NPDL Functional MRI Analysis Manual

Release 1.0

Connor Lane

February 22, 2016

CONTENTS

1 Data organization **1**

1.1 Exploring the Godzilla server 1

1.2 The scan log 1

1.3 Transferring data for analysis 2

2 Surface reconstruction **4**

2.1 Surfaces 4

2.2 Introduction to Freesurfer 5

2.3 Running `recon-all` 6

2.4 Inspecting outputs 6

2.5 Common reconstruction errors 8

2.6 Quality control record keeping 9

2.7 Post reconstruction processing 10

CHAPTER

ONE

DATA ORGANIZATION

1.1 Exploring the Godzilla server

Once data is collected, it's sent to the KKI Godzilla server for permanent storage. The first step in analysis then is to retrieve the data from Godzilla. Like our lab server, Godzilla is a UNIX machine, so we will use standard command line tools to log onto the server and transfer data.

To log onto the server, you need to contact system administrator at KKI and request an account. When you have an account, you can log on using ssh and begin exploring.

To simplify the log-in process, you should set up public key authentication. For a description of what this is and how to do it, see this tutorial[1].

Once you're logged in, navigate to /g4/mbedny. This is our lab folder on Godzilla. All of our studies are stored in this folder (e.g. BSYN, BRAILLE). Within each study folder, there will be a folder for each subject. And within each subject's folder, a par and rec file for each collected run, plus an MR folder, containing DICOM images for the MPRAGE scan:

```
/g4/mbedby/
  |- BSYN/
    |- BSYN_S_01/
      |- bsyn_s_01_3_1.par
      |- bsyn_s_01_3_1.rec
      |- bsyn_s_01_4_1.par
      |- bsyn_s_01_4_1.rec
    |- BSYN_S_02/
  |- BRAILLE1/
    |- BRAILLE1_CB_01
      |- MR/
        |- 1.3.46.670589.11.24058.5.0.1788.2014053014171806001
        |- 1.3.46.670589.11.24058.5.0.1788.2014053014171865002
      |- braille1_cb_01_3_1.par
      |- braille1_cb_01_3_1.rec
```

Typically, runs 1, 2 are the survey and reference scans respectively. These scans are not usually sent to Godzilla during data collection. The first scan we keep is usually run 3, the MPRAGE.

1.2 The scan log

For each scanning session you must keep a scan log documenting the events of the session. The scan log is how we associate scanner runs with behavioral files. Without the scan logs we could not run any analyses.

[1] https://macnugget.org/projects/publickeys/

You should keep both a written scan log and an electronic version. The format for the electronic version is as follows:

```
# Study: BSYN
# Subject ID: BSYN_S_01
# Scanner ID: BSYN_04
# Registration ID: 1403250900
# Date: 3/25/14 9:00
# Scanner: MR1 32ch
# Scanned by: TB

3 mprage
4 bsyn_01
5 bsyn_02
6 bsyn_03
7 bsyn_04 # participant got out to use the restroom.
8 bsyn_05
9 bsyn_06

# Notes:
# - Volume set to 2.5
# - Good performance on average
# - Subject a little claustrophobic for the first scan
```

The scan log starts with a header containing info about the scan session: the study name, the subject ID, the scanner subject ID (which may or may not be different), etc.

- Each line of the header must start with “#”.
- The “key” for each line must be spelled exactly as shown.
- The keys must be separated from their values with a colon.

The next section of the scan log is a two-column matrix consisting of run number, run name pairs. The run number is the number of the scan, as it was collected. The run name typically has the format `{task}_{run num}`, with the run number being zero-padded to two places.

Last, there is an optional *Notes* section, where you can record miscellaneous information about the session or the participant. Each line of notes should start with “#”.

1.3 Transferring data for analysis

Transferring data is a three part process:

1. Fetch the par and rec files from Godzilla.
2. Convert the par/recs to gzipped Nifti files.
3. Rename the converted files to something more convenient than the default scanner names.

All of these steps are accomplished with the `parfetch` command, which is part of the lab’s suite of scripts

```
Usage: parfetch [options] <scan-log>

Fetch par and rec files from the scanner file server and convert
to gzipped nifti. File organization on the server is assumed to
follow the convention:

  {lab dir}/{study dir}/{subject ID}/*_{run #}_{acq #}.*
```

```
Arguments:
  <scan-log>   Scan log text file. Describes how files should be
               renamed. First column is run number, second column
               is new name. You may optionally specify the Study,
               Subject ID, and/or Scanner ID in a comment line
               (starting with #). All other lines starting with #
               will be ignored. See below for an example of the
               proper format.

Options:
  --study <study>     Name of study on server. Read from the scan log
                      by default (needs a '# Study: XXXX' line).
  --sub <scan-sub>    Scanner subject ID on server. Read from the scan
                      log by default (needs a '# Scanner ID: XXXX' line).
  --out <outdir>      Directory to put converted data. If this option
                      is not specified, the converted data will be placed
                      in {subject ID}/raw, in the working directory,
                      where {subject ID} is read from the scan log.
  --u <user>          Name of server user [default: clane9].
  --labdir <dir>      Lab directory on server [default: /g4/mbedny].
  --no-clean          Don't delete redundant rec files.
```

First `parfetch` reads the scan log for the scan session to determine where the data is located on Godzilla, and where it should be placed on the lab server. It uses the "Study" and "Scanner ID" values to determine where the data is located, and it uses the "Subject ID" value to decide where to put the data (defaulting to `{Subject ID}/raw` in the working directory.

Next, the command transfers the data to the lab server using the `scp` command. For this part to work it is essential that you can access Godzilla. And if you have public-key authentication set up, you won't have to enter your password. Next, `parfetch` uses `dcm2nii` to convert the data to gzipped Nifti files. See the Mricron[2] site for details on this part. Last, `parfetch` renames the converted files according to the names given in the second column in the scan log.

If we were to run `parfetch` on the example scan log above, the resulting raw folder would be structured like this:

```
BSYN_S_01/
  |- raw/
    |- bsyn_01.nii.gz
    |- bsyn_02.nii.gz
    |- mprage.nii.gz
    |- par/
      |- bsyn_04_3_1.par
      |- bsyn_04_4_1.par
      |- MR/
    |- parfetch.log
    |- sl.txt
```

[2] http://www.mccauslandcenter.sc.edu/mricro/mricron/dcm2nii.html

CHAPTER

TWO

SURFACE RECONSTRUCTION

2.1 Surfaces

Most of our analyses are performed on data that have been mapped to the cortical surface. Surface-based data files come in mainly two kinds: anatomical surfaces, and metric overlays. Anatomical surfaces are like high-res T1 images. They are used to represent the shape of the cortical sheet, all of the sulci and gyri. Anatomical surfaces are geometrical structures made up of triangular faces that are "glued" together along the edges to form a closed "mesh". The points of the triangular faces are called vertices. Vertices are the surface analogues to voxels. They are the units of analysis.

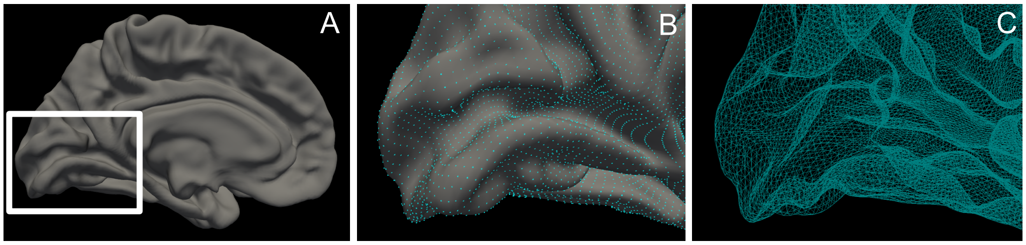

Figure 2.1: Renderings of an anatomical surface. **(A)** The entire surface shown from the medial view. V1 is boxed. **(B)** A close-up of V1, with vertices shown in teal. **(C)** Another close-up of V1, this time showing the mesh structure.

Metric overlays, on the other hand, are more like BOLD images. Metrics store the actual functional data. At bottom, metrics are mappings between the vertices of an anatomical surface and some numerical data. The numerical data assigned to each vertex can either be a vector, like a functional time series, or a single scalar, such as a z or t value. Metrics are also commonly used to represent ROIs, by assigning vertices within the ROI the value 1, and all other vertices 0. One important thing to remember is that metrics contain no anatomical structure. This is unlike standard BOLD images, where the spatial relationships between data points are encoded in the file format itself.

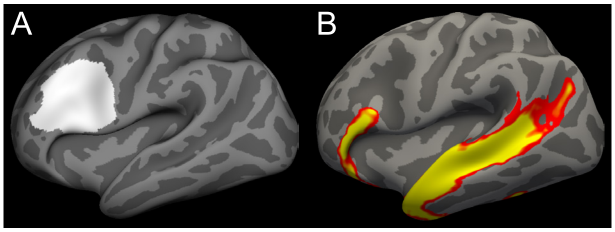

Figure 2.2: Examples of metric overlays. **(A)** An ROI in inferior frontal cortex, represented with 1's and 0's. **(B)** A z-stat image for a language contrast.

2.2 Introduction to Freesurfer

All of our surface-based analyses depend a lot on the Freesurfer software package. Freesurfer is basically a set of command-line tools for generating and manipulating surface files. Generating anatomical surfaces from T1 images is probably the most important thing we use Freesurfer for. The command for that is called `recon-all`. Some other important uses for Freesurfer include:

- Volume and surface visualization (freeview[1], tksurfer[2])
- Volume and surface file conversion (`mri_convert`, `mris_convert`)
- Surface-informed volume registration (`bbregister`)
- Surface manipulation (`mri_surf2surf`)
- Surface-based group fMRI statsistics (`mri_glmfit`)

Freesurfer uses a centralized workspace called the *Subjects Directory* for carrying out most of its processing. The location of the Subjects Directory is determined by the shell environment variable `SUBJECTS_DIR`. Within the Subjects Directory there are going to be many individual *subject folders*. These folders are generated when you run `recon-all`. Their purpose is to store all of the output surface files and intermediate volume files created by `recon-all`. For example, the default Freesurfer Subjects Directory (`$FREESURFER_HOME/subjects`), contains a subject called "bert":

```
bert/
  |- bem/
  |- label/
  |- mri/
    |- brain.mgz
    |- orig.mgz
    |- T1.mgz
  |- scripts/
    |- recon-all.log
    |- recon-all-status.log
  |- src/
  |- stats/
  |- surf/
    |- lh.pial
    |- rh.pial
    |- lh.sphere.reg
    |- rh.sphere.reg
    |- lh.white
    |- rh.white
  |- tmp/
  |- touch/
  |- trash/
```

The mri and surf folders contain the output volume and surface files respectively. For example, T1.mgz is a copy of the raw T1 volume, after intensity bias correction has been applied. lh.white is an anatomical surface that follows the grey/white matter boundary in the left hemisphere.

Note: Freesurfer uses a custom file format to represent MRI volumes and surfaces. These files usually have the extension `.mgh` or `.mgz` (compressed). You can convert these files to more standard formats like Nifti and Gifti using `mri_convert` and `mris_convert` respectively.

[1] http://freesurfer.net/fswiki/FreeviewGuide
[2] https://surfer.nmr.mgh.harvard.edu/fswiki/tksurfer

A good way to get familiar with all of the Freesurfer output files is by following the Freesurfer output inspection tutorial[3], either with the tutorial data or your own. This is also a good way to get familiar with the primary Freesurfer visualization tool, freeview[4].

When you open a new terminal, `SUBJECTS_DIR` will probably be set to `$NPDL_SCRIPT_DIR/subjects`. This is a lab-specific folder that contains the three *group average* subjects we use most often: fsaverage, 32k_fs_LR, and 164k_fs_LR. fsaverage comes packaged with Freesurfer and is based on the MNI 305 template. 164k_fs_LR and 32k_fs_LR are derived from it. It is useful to have the `SUBJECTS_DIR` set this way, so you can have easy access to these subject folders. However, when it is time for you to do your own analyses, you will need to create a Subjects Directory inside your analysis folder to hold the surface reconstruction folders for your study subjects.

2.2.1 Additional Freesurfer resources

- The Freesurfer course materials: https://surfer.nmr.mgh.harvard.edu/fswiki/FsTutorial
- The Freesurfer youtube channel: https://www.youtube.com/channel/UCruQerP8aa-gYttXkAcyveA
  - In particular, the smoothing and registration talk, which is all about the benefits of surface-based analysis: https://www.youtube.com/watch?v=8WPvXoORoAw

2.3 Running `recon-all`

Freesurfer's entire surface reconstruction pipeline is completely automated. It only takes one command to get the process going, e.g.:

```
recon-all -subject BLAH_S_01 -i T1.nii.gz -all
```

This command will create a subject folder called BLAH_S_01 inside the Subjects Directory and begin populating it with BLAH_S_01's anatomical surfaces. The only input is T1.nii.gz, the high resolution T1 volume. There are about 34 steps in the entire processing stream, starting with intensity normalization and skull stripping, and going through surface creation, surface inflation, and surface-based registration to fsaverage. Altogether it can take up to 10 hours to complete (on our machine it's usually about 6 hours). The `-all` option tells recon-all to do all 34 steps.

You can view the complete description of all processing steps either by looking at the `recon-all` help message, or by navigating to the online Freesurfer analysis pipeline overview[5]. If you're looking for more detail, try the recon-all table[6] which contains a complete outline of all the reconstruction steps, and the sub-commands involved.

Note: Make sure to set the `SUBJECTS_DIR` environment variable properly before you start your analyses. Otherwise `recon-all` will try to write to the default Subjects Directory, which will likely cause a permissions error.

2.4 Inspecting outputs

The critical outputs of the Freesurfer reconstruction process are four surfaces (two for each hemisphere). The left and right "white" surfaces mark the grey/white matter boundary. The left and right "pial" surfaces

[3] http://freesurfer.net/fswiki/FsTutorial/OutputData_freeview
[4] http://freesurfer.net/fswiki/FreeviewGuide
[5] http://freesurfer.net/fswiki/FreeSurferAnalysisPipelineOverview
[6] http://freesurfer.net/fswiki/ReconAllTableStableV5.3

mark the CSF/grey matter boundary. The goal of data inspection is to check that the volume between these surfaces includes all and only grey matter.

We visually inspect these surfaces overlaid onto the high-res anatomical image. To do this, we load the surfaces and the high-res anatomical into the Freesurfer viewer, freeview[7]. Freeview can be opened at the terminal by typing `freeview`. You can then use the GUI interface to load the surfaces and T1 image.

Alternatively, you can specify the images as command-line arguments to the `freeview` command.

To make loading images easier, we have the command `checksurf`, which only requires a subject ID argument (provided the `SUBJECTS_DIR` is set properly):

```
checksurf $subj
```

All `checksurf` does is encapsulate what would be a very long `freeview` call: loading the original T1, the brain-extracted T1, the left and right white and pial surfaces, as well as the left and right inflated surfaces

```
freeview \
  -v \
  $SUBJECTS_DIR/$subj/mri/brainmask.mgz \
  $SUBJECTS_DIR/$subj/mri/orig.mgz:visible=0 \
  -f \
  $SUBJECTS_DIR/$subj/surf/lh.white:edgecolor=blue:edgethickness=2 \
  $SUBJECTS_DIR/$subj/surf/rh.white:edgecolor=blue:edgethickness=2 \
  $SUBJECTS_DIR/$subj/surf/lh.pial:edgecolor=green:edgethickness=2 \
  $SUBJECTS_DIR/$subj/surf/rh.pial:edgecolor=green:edgethickness=2 \
  $SUBJECTS_DIR/$subj/surf/lh.inflated:edgethickness=0:visible=0 \
  $SUBJECTS_DIR/$subj/surf/rh.inflated:edgethickness=0:visible=0
```

To check the surface boundaries, you will want to scroll through each slice of the image, looking for places where too much or too little has been included as “grey matter”. You should pick either the sagittal, coronal, or axial view from the top menu bar, and work your way through the entire image. You can scroll the slices using “Fn+Up Arrow” and “Fn+Down Arrow”. Some other useful shortcuts are:

- Shift+Left click drag: Control brightness/contrast of the anatomical image
- Mouse scroll: Zoom in/out
- Ctrl+Left click: Zoom in at cursor
- Ctrl+Right click: Zoom out at cursor
- Left/Right/Up/Down Arrow: Move FOV

These shortcuts and others can also be found in Help –> Quick Reference in the Freeview window.

Some general pointers for checking surface accuracy:

- About 1-3 seconds per slice tends to be a good pace.
- I prefer to holistically watch the entire cortex as I scroll through, rather than fixate on each individual sulcus and gyrus.
- When you see something that doesn’t look right, try clicking on the spot and changing views. Getting a different perspective usually helps you decide if what you’re seeing is really an error.
- It also helps to pass over a problem area a few times, scrolling back and forth. Getting a better idea of the context around a potential reconstruction error will usually help you decide what to do about it.

[7] http://freesurfer.net/fswiki/FreeviewGuide

2.5 Common reconstruction errors

Most of the time the surfaces produced by Freesurfer are just fine and don't need any intervention (say ~50% of subjects). When there are surface problems, they'll usually fall into one of the following categories.

Note: Many of these errors are covered in the Freesurfer failure modes talk[8], and the fixes are described in the Freesurfer troubleshooting tutorial[9]. **When unsure, defer!** Freesurfer is smart, and objective. If you're not sure that what you're seeing is an error, just let it be.

2.5.1 Bad white matter segmentation

The segmentation of white matter from grey matter relies on white matter voxels having a consistent intensity. When you get a region of white matter that is especially bright or dark, this can cause Freesurfer to mislabel it. You'll often see this problem in the long, thin gyri located in anterior temporal lobe.

To fix this kind of error, you'll need to place some *control points* on or around the mislabeled voxels, and re-run `recon-all`. For a full description of the steps to take, see the Freesurfer control points tutorial[10].

Note: This type of error is often visible in several adjacent slices. If you see what looks like a white matter segmentation error in one isolated slice, you should probably leave it be. This is most likely just Freesurfer making a difficult partial-volume decision.

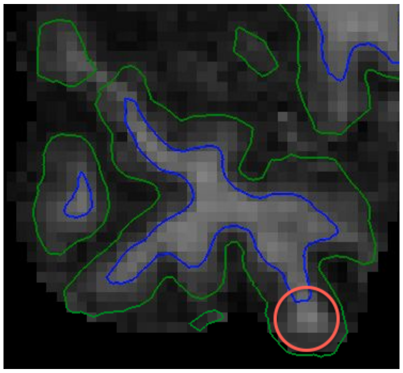

Figure 2.3: Example of bad white matter segmentation. A cluster of white matter voxels are classified as grey matter. It is difficult to see here, but the circled white matter voxels have intensity values well below the expected value, 110.

2.5.2 Skull segmented as grey matter

The skull stripping step can sometimes fail to remove parts of the skull or dura, especially around the eyes, around orbitofrontal cortex, and near the top of the head. This extra skull can sometimes get mislabeled as grey matter, which you'll have to fix by erasing the voxels manually. For a full description of the steps to take, see the Freesurfer skull-strip fix tutorial[11].

Note: Most of the time skull strip errors won't cause surface errors. No need to do anything in this case.

[8] http://surfer.nmr.mgh.harvard.edu/pub/docs/freesurfer_failuremodes_ani.ppt

[9] https://surfer.nmr.mgh.harvard.edu/fswiki/FsTutorial/TroubleshootingData

[10] https://surfer.nmr.mgh.harvard.edu/fswiki/FsTutorial/ControlPoints_freeview

[11] https://surfer.nmr.mgh.harvard.edu/fswiki/FsTutorial/SkullStripFix_freeview

The surfaces are all we care about.

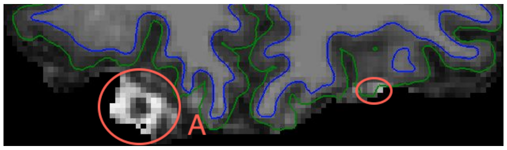

Figure 2.4: Skull strip errors that don't affect the cortical surface.

2.5.3 White Matter Lesions

We sometimes see subjects with small lesions scattered in their white matter. These lesions look like dark spots in the white matter volume. They are more common in older subjects, and usually having one means having many. When these lesions occur near the grey/white matter boundary, they can cause the white surface to errantly "dip" into (what should be) white matter.

Adding control points won't help in this situation. You can only use control points to correct intensity issues due to scanner bias. Putting a control point on a voxel tells Freesurfer "This voxel is white matter, so you should scale it's intensity (and that of neighboring voxels) to match other white matter voxels." Since lesion voxels don't actually contain white matter, adding control points here is the wrong choice. Instead, you will have to manually fill in the lesion area in the subject's wm.mgz volume. For complete instructions, see the Freesurfer white-matter edit tutorial[12].

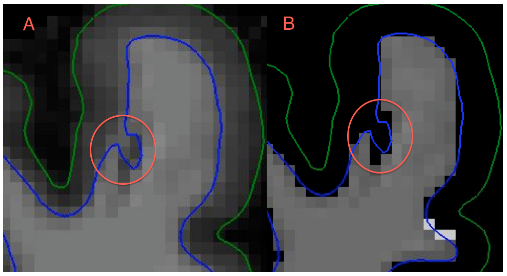

Figure 2.5: A white matter lesion causing a reconstruction error. **(A)** The error shown on the brainmask.mgz volume. **(B)** The error shown on the wm.mgz volume. Note how implausible the surface curvature is here. Also, see that the error is more obvious in **(B)** than in **(A)**.

2.5.4 Midline weirdness

The paths that the surfaces take through the midline structures and the corpus callosum are usually pretty nonsensical. **This is to be expected.** Don't worry about it. The lines here are more or less arbitrary, since there is no cortex to follow.

2.6 Quality control record keeping

It's very important to take notes while inspecting your data. This way you can return months later and feel confident that the quality assurance was done right. As a lab, we try to stick to a common format for quality control record keeping.

[12] https://surfer.nmr.mgh.harvard.edu/fswiki/FsTutorial/WhiteMatterEdits_freeview

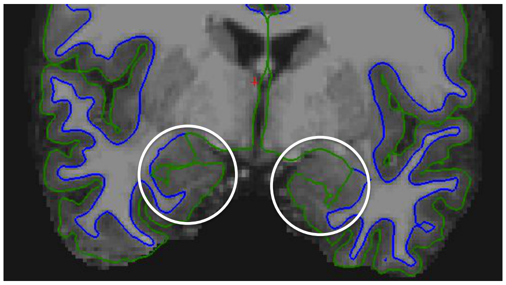

Figure 2.6: Uninterpretable pial surface through midline subcortical structures.

The quality control spreadsheet, usually called *recon_QC.xls* has two tabs. The first tab, called *Recon check*, is for keeping track of who checked which subjects when, and whether the reconstruction was ok, or required fixing.

Subject	Recon OK	Date	Checked by
BSYN_CB_02	yes	8/4/15	CL
BSYN_S_02	no	8/4/15	CL

The second tab, called *Recon errors*, is for keeping track of errors that you find in the surface reconstruction. Each row corresponds to an error. In the columns, you record the subject that you found the error in, a description of the error, the voxel coordinates for the error, what fix you applied (if any) and whether the fix worked. Keep in mind that this sheet is both for errors that need intervention, and for problem spots you just want to note, but don't need any further action. If you're not sure how to describe an error you see, or how to fix it, you should still take down at least the voxel coordinates. This way you can return to it later.

Subject	Error	Coordinates	Fix	Fixed
BSYN_CB_02	Questionable partial-volume decision	[87, 153, 87]	N/A	N/A
BSYN_S_02	Skull included around right eye	[154, 144, 175]	N/A	N/A
BSYN_S_02	Skull included in right orbital frontal	[160, 151, 156]	Removed extra skull	yes

2.7 Post reconstruction processing

After Freesurfer reconstruction is complete, some additional surface processing is required before you can move on to later analysis steps. These steps are carried out using the `postrecon` script, and include:

- Nonlinear registration of the subject's T1.mgz with the MNI 152 template using FSL's `fsl_anat`.
- Converting primary surfaces to Gifti format using `mris_convert`.
- Constructing HCP-style midthickness and inflated surfaces using `wb_command -surface-average` and `wb_command -surface-generate-inflated`.
- Downsampleing Gifti surfaces to the 32k_fs_LR mesh using `wb_command -surface-resample`. This reduces number of vertices from >100K to ~32K. After downsampling, each surface face occupies ~2mm^2, which more closely follows the resolution of the raw data.

`postrecon` only requires a subject ID argument:

```
postrecon $subj
```

Subject folders are modified in place. The results of the volumetric registration are written to mri/T1. The generated surfaces are added to the surf folder.