slice in each pane will be the same. However, if the slice number is increased in one pane, the slice number will be increased in all other panes.

4.2 Cursor and mouse information

The bottom of the Tools Window displays information about the voxel under the cursor (on the left) and the mouse (on the right). The information displayed depends on the kind of data loaded. If only an anatomical volume is loaded, only the volume index coordinates, volume Talairach coordinates, and volume value will be displayed. (If for some reason a Talairach transform could not be found, RAS (Right Anterior Superior) coordinates will be displayed instead.)

As other data are loaded, more information will appear in this area. For example, if a functional overlay is loaded, the functional overlay value will appear in this area. Alternatively, the data shown can be changed in the **View->Information** menu.

5.0 Saving images

You can save the contents of the Display Window to a picture file. Tkmedit uses the RGB format, a simple format readable by most image programs. To save a single picture, choose **Tools->Save RGB...** You may want to turn off the cursor or other view items in the **View** menu before saving your picture.

You can also save a series of images with the **Tools->Save RGB Series...** command. This command steps through slices in the current orientation and saves an image of each. You choose a directory and prefix for the file names as well as the range of slices to step through. For example, if you choose /tmp for the directory, "image" for the prefix, and 5 and 8 for the start and end, it will create /tmp/image005.rgb, /tmp/image006.rgb, /tmp/image007.rgb, and /tmp/image008.rgb.

6.0 Tips

- Most volume input operations can take many different kinds of volumes. If the volume is a bfloat or bshort volume, select one of the .bfloat or .bshort files. If the volume is a COR volume, select the COR-.info file or just enter the directory without any file names.
-

  Tkmedit uses the Tk toolkit (http://www.tcl.tk) for its interface (the 'Tk' in 'tkmedit'). A nice feature of this toolkit is that all the menus are detachable and can become a separate window that stays up all the time. This is particularly useful when using the View menu. Just click on the top line of the menu when it comes down from the menu bar, and a window will take its place.
- Learn the keyboard shortcuts for volume navigation.
  - The 'x', 'y', and 'z' keys will change orientations.
  - The arrow keys will change slices.
  - You can zoom in with Ctrl + (Ctrl-Shift =) and out with Ctrl -.
  - You can also jump to the Navigation Tool by hitting the 'n' key and click around the volume, then return to the tool you were using previously.
- If you are using tkmedit with the expected directory structure at the NMR-MGH center, tkmedit will automatically look in certain places for files. See the File Name Substitution section.
- Find yourself using certain toolbars? You can set environment variables to automatically hide or show without using the View menu. Set TKMEDIT_TOOLBAR_MAIN, TKMEDIT_TOOLBAR_NAV, or TKMEDIT_TOOLBAR_RECON to 0 or 1 to hide or show the main, navigation, or reconstruction toolbars, respectively.
- All file dialog boxes have a menu at the top which you can use to jump to specific directories to find data quickly. Additionally, you can specify your own shortcut directory by setting the environment variable FREESURFER_DATA.

2008-06-01 20:47

Working with data

1.0 Anatomical volumes

1.1 Main and aux volumes

Tkmedit can hold two anatomical volumes in memory at the same time, called the Main volume and the Aux volume. Although it can not display them together at the same time, clicking on a voxel in one volume will display that value at that voxel in both volumes.

The Main volume is considered the central data object in tkmedit. All other data is displayed relative to the Main volume. The subject name and Main volume name are displayed in the title bar of the Display Window. If an Aux volume is loaded, its name is displayed in parentheses next to the Main volume name in the title bar.

If both volumes are loaded, you may select which volume is displayed by choosing Main Volume or Aux Volume in the View menu, or by pushing in the appropriate button (for Main and for Aux) on the Main Toolbar. The currently shown volume will be surrounded by double asterisks in the title bar, and its value in the Tool WIndow is shown in double asterisks. If only the Main volume is loaded, asterisks will not be displayed.

You can go to a specific location in the volume by using the Tools, Goto Point... command, or by typing the coordinates into one of the fields in the Tools Window where the coordinates are displayed and pressing the return key.

1.2 Display transform

Display transforms can be applied to each volume to transform its position. This transform only affects the way the volume is drawn in the Display Window. All coordinates displayed in the Tools Window will not be affected by this transform.

The file must be an LTA (*.lta) or XFM (*.xfm) transform type. To load a transform for the Main volume, choose File, Transforms, Load Transform for Main Volume... from the menu and enter the file name of the transform. To unload it, choose File, Transforms, Unload Transform for Main Volume. There are corresponding items for the Aux volume.

1.3 Display options

Volume display in tkmedit is pretty basic, but there are options for configuring the brightness and contrast of each volume. Choose View, Configure, Brightness / Contrast... to open a dialog with four sliders. The top two control the brightness and contrast of the Main volume and the bottom two the Aux volume.


You may also turn off the volume display by unchecking View, Anatomical Volume. This is useful if you want to look at other types of data with the anatomical volume in the background. Neither the Main nor Aux volume will be displayed if this item is not checked.

You can display a maximum intensity projection by checking the View, Maximum Intensity Projection item. This displays the voxel with the highest value in the current orientation, and is the same for every slice in an orientation.

1.4 Editing

Editing is done with the Edit Tool which can be activated by pushing the button in the Main Toolbar, by choosing Tools, Edit Voxels, by pressing the 'a' key.

There are settings for two 'colors' for this tool, each bound to a button when this tool is active. These settings can be choosing Tools, Configure Volume Brush.... This tool, along with the other editing tools, uses the global brush settings available in Tools, Configure Brush Shape... and in the Reconstruction Toolbar. These include the shape of the brush (circular or square), its radius, and whether or not it acts in all three dimensions ( ). You can also select whether your edits affect only the Main volume or both the Main and Aux volumes. (Most users will only want to edit one volume.) If the Main volume is the only target and even if the Aux volume is visible, the Main volume will still be edited.

This brush uses a threshold that allows you to specify which voxel values to be affected. This allows you to change ranges of values in an area to a new value, useful in cleaning up large anatomical areas.

There are a few other operations you can use to edit the volume. In the Tools, Volume menu there are commands for Thresholding, Flipping, and Rotating the volume. These are self-explanatory and can be used to fix misoriented data. These operations only affect the Main volume only.

The Smart Cut command is designed for removing entire chunks of volume based on the position of the cursor. It will create a cutting plane perpendicular to the current viewing orientation based on the closest edge of the window and remove everything from that plane to the edge. For example, if you place the cursor near the bottom-middle of the screen and activate the command, it will remove everything from the cursor to the bottom edge.

1.5 Undoing

There are three ways to undo an edit. All of them only work with the Main volume, as it is the only one that can be edited. The first is with the standard Edit, Undo Last Edit menu item. This action will undo the last edit, and is the quickest and easiest way to undo single edits.

Another way is to use the Snapshot Volume tool in the Tools, Volume menu. Take a snapshot of the volume to save a copy of its contents in memory by choosing Edit, Take Snapshot of Volume. To restore the volume to this state, use the Edit, Restore Volume to Snapshot tool. This method is ideal for making 'milestone' versions of the the volume state. There are buttons for these commands on the Reconstruction Toolbar: for taking a snapshot, and for restoring to it.

The third way is to use the Undo Volume. This can be envisioned as a separate volume containing all the voxels that have been changed. You can view these voxels by checking View, Undoable Voxels; they will be drawn with a blue highlight. By holding down the shift key and clicking with mouse button 2 with the Edit Voxel tool active on an undoable voxel in Display Window, you can undo all contiguous undoable voxels, like an undo flood fill. This is best suited for undoing contiguous areas of voxels.

1.6 Saving

After all your careful edits, you will want to save the changes to your data. tkmedit writes anatomical volumes out as COR- files. Choose File, Save Main Volume to overwrite the original source volume or File, Save Main Volume As... to specify a new location. You can save the Aux volume with corresponding commands in the File, Aux Volume submenu.

2.0 Control points

Control points are used in surface construction. Every subject has an associated control.dat file in their tmp directory; this is the list of control points, in TalairachSpace, that is read in to tkmedit automatically upon start.

2.1 Display options

Control points appear as green cross hairs in the Display Window. They can be turned off by unchecking **View->Control Points**.

2.2 Making and deleting control points

Control points can be edited with the Edit Control Points tool, which can be activated by pushing the button on the Main Toolbar, by choosing **Tools->Edit Control Points**, or by pressing the 'c' key when the mouse is in the Display Window.

To make a new control point, click anywhere in the Display Window with mouse button 2. To remove a control point, click near it with mouse button 3. Button 3 will remove the closest control point to the voxel clicked, within 3 voxels.

2.3 Saving

Whenever a new control point is made, it is automatically written to the control.dat file. However, removed control points are not removed from the file until you choose **File->Save Control Points** or quit normally with the **File->Quit** command.

3.0 Selections

Selections are also called labels or ROIs. You can read or write selections from and to label files or create them from within tkmedit. Note that there is only one selection in tkmedit, even though it doesn't have to be contiguous.

3.1 Display options

Selections are drawn as a green overlay. This overlay can be turned off by unchecking **View, Selection / Label**.

3.2 Selecting and deselecting

You can select and deselect voxels with the Select Voxels Tool, which can be activated by pushing the button on the Main Toolbar, by choosing **Tools, Select Voxels**, or by pressing the 's' key.

This tool is a brush tool, so it uses the global brush settings available in **Tools, Configure Brush Shape**... and in the Reconstruction Toolbar. You can think of selecting and deselecting voxels as painting your selection. To select voxels, use mouse button 2 to draw your selection and button 3 to erase it.

You can also use a flood fill to select and deselect contiguous areas. Open the configuration dialog box with the **Tools, Configure Flood Select**...command. Here, you have options for the making the fill 2D or 3D, settings the source, and the fuzziness and distance parameters.

-

   Source Volume: Specifies which volume to use when looking for similar voxels, so that the Main, Aux, or segmentation volumes can be used to look for regions.

   Fuzziness: Determines the level of similarity to consider when finding regions. If it is 0, only voxels with exactly the same value as the voxel clicked will be set. If it is greater than 0, this is the maximum value difference from the clicked voxel that a contiguous voxel can have to be considered in the same region.

   Distance: Determines the maximum size of the region to fill from the clicked voxel. If it is 0, there will be no distance limit.

   3D: Determines if the fill is in-plane only or will fill in three dimensions.

Once you have configured your fill, perform it with shift-mouse button-2 or 3; 2 will select an area, and 3 will deselect an area.

3.3 Saving and loading

Tkmedit reads and writes label files. This file format is basically a list of points in RAS coordinate space with some header information. They can be loaded into tksurfer if they intersect with a surface.

4.0 Surfaces

Tkmedit can read and display 3D surface files. The surfaces are intersected with the displayed slices and drawn overlaid onto the anatomical data.

4.1 Surface configurations

There are three possible surface configurations: Main, Original, and Pial. When you first load a surface, it is read into the Main configuration, and tkmedit looks for the other two configurations with similar names.

You can manually load in another surface as the Original or Pial configuration. All three configurations can be displayed at the same time, so you can compare how a surface looks in relation to another one.

4.2 Loading and unloading

Use the **File, Load Main Surface...** command to load in the Main configuration. This will also try and load Original and Pial versions of this surface. Use the **File, Load Surface Configuration...** submenu to load the **Original Vertices** or **Pial Vertices** manually.

Surfaces take up a lot of memory and they some time to intersect and draw. To release these resources when you are done with a surface, use the File, Unload Surface command.

4.3 Display options

Each configuration is drawn in a different color. By default, the Main surface is drawn in yellow, the Original surface in green, and the Pial surface in red. You can change these colors in the **View, Configure..., Surface...** dialog box. You can also configure the width of the line used to draw the surface.

You can turn off surface display by unchecking the **View, Main Surface, View, Original Surface**, and **View, Pial Surface** items. You can also hide and show the surfaces with the buttons on the Reconstruction Toolbar (M Main surface, O Original surface, and P Pial surface). You can display the actual vertices of the surface by checking the **View, Surface Vertices** option.

There are two ways of determining which points to draw when intersecting the surface with the current slice; by using interpolation, and by using projection. Using interpolation will produce a smoother picture and a truer view of the surface intersection. However, the vertices that are drawn will be the interpolated intersection points of an edge and the slice, not the actual vertex. When projection is used, the vertices drawn will appear in their actual place in-plane. This diagram illustrates the methods better:
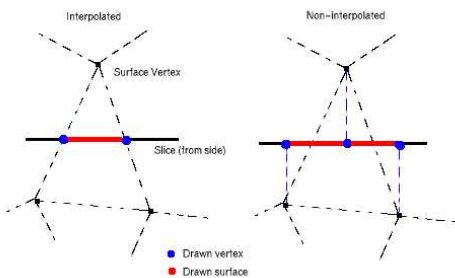
When **View, Interpolate Surface Vertices** is checked, the interpolation method is used. When it is unchecked, the projection method is used.

4.4 Finding and showing vertices

Tkmedit has a few tools for finding and displaying specific surface vertices. If you know the vertex number of the vertex you are interested in, you can use the **Tools, Surface, Find Main Vertex...** command to enter the number and go to that vertex. Note that if the edge width is smaller than the slice width, the vertex may be 'embedded' in the slice and the cursor will not appear right over a vertex. There are corresponding commands for the Original and Pial surfaces as well.

You can also have tkmedit find the surface vertex closest to the cursor with the **Tools, Surface, Show Nearest Main Vertex** command. The vertex information will be printed to the console. There are corresponding commands for the Original and Pial surfaces as well.

5.0 Functional volumes

Functional data can be viewed in tkmedit as an overlay onto the anatomical data (or any other volumetric data) or in a graph, as in a time course. Tkmedit can hold one volume for each purpose in memory at the same time. The volumes can be the same, i.e. a volume can be viewed as an overlay and as a time course, or separate.

5.1 Loading data

Loading Overlay and Time Course data is similar. Use the **File, Load Overlay Data**... or **File, Load Time Course Data**... command to load in a functional volume.

You can specify a registration file to use. If you do not, tkmedit will look in the same directory as the bfile data for a file called register.dat. A registration file is a matrix that defines the translation between anatomical RAS space and functional RAS space. Tkmedit uses this to align the functional volume with the anatomical volume.

The Time Course volume can also have an offset volume. This must be specified from the command line. See the reference section for details.

5.2 Overlay

5.2.1 Display options

The Overlay is displayed, as implied in its name, as a translucent overlay in the Display Window. Its default state uses a heat color scale, in which negative values are drawn in blue and positive values are drawn in red. A threshold is used to limit the values shown and define a color ramp so that more values close to the maximum or negative maximum are red or blue and values close to the minimum or negative minimum are yellow or teal.

The Overlay display can be turned off by unchecking the **View, Functional Overlay** option. Other aspects of the Overlay display are configured in the dialog box available in **View, Configure..., Functional Overlay**.... The following options are shown:

Time Point / Condition: Changes the currently shown time point or condition in the Display Window. The range is shown in parentheses next to the label. Note that a header file must be present in order to specify the division of data between time points in conditions, otherwise tkmedit assumes multiple time points and no conditions.

Truncate negative / positive values: Check to turn off the display of negative or positive values.

Reverse values: Check to display negative values in red and positive values in blue.

Grayscale: Check to use a grayscale color scale instead of the normal blue to red heat color scale. The lowest, most negative values will be shown in black and the highest, most positive values in white.

Opaque: Check to make the Overlay opaque.

Ignore threshold: Check to display all the values in the volume with a linear threshold, disabling the threshold values.

Threshold minimum / midpoint / slope: Configures the heat scale display. The minimum is the lowest value shown. The maximum is defined by ((1/slope) + midpoint).

Note that the Apply button must be clicked for changes to take effect.

The **View, Mask to Functional Overlay** option is available to only draw areas of the anatomical volume that are in the valid space of the Overlay volume. This gives you a better sense of the entire Overlay volume and how it relates to the anatomical volume.

5.2.2 Registration

If enabled, you can use tkmedit to register the Overlay volume, or align the Overlay volume with the anatomical data. This changes the registration file. Use the **Tools, fMRI, Register Functional Overlay** command to bring up a dialog box containing arrow buttons. Use these buttons along with the associated parameters to move the Overlay volume.

You can save your changes to the registration file with the **File, Save Overlay Registration** item. You can restore the registration to its original state with the **Tools, fMRI, Restore Overlay Registration**. Use the **Tools, fMRI, Set Registration to Identity** command to set the registration matrix to the identity matrix.

Note that since registration can affect the validity of any data analysis you perform, the MGH-NMR processing stream only allows registration at a specific point in the stream. Registration may not be available at all times. It can be enabled from the command line at start-up time. See the reference section for details.

5.3 Time course

5.3.1 Graph

The Time Course is displayed in a separate graph window. The graph will only be shown if the Time Course volume has more than one time point. Clicking on a voxel in the Display Window finds the corresponding functional voxel based on the registration and displays the values at that voxel for all time points in the graph window.

If there are multiple conditions defined in the header file, each condition will show up as a separate line in the graph. The legend on the right side of the graph window displays what line corresponds to what condition. By moving the mouse over a condition label in the legend, the corresponding line in the graph will be highlighted. This is useful for picking out one condition in a crowded graph.

You can also zoom into the graph. Click and drag with button 2 to draw a box around the area of interest. The graph will be resized around that box. Click with button 3 to zoom back out.

If the same volume is being used for the Overlay and the Time Course, clicking in the graph window will change the time point in the Overlay display to what x point was clicked. You can use this method to navigate the Overlay volume.

The selection / label can be used to specify an average area to graph. To graph a selection / label as an average, draw or load a selection and choose **Tools, fMRI, Graph Current Selection**. The title at the top of the graph will change to show that a selection is being graphed instead of a single voxel. The contents of a selection can be printed to a text file in a chart form with the **Tools, fMRI, Print Time Course Summary to File**... command. You can use this data in a separate graphic program.

The contents of the graph window can be written to a Postscript file with the **Tools, fMRI, Save Time Course Graph to Postscript File**... command. It can then be included in another document or printed.

5.3.2 Display options

The graph can be configured with the **View, Configure...**, **Time Course Graph** dialog box. The top part of this dialog is a list of mappings between colors and conditions. You will see as many rows as the Time Course volume has conditions. You can assign a color to each condition with the number field on the right or hide colors / conditions with the checkbox on the left. By default, condition 0 is hidden. You can also enter a name for the color in the column on the left. This name will show up in the graph window in the legend on the right.

The bottom part has the following options:

Show error bars: If this box is checked and error data is present, vertical errors bars will be drawn at each data point, signifying the possible error for each value.

Automatically size graph: If checked, the graph axes will automatically resize according to the range of values. If unchecked, the axes will stay static.

Subtract pre-stim average: If checked, the average value of the values before the stimulus (time second = 0) for each condition is subtracted from all values. This effectively adjusts the y axis to normalize for pre-stim values.

Show percent change: If an offset volume is available, this will show the percent change for each value instead of the raw value itself.

Number of pre-stim points: Originally defined by the header file, this value lets you adjust the time = 0 value.

Time resolution: This is defined by the header file and is used to calculate how many seconds are in each time point.

Note that the Apply button must be clicked for changes to take effect.

6.0 Segmentations

A segmentation is a volume whose values represent an index of an anatomical structure or label to which the corresponding voxel in the main anatomical volume belongs. The structures are listed in a lookup table. This table also specifies the colors the structures should appear in, and is more commonly called a color table.

6.1 Loading

Both a volume and a color table must be loaded to view a segmentation. A segmentation data file is treated the same as a normal anatomical volume, so it can be a COR- volume or any other supported file type. The color table assigns a name and color to each index. You can load both items by choosing **File, Load Segmentation...**.

You can create a new segmentation with the **File, New Segmentation...** command. You will still need to enter a color table to use.

You can import a surface annotation with the **File, Import Surface Annotation as Segmentation...** command. A surface annotation is a segmentation of the surface, so that every vertex in the surface has an associated label. Tkmedit will look up the corresponding voxel for each vertex and assign the label value to that voxel in a new segmentation.

Like the anatomical volumes, tkmedit can have a Main or Aux segmentation. Each can have its own color table. To load Aux segmentations use the commands in **File, Aux Segmentation**. As with the anatomical volumes, only the Main segmentation volume can be edited.

6.2 Display options

Segmentations are drawn as a colored overlay in the Display Window. The colors are defined in the color lookup table specified at load time. The opacity of the overlay can be configured in the dialog box brought up by choosing **View, Configure..., Segmentation Display...**. overlay can be hidden by unchecking View, Segmentation Overlay. You can toggle between the Main and Aux segmentations with the **View, Aux Segmentation Overlay** command.

If the **Segmentation Label Volume Count** option in the **View** menu is checked, when the Edit Segmentation tool is active and the left button is used to set the cursor, tkmedit will count the number of contiguous voxels in the label clicked and display the result in the Cursor Information area in the Tools Window. This operation may take anywhere from 1 to 10 seconds depending on the size of the label clicked, so be aware that tkmedit may seem unresponsive for this length of time after setting the cursor with this option activated.

6.3 Editing

A segmentation can be edited with the Edit Segmentation tool, which can be activated by pushing the

button on the Main Toolbar, by choosing **Tools, Edit Segmentation**, or by pressing the 'g' key.

By clicking with button 2 with this tool, you can 'paint' a label in the segmentation. The color used can be set in the dialog box brought up by choosing **Tools, Configure Segmentation Brush...**. This tool is a brush tool, so it uses the global brush settings available in **Tools, Configure Brush Shape...** and in the Reconstruction Toolbar. Clicking with button 3 will 'erase' segmentation labels (by setting the label values to 0).

The painting function is similar to the Edit Voxels tool: the clicked voxel, and those around it depending on the brush radius, are set to the value specified in the **Configure Segmentation Brush** dialog. Clicking on a voxel with button 2 and the control and shift key held down will set the current painting color to the color of the clicked voxel. Edits can be undone with the **Edit, Undo Last Edit** menu item.

A flood fill is also available, activated by shift-clicking mouse button 2 or 3. This tool fills areas with values equal to or similar to the voxel clicked. It looks for contiguous areas in the source volume to fill in the segmentation with the selected color. It is configured in the **Configure Segmentation Brush** dialog. It has parameters for 3D, the source volume, fuzziness, distance.

Source Volume: Specifies which volume to use when looking for similar voxels, so that the Main, Aux, or segmentation volume can be used to look for regions. **Fuzziness:** Determines the level of similarity to consider when finding regions. If it is 0, only voxels with exactly the same value as the voxel clicked will be set. If it is greater than 0, this is the maximum value difference from the clicked voxel that a contiguous voxel can have to to be considered in the same region. **Distance:** Determines the maximum size of the region to fill from the clicked voxel. If it is 0, there will be no distance limit. **3D:** Determines if the fill is in-plane only or will fill in three dimensions.

A fill can be undone with the **Edit, Undo Last Edit** menu item.

6.4 Saving

To save changes to a segmentation, choose **File, Save Segmentation** to overwrite the original segmentation volume or File, Save Segmentation As... to specify a new directory in which to write the COR- volume.

7.0 DTI

Diffusion Tensor Images can be viewed as an overlay. The red, blue, and green colors represent the direction of the vectors.

7.1 Loading

DTI volume data sets consist of two volume: a three frame eigenvector volume (EV volume) and a fractional anisotropy volume (FA volume). These must be specified in the **File, DTI, Load DTI Volumes...** dialog box. There will be a considerable delay while loading DTI volumes as tkmedit does some pre-processing.

The eigenvector volume describes the directionality of diffusion in a particular direction (with the first EV being the direction of greatest mean diffusion within each voxel), and the FA volume describes how homogeneous the diffusion is in a particular direction (high values mean that diffusion is primarily occurring in a single direction- and FA is calculated from something like the variance of the three eigenvalues).

7.2 Display options

Head points are displayed in full color. The red, green, and blue spectrums represent the directionality of the EV volume. The opacity of the overlay can be configured in the dialog box brought up by choosing **View, Configure..., DTI Display...**. The overlay can be hidden by unchecking **View, DTI Overlay**.

8.0 Head points

Head points data represent the location sensors during an MEG/EEG scan.

8.1 Loading

Head points data consist of two files: a list of points and their labels, and a transform file. These must be specified in the **File, Load Head Points...** to load head points data.

8.2 Display options

Head points are displayed as green diamonds in the Display Window. They can be hidden by unchecking **View, Head Points**. You can select a head point by clicking near it. The selected head point is drawn in red and its label is shown in the Tools Window.

When you are viewing the maximum intensity projection (by checking **View, Maximum Intensity Projection**), you will see all the head points. Note that in this view you can select head points in all planes, not just the current one.

8.3 Editing

You can edit the label of the currently selected head point by choosing **Tools, Head Points, Edit Current Head Point Label....** Enter the new name in the dialog box. You will need to click the point again in the Display Window to see the new name in the Tools Window.

You can also modify the registration transform. Use the **Tools, Head Points, Register Head Points...** command to bring up a dialog box containing arrow buttons. Use these buttons along with the associated parameters to move the head points. It is easiest if you do this in the maximum intensity projection view.

8.4 Saving

Save changes to the head points labels by choosing **File, Save Head Points**. Save changes to the registration transform by choosing **File, Save Head Points Transform**. Both commands overwrite the original files.

2008-06-01 20:47

Reference

1.0 Mouse commands

Mouse Controls for Tools

	Key Shortcut	Button 1	Button 2	Button 3
Navigation Tool	n	Clicking and dragging pans the view across the current slice. No effect when the zoom level is 1.	Clicking once in the top half of the window increases the slice by 1, clicking in the bottom decreases the slice. Dragging up increases the slice, dragging down decreases it.	Clicking once in the top half of the window zooms in, clicking in the bottom zooms out. Dragging up zooms in continuously, dragging down zooms out.
Select Voxel Tool	s	Clicking sets the cursor.	Clicking adds a voxel to the selection. Sets the cursor when the button is released. Shift+click performs a flood select.	Clicking removes a voxel from the selection. Sets the cursor when the button is released.Shift+click performs a flood deselect.
Edit Voxel Tool	a	Clicking sets the cursor.	Clicking edits a voxel with the Button 2 settings. Shift+click undoes a undoable voxel region. Sets the cursor when the button is released.	Clicking edits a voxel with the Button 3 settings.
Edit Segmentation Tool	g	Clicking sets the cursor. If the ROI Volume Count display option is on, displays the ROI volume in the Tools Window.	Clicking paints a segmentation voxel. Shift+click performs a flood fill. Shift+Ctrl+click sets the color to the color of the voxel clicked. Sets the cursor when the button is released.	Clicking erases a segmentation voxel (sets to label 0). Sets the cursor when the button is released. Shift+click performs a flood erase.
Edit Control Points Tool	c	Clicking sets the cursor.	Clicking makes a new control point. Sets the cursor when the button is released.	Clicking removes nearest control point. Sets the cursor when the button is released.

Mouse Controls for Zooming

Control+Button 1:	Zoom in, doubling the zoom level
Control+Button 2:	When zoomed in, sets the cursor and re-centers the view around the new cursor without changing the zoom level.
Control+Button 3:	Zoom out, halving the zoom level

2.0 Menus

2.1 File

Load Main Volume...	Loads the Main volume from a directory of COR- files.
Load Aux Volume...	Loads the Auxiliary volume from a directory of COR- files.
Import Main Volume from File...	Reads a specified file as the Main volume.
Import Aux Volume from File...	Reads a specified file as the Aux volume.
Load Transform for Main Volume...	Loads an LTA or XFM file to use as the display transform for the Main volume.
Load Transform for Aux Volume...	Loads an LTA or XFM file to use as the display transform for the Aux volume.
Unload Transform for Main Volume	Removes the display transform for the Main volume.
Unload Transform for Aux Volume	Removes the display transform for the Main volume.
Save Main Volume	Saves the Main volume in its original COR- file directory.
Save Main Volume As...	Saves the Main volume in a new directory.
Load Main Surface...	Loads a surface overlay (e.g. orig.?h, white.?h, pial.?h), looking for other configurations and loading them available.
Load Surface Configuration	Loads an alternate (Original or Pial) surface configuration.
Unload Surface	Removes the surface and all configurations.
Load Overlay Data...	Reads in coregistered functional data for in-plane overlay.

Load Time Course Data...	Loads in coregistered functional data for graphing over multiple time points.
Save Overlay Registration	If registration has been enabled, this will save the registration to the register.dat file.
Load Segmentation...	Loads a segmentation, requiring a COR- file directory and a color lookup table file.
Save Segmentation	Saves the segmentation in its original COR- file directory.
Save Segmentation As...	Saves the segmentation in a new directory.
Load Label...	Reads in a cortical label file as the selection.
Save Label As...	Saves the selection as a cortical label file.
Load Head Points....	Loads a head points transform file and a list of MEG/EEG sensor points.
Save Head Point Transform	Saves an edited head points transform file.
Save Head Points	Saves head points.
Save Control Points	Saves control points to control.dat file. (Done automatically.)
Quit	Exits the program.

2.1.1 Load surface configuration

Original Vertices	Load the Original vertex set for an existing surface
Pial Vertices	Load the Pial vertex set for an existing surface

2.1.2 Transforms

Load Transform for Main Volume...	Load a display transform for the Main anatomical volume.
Load Transform for Aux Volume...	Load a display transform for the Aux anatomical volume.
Unload Transform for Main Volume	Unload a display transform from the Main anatomical volume.
Unload Transform for Aux Volume	Unload a display transform from the Aux anatomical volume.

2.1.3 Label

Load Label...	Load a label file and add it to the current selection.
Save Label As...	Save the current selection as a label file.

2.1.4 GCA

Load GCA	Load GCA volumes.
Save GCA	Save GCA volumes.
Unload GCA	Unload GCA volumes.

2.1.5 Head points

Load Head Points...	Load a list of head points.
Save Head Point Transform	Save a modified head point transform (overwrites existing).
Save Head Points	Save list of head points.

2.1.6 DTI

Load DTI Volumes...	Load DTI volumes.

2.1.7 Edit

Undo Last Edit	Undoes the last volume or segmentation edit.
Take Snapshot of Volume	Makes a copy of the current Main volume.
Restore Volume to Snapshot	Restores the Main volume to the saved copy.
Clear Label / Selection	Clears the current selection.
Clear Undo Volume	Clears the Undo Volume.

2.2 View

View Configurations	Shows a submenu of view configurations.
Tool Bars	Shows a submenu of available toolbars. Check or uncheck the toolbar in this submenu to show or hide the toolbar.
Information	Shows a submenu of available information area items such as coordinate labels. Check or uncheck the item in this submenu to show or hide the item.
Configure...	Shows a submenu of viewing characteristics that can be configured.
Anatomical Volume	Check to display the anatomical volume.
Main Volume	Toggle between showing the Main or Aux volume.

Aux Volume	
Maximum Intensity Projection	Check to display the maximum intensity projection of the current volume, or the highest values in the current plane for each voxel.
Main Surface	Check to display the Main surface configuration.
Original Surface	Check to display the Original surface configuration.
Pial Surface	Check to display the Pial surface configuration.
Surface Vertices	Check to display the surface vertices on top of the surface overlay.
Interpolate Surface Vertices	Check to use interpolation to draw surface vertices. Uncheck to use projection.
Functional Overlay	Check to show the functional overlay volume.
Functional Scale Color Bar	Check to show the color scale bar for the overlay.
Mask to Functional Overlay	Check to mask the anatomical volume to valid functional overlay space.
Show Histogram Percent Change	Check to show the window displaying changes in VLI label volumes.
Segmentation Overlay	Check to show the segmentation overlay.
Segmentation Label Volume Count	Check to count the volume of a label in the segmentation when clicking with the left mouse button when the Edit Segmentation Tool is selected.
Selection / Label	Check to show the selection.
Head Points	Check to show the head points.
Control Points	Check to show the control points.
Cursor	Check to show the cursor.
Axes	Check to show the coordinate space axes.
Undoable Voxels	Check to show the undoable voxels in the Undo Volume.

2.2.1 View configurations

Single View	Displays one pane with a single orientation.
Multiple Orientations	Displays four panes, allowing all orientations to be viewed at the same time around the cursor.
Mosaic	Displays a 4x4 grid of different slices in the same orientation.

2.2.2 Tool bars

Main	Check to display the Main toolbar, which has the following controls: Tool Selector (Navigation, Select Voxels, Edit Voxels, Edit Segmentation, Edit Control Points), View Configuration (Single, Multiple, Mosaic), Surface Display (Main, Original, Pial), Current Volume (Main, Aux)
Navigation	Check to display the Navigation toolbar, which has the following controls: Orientation (Coronal, Horizontal, Sagittal), Slice number, Zoom Out, Zoom In, Zoom Level, Save Point, Go To Saved Point, Linked Cursors
Reconstruction	Check to display the Reconstruction toolbar, which has the following controls: Brush shape (Circular, Square), 3D brush, Radius, Make Snapshot , Restore to Snapshot, TkTimer on

2.2.3 Information

Volume Index Coordinates	Check to show the volume index coordinates (0..255).
Volume RAS Coordinates	Check to show the RAS coordinates. The origin is at the center of the volume.
Volume Scanner Coordinates	Check to show the scanner coordinates, usually the RAS coordinates.
MNI Coordinates	Check to show the MNI Talairach coordinates. This was the old Talairach coordinate system, but has been replaced by a slightly modified version.
Talairach Coordinates	Check to show the Talairach coordinates, determined by the talairach.xfm file for a subject. May not be available if this file is missing.
Volume Value	Check to show the value of the Main volume voxel.
Aux Volume Value	Check to show the value of the Aux volume voxel.
Functional Overlay Index Coordinates	Check to show the functional overlay volume index coordinates.
Functional Overlay RAS Coordinates	Check to show the functional overlay RAS coordinates. The origin is at the center of the functional volume.
Functional Overlay Value	Check to show the value of the functional overlay volume voxel.
Segmentation Label	Check to show the name of the ROI in the Main segmentation that the voxel is a part of.
Aux Segmentation Label	Check to show the name of the ROI in the Aux segmentation that the voxel is a part of.
Head Point Label	Check to show the name of the head point nearest the cursor or mouse.
Surface Distance	Check to show the distance from the last cursor location to the present mouse or cursor location. (Used in setting surface distance values.)

2.2.4 Configure...

Brightness / Contrast...	Opens a dialog box in which you can set the brightness and contrast of the Main and Aux volumes.
Cursor...	Opens a dialog box in which you can change the color and shape of the cursor.
Surface...	Opens a dialog box in which you can change the color and line width of each surface configuration.
Functional Overlay...	Opens a dialog box in which you can change various characteristics of the functional overlay.
Time Course Graph...	Opens a dialog box in which you can change various characteristics of the time course graph.
Segmentation Display...	Opens a dialog box in which you can change the overlay opacity of the segmentation display.
DTI Display...	Opens a dialog box in which you can change the overlay opacity of the DTI display.

2.3 Tools

Navigate	Changes the current tool to the Navigation Tool.
Select Voxels	Changes the current tool to the Select Voxels Tool.
Edit Voxels	Changes the current tool to the Edit Voxels Tool.
Edit Segmentation	Changes the current tool to the Edit Segmentation Tool.
Edit Control Points	Changes the current tool to the Edit Control Points Tool.
Configure Brush Info...	Opens a dialog box in which you can change the size, shape, and depth of the brush.
Configure Volume Brush...	Opens a dialog box in which you can change the threshold and color values for the Edit Voxels Tool.
Configure Segmentation Brush...	Opens a dialog box in which you can change the color and fill settings for the Edit Segmentation Tool.
Configure Flood Select...	Opens a dialog box in which you can change the parameters for the flood action of the Select Voxels Tool.
Save Point	Saves the current cursor so that tksurfer can use its Go To Point function to go to that point.
Goto Saved Point	Sets the cursor to the one last saved in tksurfer for this subject.
Goto Point...	Opens a dialog box in which you can enter specific numerical values in multiple coordinate systems and go to that point.
Volume	Opens a submenu of tools and commands for anatomical data.
Surface	Opens a submenu of tools and commands for surface data.
fMRI	Opens a submenu of tools and commands for to functional data.
Head Points	Opens a submenu of tools and commands for EEG/MEG head point data.
Save RGB...	Saves the contents of the Display Window to an RGB file.
Save RGB Series...	Opens a dialog box in which you can tell tkmedit to automatically scroll through a series of slices and save their images to RGB files.

2.3.1 Volume

Threshold Volume...	A tool for changing all Main volume values above or below a specific value to a new value.
Flip Volume...	A tool for flipping the Main volume, creating a mirror image.
Rotate Volume...	A tool for rotating the Main volume any number of degrees around a main axis.
Smart Cut	A tool for setting large sections of the Main anatomical volume to 0. See the Working With Data section for details.

2.3.2 Surface

Show Nearest Main Vertex	Sets the cursor to the point of the nearest vertex on the Main surface configuration.
Show Nearest Original Vertex	Sets the cursor to the point of the nearest vertex on the Original surface configuration.
Show Nearest Pial Vertex	Sets the cursor to the point of the nearest vertex on the Pial surface configuration.
Show Nearest Main Surface Edge	Sets the cursor to the point of the nearest interpolated vertex on the Main surface configuration.
Show Nearest Original Surface Edge	Sets the cursor to the point of the nearest interpolated vertex on the Original surface configuration.
Show Nearest Pial Surface Edge	Sets the cursor to the point of the nearest interpolated vertex on the Pial surface configuration.
Find Main Vertex...	Opens a dialog in which you can enter a vertex index on the Main surface. The cursor will be set to that index.
Find Original Vertex...	Opens a dialog in which you can enter a vertex index on the Original surface. The cursor will be set to that index.
Find Pial Vertex...	Opens a dialog in which you can enter a vertex index on the Pial surface. The cursor will be set to that index.
Set Vertex Distance at Cursor	Sets the value in the Surface Distance information field for the closest vertex in the Main surface.
Average Vertex Positions...	Opens a dialog in which you can execute a command to average surface vertex positions in the Main surface.

2.3.3 fMRI

Select Contiguous Voxels by Func Value	Selects contiguous voxels to the cursor whose corresponding functional values are greater than or equal to the functional value at the cursor.
Select Contiguous Voxels by Threshold	Selects contiguous voxels to the cursor whose corresponding functional values are greater than or equal to the minimum functional threshold value.
Select Functional Voxel	Selects an entire functional voxel.
Register Functional Overlay...	A tool for aligning (registering) the functional overlay data with the anatomical data.
Restore Overlay Registration	Restores the overlay registration to its original value.
Set Registration to Identity	Sets the overlay registration matrix to the identity matrix, removing all transformations.
Graph Current Selection	Graphs the average of all voxel values in the selection in the Time Course graph window.
Print Time Course Summary to File...	Creates a file containing a table version of the time course information for the current voxel or selection.
Save Time Course Graph to Postscript File...	Saves the contents of the Time Course graph window to a Postscript file suitable for printing or including in other documents.

2.3.4 Segmentation

Select Current Label	Selects all the anatomical voxels in the segmentation label under the cursor.
Recompute Segmentation	Recomputes the Main segmentation using the Main anatomical volume and the GCA volume.
Graph Current Label Average	Graphs the average time course for all the anatomical voxels in the segmentation label under the cursor. Only works with a time course loaded.

2.3.5 Head points

Restore Head Points	Restores the head points registration to its original value.
Edit Current Head Point Label...	Edits the label of the head point closest to the cursor.
Register Head Points...	A tool for aligning the head point data with the anatomical data.

3.0 File name substitution

Whenever a file name is required to load or save data, the following system is used. First, on startup, tkmedit attempts to acquire a User Home Directory and a Subject Home Directory. The User Home Directory is always the current directory from which tkmedit was launched. The Subject Home Directory is based on the way in which the anatomical data was loaded. If the default tkmedit loading method is used, it is the concatenated contents of $SUBJECTS_DIR and the subject name. If the -f switch is used, it is the value of that parameter.

If the first character of the file name is ~ (tilde) or . (period), it is substituted with the Subject Home Directory or User Home Directory, respectively. If the first character is another alphanumeric character, a default location directory is prepended to the file name. If the first character is / (slash), it is left alone. This behavior is summarized below:

If the first character of FILENAME is....	The final file name is...
~ (tilde)	SubjectHomeDir/FILENAME
. (period)	UserHomeDir/FILENAME
/ (slash)	FILENAME
other character	SubjectHomeDir/Subdirectory/FILENAME

...where FILENAME is the file name that was input.

The subdirectory used in the last substitution depends on the file type being loaded, and is appropriate for the standard NMR center directory structure. These are:

Data Type	Subdirectory
Functional data	fmri
Head Points	bem
Surface	surf
Anatomical	mri
ROI/Label	label
RGB	image/rgb
Control Points (control.dat)	tmp
Edit (edit.dat)	tmp
Segmentation	mri
Color Table	$CSURF_DIR
Label	label
Anatomical Transform	mri/transforms

4.0 Command line options

4.1 Script

To automatically run a script, use the -tcl option, passing a script name. Tkmedit still requires anatomical data to be loaded before doing anything.

tkmedit -tcl SCRIPT_NAME

4.2 Anatomical volume

There are two ways to load anatomical data from the command line. The first is the default tkmedit mode for loading COR- volumes.

tkmedit SUBJECT IMAGE_TYPE

Where SUBJECT is a subject directory relative to the value of the SUBJECTS_DIR environment variable, and IMAGE_TYPE is a subdirectory of ${SUBJECTS_DIR}/<subject name>/mri. i.e.

tkmedit <subject name> T1

looks in ${SUBJECTS_DIR}/<subject name>/mri/T1. If there is a COR- volume in this directory, or another kind of readable data, it will be loaded as the Main anatomical volume.

Another way to load anatomical data is with an explicit path or file name using the -f switch:

tkmedit -f PATH/TO/DATA/[FILENAME]

This method looks in the given directory or file for any recognizable data. It will be loaded as the Main volume.

The Aux volume can be loaded with the -aux switch, i.e.

tkmedit <subject name> T1 -aux wm

This looks in ${SUBJECTS_DIR}/<subject name>/mri/wm for a COR- volume.

Note that tkmedit will not load without specifying a Main anatomical volume with the subject/image type method or the -f method.

4.3 Surface

To load a surface from the command line, add the name after the subject:

tkmedit <subject name> IMAGE_TYPE SURFACE

or

tkmedit -f /path/to/volume/data SURFACE

4.4 Functional volume

Overlay data is loaded with the -overlay switch and time course data with the -timecourse option, with the concatenated path and stem as the argument i.e.:

tkmedit -overlay DATA_DIRECTORY/STEM -timecourse DATA_DIRECTORY/STEM

where STEM is the portion of the bfloat file data that comes before the _xxx slice number. i.e. the stem of a file set named minsig_000.bfloat through minsig_015.bfloat is minsig.

To enable functional registration, pass -register on the command line with no arguments.

To specify a registration file that is not in the same directory as the functional volume, use the -overlay-reg or -timecourse-reg option with the registration file as the argument, i.e:

tkmedit -overlay DATA_DIRECTORY/STEM -overlay-reg /PATH/TO/REGISTRATION_FILE.DAT

To load an offset volume for the Time Course volume, specify it the same way as the Time Course volume but with the -timecourse-offset switch:

tkmedit -timecourse-offset DATA_DIRECTORY/STEM

4.5 Segmentation

Use the -segmentation switch to load a COR- volume representing a segmentation. A color file is also necessary.

tkmedit -segmentation COR_DIRECTORY COLOR_FILE

The COR_DIRECTORY is the directory of the COR- files and the COLOR_FILE is the file to use as a color lookup table.

You can specify the opacity of the segmentation overlay with the -segmentation-opacity option:

tkmedit -segmentation COR_DIRECTORY COLOR_FILE -segmentation-opacity 0.6

4.6 Head Points

Use the -headpts switch to load a head points file. This is a file of points indicating MEG/EEG sensor placement. A transform file is also required and can be specified on the command line.

tkmedit -headpts POINTS_FILE TRANSFORM_FILE

5.0 Crash recovery

If tkmedit segfaults (makes a memory access violation), and it will, and the user has edited the Main volume, the volume will be saved in the /tmp directory. This can be reloaded by loading the volume from that directory, i.e.
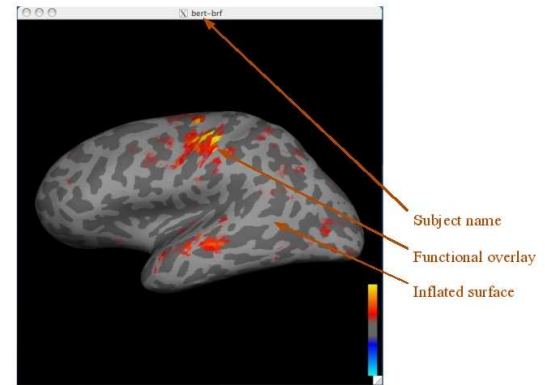
tkmedit -f /tmp

Tksurfer interface

The display window

Subject name
Functional overlay
Inflated surface

The tools window

Clear cuts
Fill uncut area
Cut plane
Cut closed line
Cut line
Redraw view
Restore view
Goto saved point
Save point
Make fill boundary
Make closed fill boundary
Custom fill
Remove selected boundary
New label from marked
Mark label
Erase label
Change label color

Row of buttons to rotate the hemisphere
Degree of rotation
Row of buttons to translate the hemisphere
Millimeters of translation

Zoom in
Zoom out
Level of zooming

Vertex info under cursor
Vertex info under mouse
File name of the functional overlay
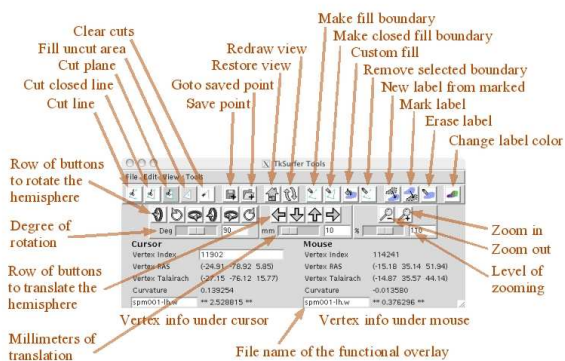
General usage

1.0 Quick start

Tksurfer can be started in any of three ways: launching it from FreeSurfer with the Surface button, using the tksurfer-sess script, and calling it from the command line. Depending on which method you use, you may see different types of data loaded, but all methods require a surface data set to be loaded. Other types of data can be loaded from the File menu.

To explore your data, you can move the mouse over the picture of the surface (the Display Window) and see information about the area under the mouse in the window on the bottom (under the Mouse section of the tksurfer Tools Window). You can change the view with the navigation buttons (under the toolbar, with icons of various arrows). To save pictures of your data, go to the Tools menu and select the Save RGB As... item near the bottom.

By default, tksurfer does not redraw the display window unless you perform a command that changes the view. This means that if another window obscures the display window and then is moved, the display window will not automatically update. This because of the time it takes to redraw an average surface. To explicitly redraw the view, just click on the Redraw button.

2.0 Viewing area

Tksurfer consists of two windows. The top window, called the Display Window, shows the currently loaded surface and its overlays. The bottom window, called the Tools Window, contains controls and feedback information. Here is the interface.

Information about the currently displayed data can be obtained by moving the mouse over the Display window or by clicking to select a vertex. As the mouse is moved in the display window, the Mouse section of the Tools window will display information about the vertex under the tip of the Mouse arrow. Information about the vertex clicked on is displayed in the Cursor section.

The view is configured with various dialog boxes and options available from the **View** menu. From this menu, you can configure the toolbars and labels available in the Tools window, various aspects of the surface, curvature, and overlays, and the time course graph window.

3.0 Navigation

Tksurfer displays the surface in a 3D environment. You may move the surface with the buttons in the movement area of the Tools window.
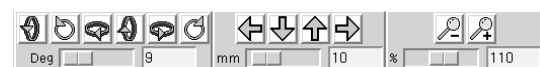
The buttons in the left section rotate the hemisphere, the ones in the middle translate it, and the ones on the right zoom in and out. The sliders adjust the number of degrees of each movement. You can also zoom in and out by holding down the control key and clicking with button 1 or 3. Button 1 zooms in, and button 3 zooms out. The vertex you click becomes the center vertex in the new view.

4.0 Cursor and marked vertices

Clicking on a vertex sets the cursor. The edges touching the vertex are drawn in a cyan color.

The cursor is drawn every time the mouse is clicked without redrawing the whole window. However, clicking also marks vertices, adding each clicked vertex to a list of marked ones. These are drawn in a white color, but only show up after the screen is redrawn. This screen shot shows two marked vertices and the cursor, at normal zoom level and zoomed in.

To see this behavior, start tksurfer with any subject and surface. Click with button 1 on the surface and see the cyan cursor appear. Now press the Redraw button (or type Alt+R while the mouse is in the display window) and see the white highlight under it. Now click around some more without hitting redraw and see the cyan cursor move. Now press Redraw, and see how all the points you previously clicked are drawn in white.

Most tools require at least two marked vertices, and some as many as four. The icons for the cut tools give hints as to how many you need. To clear the marked vertices (unmark them), click with button 3. Note that the white highlights will not disappear until Redraw is pressed.

2008-06-01 20:47

Working with data

1.0 Surfaces

Tksurfer holds one main surface. This surface structure has the surface vertex information as well as other information related to the surface, such as additional values for each vertex. Tksurfer can also hold four more surface configurations. These are additional sets of vertex information and locations, but not the other surface information. These configurations are called Inflated, White, Pial, and Original Vertices.

1.1 Loading surfaces

The main surface must always be loaded. To load more configurations, use the **File->Load Surface Configurations...** submenu. You can also load a different main surface configuration in this menu. This will replace the main vertex information but not the other surface information.

1.2 Surface display options

You can select the currently shown surface configuration with the **View->Surface Configuration** submenu. Only loaded configurations will be available.

You can enable a wireframe overlay with the **View->Wireframe Overlay** option. This will draw the surface polygon edges in blue, so you can see exactly where the edges and vertices are.

1.3 Surface related tools

There are four tools available to cut surfaces. These tools are available in the **Tools** menu and in the Main toolbar. The Cut Line tool ( ) calculates a line along all the marked vertices in the order that they were marked and cuts the vertices on that line. The Cut Closed Line tool ( ) does the same, but closes the line from the last marked vertex to the first. In the graphics below, the first image shows some marked vertices. The second shows the results after a Cut Line action, and the third shows the results after a Cut Closed Line action.

The Cut Plane tool requires exactly four marked vertices. The first three define three points on the plane and the fourth defines a point inside the plane. This image shows this process. The surface image has been scaled down to show that the entire surface has been cut away in the second image.
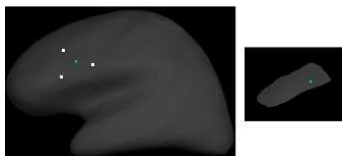
The Cut Area tool is used to cut away one incontiguous area from another. Click in the area you wish to keep and perform the action. The other area will be cut. All cut vertices can be restored with the Clear Cuts tool.

1.4 Saving surfaces

Some tools and other functions change the surface vertex information. Other tools change the vertex values and other information. To save these changes to the surface, use the **File->Save Surface** option to overwrite the original surface file and **File->Save Surface As...** to save it a new location. Note that this does not save the extra surface configurations; these are not part of the main surface structure.

2.0 Curvature

2.1 Loading curvature files

Use the **File->Curvature->Load Curvature...** item to load a curvature file.

2.2 Curvature display options

The curvature is displayed as either a green/red color scale or a binary gray background. When no overlay is displayed, the green/red color scale is used. When an overlay is active, the simpler gray is used.

The **View->Configure->Configure Curvature Display** dialog box allows you to set the midpoint and slope of the curvature color scale. This affects both the green/red and the gray display.

2.3 Saving curvature files

Use the **File->Curvature->Save Curvature** to save the curvature values and overwrite the original file, or **File->Curvature->Save Curvature As...** to save a new curvature file in a new location.

3.0 Overlay

An overlay is a set of values, 0 or 1 per vertex, displayed with a color scale. It is normally used as a functional activation overlay, but can really be any floating point value per voxel. The source volume can be a .w file, a file containing one value per vertex, previously registered with the volume. It can also be a binary volume or bfile. These are volumes of short integers (.bshort) or floating point values (.bfloat) with a corresponding registration file. Binary volumes can contain multiple time points and conditions.

Overlays can be loaded into any of five layers. Tksurfer can display one overlay layer at a time.

3.1 Loading overlay files

To load an overlay file, use the **File->Load Overlay...** item. You will be prompted for the overlay file name and the layer in which to load the file. Chose the .w file or a .bshort or .bfloat file if you are loading a binary volume.

If you are loading a binary volume, you will be prompted for the stem of the volume and a registration file. You may leave the registration file field blank if there is a registration file in the same directory as the binary volume named "register.dat."

3.2 Overlay file display options

To select which overlay to show, use the **View->Overlay Layer** submenu. The field names are automatically set to the file name loaded. You can change this name by typing a new one into the information area in the Tool window.

The display for the current overlay can be configured in the **View->Configure->Configure Overlay Display** dialog. If the data you have loaded has multiple time points or conditions, you can select which one to show with the **Time Point** and **Condition** fields. You can select the color scale to use with the radio buttons in the top area of the dialog. The **Truncate** option can be checked to turn off the display of negative values. Check the **Reverse** option to reverse the sign of the values as they are drawn in the color scale. **Inverse and Complex are reserved for future upgrades.** Also see SettingTheOverlayColorScale.

Click the **Apply** button to see your changes.

3.3 Saving overlay files

An overlay can only be written out to a .w file; binary volumes cannot be written from overlays. To write a .w file, choose **File->Save Overlay As...**, and choose a file name and layer from the dialog box.

4.0 Time course

The Time Course is a binary volume with multiple time points or conditions. The data is displayed in a graph so that all time points can be seen for a given vertex. When a vertex is clicked, the corresponding voxel is found in the binary volume. Tksurfer graphs the data for all time points at the chosen vertex.

Loading Time Course Volumes Loading binary volume Overlays and Time Course data is similar. Use the **File->Load Time Course...** command to load a time course volume. Tksurfer requires functional volumes to be in binary volume format. It will look for a header file with the same stem as the slice data or just guess the dimensions from the bfile. You need to specify the directory of the data, the stem, and optionally, a registration file. The format of a bfile name looks like this:

/path/to/data/stem_000.bfloat
/path/to/data/stem_000.bshort

The numbers go from 0 to the number of slices minus one. The 'stem' represents any text that appears before the underscore before the number in the bfile.

You can specify a registration file to use. If you do not, tksurfer will look in the same directory as the bfile data for a file called register.dat. A registration file is a matrix that defines the translation between anatomical RAS space and functional RAS space. Tksurfer uses this to align the functional volume with the anatomical volume.

5.0 Time course graph

The Time Course is displayed in a separate graph window. The graph will only be shown if the Time Course volume has more than one time point. Clicking on a vertex in the Display Window finds the corresponding functional voxel based on the registration and displays the values at that voxel for all time points in the graph window.

If there are multiple conditions defined in the header file, each condition will show up as a separate line in the graph. The legend on the right side of the graph window displays what line corresponds to what condition. By moving the mouse over a condition label in the legend, the corresponding line in the graph will be highlighted. This is useful for picking out one condition in a crowded graph.

You can also zoom into the graph. Click and drag with button 2 to draw a box around the area of interest. The graph will be resized around that box. Click with button 3 to zoom back out.

You can specify an average area to graph in two ways. You can click the vertices you want, marking them, and then choose **Tools->Graph Marked Vertices Avg** to graph the average. (Note that loading a patch marks vertices, so you can also graph the average of a path this way.) You can also load a label and graph the average of it with the **Tools->Graph Label Avg.**

The contents of the graph window can be written to a Postscript file with the **Tools->Save Graph to Postscript File...** command. It can then be included in another document or printed.

6.0 Patches

6.1 Loading patches

Use the **File->Patch->Load Patch...** item to load a patch file.

6.2 Saving patches

Use the **File->Patch->Save Patch** command to overwrite the original file, or **File->Patch->Save Patch As...** to save a new patch file in a new location.

7.0 Labels

7.1 Loading labels

Use **File->Label->Load Label...** item to load a label file.

7.2 Saving labels

Use **File->Label->Save Label** command to overwrite the original file, or **File->Label->Save Label As...** to save a new label file in a new location. the current dipole values.

8.0 Field sign

8.1 Loading field sign files

Use the **File->Field Sign->Load Field Sign...** command to load a field sign file.

8.2 Saving field sign files

Use the **File->Field Sign->Save Label** command to overwrite the original file, or **File->Field Sign->Save Field Sign As...** to save a new field sign file in a new location.

9.0 Field mask

9.1 Loading field mask files

Use the **File->Field Mask->Load Field Mask...** item to load a field mask file.

9.2 Saving field mask files

Use the **File->Field Mask->Save Field Mask** command to overwrite the original file, or **File->Field Mask->Save Field Mask As...** to save a new field mask file in a new location.

2008-06-01 20:47

Reference

1.0 Mouse commands

1.1 Mouse clicks

Button 1	Sets the cursor, drawing it in the new location, and marks the clicked vertex.
Button 2	Sets the cursor without drawing it in the new location and unmarks the clicked vertex.
Button 3	Clears marked vertices.

1.1 Mouse controls for zooming

Control+Button 1	Zooms in, doubling the zoom level
Control+Button 3	Zooms out, halving the zoom level

2.0 Menus

2.1 File

Load Surface...	Loads the main surface information, including main surface configuration.
Load Surface Configuration...	Shows a submenu with commands for loading vertex configurations.
Save Surface	Saves the contents of the entire surface structure, overwriting the original file.
Save Surface As...	Saves the contents of the entire surface structure to a new file.
Load Overlay...	Loads a .w or binary volume file into an overlay layer.
Save Overlay As...	Saves an overlay layer as a new .w file.
Load Time Course...	Loads a binary volume as a time course.
Curvature	Shows a submenu with options for loading and saving.
Patch	Shows a submenu with options for loading and saving.
Label	Shows a submenu with options for loading and saving.
Field Sign	Shows a submenu with options for loading and saving.
Field Mask	Shows a submenu with options for loading and saving.
Quit	Quits tksurfer.

2.2 Load surface configuration

Main Vertices	Loads the specified vertices from another surface file into the selected surface configuration.
Inflated Vertices	
White Vertices	
Pial Vertices	
Original Vertices	

2.3 Curvature

Load Curvature...	Loads a curvature file.
Save	Saves changes to the curvature, overwriting the original file.
Save Curvature As...	Saves the curvature into a new file.

2.4 Patch

Load Patch...	Loads a patch file.
Save Patch	Saves changes to the patch, overwriting the original file.
Save Patch As...	Saves the patch into a new file.

2.5 Label

Load Label...	Loads a label file.
Save Label	Saves changes to the label, overwriting the original file.
Save Label As...	Saves the label into a new file.

2.6 Field sign

Load Field Sign...	Loads a field sign file.
Save Field Sign	Saves changes to the field sign, overwriting the original file.
Save Field Sign As...	Saves the field sign into a new file.

2.7 Field mask

Load Field Mask...	Loads a field mask file.
Save Field Mask	Saves changes to the field mask, overwriting the original file.
Save Field Mask As...	Saves the field mask into a new file.

2.8 Edit

Undo	Undoes the cut.
Unmark All Vertices	Unmarks all vertices. Same as clicking with button 3 in the Graphics window.
Clear Label	Clears the vertices selected by loading a label.

2.9 View

Tool Bars	Shows a submenu of available toolbars. Check or uncheck the toolbar in this submenu to show or hide the toolbar.
Information	Shows a submenu of available information area items such as coordinate labels. Check or uncheck the item in this submenu to show or hide the item.
Configure	Shows a submenu of viewing characteristics that can be configured.
Surface Configuration	Shows a submenu of surface configurations. Check one to make the currently shown configuration.
Overlay Layer	Shows a submenu of overlay layers. Check one to make the currently shown layer.
Overlay	Check to display the current overlay layer.
Scale Bar	Check to display a scale bar in the window, showing the scale of 10 mm.
Color Scale Bar	Check to display a color bar in the Display window, showing the range of colors in the current overlay color scale.
Wireframe Overlay	Check to display a wireframe view of the surface edges.

2.10 Tool bars

Main	Check to make the Main toolbar visible. This toolbar contains the following controls: Cut Line, Cut Closed Line, Cut Plane, Cut Area, Clear Cuts, Save Point, Goto Saved Point, Restore Home View, Redraw View.

2.11 Information

Vertex Index	Check to show the nearest vertex index.
Distance	Check to show the distance in mm to the reported index.
Vertex RAS	Check to show the RAS coordinates of the vertex.
Vertex Talairach	Check to show the Talairach coordinates of the vertex.
MRI Index	Check to show the anatomical index of the vertex in the anatomical volume for this subject.
Vertex Normal	Check to show the normal vector of the vertex.
Spherical X, Y, Z	Check to show the spherical coordinates.
Spherical Rho, Theta	Check to show the spherical rho and theta.
Curvature	Check to show the curvature value at the vertex.
Field Sign	Check to show the field sign value at the vertex.
Overlay Layer 1-5	Check to show the overlay layer value at the vertex.
Amplitude	Check to show the value at the vertex.
Angle	Check to show the value at the vertex.
Degree	Check to show the value at the vertex.
Annotation	Check to show the value at the vertex.
MRI Value	Check to show the MRI value at the associated anatomical location for this vertex.
Parcellation	Check to show the name of the parcellation label of this vertex.

2.12 Configure

Lighting...	Opens a dialog box in which you can set the lighting parameters for the scene.
Overlay...	Opens a dialog in which you can change various characteristics of the overlay display.
Time Course...	Opens a dialog in which you can change various characteristics of the time course graph.
Curvature Display...	Opens a dialog in which you can change various characteristics of the curvature display.
Phase Encoded Data Display...	Opens a dialog in which you can change various characteristics of phase encoded data display.

2.13 Surface configuration

Main	Check to make this surface configuration the currently displayed one.
Inflated	
White	
Pial	
Original	

2.14 Overlay layer

Overlay Layer 1-5	Check to make this overlay layer the currently displayed one.

2.15 Tools

Save Point	Saves the current cursor so that tksurfer or tkmedit can use its Goto Saved Point function to go to that point.
Goto Saved Point	Sets the cursor to the one last saved in tksurfer or tkmedit for this subject.
Run Script...	Executes a tcl script.
Cut Line	Calculates a line between marked vertices and cuts the vertices in that line.
Cut Closed Line	Same as above, but closes the line between the first and last marked vertex.
Cut Plane	Requires exactly four vertices to be marked. Cuts the surface but not in the plane defined by the four vertices.

Cut Area	Requires one marked vertex and two incontiguous surface areas. Cuts the area that the marked vertex is not in.
Clear Cuts	Removes all cut vertex, restoring the surface to whole.
Send to Subject...	Same as Save Point, but allows you to enter a subject name explicitly.
Write Decimation...	Not functional, reserved for future upgrade.
Write Dipoles...	Writes a file containing the dipole values at each vertex.
Set Background Midpoint to Average	In the curvature colorscale, sets the point at which red transitions to green to the mean curvature across the whole surface, making the data more interpretable.
Fill Stats	Marks vertices based on the overlay values surrounding the cursor.
Fill Curvature	Marks the region around each selected vertex that has a similar curvature to that of the selected curvature. Used for labeling anatomical regions.
Smooth...	Iteratively smooths scalar for a user-selected number of iterations. Equivalent to convolving with a surface-based gaussian.
Inflate....	Applies the inflation force for a user-specified number of iterations, to smooth the surface.
Swap Surface Fields...	Intended for use by those who know the internal data structure of the surface. Opens a dialog box allowing you to specify two surface vertex fields to swap.
Clear Curvature	Sets the curvature value at each vertex to 0.
Graph Marked Vertices Avg	Graphs the average of marked vertices in the time course.
Graph Label Avg	Graphs the average of a label in the time course.
Save Graph to Postscript File	Saves the contents of the time course graph to a Postscript file.
Save RGB As...	Saves the contents of the Display window to an RGB file.

Glossary

anatomically derived defect: A topological defect in the cortical surface that arises from a feature of normal neuroanatomy Â to be distinguished from defects arising entirely from segmentation errors. See **topological defect**.

artifact: A feature that appears in an image but is not actually present in the imaged object.

average convexity: The signed distance that a vertex moves during the inflation process.

brain volume: The T1 volume after the skull and other non-brain structures have been removed. This volume can be viewed using tkmedit.

canonical surface: Surface-based atlas constructed from the cortical surfaces of 40 normal individuals (used for inter-subject averaging).

conversion/averaging: Process of converting and averaging multiple structural acquisitions from the native magnet format into the native FreeSurfer format (see COR files).

COR files: The native file format used by FreeSurfer to store 3D structural image data.

Euler number: After Leonhard Euler (1707-83). A topological invariant of a surface that can be computed from the number of edges, vertices and faces in a polygonal tessellation (command Âmris_euler_numberÂ). The Euler number of a sphere will equal 2; the Euler number of a surface with n handles is 2 Â 2n.

filled volume: The wm volume after separation of the left and right hemispheres and filling of each hemisphere. This volume can be viewed using tkmedit.

flattening: Producing a planar (flat) representation of a patch of the cortical surface that has minimal metric distortion.

gyrus: A fold or convolution of brain tissue (an outward folded region).

inflated surface: The smoothwm surface after inflation. This surface can be viewed using surfer.

inflation: The process of smoothing the cortex while minimizing metric distortion, so that all sulci are fully visible and surface distances are apparent to visual inspection.

intensity: Measured amount of magnetic field at a given spatial location, represented by a voxel (higher SNRÂsignal to noise ratio means voxels will have a higher intensity relative to the background noise, and appear brighter).

---

label: A particular region of interest. e.g. in tksurfer the label would be a region of interest in the surface. In tkmedit a label is a region of interest in the volume.

morphing: Computer graphics technique whereby a mapping is computed that smoothly transforms one image or surface into another.

morphometrics: The study of geometric properties of the human brain.

motion correction: Processing multiple structural volumes so that the effects of subject movement are minimized. This is typically done by aligning multiple images/volume to an initial image/volume (see conversion/averaging).

MRI volume: The three dimensional volumetric data set collected from a MRI scanner.

orig volume: The original MRI volume. This volume can be viewed using tkmedit.

orig surface: The first surface constructed by covering the labeled voxels in the filled volume. This surface can be viewed using surfer.

pial: Pertaining to the delicate pia mater which envelops the brain (gray matter). Also, the model of the pial surface (?h.pial).

pial surface: The refined estimate of the gray/CSF boundary (pial surface). This surface can be viewed using surfer.

region growing process: An algorithm that groups voxels or sub-regions into larger regions.

RF-field inhomogeneities: Spatial variations in the Radio Frequency (RF) excitation pulse. These variations result in changes in the measured intensity for a given tissue class that are related to the spatial location of the voxel.

segmentation: Labeling of tissue classes from MRI data (e.g. white matter).

smoothing: Process of producing a relatively even and regular cortical surface.

smoothwm surface: The orig surface after smoothing. This surface can be viewed using surfer.

sulcus: A groove or furrow in brain tissue (an inward folded region).

supertessellated icosahedron: Polygonal approximation to a sphere.

T1: Longitudinal relaxation constant.

T1 volume: The MRI volume after intensity normalization. This volume can be viewed using tkmedit.

T1 Weighted Image: A magnetic resonance image where the contrast is predominantly dependent on T1.

T2 Weighted Image: A magnetic resonance image where the contrast is predominantly dependent on T2.

T2: Transverse relaxation constant.

Talairach coordinate: The corresponding location in the Talairach atlas for a given point in a brain that has been coregistered with the atlas (Talairach et al, 1967).

tessellation: Covering of a surface by repeated use of a single shape.

topology: The properties of a surface related to its connectivity that are unaffected by geometric (i.e. rubber sheet) transformations.

topological defect: A portion of a surface that results in the surface topology differing from that of a sphere.

volume: A 3-D data set that typically contains either intensity information derived from the original MRI, or the results of segmenting this data into tissue classes.

voxel: The basic element of an MRI volume (analogous to a pixel in a 2-D image). The volume of a structural voxel is approximately 1 mm^3.

white surface: The refined estimate of the gray/white boundary. This surface can be viewed using surfer.

wm volume: The brain volume after white matter segmentation. This is also the volume that is manually edited. This volume can be viewed using tkmedit.

Contents

1. Techniques

Published material describing the techniques used in FreeSurfer (please cite our software!):

1.1. surface reconstruction

*

Cortical Surface-Based Analysis I: Segmentation and Surface Reconstruction, Dale, A.M., Fischl, B., Sereno, M.I., (1999). NeuroImage 9(2):179-194

Cortical Surface-Based Analysis II: Inflation, Flattening, and a Surface-Based Coordinate System, Fischl, B., Sereno, M.I., Dale, A.M., (1999). NeuroImage, 9(2):195-207.

Automated Manifold Surgery: Constructing Geometrically Accurate and Topologically Correct Models of the Human Cerebral Cortex, Fischl, B., Liu, A. and Dale, A.M., (2001). IEEE Transactions on Medical Imaging, 20(1):70-80.

Improved localization of cortical activity by combining EEG and MEG with MRI cortical surface reconstruction: A linear approach, Dale, A.M., and Sereno, M.I. (1993). Journal of Cognitive Neuroscience 5:162-176.

A Hybrid Approach to the Skull-Stripping Problem in MRI, SÃ©gonne, F., Dale, A.M., Busa, E., Glessner, M., Salvolini, U., Hahn, H.K., Fischl, B. (2004). NeuroImage, 22:1160-1075.

*

Genetic Algorithm for the Topology Correction of Cortical Surfaces, F. Segonne, E. Grimson, B. Fischl (2005). IPMI pp393--405.

1.2. inter-subject spherical averaging

*

High-resolution inter-subject averaging and a coordinate system for the cortical surface, Fischl, B., Sereno, M.I., Tootell, R.B.H., and Dale, A.M., (1999). Human Brain Mapping, 8:272-284(1999).

1.3. thickness measurement

*

Measuring the Thickness of the Human Cerebral Cortex from Magnetic Resonance Images, Fischl, B., and Dale, A.M., (2000). Proceedings of the National Academy of Sciences, 97:11044-11049.

*

Reliability of MRI-derived measurements of human cerebral cortical thickness: The effects of field strength, scanner upgrade and manufacturer, X. Han, J. Jovichich, D. Salat, A. van der Kouwe, B. Quinn, S. Czanner, E. Busa, J. Pacheco, M. Albert, R. Killiany, P. Maguire, D. Rosas, N. Makris, A. Dale, B. Dickerson, and B. Fischl, (2006) NeuroImage, 32(1):180-194.

* see also below: Rosas et al. (histology) and Kuperberg et al. (MRI) were validation studies for the thickness measures

1.4. cortical parcellation

*

Automatically Parcellating the Human Cerebral Cortex, Fischl, B., A. van der Kouwe, C. Destrieux, E. Halgren, F. Segonne, D. Salat, E. Busa, L. Seidman, J. Goldstein, D. Kennedy, V. Caviness, N. Makris, B. Rosen, and A.M. Dale, (2004). Cerebral Cortex, 14:11-22.

An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest, Desikan, R.S., F. Segonne, B. Fischl, B.T. Quinn, B.C. Dickerson, D. Blacker, R.L. Buckner, A.M. Dale, R.P. Maguire, B.T. Hyman, M.S. Albert, and R.J. Killiany, (2006). NeuroImage 31(3):968-80.

*

Cortical Folding Patterns and Predicting Cytoarchitecture, Fischl, B., N. Rajendran, E. Busa, J. Augustinack, O. Hinds, B.T.T. Yeo, H. Mohlberg, K. Amunts, K. Zilles, (2007). Cerebral Cortex 2007, doi:10.1093/cercor/bhm225.

1.5. subcortical segmentation

*

Whole Brain Segmentation: Automated Labeling of Neuroanatomical Structures in the Human Brain, Fischl, B., D.H. Salat, E. Busa, M. Albert, M. Dieterich, C. Haselgrove, A. van der Kouwe, R. Killiany, S. Klaveness, A. Montillo, N. Makris, B. Rosen, and A.M. Dale, (2002). Neuron, 33:341-355.

*

Sequence-Independent Segmentation of Magnetic Resonance Images, Fischl, B., Salat, D.H., van der Kouwe, A.J.W., Makris, N., SÃ©gonne, F., and Dale, A.M. (2004) NeuroImage 23:S69-S84.

*

Atlas Renormalization for Improved Brain MR Image Segmentation Across Scanner Platforms, Han, X. and B. Fischl, (2007). IEEE Transactions on Medical Imaging, 26(4):479-486.

1.6. related methodological papers

*

Discriminative Analysis for Image-Based Studies, Golland, P., B. Fischl, M. Spiridon, N. Kanwisher, R.L. Buckner, M.E. Shenton, R. Kikinis, A. Dale, W.E.L. Grimson, (2002) In Proc. MICCAI'2002: Fifth International Conference on Medical Image Computing And Computer Assisted Intervention. LNCS 2488:508-515.

Permutation Tests for Classification: Towards Statistical Significance in Image-Based Studies, Golland P., and B. Fischl, (2003) Proc. IPMI2003: 18th International Conference on Information Processing in Medical Imaging.

Topology Correction of Subcortical Segmentation, Segonne, F., E. Grimson and B. Fischl, (2003) MICCAI 2003.

Reliability in Multi-Site Structural MRI Studies: Effects of Gradient Non-linearity Correction on Phantom and Human Data, Jovich J, Czanner S, Greve D, Haley E, Kouwe A, Gollub R, Kennedy D, Schmitt F, Brown G, MacFall J, Fischl B, Dale A., (2006) NeuroImage, 30(2): 436-443.

Geometrically Accurate Topology-Correction of Cortical Surfaces Using Nonseparating Loops, Segonne, F., J. Pacheco, B. Fischl, (2007) IEEE Transactions on Medical Imaging, 26(4):518-529.

Phase maps reveal cortical architecture, Bruce Fischl and Lawrence L. Wald, PNAS, July 10, 2007, vol. 104, no. 28.

*

Effects of Registration Regularization and Atlas Sharpness on Segmentation Accuracy, Yeo, B.T.T., M. Sabuncu, R. Desikan, B. Fischl, P. Golland (2007). Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI 2007).

*

What Data to Co-register for Computing Atlases, B.T.T. Yeo, M. Sabuncu, H. Mohlberg, K. Amunts, K. Zilles, P. Golland, B. Fischl (2007). Proceedings of the IEEE Computer Society Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA 2007).

*

A Technique for the Deidentification of Structural Brain MR Images, Bischoff-Grethe, A., I.B. Ozyurt, E. Busa, B.T. Quinn, C. Fennema-Notestine, C.P. Clark, S. Morris, M.W. Bondi, T.L. Jernigan, A.M. Dale, G.G. Brown, and B. Fischl (2007) Human Brain Mapping, volume 28, issue 9, pp 892-903.

An MRI-based method for measuring volume, thickness and surface area of entorhinal, perirhinal, and posterior parahippocampal cortex, Feczko, E., J.C. Augustinack, B. Fischl, B.C. Dickerson, (2007). Neurobiology of Aging, 2007 Sep 10.

*

Brain morphometry with multiecho MPRAGE, van der Kouwe, A.J.W., T. Benner, D.H. Salat, and B. Fischl (2008) NeuroImage 40 (2008) 559-569.

2. Citations

Published material citing Freesurfer:

2.1. cortical thickness

*

Regional and progressive thinning of the cortical ribbon in Huntington's disease, Rosas, H.D., A.K. Liu, S. Hersch, M. Glessner, R.J. Ferrante, D.H. Salat, A. van der Kouwe, B.G. Jenkins, A.M. Dale, and B. Fischl, (2002). Neurology, 58:695-701.

*

Regionally localized thinning of the cerebral cortex in schizophrenia, Kuperberg, G.R., M.R. Broome, P.K. McGure, A.S. David, M. Eddy, F. Ozawa, D. Goff, C. West, S.C.R. WIlliams, A.J.W. van der Kouwe, D.H. Salat, A.M. Dale, and B. Fischl, (2003). Archives of General Psychiatry, 60:878-888.

Focal thinning of the cerebral cortex in multiple sclerosis, Sailer, M., B. Fischl, D. Salat, C. Tempelmann, M. Schonfeld, E. Busa, N. Bodammer, H. Heinze, and A. Dale, (2003). Brain, 126(9):1734-1744. (See also: editorial)

*

Thinning of the Cerebral Cortex in Aging, D.H. Salat, R.L. Buckner, A.Z. Synder, R.S.R. Desikan, D.N. Greve, E. Busa, J.C. Morris, A.M. Dale, and B. Fischl, (2004). Cerebral Cortex, 14:721-730.

-

A magnetic resonance imaging study of cortical thickness in animal phobia, Rauch, S., C. Wright, B. Martis, E. Busa, K. McMullin, L. Shin, A. Dale, and B. Fischl, (2004). Biological Psychiatry, 55(9):946-952.

-

Differing neuropsychological and neuroanatomical correlates of abnormal reading in early-stage semantic dementia and dementia of the Alzheimer type, Gold, B.T., D.A. Balota, M.J. Cortes, S.D. Sergent-Marshall, A.Z. Snyder, D.H. Salat, B. Fischl, A.M. Dale, J.C. Morris, and R.L. Buckner, (2005). Neuropsychologia, 43: p. 833-846.

-

Thickness of ventromedial prefrontal cortex in humans is correlated with extinction memory, Milad, M.R., B.T. Quinn, R.K. Pitman, S.P. Orr, B. Fischl, and S.L. Rauch (2005). PNAS, 102(30):10706-10711.

-

Meditation experience is associated with increased cortical thickness, Lazar, S.W., C.E. Kerr, R.H. Wasserman, J.R. Gray, D.N. Greve, M.T. Treadway, M. McGarvey, B.T. Quinn, J.A. Dusek, H. Benson, S.L. Rauch, C.I. Moore, and B. Fischl, (2005). NeuroReport, 16(17): 1893-1897.

-

Selective increase of cortical thickness in high-performing elderly - structural indices of optimal cognitive aging, Fjell, A.M., K.B. Walhovd, I. Reinvang, A. Lundervold, D. Salat, B.T. Quinn, B. Fischl and A.M. Dale (2006). NeuroImage 29:984-994.

-

Regional cortical thickness matters in recall after months more than minutes, Walhovd, K.B., A.M. Fjell, A.M. Dale, B. Fischl, B.T. Quinn, N. Makris, D. Salat, and I. Reinvan (2006). NeuroImage 31:1343 - 1351.

-

Mapping an intrinsic MR property of gray matter in auditory cortex of living humans: A possible marker for primary cortex and hemispheric differences, Sigalovsky, I.S., B. Fischl, and J.R. Melchera (2006). NeuroImage 32:1524-1537.

-

Anatomical differences in the mirror neuron system and social cognition network in autism, Hadjikhani N., R.M. Joseph, J. Snyder, H. Tager-Flusberg (2006). Cerebral Cortex 16(9):1276-82.

-

Different regional patterns of cortical thinning in Alzheimer's disease and frontotemporal dementia, Du, AT, N. Schuff, J.H. Kramer, H.J. Rosen, M.L. Gorno-Tempini, K. Rankin, B.L. Miller and M.W. Weiner (2007). Brain 130:1159-1166.

-

233

Differential effects of aging and Alzheimer's disease on medial temporal lobe cortical thickness and surface area, Dickerson B.C, E. Feczko, J.C. Augustinack, J. Pacheco, J.C. Morris, B. Fischl, R.L. Buckner (2007). Neurobiology of Aging, 2007 Sep 13.

-

Thickening in the somatosensory cortex of patients with migraine, DaSilva, A.F.M., C. Granziera, J. Snyder and N. Hadjikhani (2007). Neurology 69:1990-1995.

-

Detection of cortical thickness correlates of cognitive performance: Reliability across MRI scan sessions, scanners, and field strengths, Dickerson, B.C., E. Fenstermacher, D.H. Salat, D.A. Wolk, R.P. Maguire, R. Desikan, J. Pacheco, B.T. Quinn, A. Van der Kouwe, D.N. Greve, D. Blacker, M.S. Albert, R.J. Killiany, and B. Fischl (2008). NeuroImage 39(1):10-8.

-

Regional neocortical thinning in mesial temporal lobe epilepsy, McDonald C.R., D.J. Hagler Jr., M.E. Ahmadi, E. Tecoma, V. Iragui, L. Gharapetian, A.M. Dale, E. Halgren (2008). Epilepsia. 2008 Feb 7.

-

Cerebral cortex and the clinical expression of Huntington's disease: complexity and heterogeneity, Rosas, D.H., D.H. Salat, S.Y. Lee, A.K. Zaleta, V. Pappu, B. Fischl, D. Greve, N. Hevelone and S.M. Hersch (2008). Brain doi:10.1093/brain/awn025.

-

Regional neocortical thinning in mesial temporal lobe epilepsy, McDonald, C.R., D.J. Hagler, Jr., M.E. Ahmadi, E. Tecoma, V. Iragui, L. Gharapetian, A.M. Dale, and E. Halgren (2008). Epilepsia 48 (2), 211-219.

2.2. volumetric

-

Age does not increase rate of forgetting over weeks neuroanatomical volumes and visual memory across the adult life-span, Fjell, A.M., K.B. Walhovd, I. Reinvang, A. Lundervold, A.M. Dale, B.T. Quinn, N. Makris, and B. Fischl, (2005). Journal of the International Neuropsychological Society, 2-15(1):2-15.

-

Cortical volume and speed-of-processing are complementary in prediction of performance intelligence, Walhovd, K.B., A.M Fjell, I. Reinvang, A. Lundervold, B. Fischl, B.T. Quinn, N. Makris, and A.M. Dale, (2005). Neuropsychologia, 43(5):704-713.

-

Effects of age on volumes of cortex, white matter and subcortical structures, Walhovd, K.B., A.M. Fjell, I. Reinvang, A. Lundervold, A.M. Dale, D.E. Eilertsen, B.T. Quinn, D. Salat, N. Makris, and B. Fischl (2005). Neurobiology of Aging 26:1261-1270. See also: Commentary: Neuroanatomical aging: Universal but not uniform.

-

234

The functional and structural significance of the frontal shift in the old/new ERP effect, Walhovd, K.B., A.M. Fjell, I. Reinvang, A. Lundervold, B. Fischl, B.T. Quinn, N. Makris, and A.M. Dale, (2006). Brain Research, 1081:156-170.

-

Detailed semiautomated MRI based morphometry of the neonatal brain: Preliminary results, Nishida, M., N. Makris, D.N. Kennedy, M. Vangel, B. Fischl, K.S. Krishnamoorthy, V.S. Caviness, and P.E. Grant (2007). NeuroImage 32:1041-1049.

-

Volumetric cerebral characteristics of children exposed to opiates and other substances in utero, Walhovd, K.B., V. Moe, K. Slinning, P. Due-TÃ¸nnessen, A. BjÃ¸rnerud, A.M. Dale, A. van der Kouwe, B.T. Quinn, B. Kosofsky, D. Greve, and B. Fischl, (2007). NeuroImage 36:1331-1344.

-

Feasibility of Multi-site Clinical Structural Neuroimaging Studies of Aging Using Legacy Data, Fennema-Notestine, C., A.C. Gamst, B.T. Quinn, J. Pacheco, T.L. Jernigan, L. Thal, R. Buckner, R. Killiany, D. Blacker, A.M. Dale, B. Fischl, B. Dickerson, R.L. Gollub, (2007). Neuroinformatics 5:235-245.

-

The Effect of Early Human Diet on Caudate Volumes and IQ, Isaacs, E.B., D.G. Gadian, S. Sabatini, W.K. Chong, B.T. Quinn, B.R. Fischl, and A. Lucas, (2008). Pediatric Research 63(3):308-314. (see also: commentary)

-

Subcortical and cerebellar atrophy in mesial temporal lobe epilepsy revealed by automatic segmentation, McDonald, C.R., D.J. Hagler, Jr., M.E. Ahmadi, E. Tecoma, V. Iragui, A.M. Dale, and E. Halgren (2008). Epilepsy Researc 2008 Mar 20.

2.3. entorhinal cortex

-

Detection of entorhinal layer II using 7 Tesla magnetic resonance imaging, Augustinack, Jean C., Andre J. W. van der Kouwe, Megan L. Blackwell, David H. Salat, Christopher J. Wiggins, Matthew P. Frosch, Graham C. Wiggins, Andreas Potthast, Lawrence L. Wald, Bruce R. Fischl, (2005). Annals of Neurology, 57(4):489-494.

2.4. cortical surface shape and folding analysis

-

Cortical Surface Shape Analysis Based on Spherical Wavelets, Yu, P., P.E. Grant, Y. Qi, X. Han, F. SÃ©gonne, R. Pienaar, E. Busa, J. Pacheco, N. Makris, R.L. Buckner, P. Golland, and B. Fischl (2007). IEEE Transactions on Medical Imaging, 26(4):582-597.

235

Cortical folding development study based on over-complete spherical wavelets Yu, P., B.T.T. Yeo, P.E. Grant, B. Fischl, P. Golland (2007). Proceedings of the IEEE International Conference on Computer Vision (MMBIA 2007).

-

Abnormal cortical folding patterns within Broca's area in schizophrenia: Evidence from structural MRI, Wisco, J.J., G. Kuperberg, D. Manoach, B.T. Quinn, E. Busa, B. Fischl, S. Heckers, A.G. Sorensen (2007). Schizophrenia Research, 94(1-3):317-327.

-

Accurate prediction of V1 location from cortical folds in a surface coordinate system, Hinds, O.P., N. Rajendran, J.R. Polimeni, J.C. Augustinack, G. Wiggins, L.L. Wald, H.D. Rosas, A. Potthast, E.L. Schwartz and B. Fischl (2007). NeuroImage 2007.10.033.

2.5. surface-based analysis

-

A human parietal face area contains aligned head-centered visual and tactile maps, Sereno, M.I. and RS Huang, (2006). Nature Neuroscience 9:1337-1343.

-

Enhanced Cortical Activation in the Contralesional Hemisphere of Chronic Stroke Patients in Response to Motor Skill Challenge, Schaechter, J.D., and K.L. Perdue (2007). Cerebral Cortex, 2007 Jun 29 [Epub ahead of print].

2.6. cortical flat-patches

-

Attentional modulation of human auditory cortex, Petkov, C.I., X. Kang, K. Alho, O. Bertrand, W.E. Yund, and D.L. Woods, (2004). Nature Neuroscience 7:658-663.

2.7. FS-FAST

-

Faces and objects in macaque cerebral cortex, Tsao, D.Y., W.A. Freiwald, T.A. Knutsen, J.B. Mandeville and R.B.H. Tootell, (2003). Nature Neuroscience 6:989-995.

2008-06-01 20:45

236

FsTutorial Snapshot

Snapshots are periodically made of the FreeSurfer Tutorial, for those who want a print version, or a static copy of the HTML files. Bear in mind that it is possible for the snapshots to be out-of-date with the wiki-based source files, which undergo small improvements. If an updated snapshot is desired, send a request to the freesurfer mailing list
`<freesurfer AT nmr DOT mgh DOT harvard DOT edu>`.

Here is the tutorial in PDF format:

-
    FreeSurferTutorial-2008-05-31_N.pdf (single page format)
-
    FreeSurferTutorial-2008-05-31_N_4pp.pdf (four-per-page format)

Here is a static copy (html format) of the tutorial:

-
    FreeSurferTutorial-2008-05-31-html.tar.gz

To install the static copy, type:

`tar zxvf FreeSurferTutorial-2008-05-31-html.tar.gz`

and look for the file `index.html` in the `FsTutorial` directory.

Note: The output from the snapshot tools is sub-optimal (poor formating, lack of page numbers, etc.), so apologies for this!

NMR Center internal: how to create a new snapshot

2008-06-01 20:47

Acknowledgments

Contributors to the creation of FsTutorial:

BWH:

- Wendy Plesniak
- Nicole Aucoin
- Kathryn Hayes
- Haiying Liu
- Steve Pieper

MGH:

- Bruce Fischl
- Evelina Busa
- Doug Greve
- Jenni Pacheco (special thanks to Jenni for all the hard work!)
- Brian Quinn
- Nick Schmansky
- Kevin Teich

2008-06-01 20:47

BWH:

- Wendy Plesniak