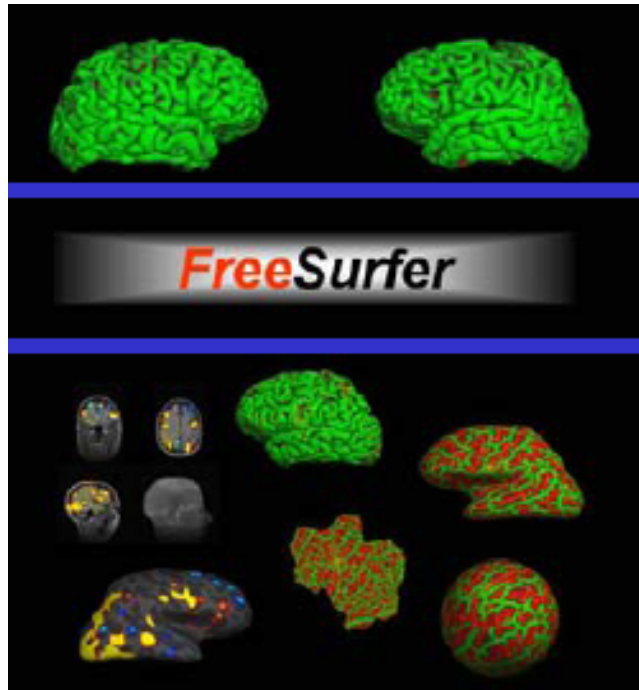# FREESURFER
Processing & editing overview

Prepared by the Wig Neuroimaging Laboratory

Neil K. Savalia, Phillip F. Agres, Micaela Y. Chan, Claudia Carreno, Linh T. Nguyen & Gagan S. Wig

April 2023 (v2.1); August 2015 (v1)

https://www.wigneurolab.org/resources

TABLE OF CONTENTS

2023 updates from previous version of this manual:

This manual is an update to the original manual that was referenced in Savalia et al. (2017) *Human Brain Mapping* (**FREESURFER Processing & editing overview – prepared by the Cognitive Neuroimaging Laboratory**; October 2015). The updates include reducing lab-specific settings, additional details regarding cerebellum edits, and some updates based on newer versions of FreeSurfer/FreeView.

I. INTRODUCTION TO FREESURFER

FreeSurfer is an open-source software package comprising automated tools for reconstruction of the brain's cortical surface and anatomical segmentation of brain structures from structural MRI data. FreeSurfer tools are developed by Bruce Fischl, Doug Greve, and others at the Martinos Center for Biomedial Imaging.

FreeSurfer can be run from a computer/server equipped with Linux or Mac OS X operating systems (or a Windows computer via VirtualBox). The operations that FreeSurfer runs are resource-intense, consuming large amounts of processor time, memory and disk space. Researchers should run FreeSurfer from the most powerful machine available to them, and consistently use the same machine for processing a given study.

FreeSurfer allows researchers to create robust two-dimensional surface renderings of individuals' cortical surfaces, which allow highly accurate inter-subject registration and analyses of cortical folding. FreeSurfer will also quantify a variety of morphometric brain measures such as whole brain volume, cortical thickness (global and regional), and white matter or ventricular volume. For instance, many researchers employ FreeSurfer tools to obtain sub-cortical regional volume and thickness estimates (e.g., hippocampal volume, frontal cortex thickness).

Given these features, FreeSurfer is a very useful research tool for guiding analysis of brain structure (and as a consequence, functional localization). Depending on the researcher's questions, however, it is imperative to verify the quality of FreeSurfer's automated pipelines. In some instances, users must impose manual intervention or specific alterations to the processing pipeline to obtain clean estimates of the data of interest.

This document is meant to summarize the FreeSurfer processing pipeline for automated anatomical segmentation and surface generation. From there, a guide for checking and editing FreeSurfer outputs is outlined. The instructions and tips are devised primarily for users processing FreeSurfer data in Wig Neuroimaging Laboratory (Wiglab; https://www.wigneurolab.org/) environment. However, many of the concepts introduced here should be usable to others.

For extra information/questions left unanswered, please check these very helpful resources:
- Martinos Center for Biomedical Engineering: https://www.martinos.org/
- FreeSurfer Wiki: https://freesurfer.net/fswiki
- FreeSurfer Mailing List: https://surfer.nmr.mgh.harvard.edu/fswiki/FreeSurferSupport
- FreeSurfer YouTube Channel: https://www.youtube.com/@freesurfer7634

How to use this manual:

Skip to section IV if you need a refresher on how to do manual edits. Section III goes over common failures that can be fixed using techniques shown in Section IV.

In the rest of this document, command line scripts that should be typed into the terminal or written in a bash script will be written with `courier font in blue`, and enclosed in square brackets [] if its in-line with other text (e.g., [`echo hello world`])

II. RUNNING FREESURFER: AUTOMATED RECONSTRUCTIONS

One of the central FreeSurfer commands to familiarize with is '**recon-all**', a command line function that creates an automated cerebral reconstruction (a.k.a., "recon") of an input anatomical brain scan.

Recon-all outputs a variety of anatomical measures useful for multiple aspects of neuroimaging research (e.g., anatomical segmentations, cortical thickness values, brain volume estimates, tissue type masks and 2D surfaces). As such, the formation of highly accurate reconstructions is a fundamental step in processing our data.

The remainder of this section will sketch out the way recon-all works, how to perform cortical reconstructions in the Wiglab environment, and some additional pointers. Keep in mind that this automated portion of FreeSurfer is just the *tip of the iceberg* of what using FreeSurfer involves…

II - A. Processing Flow

Recon-all functions by taking an input anatomical scan and running it through multiple FreeSurfer functions (see **Table 1**).

Typing "recon-all" into the command line when logged into Golgi (our data processing server, running Linux) will return an overview of the command and its arguments, including some options that are beyond the scope of this tutorial. See *Expert Techniques* or look up the recon-all Dev table on the FreeSurfer wiki for further information

TABLE 1 – RECON-ALL PROCESSING STAGES SUMMARY

autorecon1
1. Motion correction & conforming
2. Talairach transformation computation
3. Non-parametric Non-uniform Intensity Normalization (N3)
4. Intensity Normalization 1
5. Skull Stripping

autorecon2
6. EM (GCA) Registration
7. CA Intensity Normalize
8. CA Non-linear Registration
9. CA Label of Subcortical Structures
10. Intensity Normalization 2 **(start here for control points edits [`-autorecon2-cp`)**
11. Mask Brain Final Surface
12. White Matter Segmentation
13. Cut/Fill with aseg **(start here for wm-edits [`-autorecon2-wm`])**
14. Tessellation (begins per-hemisphere operations)
15. Smooth 1
16. Inflate 1
17. Quick Sphere Unfolding
18. Automatic Topology Fixer
19. Make WM and Cortical Surfaces **(start here for pial edits [`-autorecon-pial`])**
20. Smooth 2
21. Inflate 2
22. Make Curvature (Gaussian and Mean)
23. Make Curvature Stats

Checking autorecon2 output surfaces:
a) If control points added, run -autorecon2-cp (10-23)
b) If wm.mgz edited, run -autorecon2-wm (15-23)
c) If correcting pial surface, run -autorecon2-pial (19-23)

autorecon3
24. Spherical Mapping
25. Spherical Registration
26. Jacobian White Surface Distortion Calculation
27. Map Average Curvature to Subject
28. Cortical Parcellation 1 - Desikan/Killiany atlas
29. Pial Surface Generation
30. Cortical Ribbon Mask
31. Cortical Parcellation Statistics
32. Cortical Parcellation 2 – Destrieux atlas
33. Cortical Parcellation Statistics 2
34. Cortical Parcellation 3 – DTK atlas
35. Cortical Parcellation Statistics 3
36. Computes surface-wise gray/white contrast
37. Relabel Hypointensities (volumetric)
38. Add Cortical Parcellation Mapping to ASEG
39. Compute Statistics of Segmentations
40. Add WM Parcellation to ASEG + Stats
41. Add Broadman Area Labels

This table is based on FreeSurfer 6.0 recon-all table (see other version table on fswiki):
https://surfer.nmr.mgh.harvard.edu/fswiki/ReconAllTableStableV6.0

(https://surfer.nmr.mgh.harvard.edu/fswiki/ReconAllDevTable).

Performing the full, automated reconstruction is generally done by typing the recon-all function and various arguments into the Terminal command line.

```
$ recon-all -i <input> -all -sd <subjdir> -s <subjID>]
```

II - B. Wiglab Method & Output Structure

When operating in the Wiglab environment (e.g., using Golgi), `recon-all` is run as part of a conserved processing stream imbedded in our overall MRI preprocessing method (i.e., a switch for performing FS reconstruction is set to true/false in subject parameter files).

To familiarize yourself with the FreeSurfer output structure, check out the FreeSurfer's example subject's data in *$FREESURFER_HOME/subjects/bert* and review output data laid out on FreeSurfer's wiki (https://freesurfer.net/fswiki/FsTutorial/OutputData_freeview). The example subject, bert, is also a good benchmark for what are considered 'good' outputs.

II - C. Other

The steps of recon-all can be run in stages (e.g., the different phases of autorecon), or even step-wise (i.e., for careful control). We suggest running the commands in three pre-defined stages (see 'autorecon' annotations in Table 1) for data that is thought to be prone to FreeSurfer errors (e.g., input anatomical scan of low quality, poor grey/white matter contrast, presence of anatomical abnormalities, possible head-movement suspected during the scan based on data flagging; see Savalia et al. 2017 [doi: 10.1002/hbm.23397]).

The autorecon portions of recon-all can be performed from the Terminal command line when logged into Golgi. Do this by submitting it to the SGE/Slurm cluster or use a detached persistent terminal (e.g., Tmux)/remote desktops (e.g., VNC/X2go). Don't just run it through an *ssh* session on the terminal in case the ssh tunnel is unintentionally terminated:

1. Set the *SUBJECTS_DIR* variable to the 'reconall' folder in a subject's preprocessed directory (by default it is set to [*/usr/local/freesurfer/subjects*]).

2. Call recon-all using the specific 'autorecon' directive desired, e.g.:
   `$ recon-all -i <input> -autorecon1 -sd $SUBJECTS_DIR -s <subjID>`

The remainder of this tutorial will continue as if FreeSurfer's recon-all was run as part of Wiglab's processing pipeline. However, the following steps cover information that is useful for almost any method of running recon-all. Specifically, the information related to **reviewing your recon, trouble-shooting issues and manual editing techniques** *will be useful for any individual who wants to further their understanding of FreeSurfer. As you continue learning, please feel free to contribute to refining this manual for fellow users.*

III. REVIEWING & TROUBLESHOOTING

III - A. Hard Failures

Hard failures in recon-all involve the process flow quitting out before finishing. These sorts of failures are infrequent, but nonetheless can occur at any point. These sorts of failures will occur predominantly for two reasons:

(1) <u>A computer issue</u>: lack of disk space/RAM, improper file permissions, invalid directory paths, a broken pipe, or another means of unintentional user error. These issues are easily diagnosed and resolved by back tracking over your input commands and digging through **recon-all.log** or **reconall.error** files that are dropped in the subject's output *scripts/* folder.

(2) <u>Data quality issue</u>: artifacts, heavy movement during anatomical T1 scan, large pathologies, etc. that lead to recon-all aborting processing mid-stream (e.g., 'Automatic Topology Fixer' cannot reconcile massive surface failures, etc.). Diagnose these sorts of issues by checking through the subject's **reconall.log** and **reconall.error** files to localize the error (in *scripts/* folder), verifying the quality of input images and images from the last successful step (in *mri/* folder), and re-running the failing step with potentially new/more appropriate options. See Savalia et al., 2017 for approaches to assessing T1 data quality.

As the above makes clear, these problems can require digging through recon-all log and error files as well as decoding unusual problems with FreeSurfer. For instance, the outputs in your participant's output directory look incomplete you may decide to check the last few lines of a recon-all.log file to determine what steps occurred last. For a participant you have run recon-all on just once, you can check the last 25 lines of their log file: [*$: tail -25 recon-all.log*]. With a participant you have recon-all on several times (e.g., multiple iterations of autorecon), you may wish to scan the log file for every instance of "exited with ERRORS" and include the 10 lines before and 10 lines after a match: [*$: cat recon-all.log | grep -B 10 -A 10 "exited with ERRORS"*].

Some solutions to hard failures require customized or step-wise recon-all command flow procedures. Therefore, navigating these issues requires a working knowledge of the FreeSurfer process flow and a capacity for navigating the Terminal window/Golgi.

Do not hesitate to ask for help if you are having trouble: approach someone with a bit more experience or send a message to the experts of the FreeSurfer mailing list!

III - B. Soft Failures

Soft failures occur when recon-all finishes processing your data but requires modifications or manual intervention. These errors tend to occur most often with poor input anatomical scans (e.g., high in-scanner movement; see Reuter et al., (2015) and Savalia et al., (2017) for helpful discussion of motion artifacts in FreeSurfer) and/or with variation in brain structure significantly beyond that observed in healthy young-adulthood (e.g., age-related atrophy, structural abnormalities, etc.).

Importantly, since recon-all runs to completion in the case of soft failures, the subject's *scripts/* directory will contain a **reconall.done** file and the **recon-all.log** file will end with a line mentioning that processing "*finished without errors*". However, script completion and an absence of automatically detected errors can be misleading.

As such, it is vital to follow a number of meticulous steps (see *Section III. C*) and vigilantly check every subject's recon-all outputs. Soft failures can be attributable to many sources, but they usually include one or more of the following:

(1) <u>Skull-stripping issue</u>: a failure to properly mask out the non-brain portions of an input anatomical T1 scan can lead to inaccuracy in FreeSurfer processing. Skull stripping may be overly aggressive or may be not aggressive enough, depending on the nature of the input anatomical scan (see examples in **Fig 1**).
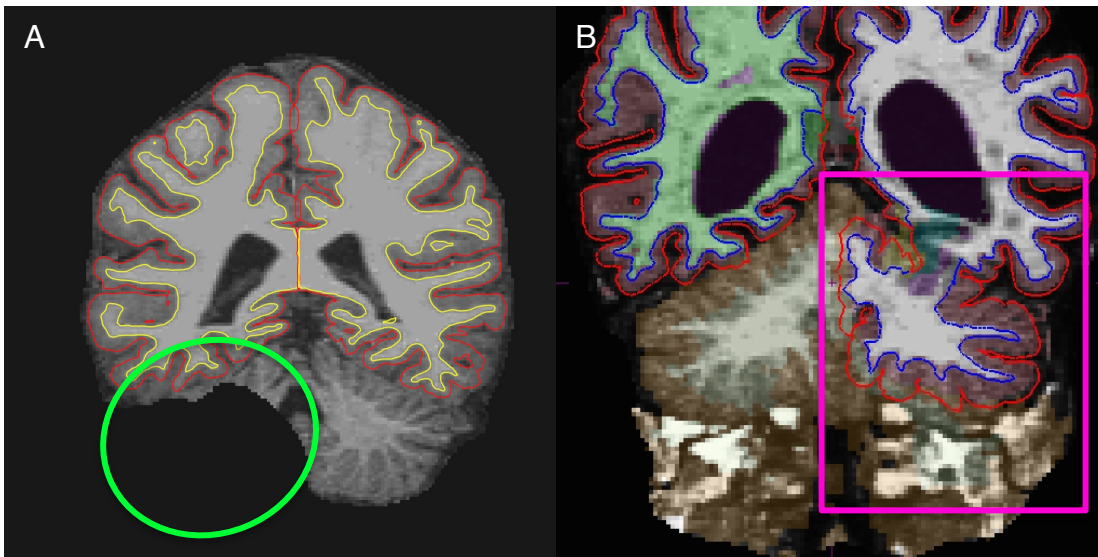

Fig 1. Downstream problems resulting from skull strip errors. *Skull stripping issues will often result in relatively poorer intensity normalization and anatomical segmentation. Shown above are: (A) an example of overly aggressive skull stripping, which cuts off portions of the cerebellum (green oval), and (B) an example of skull stripping that does not remove enough tissue beneath the brain. The colored overlays in (B) show a downstream problem where anatomical labels are attributed to non-brain structures and where cortical surfaces include regions of cerebellum (pink box).*

(2) <u>Intensity normalization issue</u>: the intensity normalization steps help downstream FreeSurfer functions distinguish between grey and white matter tissue, but are very dependent on a input T1 image that contains sufficient grey/white contrast. Inconsistent and erratic intensity normalization of anatomical T1 scans can lead to faulty tissue-type classifications, and substantial inaccuracies in segmentation and anatomical surface generation (**Fig 2**).
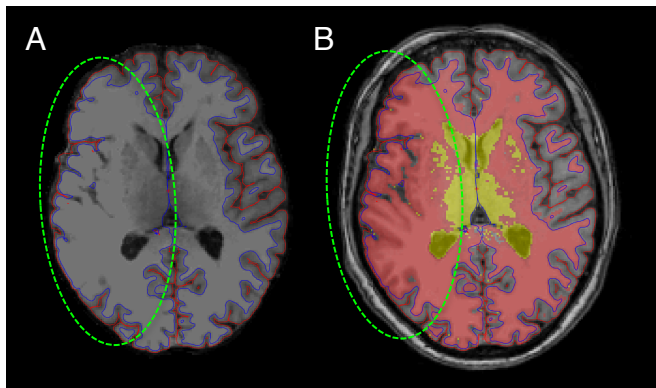

Fig 2. Poor intensity normalization gives faulty white matter segmentation and surface generation. *The images to the left depict instances where (A) poor and inconsistent intensity normalization severely muddles the distinction between grey and white matter (green circles). This leads to (B) anomalous white matter segmentation (heat map) and ultimately poor surface generation in the regions with poor intensity normalization.*

(3) <u>Segmentation issue</u>: FreeSurfer may inaccurately segment the input anatomical scan for various reasons alluded to above. This can lead to faulty output label files, inaccurate extraction of cerebral grey matter, and usually a flawed surface mapping of the input anatomy (refer to **Fig 1B** for an example of this).

(4) <u>Pial surface issue</u>: pial surfaces are generated to help determine the outer extent of grey matter present in a T1 anatomical scan. Misplaced pial surfaces will lead to inaccurate surface mapping and poor-quality curvature, gray-matter thickness and volumetric segmentation statistics (**Fig 3**).
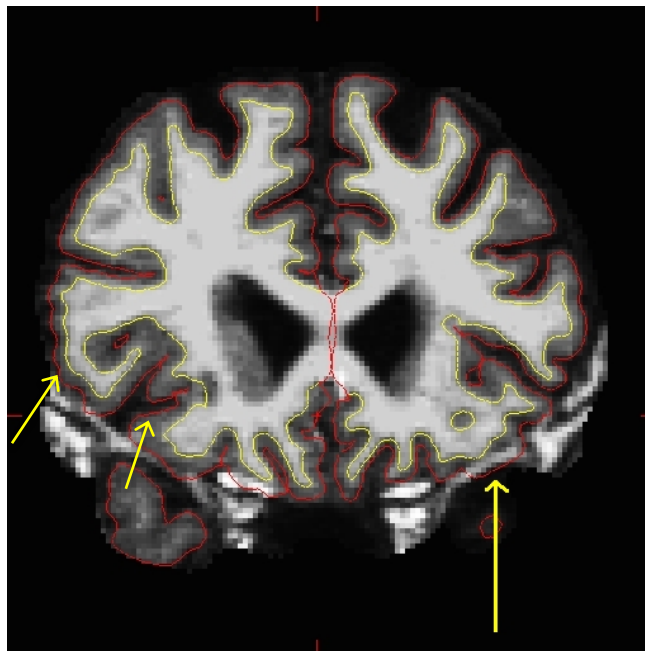

Fig 3. Pial surface includes non-grey matter locations. *Pial surfaces may include locations that do not represent grey matter signal (yellow arrows).*

(5) Topological issue: a subject's anatomical topology gets mapped from the 3D volumetric space (T1 image) to a two-dimensional surface, but often results in surface defects (e.g., 'bridges', 'holes'). While almost all defects are resolved with recon-all's 'Automatic Topology Fixer', some may be too large for the automated processing to resolve. Persistent defects will give inaccurate anatomical surfaces (**Fig 4**) and poor group registration of functional or anatomical data.
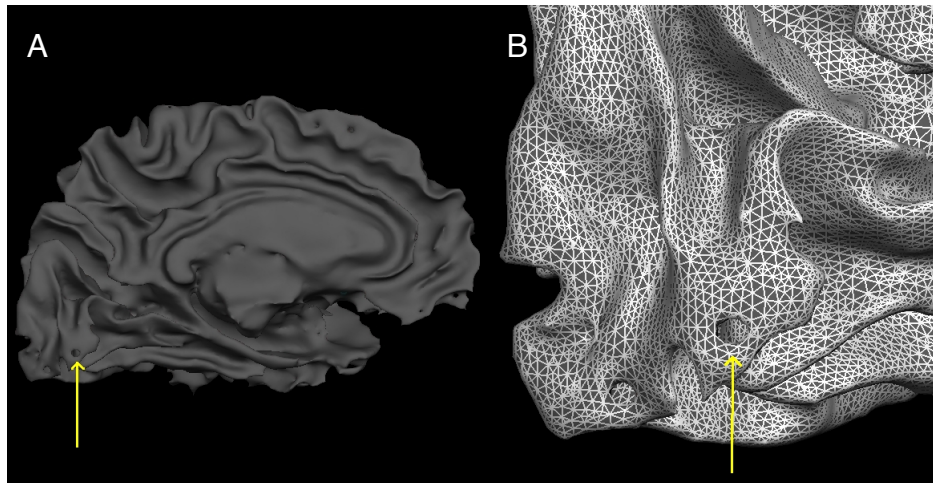

Fig 4. Surface defects. Though FreeSurfer's automatic topology fixer is fairly robust, (A) some large errors in surface generation can persist as anomalous 'bridges' or 'holes' (yellow arrows). The right-side image (B) shows the zoomed in surface mesh that clearly misses a region of occipital cortex that downstream surface-based analyses could easily gloss over.

A common strand in each of the soft failures is that an issue allows complete generation of volumetric segmentation and surface mapping, but that the results are faulty in some way. For both of these cases, fixes are implemented by imposing manual edits to a set of volumetric images (see *Section IV*). Even to fix output surfaces only, quality control is imposed at the volume level followed by a regeneration of (hopefully) more correct outputs.

Our lab has encountered a few instances of large soft failures that have been ameliorated by simply reprocessing an individual's data through the entire recon-all command. If you see something drastically wrong with your recon that cannot be pinpointed, it's not a bad idea to give FreeSurfer a second shot at it.

III - C. Reviewing Your Recon

As mentioned above, each recon needs a thorough check to assure good output quality. For inspecting an individual's data in the Wiglab environment, the following steps are also meant to diagnose soft failures and construct a customized procedure for manual intervention on that individual's data. At minimum, these are the passive checks that we recommend:

(1) Investigate outputs in scripts folder: The files in this folder are helpful for checking various aspects of a recon job and can be traversed with a few simple entries to the Terminal command line. While we mentioned the last line of the log file will indicate whether there were any process-stopping errors in recon-all, it is useful to know how to spot check a couple pieces:

   a. Recon-all errors – "did recon-all finish without automatically detected issues?"

To quickly check whether the input subject finished without errors, run the following: [`$: tail -1 reconall.log`]. The reconall.log file is also an information-rich output that is very useful in trying to pinpoint both soft & hard failures.

b. Recon-all completion – "how far along is my recon?"
With preprocessing and FreeSurfer jobs being sent to the server grid, you may want to monitor the progress of your recon. If your recon is processing, a file with the extension ".IsRunning" will be present in the subject's `scripts/` folder, otherwise the reconall.done file will be present. To check recon-all progress, users can look at the reconall.cmd file, which shows all the commands that FreeSurfer has run on the input data (e.g., [`$: cat reconall.cmd`]).

c. Defect corrections – "are there any massive surface defects?" (optional)
It can be helpful to get a sense of how troublesome a recon job is for the automated steps to run through. One way to do this is to check how many defects the 'Automatic Topology Fixer' has to correct, and how large they are. Running the following: [`$: cat reconall.log | grep "DEFECT"`] will list the defects present in surface generation, and their sizes in vertices ("Vertex Area" can be checked in log files, ~ .6mm^3). The presence of many defects with more than a few thousand vertices each is a good indicator that the volumetric segmentation may be rough and require close attention. With more experience navigating these log files, you'll be able to use this defect information to pinpoint specific zones to target in manual edits.

(2) <u>Check Talairach transform and edit if necessary</u>: In the more recent versions of FreeSurfer, the Talairach transform has become a pretty robust step in the recon-all processing stream. However, a quick check of the success of the Talairach transform can provide useful information on whether a subject may need closer attention.

To extract QC measures related to Talairach transforms, enter the following into the Terminal command line: [`$: cat reconall.log | grep "TalAviQA"`], two values will be returned. The first is the spatial correlation of the blurred input anatomical image and the Talairach atlas representative target image ("eta"; values above 0.96 generally considered passable). The second number ("atlas_transform_error"; values < 15 are generally considered passable) is computed on the basis of registration objective function curvature in 12-parameter space and compared to values derived from a training set.

Finding values of 'eta' and 'atlas_transform_error' outside of the normal bounds may indicate that the subject's transform has a major issue that needs fixing (e.g., eta << 0.96, atlas_transform_error >> 15; see *Section IV.D1*), or a subject whose FreeSurfer outputs will require especially careful attention (e.g., anatomy just does not line up super well with the target atlas). However, beware that these numbers may appear more disconcerting than the data they reflect when dealing with datasets with individuals advanced in age or clinical populations. Edits to the Talairach transform can be performed by following the steps outlined on the FreeSurfer wiki: https://surfer.nmr.mgh.harvard.edu/fswiki/FsTutorial/Talairach_freeview

(3) <u>Check various volumetric and surface outputs</u>: Arguably one of the most important steps is reviewing your recon; the manner in which you approach checking your FreeSurfer outputs can make or break your final data quality. We recommend taking a thorough look at output volumes and surfaces to guide your overall editing process.

Typing the following lines into the Terminal window when logged into Golgi will open up the Freeview application and display several of the volumetric and surface outputs in a recommended arrangement (**Figs 5-7**):

```
$: sub=<subid>
$: SUBJECTS_DIR=<Study_Dir>/"$sub"/fs/reconall/"$sub"
$: freeview -v "$SUBJECTS_DIR"/mri/aparc+aseg.mgz:colormap=lut:opacity=0.75
 "$SUBJECTS_DIR"/mri/T1.mgz "$SUBJECTS_DIR"/mri/brainmask.mgz
 "$SUBJECTS_DIR"/mri/brain.finalsurfs.mgz
 "$SUBJECTS_DIR"/mri/wm.mgz:colormap=heat:opacity=0.4 -f
 "$SUBJECTS_DIR"/surf/lh.pial:edgecolor=red:edgethickness=1
 "$SUBJECTS_DIR"/surf/rh.pial:edgecolor=red:edgethickness=1
 "$SUBJECTS_DIR"/surf/lh.white:edgecolor=blue:edgethickness=1
 "$SUBJECTS_DIR"/surf/rh.white:edgecolor=blue:edgethickness=1
```
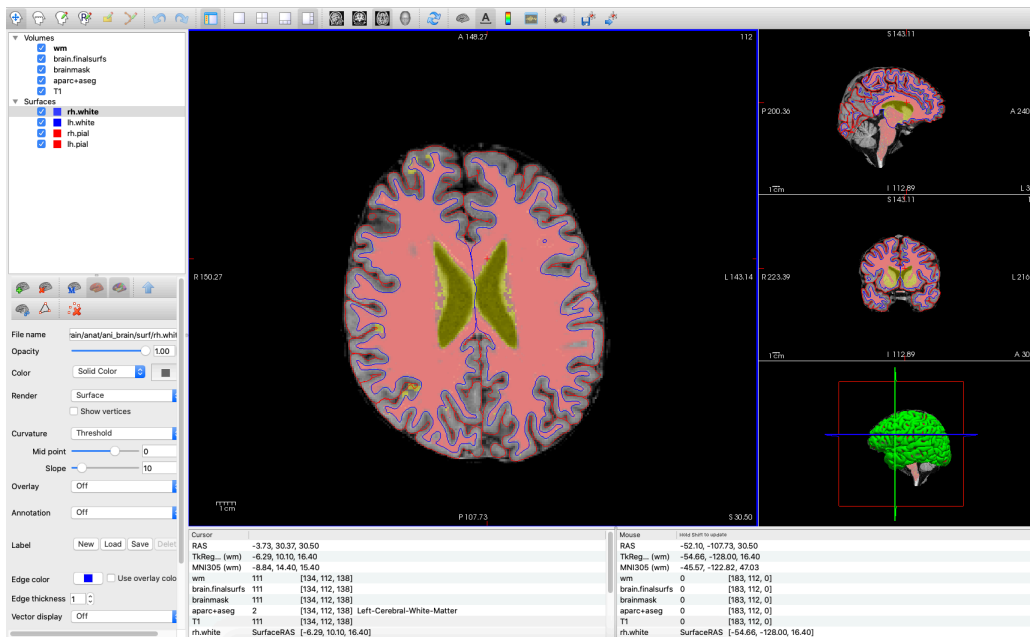

Fig 5. Freeview window using command listed in III.C3. The window displays the input anatomical image (T1.mgz), a skull-stripped output (brainmask.mgz), a white matter volume parcellation (wm.mgz), a volume-based anatomical parcellation (aparc+aseg.mgz), the intensity-normalized image that is input to surfaces generation (brain.finalsurfs.mgz), and white (blue outlines) and pial (red outlines) surface renderings for both hemispheres.

To check the loaded images, flip through each slice of the volumetric outputs (the axial slice is very helpful, but check other orientations as well). Keep in mind that *you are viewing a three-dimensional structure using a two-dimensional picture.* It is often helpful to scroll through a couple slices quickly to gain an idea of what the brain and the surfaces being mapped to the brain are doing. Something may look weird on one slice but scrolling through the next couple you realize that the particular image is correct but just happened to look weird in that one orientation.

Scroll through your images looking for locations properly mapped onto your anatomical data in the input T1.mgz file (**Fig 6A**). First, check whether the brainmask.mgz (**Fig 6C**) adequately removes the skull and other non-brain tissue in all locations. Problems with skull stripping will often require manual edits (see *Section IV.C*). Second, check how well the white matter parcellation (wm.mgz; **Fig 5**, red/yellow overlay and **Fig 6B**) labels white matter tissue. For the purposes of FreeSurfer, the wm.mgz file should be thought of as a mask containing everything interior the innermost layer of the grey matter (e.g., ventricles and subcortical structures get lumped into wm.mgz). The wm.mgz is prone to both false-positives (labeling cortical grey

matter as white) and false-negatives (excluding locations that are actually white matter), so pay close attention for instances of either that may require edits (see *Section IV.D*).

The aparc+aseg.mgz (**Fig 6D**) and brain.finalsurfs.mgz (similar to **Fig 6C**) files are somewhat less essential to check for our purposes, but they contain useful information. The aparc+aseg file contains a projected segmentation of anatomical structures across the brain, and can be useful for obtaining structurally specific volumetric data. The Wiglab uses the overall grey matter label in addition to white matter and cerebrospinal fluid labels that are generated here.
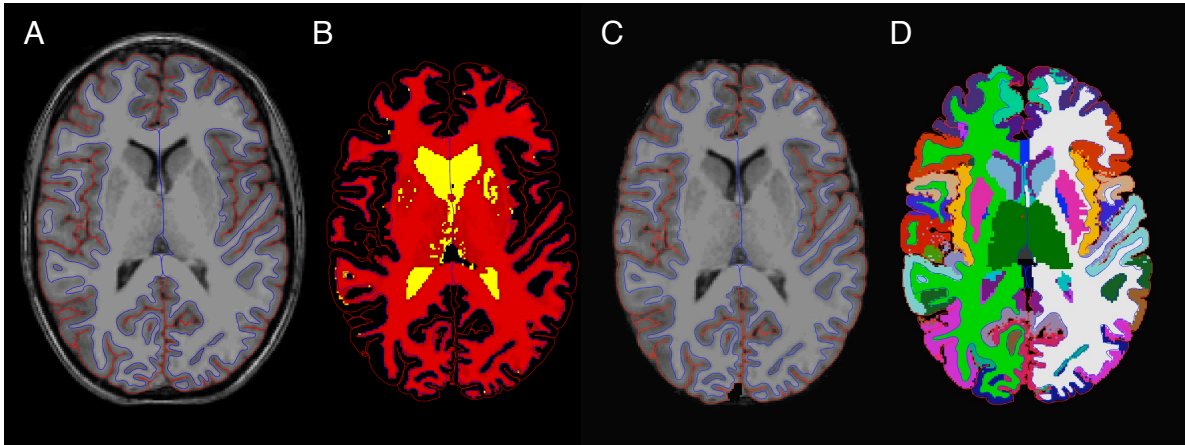
Fig 6. Important images to check in Freeview. The above panel displays examples of an (A) input anatomical image (T1.mgz), (B) a white matter segmentation (wm.mgz), (C) a skull-stripped anatomical image (brainmask.mgz; brain.finalsurfs.mgz is the same image with a second round of intensity normalization), (D) a volume-based anatomical segmentation (aparc+aseg.mgz). Each image depicts the final, corrected white (blue lines) and pial (red lines) surfaces for the input data.

The brain.finalsurfs.mgz file contains the immediate input for surface generation. Any downstream defects in the final surface outputs will result from this image, so assuring quality (e.g., brain extraction, good intensity normalization, etc.) and good fit with the wm.mgz are helpful if soft errors are persistent (see *Section VI.A*). The outlines of white matter and pial surfaces are projected along the volumetric images in blue and red outlines, respectively. The white matter surface represents the outer extent of the white matter volumetric parcellation, and the pial surface represents the outer extent of cortical grey matter. Defects that manifest at the surface level originate with problems in the volumetric images (i.e., usually brainmask.mgz and wm.mgz). Keep a special eye out for locations where the white surface outline is not aligned with the wm.mgz file (e.g., over-inclusions, exclusions), and locations where the pial surface is not flush with the outer face of cortical grey matter (e.g., includes skull tissue) as these will require edits. Three-dimensional renderings of the subject's surfaces can also be viewed in Freeview (see **Fig 7**) to check for 'holes' and 'bridges' that FreeSurfer did not correct, though this is better visualized with tksurfer (see *Section III.D2*).
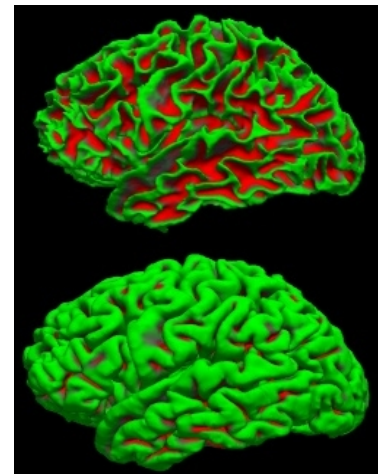

Fig 7. FreeSurfer lh.white (top) and lh.pial (bottom) surfaces. The above images are examples of the 3D renderings of two left hemisphere surfaces that FreeSurfer generates for individual subjects in recon-all.

(4) <u>Go make the necessary edits</u>: following instructions in *Section IV*, go forth in making the required edits for generating correct volumetric segmentation and surfaces.

*Researchers new to the editing process will find it extraordinarily difficult to finish their checking/editing within 20-30 minutes. Initially, it can be very tempting to **"over-edit".** However, for an effective editor there are vastly diminishing returns after this time window. It is absolutely necessary to build a level of tolerance for some of the errors that are present in automated segmentations. For instance, **when a surface defect appears in one slice but is resolved within one-to-two slices in any direction, that defect does not require intense user intervention.** Instead, work on catching and fixing the larger defects that introduce drastic errors into your final FreeSurfer outputs.*

In most instances, you will want to create a master study-specific documentation that lays out your editing process for a given participant. Ideally, researchers should first document any problems with FreeSurfer outputs–if any–from the automated recon (e.g., defective wm.mgz, brainmask.mgz or brain.finalsurfs.mgz, aparc+aseg.mgz, incorrectly generated white matter and pial surfaces). It is also good practice to note the edits that are being recommended during the first pass through the data.

Once edits have been implemented, researchers should re-check the individual's outputs as in *Section III.C3*, note whether the issues have been resolved, and make certain that additional problems have not come about.

IV. EDITING PROCESS

<u>Overview</u>

FreeSurfer edits can be completed using two software, *FreeView* and *TKMedit*. Currently, FreeView is the software primarily being updated, and we recommend doing most edits through FreeView. MGH provides a good resource on basic navigation around FreeView: https://surfer.nmr.mgh.harvard.edu/fswiki/FreeviewGuide/FreeviewGeneralUsage/FreeviewNavigation

The editing process is separated into (A) adding control points, (B) white-matter edits, and (C) pial edits. We recommend that A and B be done in FreeView for ease of operation and un-do functions. For pial-edits, some editors that was previously trained in TKMedit finds it faster to edit pial in the older software, so we provide the guide to both. The setup to do pial edit (editing brainmask.mgz), and white matter edit (editing wm.mgz) in FreeView is very similar so a brief overview on using FreeView for pial edit is at the end of the pial edit section.

IV - A. Adding Control Points

A commonly occurring issue with respect to white matter (WM) and grey matter (GM) segmentation is the labeling of WM segments as GM, or a section of GM (typically a portion of a gyrus) being excluded from the pial surface. FreeSurfer creates these surfaces as a function of relative brightness contrasts. Therefore, borders that have relatively low contrast may be excluded from a given surface. One method of increasing the contrast in these areas is to "brighten" certain voxels in a volumetric image by adding Control Points. This method predominantly helps FreeSurfer better classify segments of brain tissue that should be included in the white matter surface (a control point will change a voxel's intensity value to 110, FreeSurfer's normalized value for white matter). Control Points are most effective for editing smaller, precise areas on the surface; to edit large areas of WM, it is best to edit the wm.mgz file (see section B).

- If editing a previously created point set, create a dated backup of control points file.

```
cp "$SUBJECTS_DIR"/tmp/control.dat "$SUBJECTS_DIR"/trash/control_backup_[date].dat
```

(1) To add Control Points, FreeView is most user-friendly. Open the FreeView window for the current subject by using the command introduced above in Section III.C.3.

```
$: sub=<subid>
$: SUBJECTS_DIR=<STUDYDIR>/

"$sub"/fs/reconall/"$sub"
$: freeview –v "$SUBJECTS_DIR"/mri/aparc+aseg.mgz:colormap=lut:opacity=0.75
"$SUBJECTS_DIR"/mri/T1.mgz "$SUBJECTS_DIR"/mri/brainmask.mgz
"$SUBJECTS_DIR"/mri/brain.finalsurfs.mgz
"$SUBJECTS_DIR"/mri/wm.mgz:colormap=heat:opacity=0.4 –f
"$SUBJECTS_DIR"/surf/lh.pial:edgecolor=red:edgethickness=1
"$SUBJECTS_DIR"/surf/rh.pial:edgecolor=red:edgethickness=1
"$SUBJECTS_DIR"/surf/lh.white:edgecolor=blue:edgethickness=1
"$SUBJECTS_DIR"/surf/rh.white:edgecolor=blue:edgethickness=1
```

(2) When adding Control Points, it is best to have the wm and aparc+aseg volumes turned off as not to edit them, with the brain or brainmask volumes visible at the uppermost layers.
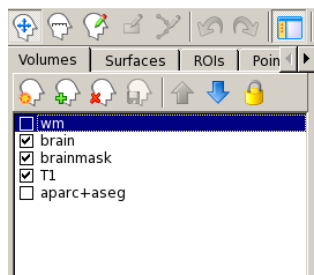

Older FreeView
(FreeSurfer 5.3)
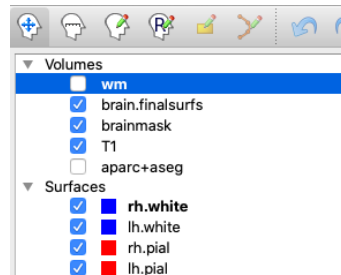
Newer FreeView
(FreeSurfer 6.0+)

Note that there are some minor differences in the layout of FreeView depending on which version is used. The biggest difference is how volume/surface/ROIs are separated in pre6.0 and 6.0+. Other illustration will mostly be from older version of FreeView

(3) Find the point set tab in the upper left corner and select "New Point Set" 🌀 (this can be found under File → New Point Set in newer FreeView).

Point Sets should be named control.dat, with brainmask as the template volume.

Just as in your reviewing process, it is best to begin in the Axial view. Some researchers prefer to edit in coronal sections, starting with the frontal lobes and progressing backwards to where the temporal lobe joins frontal/parietal cortex, then returning to temporal poles and editing backwards to the same slice.
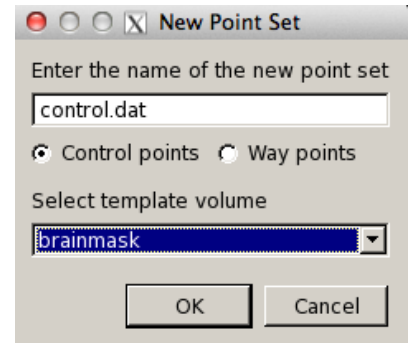
If you prefer this method, that is okay as well.

(4) Drop control points by clicking with the mouse & remove them with shift + click. You can also use undo function (ctrl + z). Holding the click down allows you to drag the control point.

When adding control points to a section of WM that is labeled GM (such as an area of WM extending into a gyrus), it is best to begin near the WM surface border (blue line), and place control points sparingly and not too close together along the path of the WM you want to be included in the new segmentation (see **Fig. 8**).

Be warned! Adding too many control points, or having control points too close together will affect the brightness of adjacent areas and may cause other structures to become labeled incorrectly.

It is very helpful to check the intensity values (displayed in the bottom right window in FreeView, values change when scrolling with mouse) of the voxels you are working with to be certain that this is a location that should be considered WM.

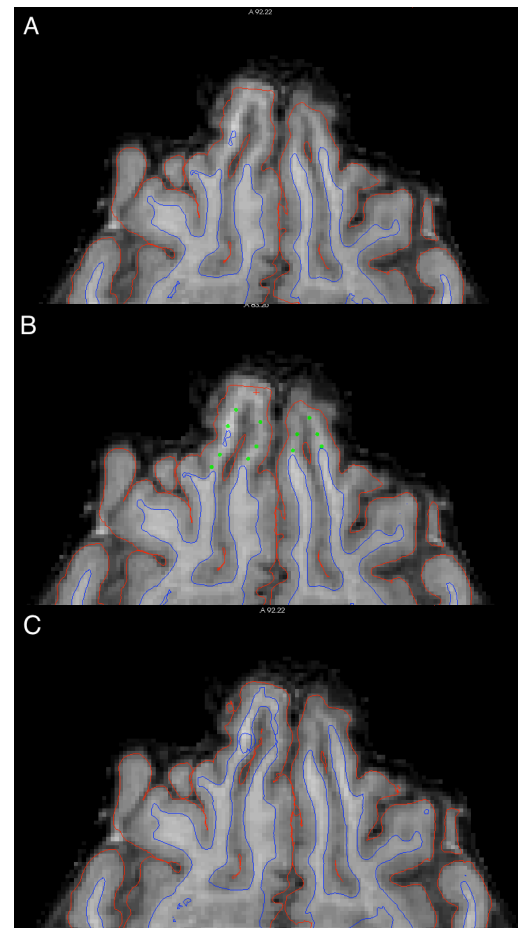| Mouse		
RAS	41.84, 22.77, 35.87	
wm	94	[93, 104, 148]
brain	94	[93, 104, 148]
brainmask	110	[93, 102, 146]
T1	97	[93, 104, 148]
aparc+aseg	41	[93, 104, 148] Right-Cerebral-White-Matter
rh.white	SurfaceRAS	[35.49, 20.28, 24.00]
	Vertex	102485 [34.97, 20.11, 24.06]
lh.white	SurfaceRAS	[35.49, 20.28, 24.00]
	Vertex	N/A
rh.pial	SurfaceRAS	[35.49, 20.28, 24.00]

Fig 8. Adding control points to capture gray matter. The OFC is a common problem area and portions of it are often excluded. Often times, the wm and pial surfaces do not capture the extent of the anatomy (A). In this scenario, you must add control points (B) beginning near the end of the wm surface, and towards pial surface, in an effort to brighten specific voxels. After recon and intensity normalization with your user-imposed edits, the surfaces should more accurately capture the tissue (C).

Typically, intensity normalization sets voxels considered WM to values of 110±20 in the typical FreeSurfer anatomical images we check (e.g., T1.mgz, brain.finalsurfs.mgz & brainmask.mgz). When making tough edits, it is a good idea to compare these values with adjacent locations, as relative contrasts can vary.

(5) The Control Points method can also be applied to gyri that are not being labeled as GM (within the pial surface border). Adding control points along the WM surface towards the apex of the gyrus will increase the relative brightness contrast and increase the likelihood that FreeSurfer will label this structure as GM.

Pro Tip #1: Sometimes it will be necessary to drop control points outside of the pial surface. Although we typically use control points to extend white matter towards the edge of the pial surface, thus extending both of them, sometimes cortex will be missing in a structure that has higher gray matter thickness and without a direct connection to white matter. This is commonly seen in the frontal and temporal poles and the insula. In these cases, you should drop control points sparingly, and in a strategic, usually central location that will create the contrast necessary for the pial surface to appropriately capture the structure.

Pro Tip #2: Portions of the orbitofrontal cortex are often not captured by the pial surface, and will need special attention. Because of its proximity to the eyes, optic nerves, and other bright tissues, voxels in the OFC can sometimes have values well over 120. Control points brighten voxels to a value of 110, so adding CP's to these voxels will not help in recapturing the surface, and may often make the problem worse by brightening other adjacent voxels.

(6) It is important to be mindful that the control points are three-dimensional and extend spherically, meaning they will affect adjacent slices in all directions. It is best to begin adding Control Points on the first slice where you observe the issue, and add them about every 2-3 slices as you continue navigating through the brain until the local issue is resolved. **<u>Do not add control points to every slice</u>**.

It is always a good idea to be conservative when adding control points, and to check your work in different views (axes) to understand your edits in a three-dimensional context. Continue navigating through all slices, in all 3 views, to review the location of your control points. You may find the need to remove or add points as you navigate through the brain.

> General rule: If a section of WM is excluded for 3 or more consecutive slices, control points should be added. Some WM exclusions are resolved within 2 slices; adding a control point there is not necessary.

(7) Remember to save control points by clicking the "Save Point Set"   button. The file should be saved in the **`$SUBJECTS_DIR"/tmp/`** folder as 'control.dat', in order for FreeSurfer to locate the file during recon. Be sure to save your work intermittently!

IV - B. Editing wm.mgz

- Prior to making any edits to wm.mgz a dated backup should be created

```
$: cp "$SUBJECTS_DIR"/mri/wm.mgz "$SUBJECTS_DIR"/trash/wm_backup_[date].mgz
```

(1) It is also best to use FreeView when making white matter edits. You will highlight the wm.mgz file in the 'Volumes' tab, and use the brain.finalsurfs.mgz image as an underlay. Make edits only to the WM volume. It can be very helpful to toggle the wm.mgz on and off to see the anatomical structures under it.
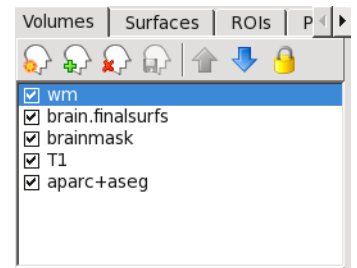
(2) Select the Voxel Edit tool [icon] in the upper left corner of the FreeView window. To "fill in" WM, the brush value should be set to 255 (yellow). It is best to leave the color map as "heat", and the other settings at their defaults. Opacity can be adjusted to best accommodate your eyes. A brush size of 2 is usually a good starting point.
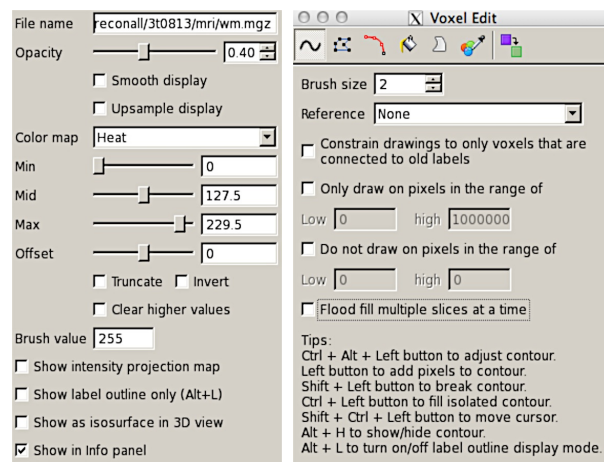
Although the WM volume is displayed in red (**110**), edits will be made with the color yellow (**255**) to distinguish your manual edits from the automated WM volume. To erase WM edits, researchers should set the brush value to **1**; FreeSurfer keeps track of manual deletions to the wm.mgz file in this way.

(3) For the types of edits outlined here, distinguishing certain structures (i.e., ventricles, subcortical structures) from WM is not necessary. The wm.mgz volume technically includes any portion of brain interior to the innermost layer of gray matter. FreeSurfer will often label 'dark' interior structures, such as ventricles, as being "outside" of the brain when it should be included in the white matter surface and wm.mgz volume. This sort of error will be very salient when the pial or white matter surfaces (red & blue lines) cut into the ventricles.
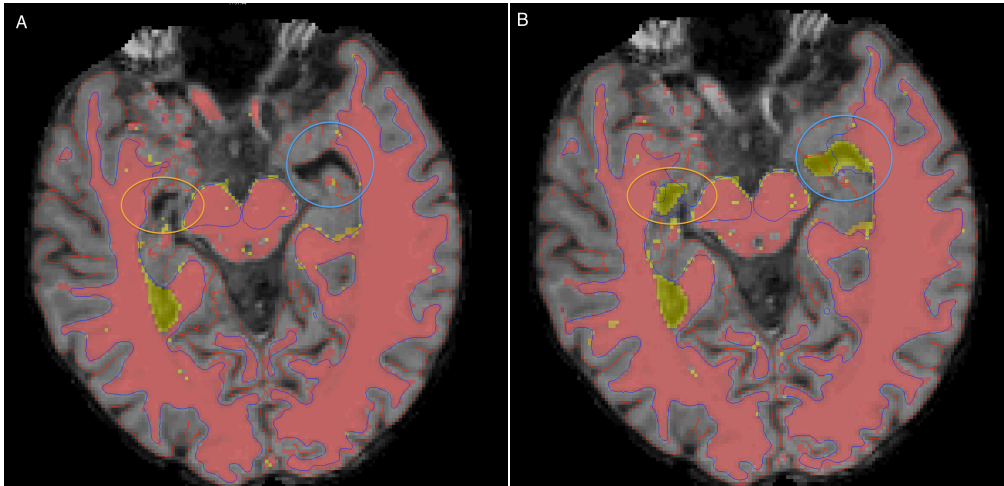

Fig 9. Filling ventricles with WM. *The inferior portions of the lateral ventricles are often excluded by the wm and pial surfaces (A), especially in older adults with more atrophy. Filling the ventricles with wm, will improve the segmentation, but often times will not capture them completely (B).*

(4) Areas requiring WM edits will often reside near the ventricles (especially in subjects that exhibit a high level of atrophy), and near the orbitofrontal cortex (e.g., the fornix, and the optic nerves).

Pro tip #1: Beware that atrophy may leave certain brains susceptible to 'opacities' in WM locations, (i.e., white matter hypointensities – WM locations that lack cell density, and appear dim). These locations are still technically WM structures, and should be included in the wm.mgz files. It is important to view the intensity values of the voxels in question, and check the underlying brain.finalsurfs.mgz and T1.mgz volumes to identify the structures.

Pro tip #2: WM edits are performed two-dimensionally; therefore, researchers should perform edits across every brain slice until the issue is resolved.

Pro tip #3: It is sometimes helpful to navigate to the first and last slice over which the issue is observed, edit those 'bounding' slices, and then 'fill in' the remaining slices in between. Keep in mind that these may begin on different slices for the left and right hemispheres.

(5) Once again, it is important to check your work in multiple views. You may find the need to delete/add more WM edits as you navigate through the brain. The horizontal axis is often most convenient and is a good starting point.

(6) Save edits by highlighting the WM volume and clicking on the "save volume" button ⊞. Always make sure you are saving your work intermittently!

IV - C. Editing brainmask.mgz & brain.finalsurfs.manedit.mgz

Editing brainmask.mgz and brain.finalsurf.manedit.mgz can be done in *TKMedit* or *FreeView*. The Wiglab has used TKMedit as it is easier to navigate and renders faster, but FreeView allows the researcher to stay within one recommended program for all types of edits. This manual will briefly mention the setup for *FreeView* for pial edits at the end of this section

Make sure to create a dated backup of the brainmask before beginning edits.

```
$: cp "$SUBJECTS_DIR"/mri/brainmask.mgz "$SUBJECTS_DIR"/trash/brainmask_backup_[date].mgz
```

The brain.finalsurfs.mgz file will rarely be edited. If the situation is dire and edits are required, however, the file needs to be copied to brain.finalsurfs.manedit.mgz in subject's mri folder in order for FreeSurfer to recognize it.

```
$: cp "$SUBJECTS_DIR"/mri/brain.finalsurfs.mgz
"$SUBJECTS_DIR"/mri/brain.finalsurfs.manedit.mgz
```

(1) Open *TKMedit* for the brainmask.mgz with the T1.mgz file as your secondary "aux" volume by entering the following line into the command window:

```
$: tkmedit -f "$SUBJECTS_DIR"/mri/brainmask.mgz -aux "$SUBJECTS_DIR"/mri/T1.mgz -surface
"$SUBJECTS_DIR"/surf/lh.white -aux-surface "$SUBJECTS_DIR"/surf/rh.white
```
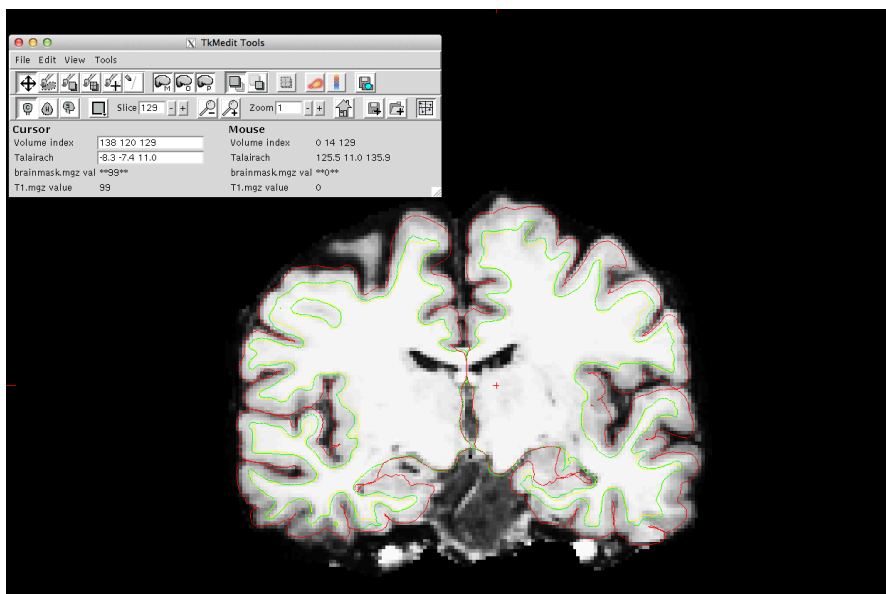

Fig 10. Typical TKMedit window, opened using the above command. Before editing, it is important to adjust the brightness and contrast of your main and aux images, and the set up your brush info. The orig surface (green line is not used for our edits and should be turned off by pressing the button.

The "Navigate Tool" ⊕ allows you to move the brain image by dragging and clicking with the mouse, and to scroll through brain slices with the arrow keys. To make edits, the "Edit Voxels Tool" must be selected.

The purpose of editing the brainmask.mgz volume is to eliminate (non-brain structures) from being incorrectly labeled as cortex (pial surface issues, see **Section III.B.4**). Pial structures can appear very bright, and the contrast created between them, and adjacent dark areas can lead to inappropriate labeling. These structures are not to be included within the pial surface and are simply edited out (given a value of *1* in the brainmask.mgz).

(2) In *TKMedit*, select 'Configure Brush Info' & 'Configure Volume Brush' under Tools tab

Select the 'Main Volume' for the target. The radius should be set to '1' as a starting point, and the brush shape is a matter of personal preference (I prefer to use square). Voxel edits can be done three-dimensionally, but extra care must be taken when editing in 3D. To start off, it is best to have the 3D button turned off.
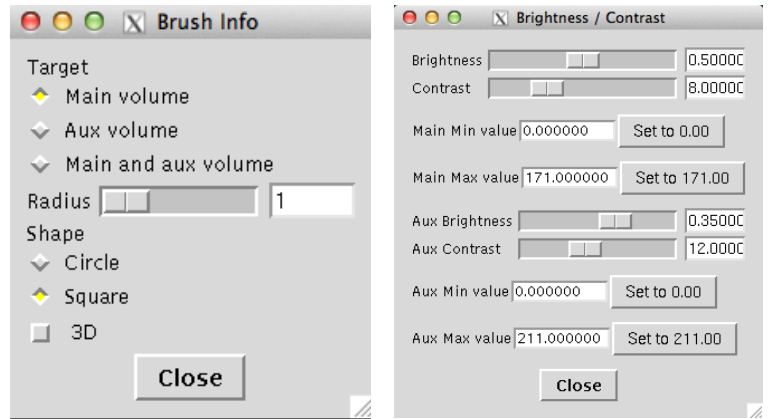
Pro Tip #1: Adjusting the brightness and contrast ('View' → 'Configure' → 'Brightness / Contrast' tab; or just 'ctrl + b') can help make the grey-matter-white-matter boundary or the grey-matter-outside-brain boundaries much clearer. Adjustments to this setting are more a matter of personal preference and should be adjusted to best allow you to identify locations that require manual intervention.

(3) In the Volume Brush Info tab, you will be able to set up two separate buttons for a three-button mouse, or 'option + click' & 'command + click' for buttons 2 and 3 respectively (this is the key mapping when using a Macintosh; this requires your XQuartz window preferences to allow 3-button mouse if you are logged into the server from a Desktop). For button 2, set the Mode to 'New Value', and set the value to **1.000000** (black). For button 3, set the Mode to 'Clone Value' and select 'Aux Volume' for the 'Clone Source'. Important: You can use button 3 to clone voxels back into your brainmask image from the T1.mgz volume if you make mistakes while editing ('ctrl + z' only undoes the last edit in TKMedit). When dealing with an overaggressive skull stripping procedure (e.g., FreeSurfer removes too much brain tissue from the brainmask.mgz), cloning voxels will help retrieve voxels containing brain tissue from the T1.mgz file and replace empty ones in the brainmask.mgz image.
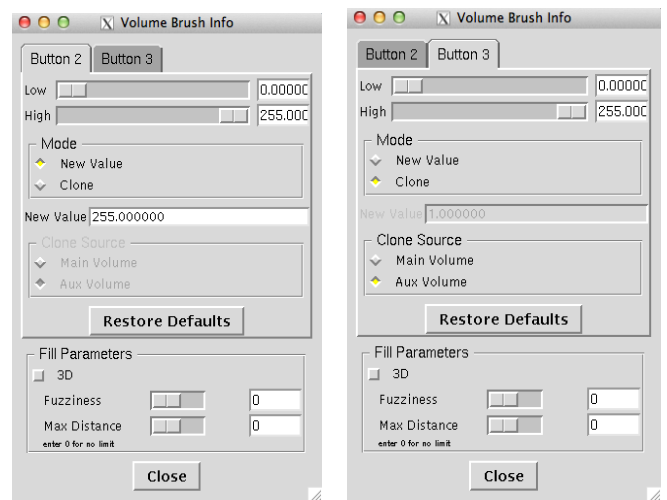
(4) Any structures that are outside of brain and are being included in the pial surface (falling within the red outline) must be edited out. Areas of the brain that require special attention are the thick sheet of dura found along the top and posterior end of the brain (falx cerebri - along the longitudinal fissure or midline) mostly affecting parietal and occipital cortices, the tentorium cerebelli (dura separating cerebral cortex from cerebellum), and the optic nerve and optic chiasm. In addition, blood vessels and other tissue can be problematic, especially near the temporal poles and subgenual cortex. If the pial surface mistakenly includes portion of the **cerebellum** within its boundary, do **not** manually erase the cerebellum by setting it to 1/0, instead, following *step (8).*

(5) At first, it may be difficult to distinguish between dura mater and cortex. The easiest way to determine the divide is to see if there is a small black space between the cortex and the dura (this may require you to zoom in quite a bit). Make sure to check the intensity values of individual voxels to differentiate between what is dura and what is cortex. Intensity values can be observed in the toolbar as you scroll across voxels. Non-brain structures are often very bright, whereas GM is slightly darker (normally below 90 on the brainmask.mgz volume). However, this rule-of-thumb can be misleading on slices that touch the edge of pial structures, where the intensity is much lower than otherwise observed. Using all three views is very common when making pial edits.
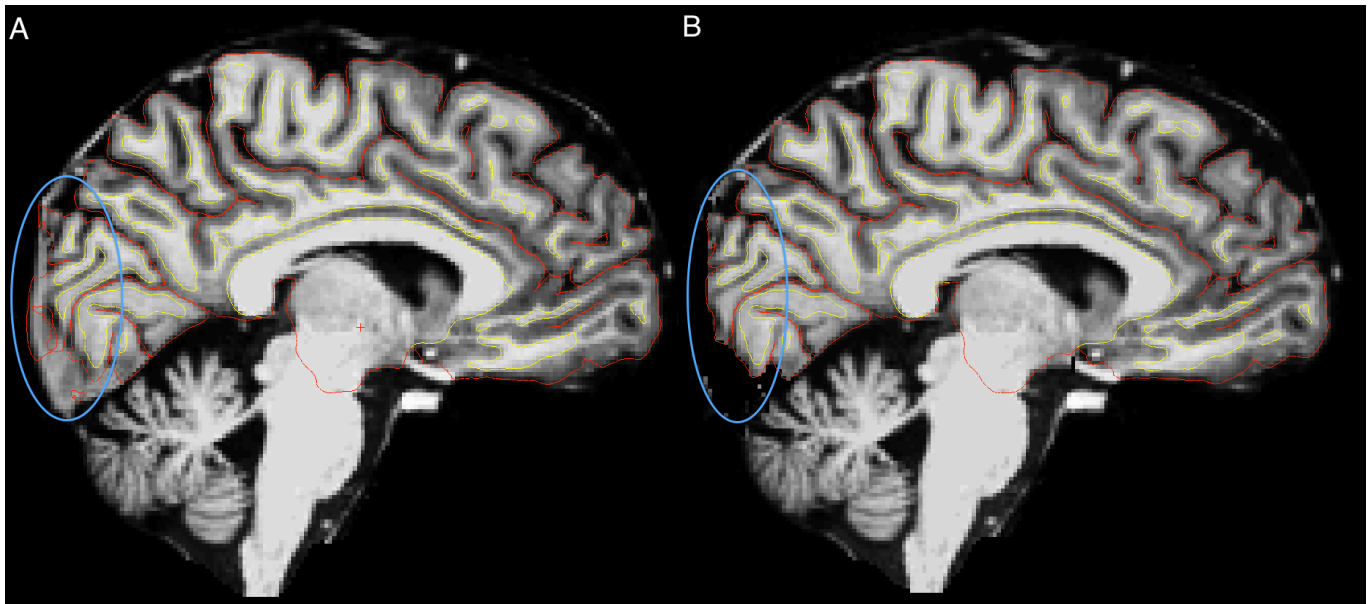

Fig 11. Removing dura mater. *The most common pial surface issue will be the inclusion of dura. In brain A we can see the pial surface extending beyond the end of the cortex and grabbing a significant amount of non-brain tissue. These types of surface errors can impact regional measures of GM thickness. In B, we see how to properly edit out the offending voxels (with a brush value of 0), and how the correct surfaces should look after recon.*

Pro Tip #2: The pial and white matter surfaces (red and yellow lines) can often be misleading and can cause us to misjudge the boundary between dura and cortex. It can be helpful to turn off the surfaces when looking at a particularly ambiguous structure, as to not be biased by the lines.

Important: Make sure not to delete actual cortex when deleting non-brain structures.

(6) Cloning: As mentioned above, sometimes FreeSurfer's skull stripping procedure can be a bit overaggressive & remove portions of cortex in the brainmask.mgz volume. In this situation, you will have to clone in voxels from the Aux (T1.mgz) volume. Additionally, if you accidentally delete a voxel of cortex during pial edits, you can clone it back in. This is important because in TKMedit you can only undo (ctrl +z) your last edit. If you have the volume brush info configured as recommended above, you will be able to clone voxels by using button 3 (command + click). Clone back the voxels as necessary, being sure not to clone dura mater that is NOT cortex into the brainmask.mgz image. If the location to be cloned is very large, researchers may prefer to clone a large region into the brainmask.mgz, and then edit out non-cortical tissue.

<u>Pro Tip #3:</u> Often times it can be helpful to actually remove some cortex while editing out dura or blood vessels. The physical boundary between the two tissue types might be obscured, and can only really be discriminated when removing some cortical tissue and seeing that you have gone too far. Once you have identified the boundary, make sure to clone back in all the voxels of cortex that you have removed.
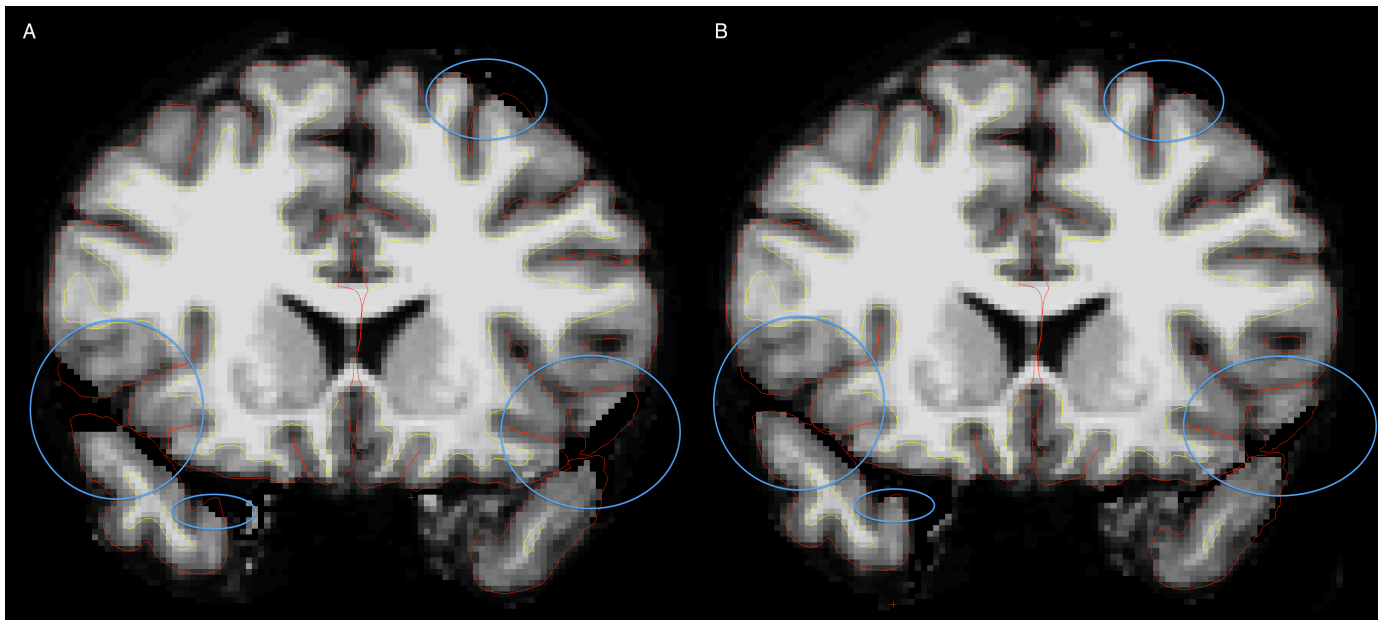

Fig 12. Cloning in cortex removed during pial edits. *A common occurrence during pial editing is accidentally removing voxels of cortex when removing dura or blood vessels. This often happens when using a large brush size or when the border between tissue types is not obvious. Brain A shows very overaggressive editing, with significant portions of cortex being removed. Brain B shows what the pial edits should look like after cloning the voxels of cortex back in. Notice that there appears to be some tissue missing in the left temporal lobe of Brain B. This is where a blood vessel that was affecting the pial surface has been removed.*

(7) Editing the brainmask.mgz is done in 2D, and researchers must progress through each individual slice. This editing process is tedious and requires a fair amount of practice to master. Zooming in closely on ambiguous areas, checking adjacent voxel intensity values, and having a good understanding of neuroanatomy all aid in the process of differentiating cortical from non-cortical structures.

- Be sure to save your work: File → Save Main Volume.

(8) Editing the **cerebellum** or persistent problems/defects. Manual edits of the cerebellum (i.e., remove cerebellum from being included in the pial surface) should be done in a file named "brain.finalsurfs.manedit.mgz." Additional persistent defects that could not be corrected by manual edits on the brainmask.mgz can also be included in this file.

First, the user creates a "brain.finalsurfs.manedit.mgz" file in a participant's mri folder by copying it directly from "brain.finalsurfs.mgz" (shown at the top of **Section IV.C**). The brain.finalsurfs.mgz/brain.finalsurfs.manedit.mgz files contain information that is very similar to the brainmask.mgz file, but have been further processed with an additional iteration of intensity normalization that includes information from control points (if present) and basic anatomical parcellation (i.e., aseg.mgz).

Edits to this image are voxel edits similar to those performed on the brainmask.mgz volume (e.g., pial edits, cloning lost voxels back into the image). For instance, it is sometimes the case that the pial surface line includes voxels containing the cerebellum.
In this case, edits to the brain.finalsurfs.manedit.mgz that set these cerebellum voxels to an intensity of '1'. Additionally, if CSF or non-brain tissue that were erased in brainmask.mgz was not being recognized by FreeSurfer during recon-all, edit them out in brain.finalsurfs.manedit.mgz may help resolve the defect.

Notes: The brain.finalsurfs.mgz and brain.finalsurfs.manedit.mgz (recon-all uses brain.finalsurfs.manedit.mgz if it exists, otherwise uses brain.finalsurfs.mgz) files serve as one of the main volumetric files used to generate the initial 2D pial and white surfaces (input to mris_make_surfaces). Given that the automated surfaces may be relatively more susceptible to editor bias when either of these volumetric images is changed manually, we strongly urge editors to exhaust the other methods of editing (i.e., control points, wm.mgz and pial edits) before turning to brain.finalsurfs.mgz. In the Wiglab, editing the brain.finalsurfs.manedit.mgz beyond removing cerebellum is a rare occurrence and we make every effort to correct pial surface issues by editing the brainmask.mgz.

Using *FreeView* for pial edit

Editing brainmask.mgz in FreeView is very similar to editing wm.mgz.

- Select brainmask.mgz under Volumes.
- **Remove voxels:** Click the Voxel Edit tool, check the recon edit box if this is a newer version of FreeView. This will automatically set the <u>brush</u> value to "**255"** (for wm edit) and <u>eraser</u> to "**1"** (for pial edit).
    - o Eraser value is applied with 'shift + left-click'
    - o Alternatively, set the brush value to "**1**" to do pial edit with just left-click. (You cannot set manual values in 'recon edit' mode, so uncheck recon edit for this option)

- **Clone voxels:** Click the Clone button (near the top right corner of the Voxel Edit window).
    - o Make sure 'recon editing' is **not** checked.
    - o At the Reference drop down menu, select T1
    - o Use left-click to copy T1 structure back into brainmask.mgz.

V. EXTRA TIPS & TRICKS

V - A. Fine-tuning a Skull-strip

Often, the default FreeSurfer skull-stripping procedure will be insufficient and leave a considerable amount of dura and other tissue in the brainmask.mgz volume. If these leftover tissues affect our surfaces, we normally correct them with pial edits. If skull-stripping is not aggressive enough and there is an excessive amount of tissue that needs to be removed, it is sometimes quicker to run the command named '**gcut**'. It will remove additional dura that may have been missed during skull stripping.

In addition, you can manually adjust the **watershed parameters** of the default skull-stripping procedure, and rerun this step using a more or less aggressive version. The default watershed is 25 percent. Increase the watershed if you need to remove more dura, and lower it if gray matter is being cut off during skull-stripping.

For more information on how to run gcut and to change watershed parameters, see the FreeSurfer wiki page for 'Fixing a bad skull strip': http://surfer.nmr.mgh.harvard.edu/fswiki/FsTutorial/SkullStripFix_freeview

B. Helpful FreeSurfer Quality Metrics

Some researchers may be interested in capturing extra objective metrics reflecting the quality of their MPRAGEs or their FreeSurfer segmentations (see Savalia et al., 2015). While there are many approaches to deriving this sort of data, several metrics are available directly within FreeSurfer.

"Euler No." and "No. of Holes" in Initial (Non-fixed) Surfaces:
The Euler number* (Euler number = 2 – 2g; where g = the total number of surface defects [holes, handles]), when calculated before any smoothing operations are applied, estimates the topological correctness of one's surface model before any surface-based corrections are imposed. The Euler number and number of holes per hemisphere is calculated in the default-processing pipeline of FreeSurfer. They can be obtained for one participant's non-fixed surfaces by a one-line command:

```
$: grep -A 2 "Computing euler" "$SUBJECTS_DIR"/scripts/recon-all.log
```

"Gray-White" and "Gray-CSF" Contrast to Noise Ratio (CNR):
CNR is the ratio of the difference in signal intensities between regions classified as different tissue-types vs. background (noise) signal. In FreeSurfer, the CNR metric assumes a Gaussian noise model (see post at https://www.mail-archive.com/freesurfer@nmr.mgh.harvard.edu/msg24216.html), where the average CNR across an image is calculated as follows:

$$gray\_white\_cnr = SQRT(gray\_mean - white\_mean) / (gray\_var + white\_var) ;$$
$$gray\_csf\_cnr = SQRT(gray\_mean - csf\_mean) / (gray\_var + csf\_var) ;$$

Values for Gray-White* and Gray-CSF CNR for each hemisphere can be obtained for a given volumetric (.mgz) file by a default FreeSurfer command line program ('mri_cnr'). For example:

```
$: mri_cnr "$SUBJECTS_DIR"/surf "$SUBJECTS_DIR"/mri/norm.mgz
```

For comparison purposes, Chalavi et al. (2012) *MRM* provide Euler No. & Gray-White CNR values obtained across multiple MR sites in comparison to *bert* (FreeSurfer sample participant).

VI. ADDITIONAL READING

Cardinale, F., Chinnici, G., Bramerio, M., Mai, R., Sartori, I., Cossu, M., Lo Russo, G., Castana, L., Colombo, N., Caborni, C., De Momi, E., & Ferrigno, G. (2014). Validation of FreeSurfer-estimated brain cortical thickness: comparison with histologic measurements. *Neuroinformatics, 12*(4), 535-542. doi: 10.1007/s12021-014-9229-2

Dale, A. M., Fischl, B., & Sereno, M. I. (1999). Cortical surface-based analysis. I. Segmentation and surface reconstruction. *Neuroimage, 9*(2), 179-194. doi: 10.1006/nimg.1998.0395

Fischl, B. (2012). FreeSurfer. *Neuroimage, 62*(2), 774-781. doi: 10.1016/j.neuroimage.2012.01.021

Fischl, B., Sereno, M. I., & Dale, A. M. (1999). Cortical surface-based analysis. II: Inflation, flattening, and a surface-based coordinate system. *Neuroimage, 9*(2), 195-207. doi: 10.1006/nimg.1998.0396

Fischl, B., Sereno, M. I., Tootell, R. B., & Dale, A. M. (1999). High-resolution intersubject averaging and a coordinate system for the cortical surface. *Hum Brain Mapp, 8*(4), 272-284.

Gronenschild, E. H., Habets, P., Jacobs, H. I., Mengelers, R., Rozendaal, N., van Os, J., & Marcelis, M. (2012). The effects of FreeSurfer version, workstation type, and Macintosh operating system version on anatomical volume and cortical thickness measurements. *PLoS One, 7*(6), e38234. doi: 10.1371/journal.pone.0038234

Han, X., Jovicich, J., Salat, D., van der Kouwe, A., Quinn, B., Czanner, S., Busa, E., Pacheco, J., Albert, M., Killiany, R., Maguire, P., Rosas, D., Makris, N., Dale, A., Dickerson, B., & Fischl, B. (2006). Reliability of MRI-derived measurements of human cerebral cortical thickness: the effects of field strength, scanner upgrade and manufacturer. *Neuroimage, 32*(1), 180-194. doi: 10.1016/j.neuroimage.2006.02.051

Kuperberg, G. R., Broome, M. R., McGuire, P. K., David, A. S., Eddy, M., Ozawa, F., Goff, D., West, W. C., Williams, S. C., van der Kouwe, A. J., Salat, D. H., Dale, A. M., & Fischl, B. (2003). Regionally localized thinning of the cerebral cortex in schizophrenia. *Arch Gen Psychiatry, 60*(9), 878-888. doi: 10.1001/archpsyc.60.9.878

Reuter, M., Tisdall, M. D., Qureshi, A., Buckner, R. L., van der Kouwe, A. J., & Fischl, B. (2015). Head motion during MRI acquisition reduces gray matter volume and thickness estimates. *Neuroimage, 107*, 107-115. doi: 10.1016/j.neuroimage.2014.12.006

Rosas, H. D., Liu, A. K., Hersch, S., Glessner, M., Ferrante, R. J., Salat, D. H., van der Kouwe, A., Jenkins, B. G., Dale, A. M., & Fischl, B. (2002). Regional and progressive thinning of the cortical ribbon in Huntington's disease. *Neurology, 58*(5), 695-701.

Savalia, N. K., Agres, P. F., Chan, M. Y. , Feczko, E. J., Park, D. C., Kennedy, K. M., & Wig, G. S. (2017). Motion-related noise in structural brain images revealed with independent estimates of in-scanner head motion. *Hum Brain Mapp,* 38(1), 472-492. doi: 10.1002/hbm.23397

Segonne, F., Dale, A. M., Busa, E., Glessner, M., Salat, D., Hahn, H. K., & Fischl, B. (2004). A hybrid approach to the skull stripping problem in MRI. *Neuroimage, 22*(3), 1060-1075. doi: Doi 10.1016/J.Neuroimage.2004.03.032

Segonne, F., Grimson, E., & Fischl, B. (2005). A genetic algorithm for the topology correction of cortical surfaces. *Information Processing in Medical Imaging, Proceedings, 3565*, 393-405.