

# recon-all步骤解析

Alex / 2018-07-15 / [free\\_learner@163.com](mailto:free_learner@163.com) / [learning-archive.org](http://learning-archive.org)

更新于2023-07-14，主要是文字排版上的更新，内容基本保持不变。

FreeSurfer提供的recon-all脚本包含很多处理步骤，根据我自己的了解介绍其中每个步骤的选项和意义。这些步骤、解释以及具体选项，可以在FreeSurfer的[ReconAllTable](#)里找到。

## 1. 原始数据格式转换

```
recon-all -i ${data_dir}/001.dcm -s ${output_dir} -sd ${output_path}
```

-i 选项表示输入文件，输入文件是DICOM或者NIFTI格式的T1加权像（可以有多个 -i 选项，对应多个输入的情况），如果是DICOM格式（有多个文件），只需要给出其中一个文件即可；-s 选项表示输出文件的目录名，-sd 选项表示输出文件的路径，也就是recon-all的处理结果会存放在 \${output\_path}/\${output\_dir} 下。

```
export SUBJECTS_DIR=${output_path}
recon-all -i ${data_dir}/001.dcm -s ${output_dir}
```

这是另一种设置输出路径的方法，通过设置 SUBJECTS\_DIR 环境变量来设置输出文件的路径。

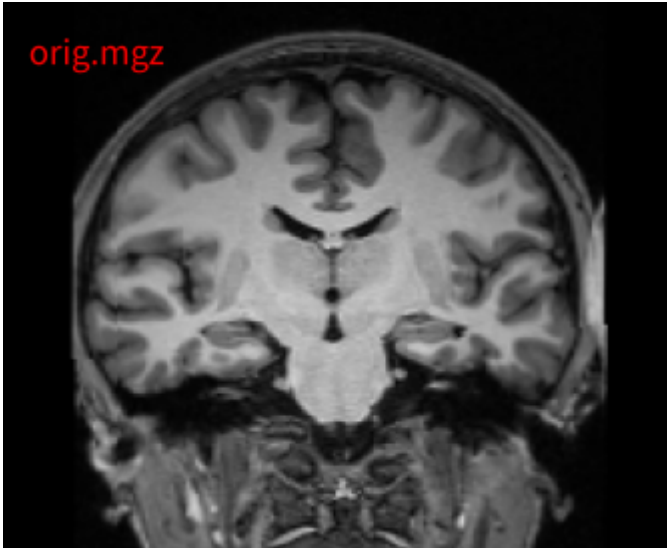
这一步会在输出目录下新建8个文件夹，分别为 label、mri、scripts、stats、surf、tmp、touch 和 trash；同时这一步将原始数据转换成MGZ格式；如果输入只有一个T1加权像，转换后的文件命名为 001.mgz，存放在 mri/orig 目录下。



## 2. 头动校正 (-motioncor)

```
recon-all -s ${output_dir} -sd ${output_path} -motioncor
```

`-motioncor` 选项表示头动校正，如果在 `mri/orig` 文件夹下有多个MGZ文件（比如 `001.mgz`、`002.mgz`，表示多个输入），则这一步将多个文件进行头动校正并平均，生成 `mri/rawavg.mgz` 文件；如果在 `mri/orig` 文件夹下只有一个 `001.mgz`文件（表示一个输入），则将直接将 `001.mgz` 复制成 `mri/rawavg.mgz`；另外这一步会调整原始文件的分辨率至 $256 \times 256 \times 128$  ( $1 \times 1 \times 1 \text{ mm}^3$ )，生成 `mri/orig.mgz` 文件。



### 3. 配准MNI模板（-talairach）

```
recon-all -s ${output_dir} -sd ${output_path} -talairach
```

`-talairach` 选项表示将原始文件线性配准到MNI305模板（`${FREESURFER_HOME}/avearge/mni305.cor.mgz`），生成转换矩阵 `mri/transforms/talairach.xfm`。

## MNI Transform File

talairach.xfm

```
% avi2talxfm
```

```
Transform_Type = Linear;
```

```
Linear_Transform =
```

```
0.911807 -0.024310 -0.023082 -5.648422
```

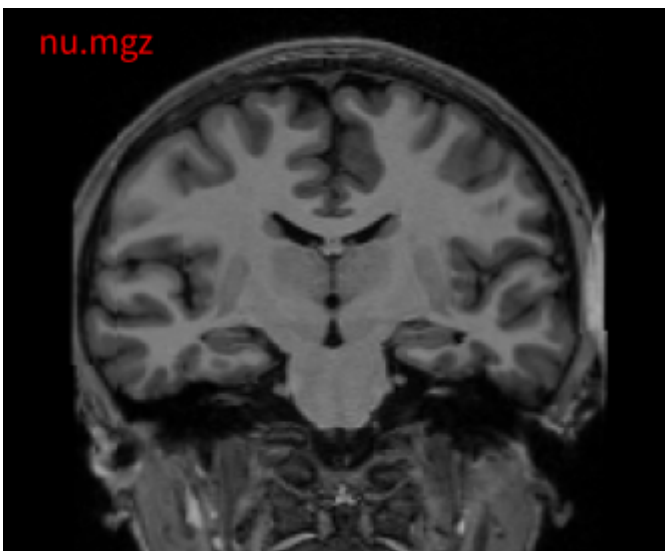
```
0.027797 1.059722 0.366447 -25.619926
```

```
0.004787 -0.230587 1.082518 -8.621109;
```

### 4. 偏差校正 (-nuintensitycor)

```
recon-all -s ${output_dir} -sd ${output_path} -nuintensitycor
```

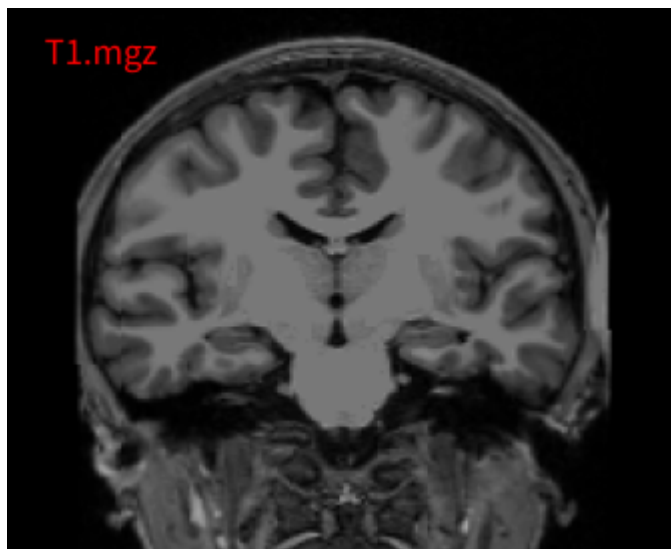
偏差校正就是校正由于磁场不均匀性引起的信号偏差，校正以后生成的文件名为 `nu.mgz`。



### 5. 信号标准化 (-normalization)

```
recon-all -s ${output_dir} -sd ${output_path} -normalization
```

这一步将白质区域的信号标准化为110左右，输出文件为 `T1.mgz`。



## 6. 颅骨剥离 (-skullstrip)

```
recon-all -s ${output_dir} -sd ${output_path} -skullstrip
```

颅骨剥离即去除非脑（non-brain）的组织，recon-all的颅骨剥离还是基于模板信息，这里使用的模板为 `$FREESURFER_HOME/average/RB_all_withskull_2016-05-10.vc700.gca`，生成 `brainmask.mgz` 文件。



## 7. 配准GCA模板 (-gcareg)

```
recon-all -s ${output_dir} -sd ${output_path} -gcareg
```

将偏差校正后的图像（`nu.mgz`）配准到GCA模板（`$FREESURFER_HOME/average/RB_all_2016-05-10.vc700.gca`），生成配准文件 `transforms/talairach.lta`。

```
# transform file transforms/talairach.lta
# created by huyang on Wed Jul 11 08:40:49 2018

type      = 0 # LINEAR_VOX_TO_VOX
nxforms   = 1
mean      = 125.0000 129.0000 125.0000
sigma     = 10000.0000
1 4 4
9.231123328208923e-01 -1.875844597816467e-02 4.137422889471054e-02 7.067777633666992e+00
7.795047014951706e-03 1.130222439765930e+00 1.920714676380157e-01 -4.282280731201172e+01
-3.712830692529678e-02 -2.209463566541672e-01 1.032298088073730e+00 1.375670433044434e+01
0.000000000000000e+00 0.000000000000000e+00 0.000000000000000e+00 1.000000000000000e+00
```

## 8. 调整信号 (-canorm)

```
recon-all -s ${output_dir} -sd ${output_path} -canorm
```

由于这一步FreeSurfer并没有提供详细的解释，我猜测这一步和第5步的区别是在去除颅骨以后，再次根据GCA模板调整信号值，输出为 `norm.mgz`。



## 9. 非线性配准GCA模板 (-careg)

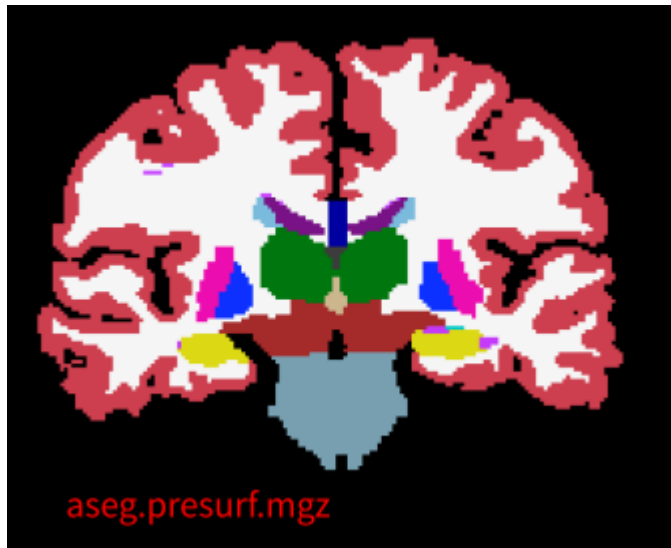
```
recon-all -s ${output_dir} -sd ${output_path} -careg
```

将 `norm.mgz` 非线性配准到GCA模板（`$FREESURFER_HOME/average/RB_all_2016-05-10.vc700.gca`），生成配准文件 `transforms/talairach.m3z`。

## 10. 标记皮下核团 (-calabel)

```
recon-all -s ${output_dir} -sd ${output_path} -calabel
```

根据GCA模型标记出皮下核团（以及胼胝体）的区域，生成 `aseg.presurf.mgz`。



#### 11. 信号标准化 (-normalization2)

```
recon-all -s ${output_dir} -sd ${output_path} -normalization2
```

目的与第5步相同，即将白质区域信息标准化110左右；不同之处在于使用颅骨剥离和皮下核团标记后的结果优化这一步骤，生成 `brain.mgz`。



#### 12. -maskbfs

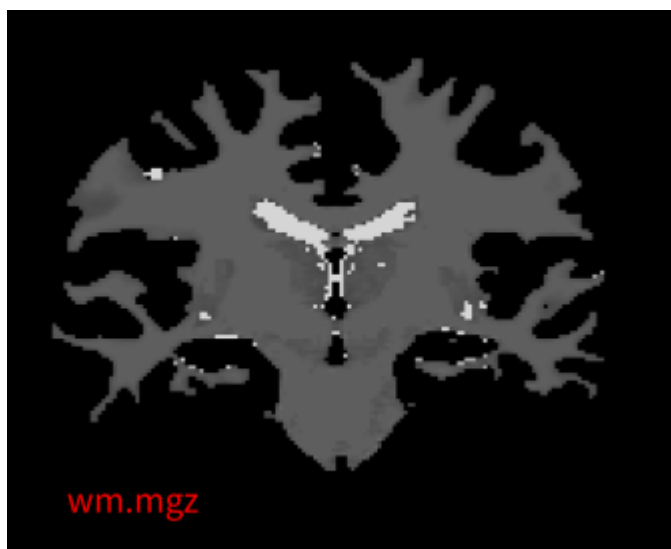
```
recon-all -s ${output_dir} -sd ${output_path} -maskbfs
```

这一步不太清楚有什么作用，生成 `brain.finalsurfs.mgz`，跟 `brain.mgz` 似乎是完全一样的。

### 13. 白质分割 (-segmentation)

```
recon-all -s ${output_dir} -sd ${output_path} -segmentation
```

白质分割就是找到白质和灰质的边界，所以这里的“白质”也包括脑室和皮下核团等区域，生成 `wm.mgz`。



### 14. Cut-Fill (-fill)

```
recon-all -s ${output_dir} -sd ${output_path} -fill
```

去掉中脑（mid-brain）、分开左右脑；将左脑数值调整为255，右脑数值调整为127；输出为 `fiil.mgz`。

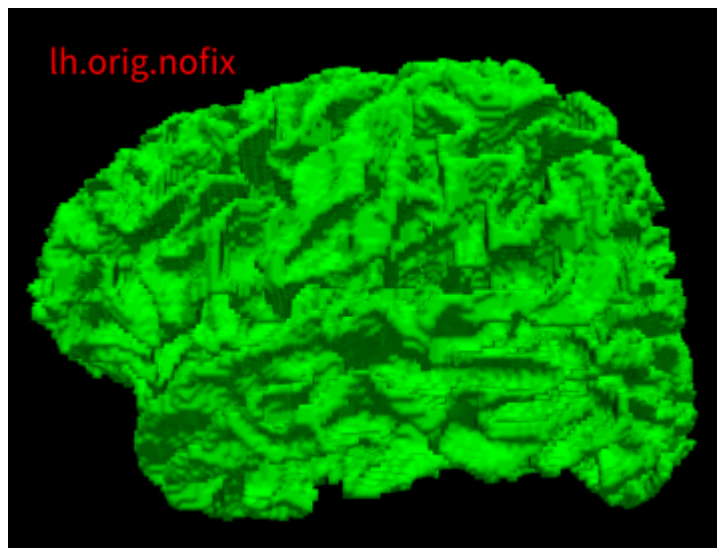


### 15. 创建灰白质皮层 (-tessellate)

```
recon-all -s ${output_dir} -sd ${output_path} -tessellate
```

用小三角形覆盖灰白质的边界（即内皮层），生成原始的左右半球皮层

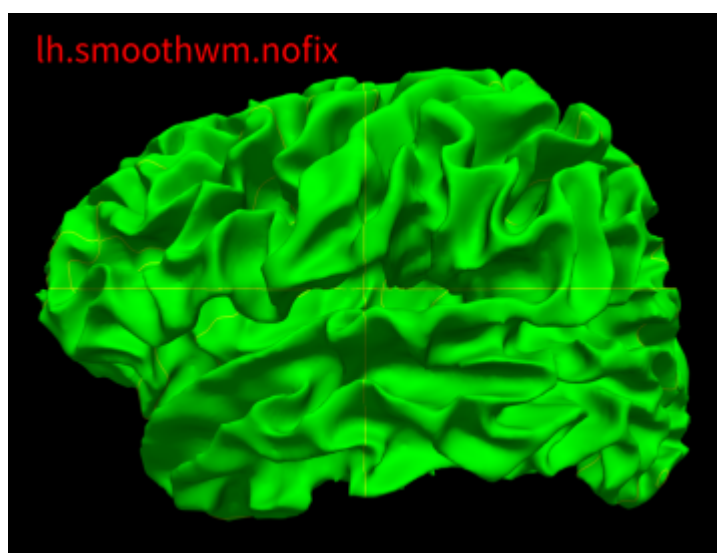
lh.orig.nofix/rh.orig.nofix。



#### 16. 平滑 (-smooth1)

```
recon-all -s ${output_dir} -sd ${output_path} -smooth1
```

上一步生成的皮层是非常不平滑的，因为灰白质的边界在体素上是相互垂直的，因此对原始的皮层进行平滑，生成 lh.smoothwm.nofix，l 表示l或r。

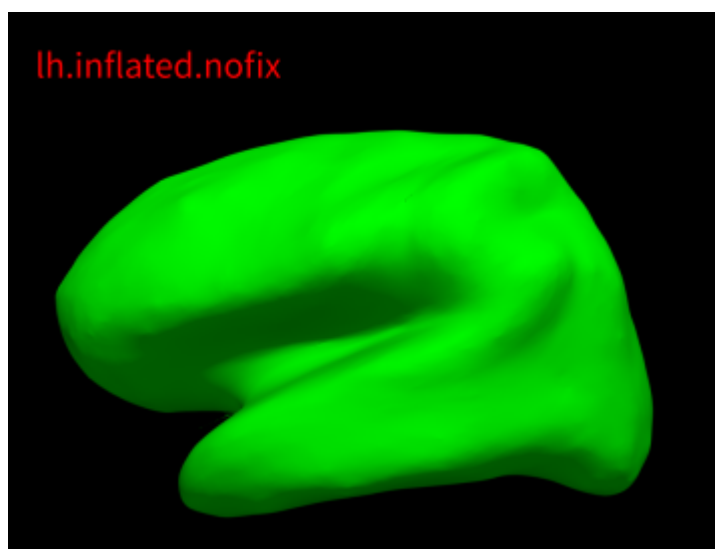


#### 17. inflation (-inflate1)



```
recon-all -s ${output_dir} -sd ${output_path} -inflate1
```

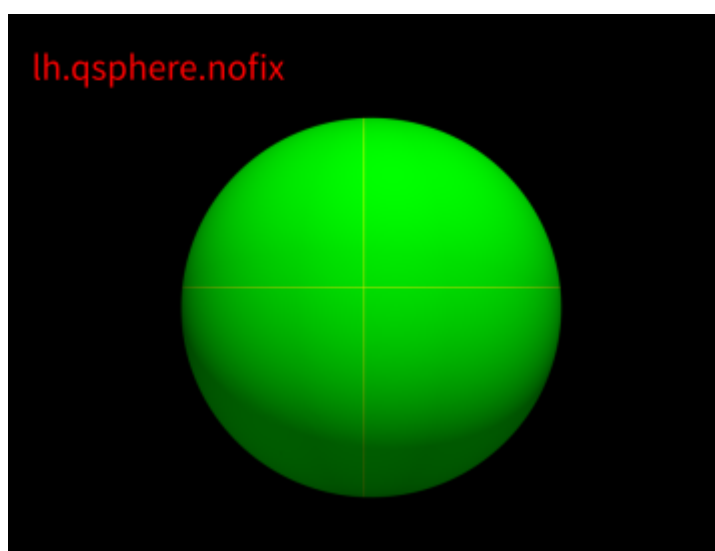
将皮层膨胀起来，使得折叠的沟能显示出来，生成 `?h.inflated.nofix`。



## 18. 球化 (-qsphere)

```
recon-all -s ${output_dir} -sd ${output_path} -qsphere
```

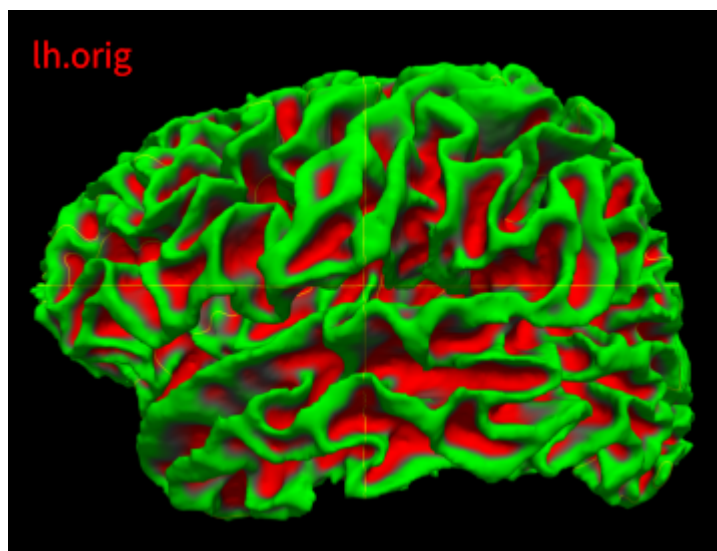
将上一步膨胀后的皮层进一步“变成”球形（具体的数学原理并不清楚），用于后面自动化修补皮层的缺陷，生成 `?h.qsphere.nofix`。



## 19. 修补皮层缺陷 (-fix)

```
recon-all -s ${output_dir} -sd ${output_path} -fix
```

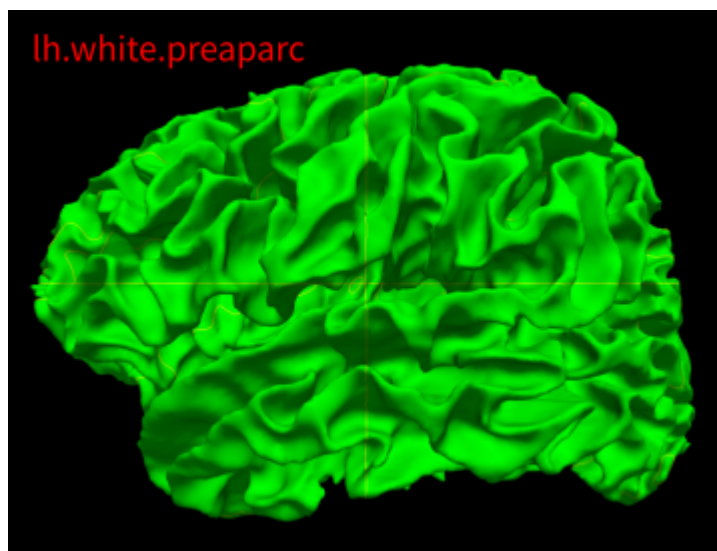
修补原始皮层的拓扑学缺陷，生成 `h.orig`。

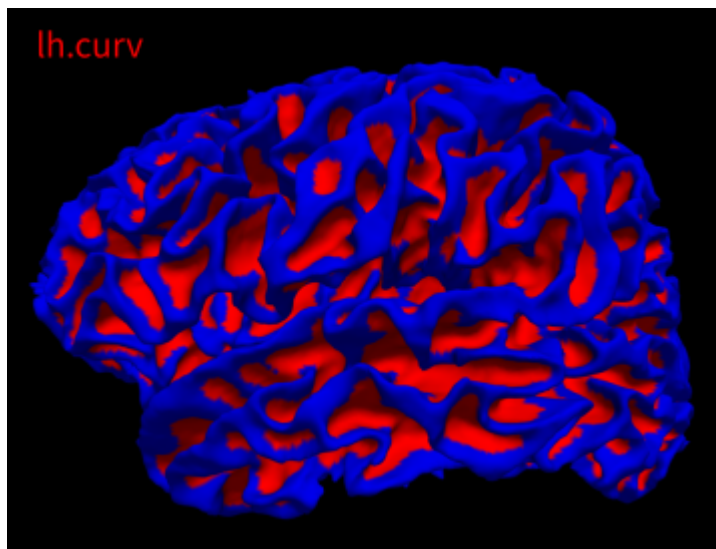


## 20. 细化白质皮层 (-white)

```
recon-all -s ${output_dir} -sd ${output_path} -white
```

通过微调原始皮层，使皮层沿着灰白质信号梯度的方向，从而细化白质皮层，生成 `h.white.preaparc` 以及皮层曲率 `h.curv`。

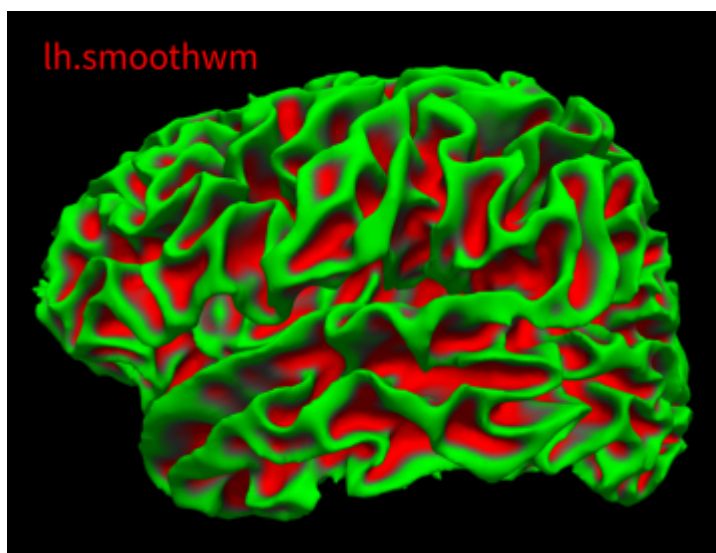




## 21. 平滑 (-smooth2)

```
recon-all -s ${output_dir} -sd ${output_path} -smooth2
```

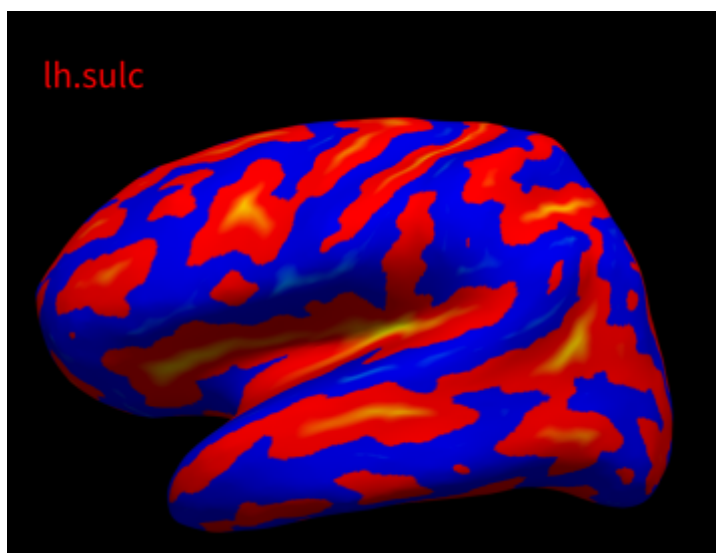
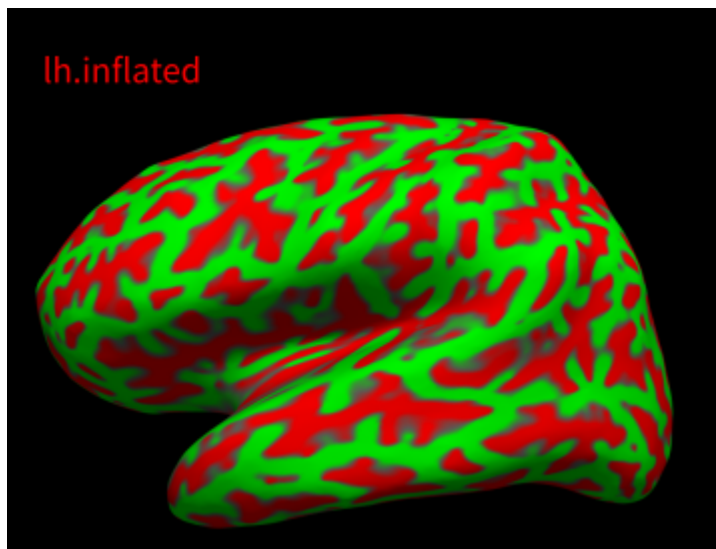
同步骤16，生成 `?h.smoothwm`。



## 22. inflation (-inflate2)

```
recon-all -s ${output_dir} -sd ${output_path} -inflate2
```

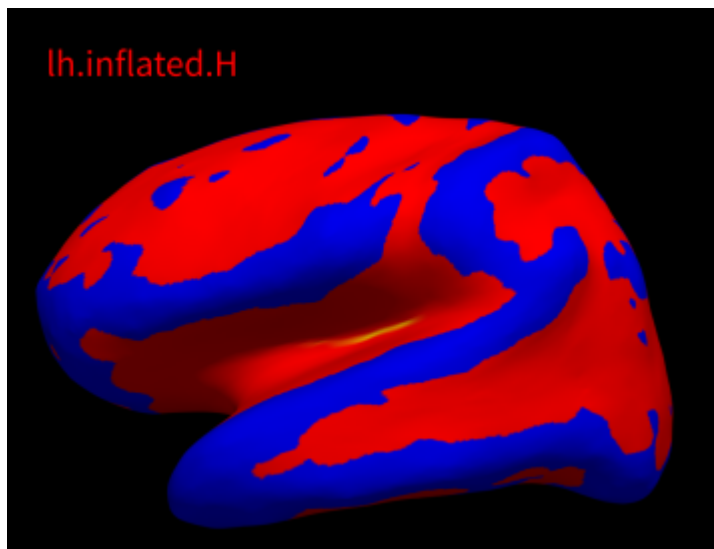
同步骤17，生成 `?h.inflated` 和 `?h.sulc`。



### 23. 曲率计算 (-curvHK)

```
recon-all -s ${output_dir} -sd ${output_path} -curvHK
```

计算平均曲率和高斯曲率，生成 `?h.white.H/?h.white.K` 以及 `?h.inflated.H/?h.inflated.K`，`.H` 表示平均曲率文件，`.K` 表示高斯曲率文件。



## 24. 曲率统计指标 (-curvstats)

```
recon-all -s ${output_dir} -sd ${output_path} -curvstats
```

生成一些关于曲率的统计指标 `stats/?h.curv.stats`。

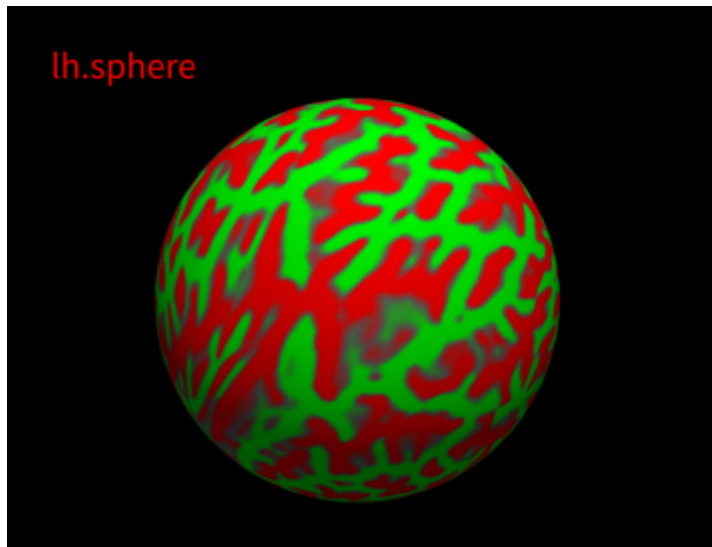
```
Raw <mean> +- <std> (using 'lh.curv'):
```

Raw Min:	-4.90902	at vertex 110332
Raw Max:	4.24829	at vertex 112560
Raw Curvature Calculation Type:	discrete	
Raw Mean Vertex Separation (Whole Surface):	0.85405 +- 0.22505	mm
Raw Total Surface Area:	96610.17969	mm^2
Raw Total Number of Vertices:	157683	
Raw Average Vertex Area (Whole Surface):	0.61269	mm^2
Raw Natural Surface Integral:	-1718.86230	
Raw Rectified Surface Integral:	12697.10840	
Raw Positive Surface Integral:	5489.12305	
Raw Negative Surface Integral:	7207.98535	lh.curv.stats

## 25. 球化 (-sphere)

```
recon-all -s ${output_dir} -sd ${output_path} -sphere
```

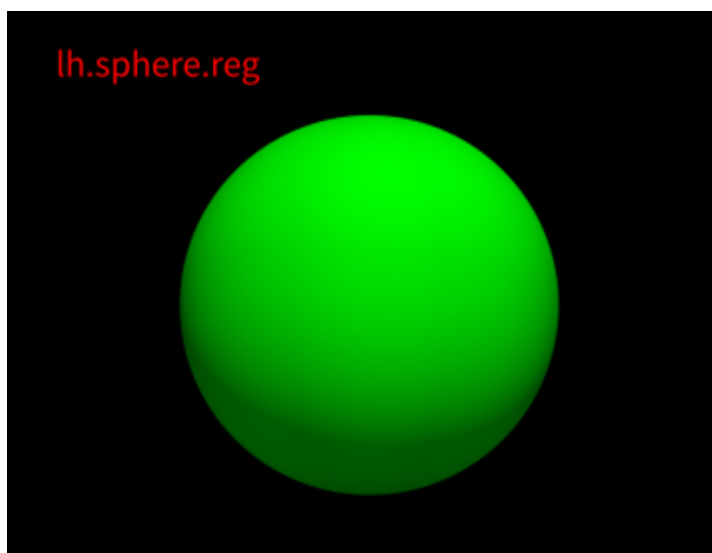
类似于第18步，用于下一步配准到球形模板上 (atlas)，生成 `?h.sphere`。



## 26. 配准到球形模板 (-surferg)

```
recon-all -s ${output_dir} -sd ${output_path} -surferg
```

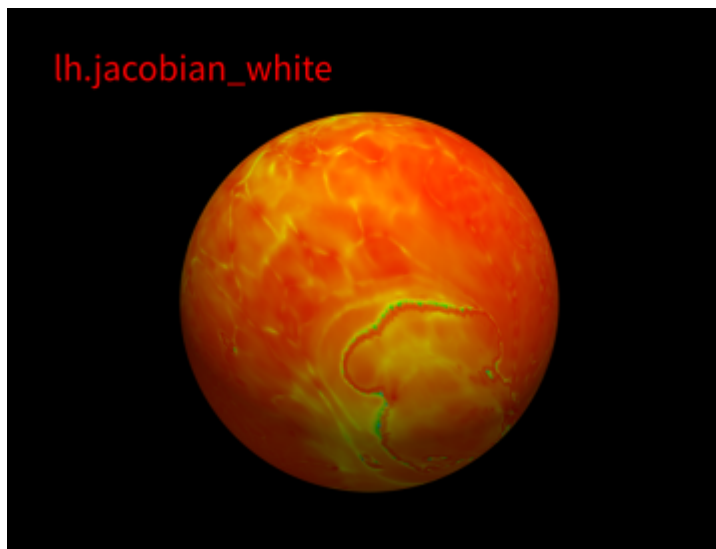
将个体球化的皮层配准到球形的模板，首先使用 `?h.sulc` 进行粗配准，然后利用 `?h.curv` 进行精配准，生成 `?h.sphere.reg`。



## 27. 变化率 (-jacobian\_white)

```
recon-all -s ${output_dir} -sd ${output_path} -jacobian_white
```

计算白质皮层在配准到球形模板中的变形程度，生成 `?h.jacobian_white`。



## 28. 组水平曲率 (-avgcurv)

```
recon-all -s ${output_dir} -sd ${output_path} -avgcurv
```

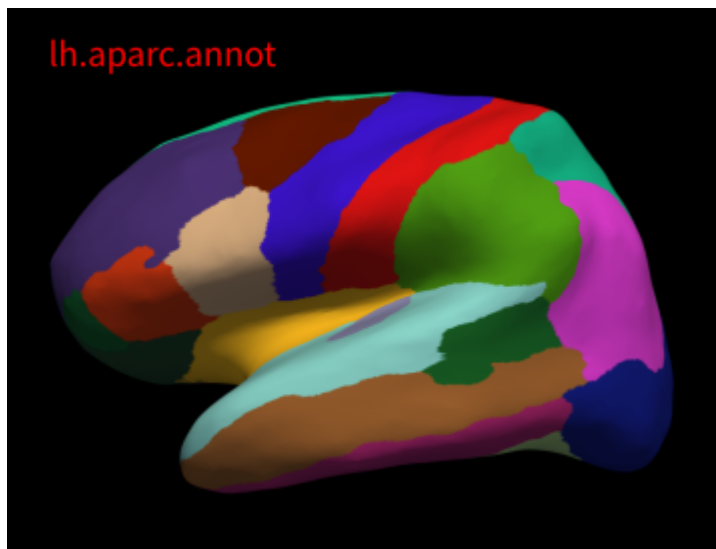
根据上一步的配准结果，将模板的曲率转换到个体水平，生成 `?h.avg_curv`。



## 29. 皮层标记 (-cortparc)

```
recon-all -s ${output_dir} -sd ${output_path} -cortparc
```

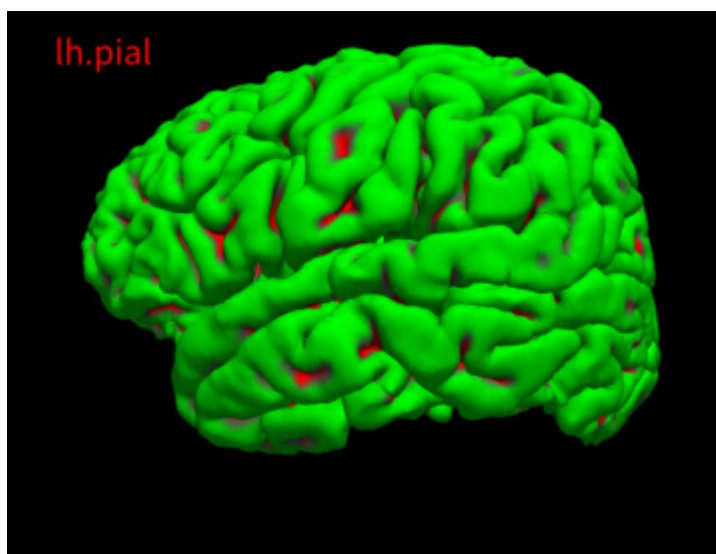
将皮层根据Desikan-Killiany Atlas分区模板标记出不同区域，生成 `label/?h.aparc.annot`。



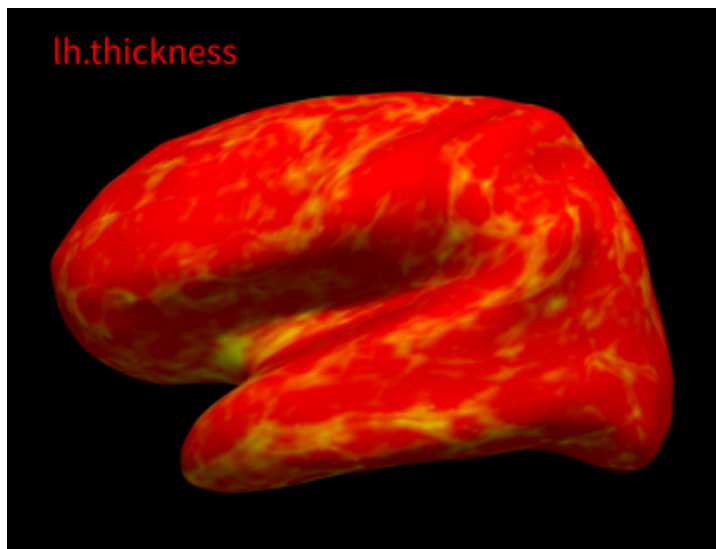
### 30. 灰质-脑膜皮层 (-pial)

```
recon-all -s ${output_dir} -sd ${output_path} -pial
```

生成灰质与脑膜边界的皮层（即外皮层） `?h.pial` 以及相应的曲率、面积文件 `?h.curv.pial/?h.area.pial`，通过外皮层和内皮层，得到皮层厚度文件 `?h.thickness`。



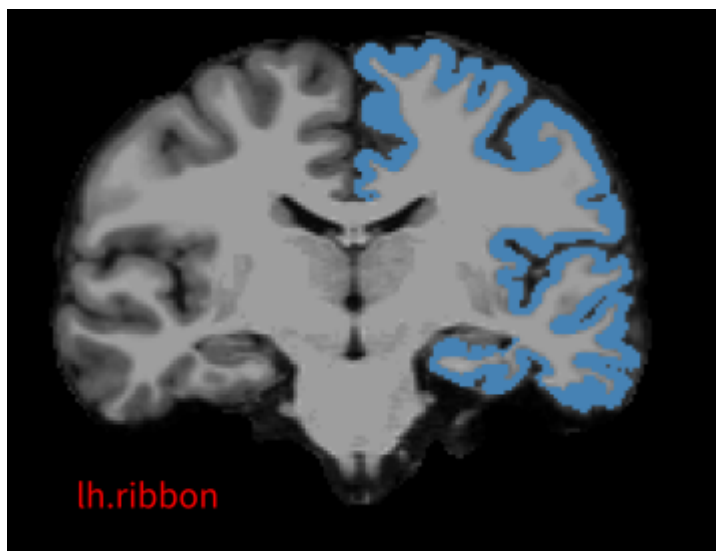




### 31. 灰质mask (-cortribbon)

```
recon-all -s ${output_dir} -sd ${output_path} -cortribbon
```

根据内外皮层，生成整个灰质区域的mask文件 `h.ribbon.mgz/ribbon.mgz`。



### 32. 皮层分区统计 (-parcstats)

```
recon-all -s ${output_dir} -sd ${output_path} -parcstats
```

根据第29步的分区结果，计算各个区域的结构指标，比如皮层厚度、平均曲率等，生成 `stats/?h.aparc.stats`。

lh.aparc.stats

# ColHeaders	StructName	NumVert	SurfArea	GrayVol	ThickAvg	ThickStd	MeanCurv
bankssts		1458	1095	2550	2.490	0.433	
caudalanteriorcingulate		963	664	1605	2.292	0.516	
caudalmiddlefrontal		4512	2971	8920	2.720	0.466	
cuneus		3016	1935	3605	1.820	0.431	
entorhinal		708	506	1928	3.171	0.858	
fusiform		4810	3300	9967	2.754	0.512	
inferiorparietal		7887	5306	14683	2.561	0.510	
inferiortemporal		6018	4173	13522	2.822	0.690	

### 33. 皮层标记 (-cortparc2)

```
recon-all -s ${output_dir} -sd ${output_path} -cortparc2
```

同步骤29，将皮层根据Destrieux Atlas分区模板标记出不同区域，生成 ?

h.aparc.a2009s.annot 。

### 34. 皮层分区统计 (-parcstats2)

```
recon-all -s ${output_dir} -sd ${output_path} -parcstats2
```

根据第33步的分区结果，计算各个区域的结构指标，比如皮层厚度、平均曲率等，生成 stats/?

h.aparc.a2009s.stats 。

### 35. 皮层标记 (-cortparc3)

```
recon-all -s ${output_dir} -sd ${output_path} -cortparc3
```

同步骤29，将皮层根据DKT Atlas分区模板标记出不同区域，生成 ?

h.aparc.DKTatlas40.annot 。

### 36. 皮层分区统计 (-parcstats3)

```
recon-all -s ${output_dir} -sd ${output_path} -parcstats2
```

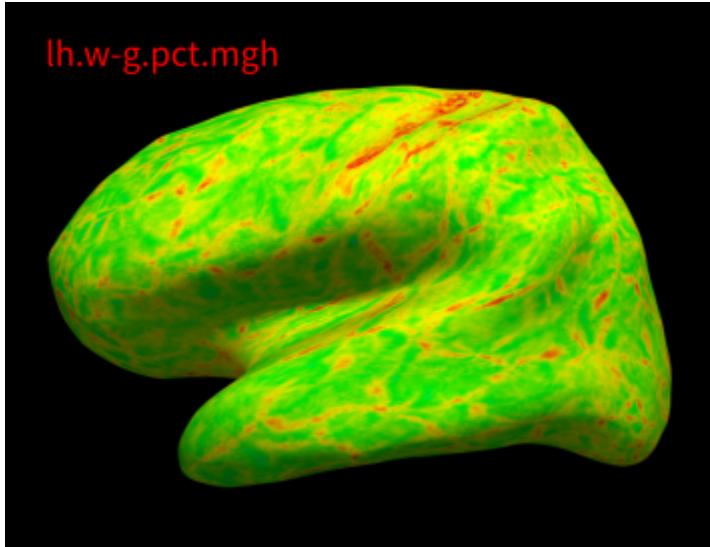
根据第35步的分区结果，计算各个区域的结构指标，比如皮层厚度、平均曲率等，生成 stats/?

h.aparc.a2009s.stats 。

### 37. 灰白质比例 (-pctsurfcon)

```
recon-all -s ${output_dir} -sd ${output_path} -pctsurfcon
```

计算灰质和白质的比例： $pct = 100 * (W - G) / [0.5 * (W + G)]$ ，生成 `surf/?h.w-g.pct.mgh` 和 `stats/?h.w-g.pct.stats`。



### 38. -hyporelabel

```
recon-all -s ${output_dir} -sd ${output_path} -hyporelabel
```

这一步FreeSurfer的wiki也没有做明白的解释，生成 `aseg.presurf.hypos.mgz`。

### 39. volume标记 (-aparc2aseg)

```
recon-all -s ${output_dir} -sd ${output_path} -aparc2aseg
```

根据皮层标记的结果标记volume，即将分区从皮层（surface）转换到体积（volume）上，是同一信息在不同数据形式上的表征，对应于不同分区模板，分别生成 `aparc+aseg.mgz`、`aparc.a2009s+aseg.mgz` 和 `aparc.DKTatlas+aseg.mgz`。



#### 40. 皮下核团分区统计 (-segstats)

```
recon-all -s ${output_dir} -sd ${output_path} -segstats
```

计算皮下核团分区结构指标统计值，包括体积均值、标准差等，生成 `stats/aseg.stats`。

#### 41. 白质分区 (-wmparc)

```
recon-all -s ${output_dir} -sd ${output_path} -wmparc
```

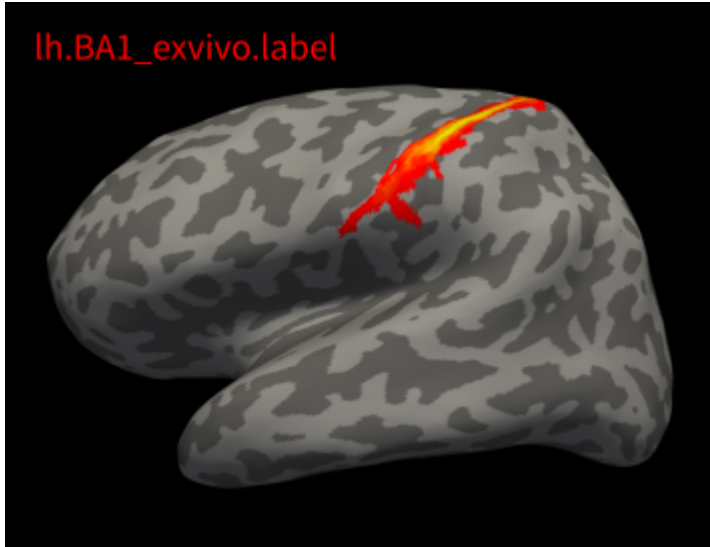
对白质进行分区，并计算各分区的统计指标，生成 `wmparc.mgz` 和 `wmparc.stats`。



#### 42. Brodmann分区 (-balabels)

```
recon-all -s ${output_dir} -sd ${output_path} -balabels
```

生成Brodmann分区的标签（label）文件 `label/?h.BA*_exvivo.label`；标签文件和分区文件（annot）的区别在于一个标签文件只表示一个区域，一个分区文件包含了所有区域的分区信息。



## 总结

从目前我了解的情况来看，recon-all的处理包括volume和surface两部分，volume处理的结果放在mri文件夹下，surface处理的结果放在surf文件夹下，label文件夹里放着分区和标签文件，stats文件夹存放的各个分区结构指标的统计值；重建皮层的思路是获得白质与灰质边界（内皮层），为了获得灰白质边界，就需要颅骨剥离、组织分割等处理；重建了灰白质皮层和灰质-脑膜皮层（外皮层）才能计算一些结构指标，比如皮层厚度。还有很多不太清楚的地方，比如什么是GCA模型，不同文件的格式区别（lh.white/lh.thickness）等。