

第1部 (3)

recon-all -autorecon3

筑波大学医学医療系
精神医学

根本 清貴

本勉強会のルール

- ターミナルでタイプするものは、青色で表示
 - 例: `freeview -v bert/mri/T1.mgz`
- 入力する必要のないコマンドは、紫色で表示
- スクリプトに記載してある内容は緑色で表示
- ショートカットキーの組み合わせは、`Ctrl+C` のように水色で色づけ
- GUIでの動作は、`View → Show Control Panel` のように紫で色づけ
- # 以降は、解説でありタイプする必要はない
- 「フォルダ」=「ディレクトリ」

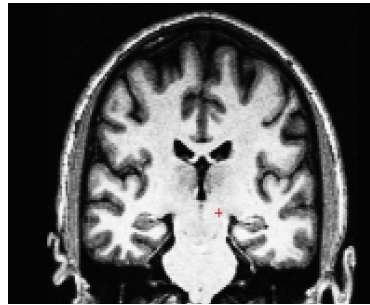
本セクションの目標

- `recon-all -autorecon3` の各ステップを理解する
- 実際にどのようなファイルが生成されるのかを確認する
- これが理解できれば、どのようなエラーが生じるかがわかる

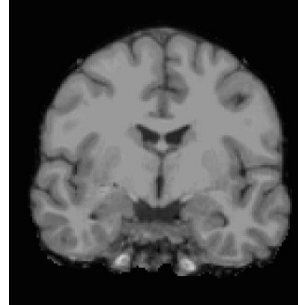
本セクションのチートシート

- `nisg-202001/docs` にある `ex3.html` をダブルクリック
- コマンドの意味も記載されている

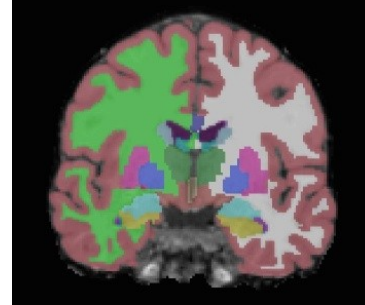
recon-allの概要



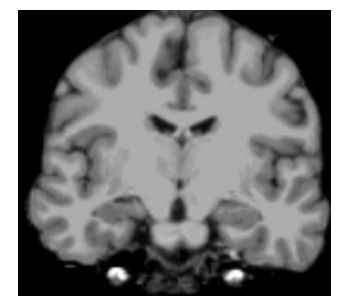
入力画像
T1強調画像



頭蓋骨除去



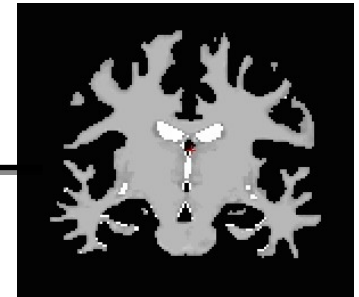
Volumeの
ラベリング



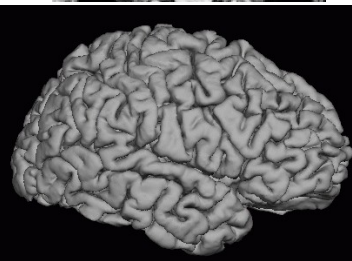
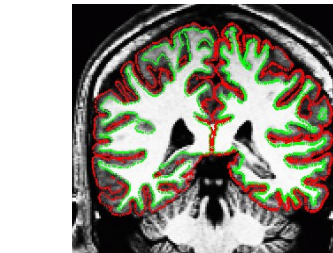
信号値の
正規化

-autorecon1

-autorecon2 (下に続く)



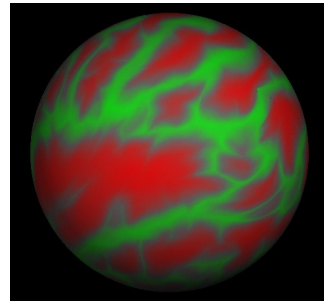
白質の分離



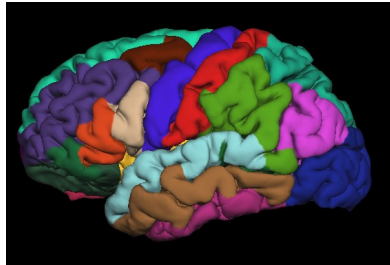
Surfaceの抽出

-autorecon2

FreeSurferスライドを改変



Surfaceアトラスへの
レジストレーション



脳回のラベリング

-autorecon3



統計!

本セクションのリソース

- recon-all のソースを精読
 - 最終的にもっとも信頼できるのは、プログラム本体
- <https://surfer.nmr.mgh.harvard.edu/fswiki/ReconAllTableStableV6.0> を参照

SUBJECTS_DIRの設定

- Lin4Neuroの場合

`nisg=/media/sf_share/nisg-202001`

- MacOSの場合

`nisg=~/.git/nisg-202001`

`SUBJECTS_DIR=$nisg/subjects`

`cd $SUBJECTS_DIR/ernie`

recon-all -autorecon3 で 使われるdirectives

- 24 -sphere
- 25 -surfreg
- 26 -jacobian_white
- 27 -avgcurv
- 28 -cortparc
- 29 -pial
- 30 -cortribbon
- 31 -parcstats
- 32 -cortparc2
- 33 -parcstats2
- 34 -cortparc3
- 35 -parcstats3
- 36 -pctsurfcon
- 37 -hyporelabel
- 38 -aparc2aseg
- 39 -apas2aseg
- 40 -segstats
- 41 -wmparc
- 42 -balabels

24. -sphere

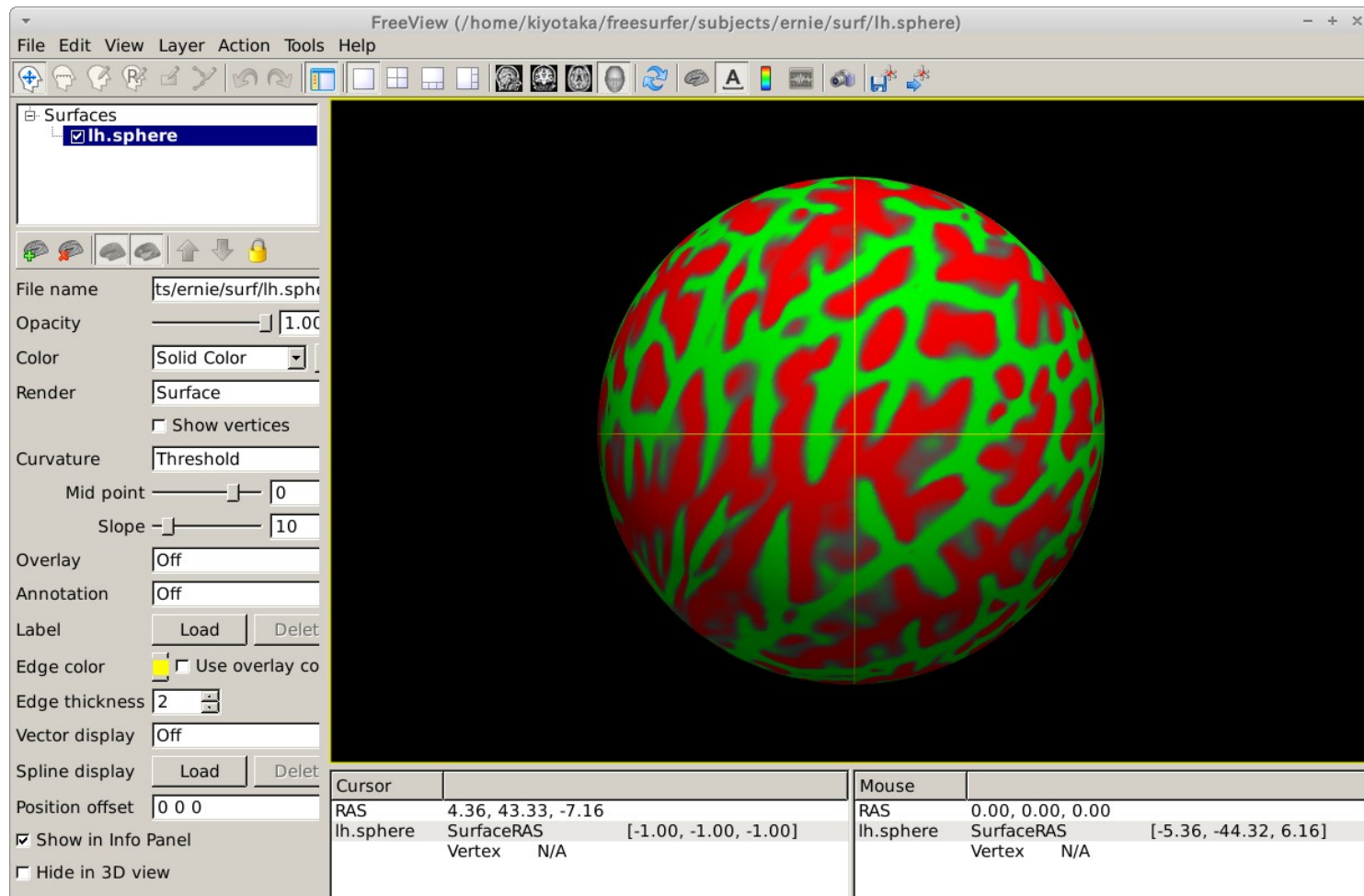
- -autorecon2 で作成された ?h.inflated と ?h.smoothwm を用いて球体に変形
 - 球体アトラスにあわせるために必要なプロセス
 - 入力画像: ?h.inflated, ?h.smoothwm
 - 出力画像: **?h.sphere**

lh.sphere の確認

```
cd $SUBJECTS_DIR/ernie/surf
```

```
freeview -f lh.sphere \  
-layout 1 -viewport 3d
```

lh.sphere



- inflated 画像が球体に変形したイメージを持つと理解しやすいかも

25. -surfreg

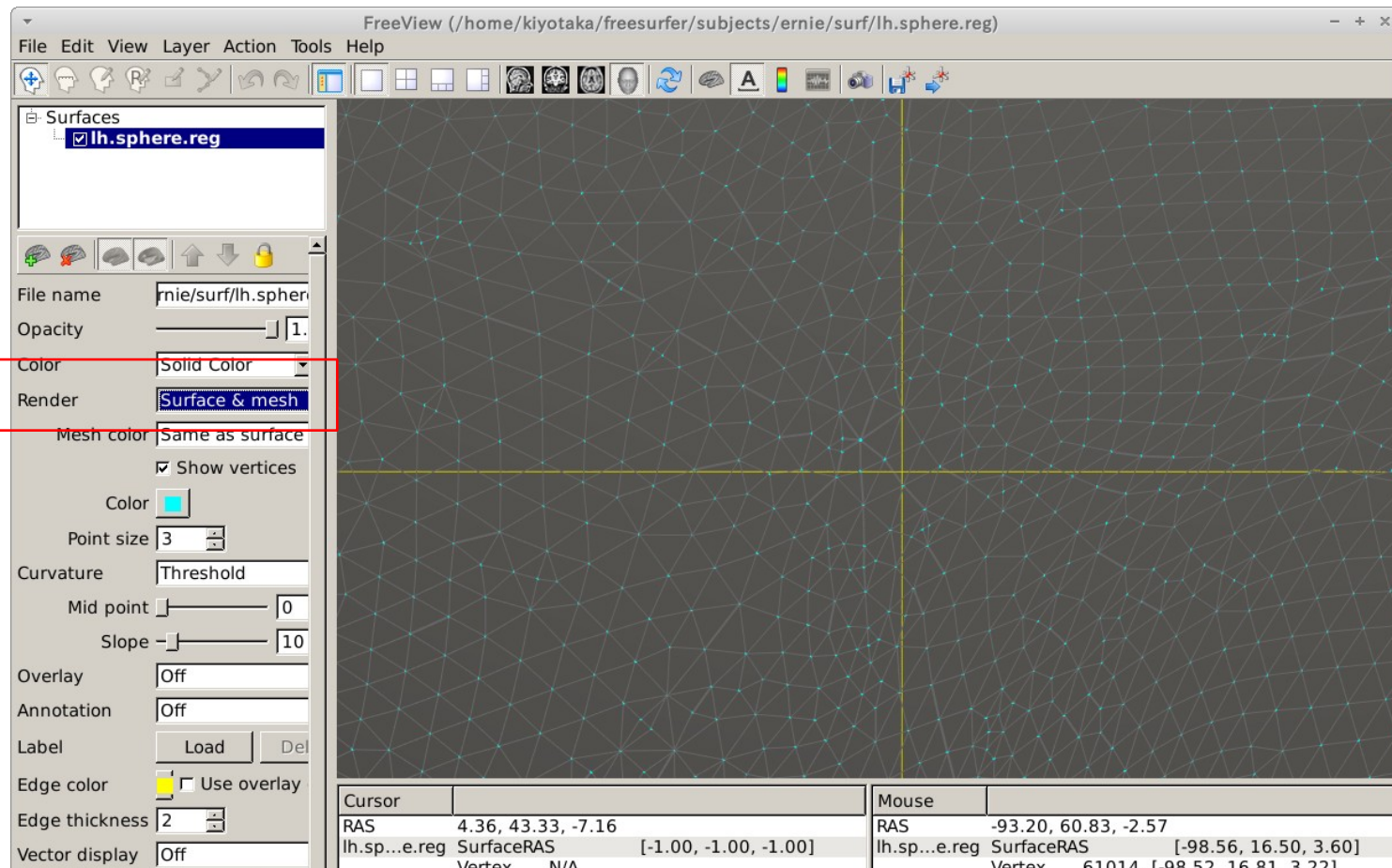
- `?h.sphere` を球体アトラスにあわせこみ
 - `?h.sulc` を使って、脳溝のパターンをもとに球体アトラスにおおまかにあわせ、`?h.curv` を使って、さらにあわせこむ
 - 入力画像: `?h.sphere`, `?h.smoothwm`
 - 出力画像: `?h.sphere.reg`

lh.sphere.reg の確認

```
freeview -f \  
  lh.sphere.reg:vertex=1 \  
  -layout 1 -viewport 3d -zoom 10
```

- lh.sphere.reg の頂点を表示
- 10倍に拡大

lh.sphere.reg



- Renderから、**Surface & mesh** を選択
- 三角形が維持されていることを確認

頂点数が変わっていないことの確認

- `lh.smoothwm`, `lh.inflated`, `lh.sphere`, `lh.sphere.reg` の頂点数を確認

`mrisc_euler_number lh.smoothwm`

`mrisc_euler_number lh.inflated`

`mrisc_euler_number lh.sphere`

`mrisc_euler_number lh.sphere.reg`

頂点数は維持されている

`mrisc_euler_number lh.smoothwm`

$$- v - e + f = 153341 - 460017 + 306678$$

`mrisc_euler_number lh.inflated`

$$- v - e + f = 153341 - 460017 + 306678$$

`mrisc_euler_number lh.sphere`

$$- v - e + f = 153341 - 460017 + 306678$$

`mrisc_euler_number lh.sphere.reg`

$$- v - e + f = 153341 - 460017 + 306678$$

- 見た目の形状は変わりこそすれ、三角形は変わっていない!
- これがSurface Based Morphometryのキモ

26. -jacobian_white

- white surface を球面アトラスに非線形変換を行う際にどれだけ歪ませる必要があったかを計算
 - curvと同じフォーマットになっている
 - 入力画像: ?h.white.preaparc, ?h.sphere.reg
 - 出力ファイル: **?h.jacobian_white**
- Jacobian
 - >1 進展; $=1$ 変化なし <1 縮小

Jacobian determinant の確認

```
cd .. # $SUBJECTS_DIR/ernie に移動
```

```
freeview -v mri/brain.mgz -f \  
surf/lh.smoothwm:overlay=lh.jacobian_white \  
-layout 2 -viewport coronal
```

- lh.jacobian.white は画像ではなく、信号値をもっている情報のため、overlayする
 - Freeviewのoverlayは画像のoverlayではないことに注意

jacobian_white

FreeView (/home/kiyotaka/freesurfer/subjects/ernie/surf/lh.smoothwm)

File Edit View Layer Action Tools Help

Volumes
└ ☒ brain
Surfaces
└ ☒ lh.smoothwm

File name: /nie/surf/lh.smoothwm
Opacity: 1.00
Color: Solid Color
Render: Surface
Show vertices: ☐
Curvature: Binary
Mid point: 0
Overlay: lh.jacobian_white
Configure Remove
Annotation: Off
Label: Load Delete
Edge color: ☒ Use overlay color
Edge thickness: 2
Vector display: Off
Spline display: Load Delete
Position offset: 0 0 0
Show in Info Panel: ☒
Hide in 3D view: ☐

Cursor

RAS	-25.28, 44.32, 2.21
TkRe...ain)	-30.64, 0.00, 8.38
Talai...rain)	167.12, 107.18, 79.59
brain	67 [159, 120, 128]
lh.s...thwm	SurfaceRAS [-30.64, 0.00, 8.38]
Vertex	98649 [-30.52, -0.02, 8.63]
lh.jacobian_white	1.72175

Mouse

RAS	-50.41, 44.32, -5.60
TkRe...ain)	-55.77, 0.00, 0.57
Talai...rain)	194.77, 115.82, 77.52
brain	63 [184, 127, 128]
lh.s...thwm	SurfaceRAS [-55.77, 0.00, 0.57]
Vertex	98750 [-55.30, -0.70, 1.00]
lh.jacobian_white	1.59113

- 移動距離が多そうなところな境界にマウスをもっていき、そのjacobian whiteを確認

27. -avgcurv

- 球面アトラスの曲率を個人の脳に対してリサンプリングを行う
 - 平均的な脳回のfoldingパターンを表示できる
 - 入力画像: `?h.sphere.reg`
 - 出力ファイル: `?h.avg_curv`

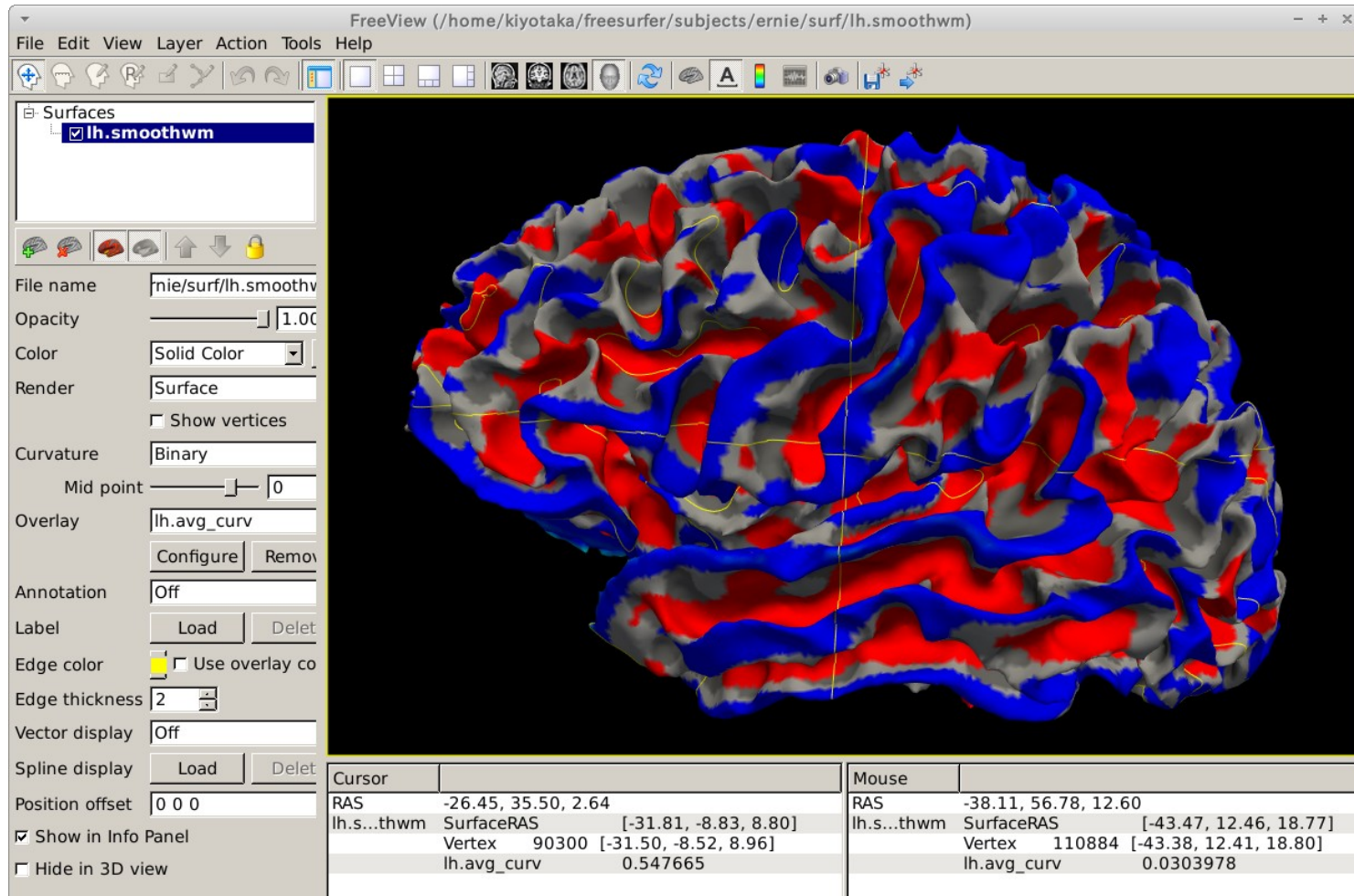
平均的な folding pattern を個人の white surface に表示

```
cd surf # $SUBJECTS_DIR/ernie/surf に移動
```

```
freeview -f lh.smoothwm:overlay=lh.avg_curv \  
-layout 1 -viewport 3d
```

- lh.avg_curv は画像ではなく、信号値をもっている情報のため、overlayする

lh.avg_curv



- アトラスの脳回が青、脳溝が赤で表示されている
- あくまでもアトラスであり、個人には完全に合致していない

28. -cortparc

- 脳領域のラベルを、個々の領域に割り当てる
 - ここでは、Desikan Killiany (DK) アトラスを割り当てる
 - 三角形の数は変わっていない → smoothwm, inflated, sphere いずれにも適用できる
 - 入力画像: `aseg.presurf.mgz`,
`?h.cortex.label`, `?h.sphere.reg`
 - 出力ファイル: `label/?h.aparc.annot`

DK アトラスの確認

- DKアトラスのラベルを確認
- annotationファイルは、指定する際に lh, rh はつけなくてよい
 - 指定されたファイルから左右を自動で判定している
- lh.smoothwm, lh.inflated, lh.sphere すべてで見てみる

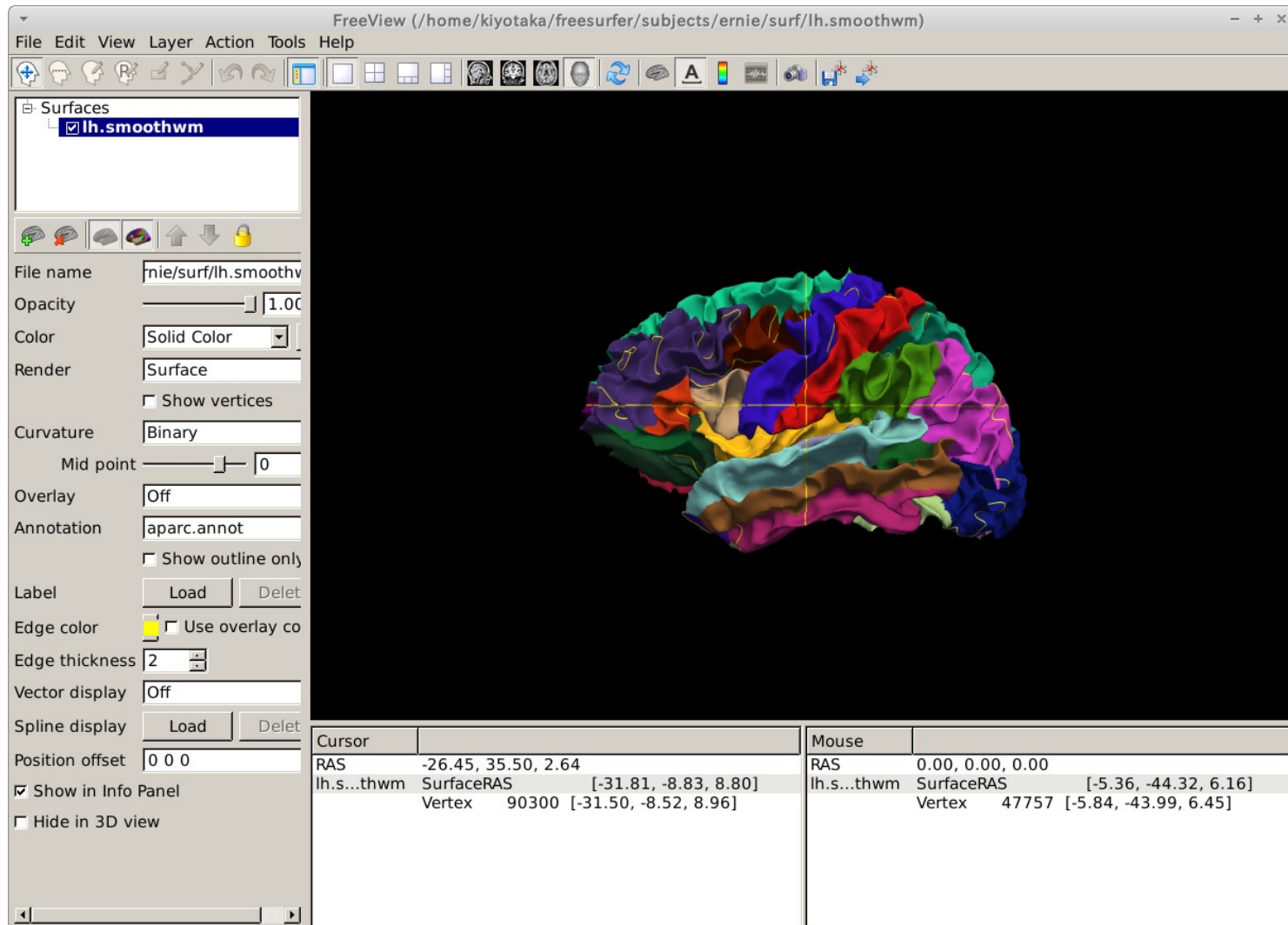
```
freeview -f lh.smoothwm:annot=aparc.annot
```

```
freeview -f lh.inflated:annot=aparc.annot
```

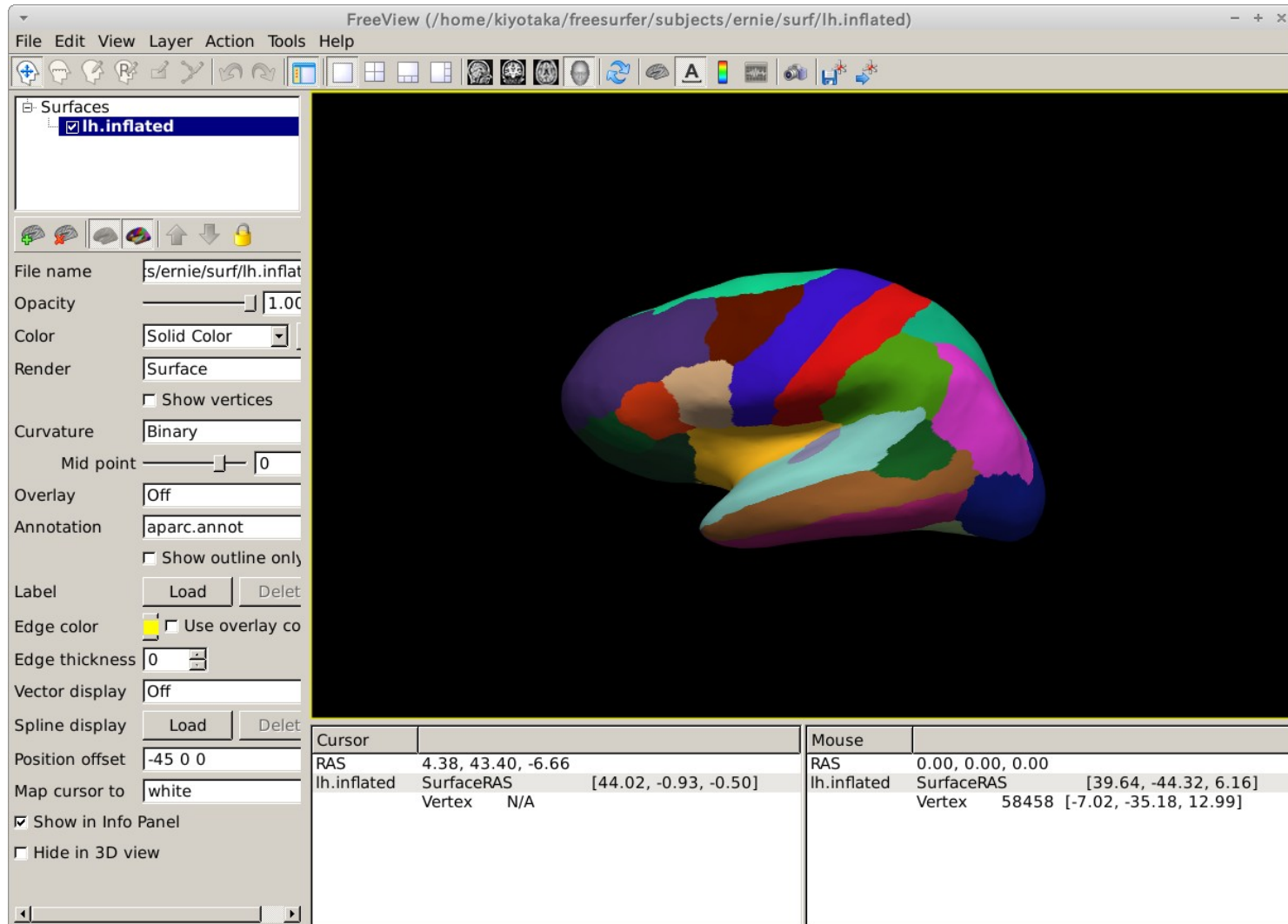
```
freeview -f lh.sphere:annot=aparc.annot
```

- freeview は前回のレイアウトを記憶している
 - 前回と同じであれば layout など指定しなくてもよい

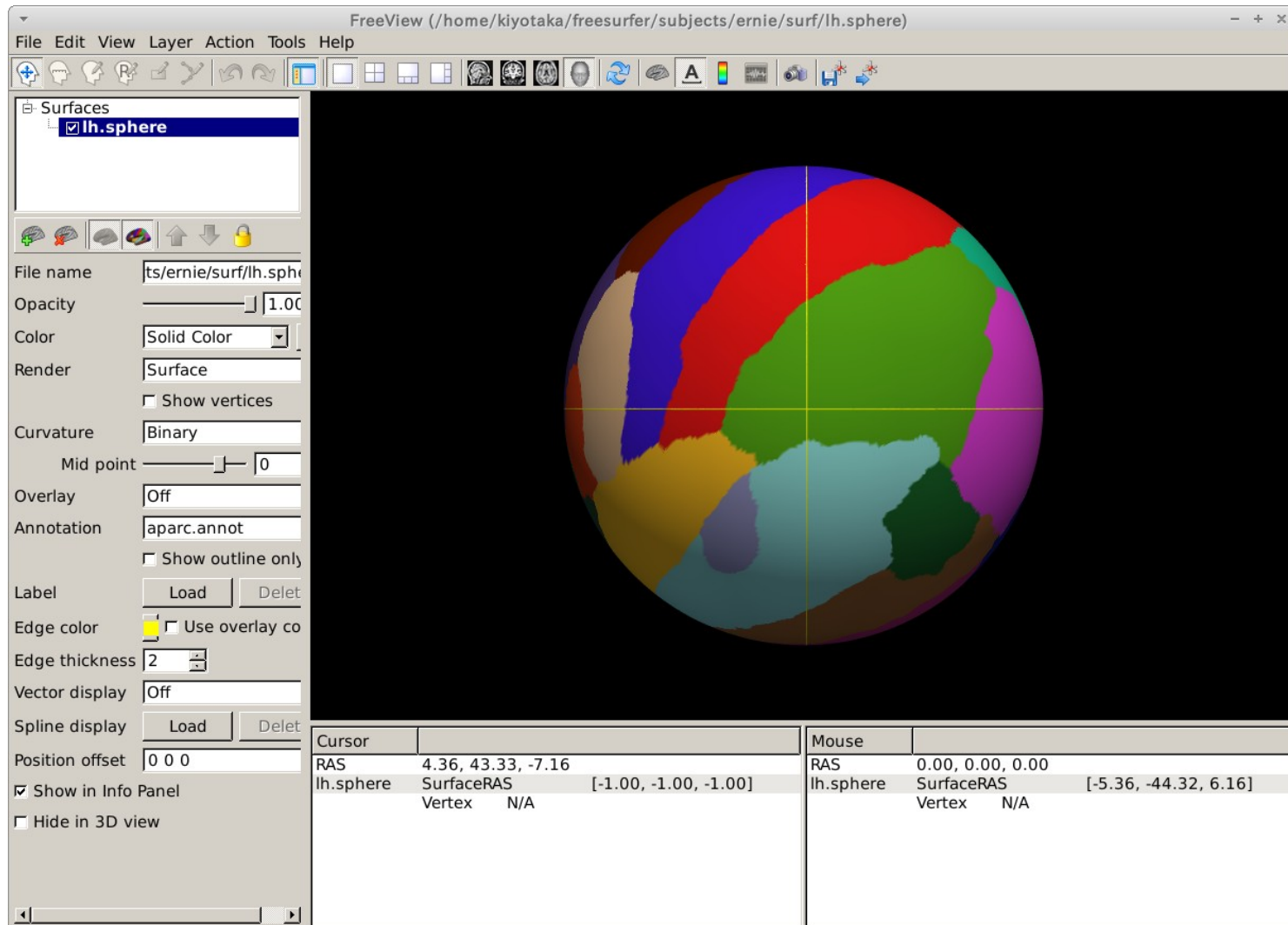
DK atlas on smoothwm



DK atlas on inflated



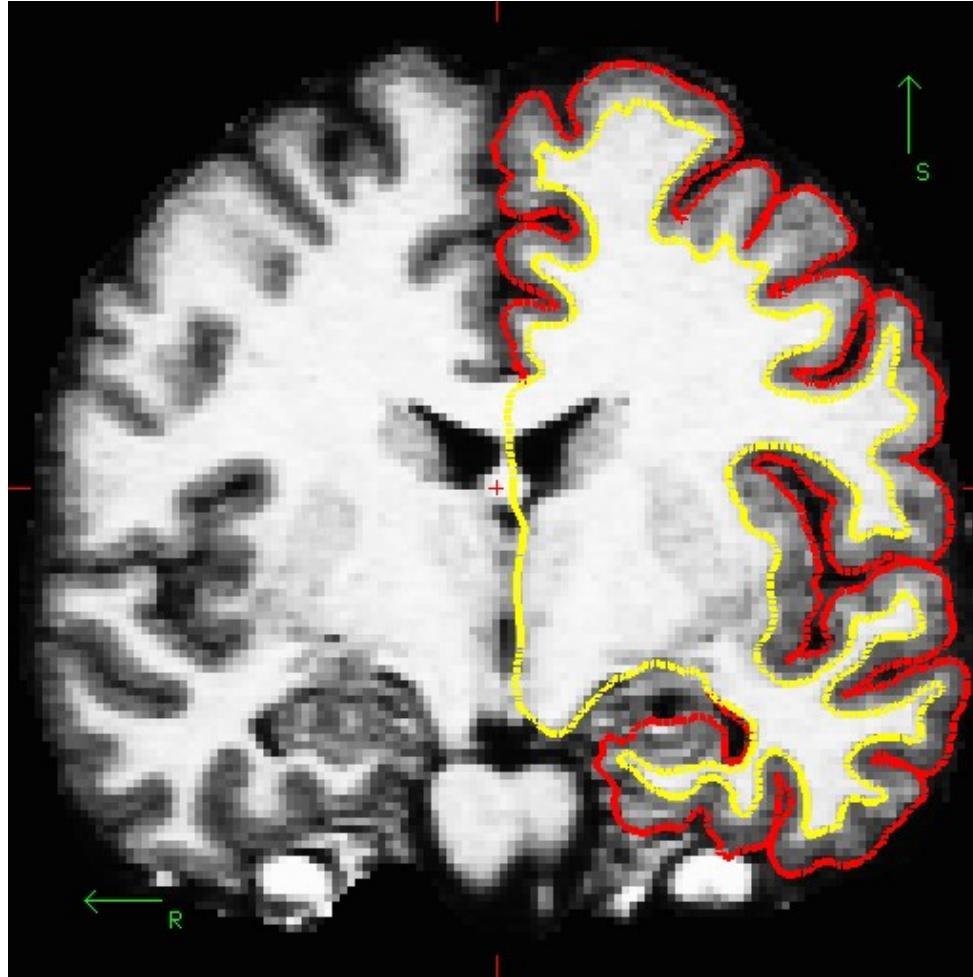
DK atlas on sphere



29. -pial

- white surface 画像を元に、pial surface 画像を生成する
 - pial: 軟膜 灰白質と軟膜の境界＝脳表
 - pial surface から曲率、面積、皮質厚も求める
 - recon-all のハイライト
 - 入力画像: `aseg.presurf.mgz`,
`brain.finalsurfs.mgz`, `wm.mgz`,
`filled.mgz`, `?h.oirg`, `?h.aparc.annot`
 - 出力ファイル: `?h.pial`, `?h.curv.pial`,
`?h.area.pial`, `?h.thickness`

軟膜のSurfaceは白質のSurfaceから生成



pial surface の確認

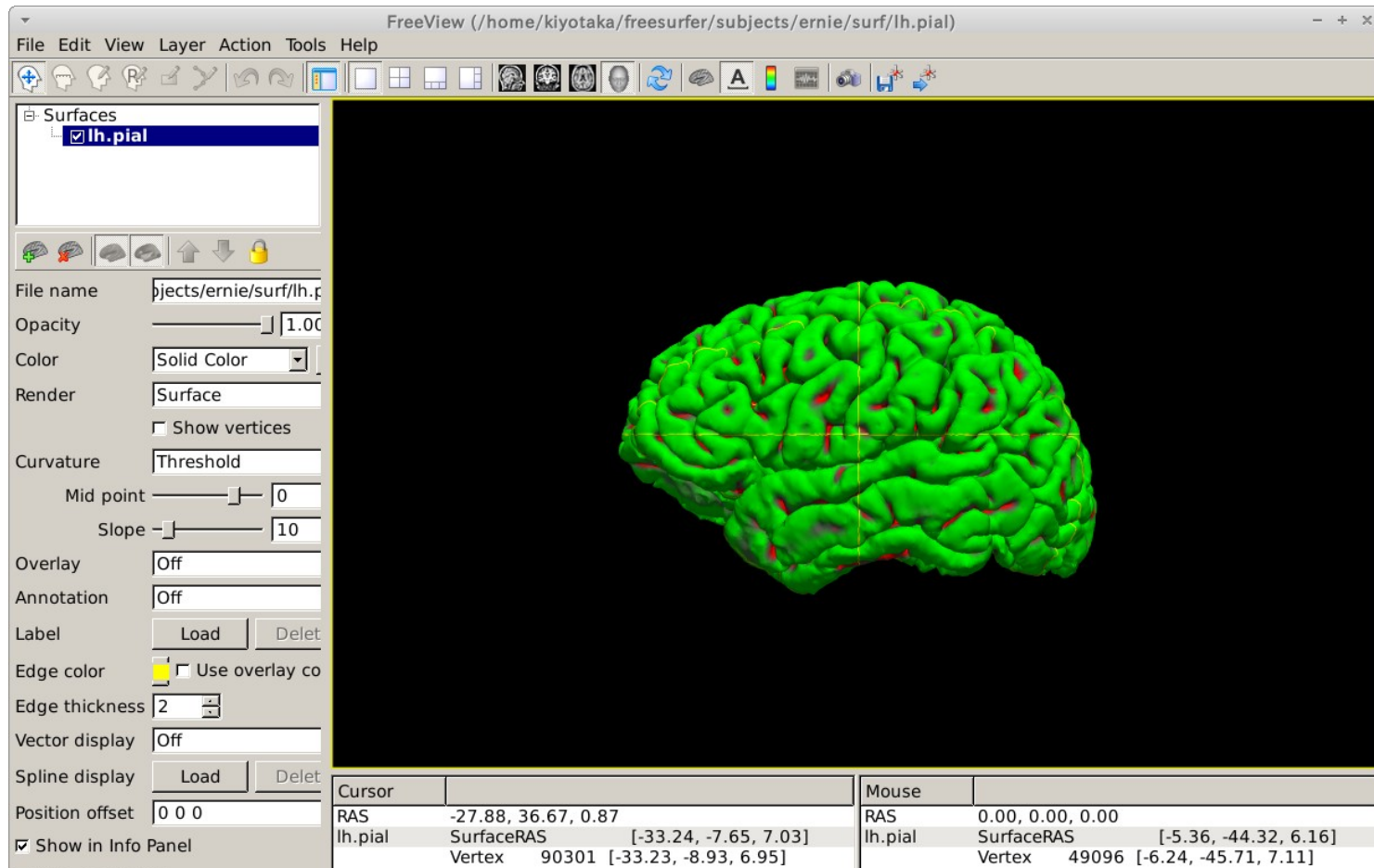
- pial surfaceを見てみる
 - curv, area, thicknessはすべてoverlayで指定する
- thicknessがよく使われるため、thicknessを確認する

```
freeview -f lh.pial
```

```
freeview -f \
```

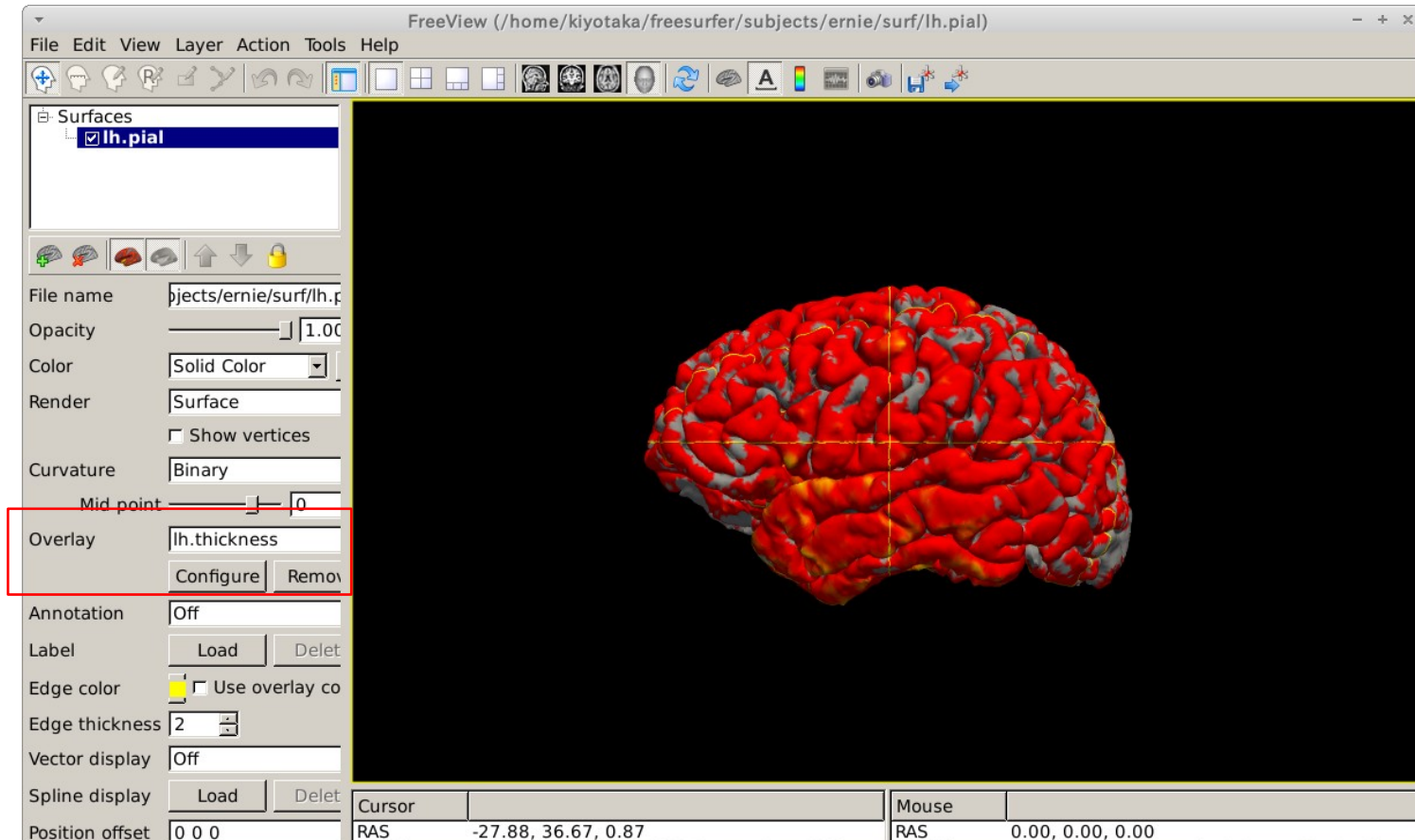
```
lh.pial:overlay=lh.thickness
```

lh.pial



- 緑と赤は曲率を意味
 - 緑が凸、赤が凹

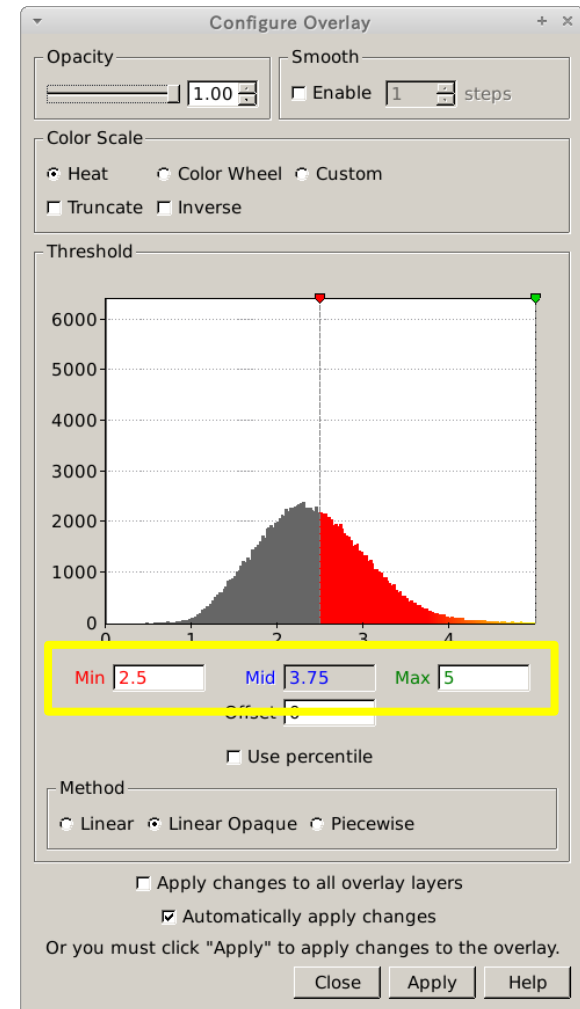
lh.thickness



- thicknessがoverlayされたが、色がついていないところがある
- こういう時は、左側メニューの Overlay → Configure をクリックして設定を確認

configure overlay

- 現時点の設定は、最小値 2.5 最大値 5 の設定
- ヒストグラムを見ると、0.5 – 4.5 あたりに分布している
- 皮質厚であるから、そのぐらいが適当 (4.5mmは厚すぎる気もするが…)



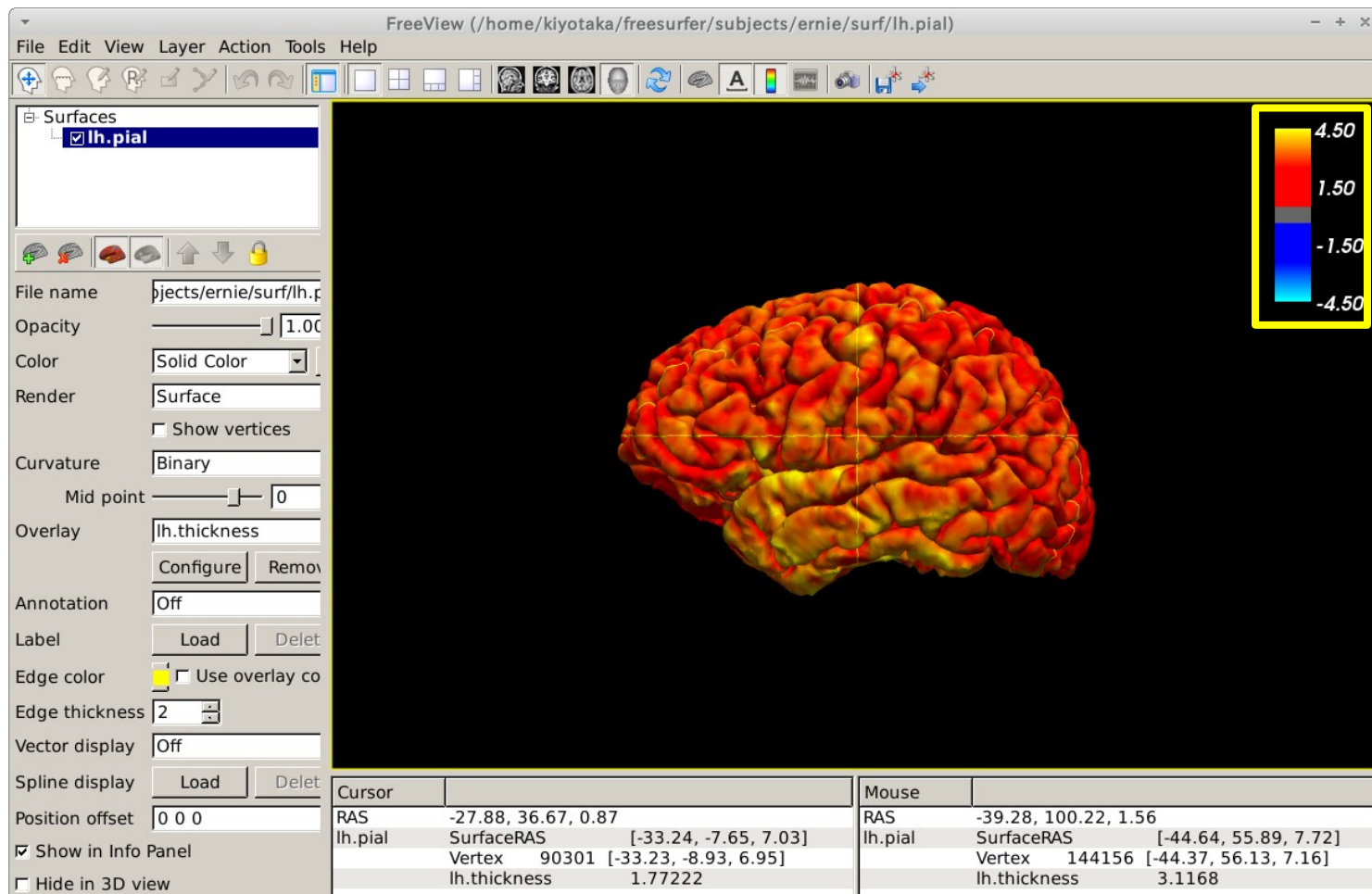
thickness の確認

- thickness を閾値を設定して確認

```
freeview -f \  
lh.pial:overlay=lh.thickness:overlay_  
threshold=0.5,4.5
```

- overlay_threshold=最小値, 最大値
 - スペースを入れないことがポイント
 - スペースを入れるとコマンドラインが途切れる

適切な設定でのlh.thickness



- メニューの "View" → "Show Color Scale" でカラースケールを表示できる

30. -cortribbon

- 皮質のマスク画像を生成
 - cortical ribbon: 皮質(灰白質)だけを取り出すとリボンのように見えるため
 - `?h.ribbon.mgz` を生成後、以下の値を割り当てた `ribbon.mgz` を生成する
 - 左白質: 2; 左皮質: 3; 右白質: 41; 右皮質: 42
 - 入力画像: `aseg.presurf.mgz`, `?h.white`, `?h.pial`
 - 出力画像: `?h.ribbon.mgz`, `ribbon.mgz`

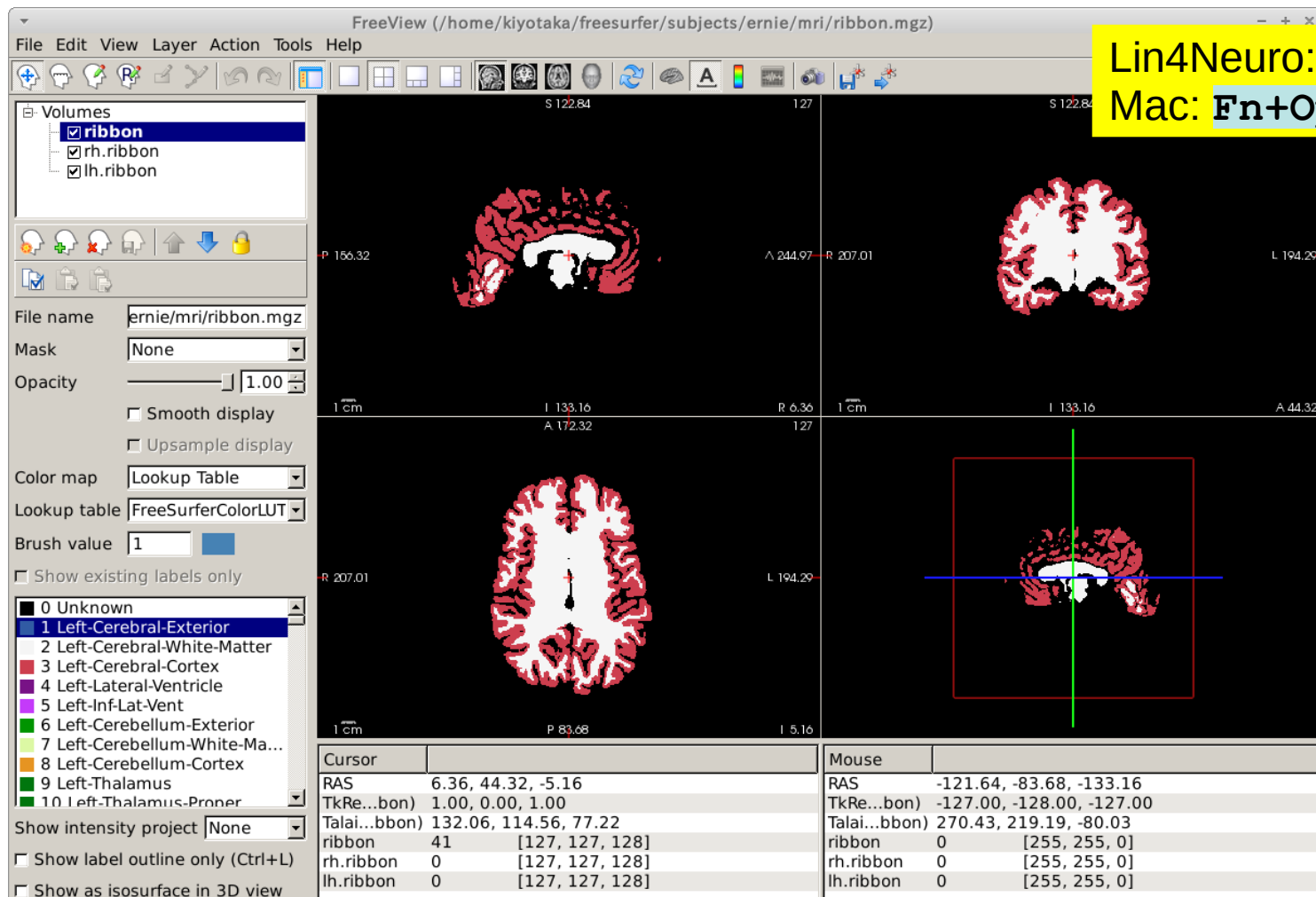
cortical ribbon の確認

- ribbonを確認

```
cd ../mri
```

```
freeview -v lh.ribbon.mgz \  
rh.ribbon.mgz \  
ribbon.mgz:colormap=lut \  
-layout 2 -viewport sagittal
```

適切な設定でのlh.thickness



Lin4Neuro: **Alt+C**
Mac: **Fn+Option+C**

- 様々な領域をクリックしながら信号値を確認
- レイヤーを入れ替えて左半球と右半球の皮質も確認

31. -parcstats

- 皮質の統計値をDKアトラスに基づいて計算
 - 1. 領域名 2. 頂点数 3. 全脳表面積 (mm²) 4. 全灰白質容積 (mm³) 5. 平均皮質厚 (mm) 6. 皮質厚の標準誤差 (mm) 7. 平均曲率 8. ガウス曲率 9. folding index 10. intrinsic curvature index
 - 入力画像: ?h.aparc.annot, wm.mgz, ribbon.mgz, ?h.white, ?h.pial, ?h.thickness
 - 出力ファイル: stats/?h.aparc.stats, label/aparc.annot.ctab
- aparc.stats はROI解析のセクションで扱う

32. -cortparc2

- 脳領域のラベルを、個々の領域に割り当てる
 - ここでは、Destrieux アトラスを割り当てる
 - 三角形の数は変わっていない → pial, smoothwm, inflated, sphere いずれにも適用できる
 - 入力画像: `aseg.presurf.mgz`,
 `?h.cortex.label`, `?h.sphere.reg`
 - 出力ファイル: `label/?h.aparc.a2009s.annot`

Destrieux アトラスの確認

- Destrieux アトラスのラベルを確認
- annotationファイルは、指定する際に lh, rh はつけなくてよい
 - 指定されたファイルから左右を自動で判定している
- lh.pialで見てみる

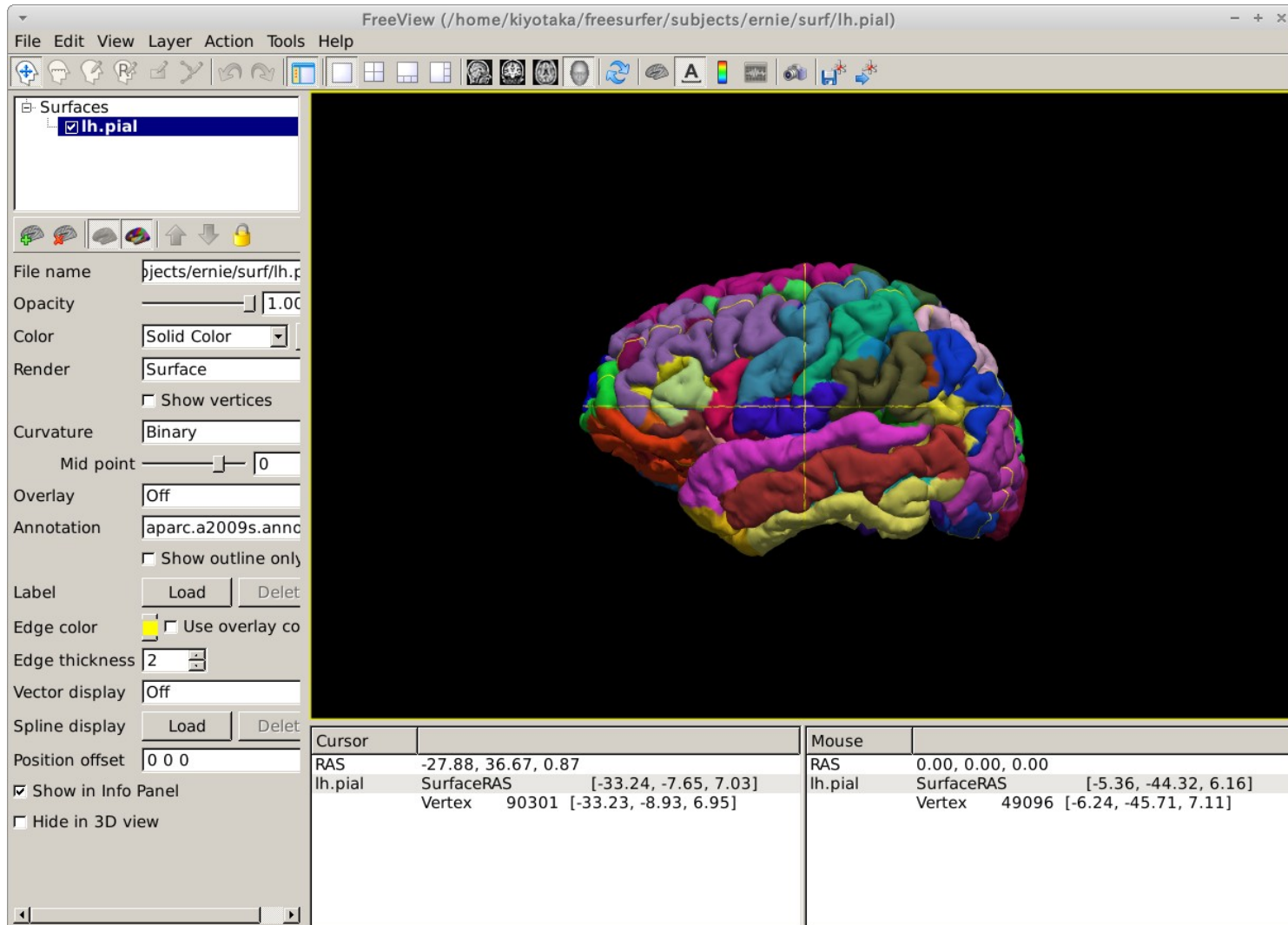
```
cd ../surf
```

```
freeview -f \
```

```
lh.pial:annot=aparc.a2009s.annot \
```

```
-layout 1 -viewport 3d
```

Destrieux atlas on pial



33. -parcstats2

- 皮質の統計値をDestrieuxアトラスに基づいて計算
 - 1. 領域名 2. 頂点数 3. 全脳表面積 (mm²) 4. 全灰白質容積 (mm³) 5. 平均皮質厚 (mm) 6. 皮質厚の標準誤差 (mm) 7. 平均曲率 8. ガウス曲率 9. folding index 10. intrinsic curvature index
 - 入力画像: ?haparc.a2009s.annot, wm.mgz, ribbon.mgz, ?h.white, ?h.pial, ?h.thickness
 - 出力ファイル: stats/?h.aparc.a2009s.stats, label/aparc.annot.a2009s.ctab
- aparc.a2009s.stats はROI解析のセクションで扱う

34. -cortparc3

- 脳領域のラベルを、個々の領域に割り当てる
 - ここでは、Desikan-Killiany-Tourville (DKT) アトラスを割り当てる
 - 三角形の数は変わっていない → pial, smoothwm, inflated, sphere いずれにも適用できる
 - 入力画像: `aseg.presurf.mgz`,
`?h.cortex.label`, `?h.sphere.reg`
 - 出力ファイル: `label/?h.aparc.DKTatlas.annot`

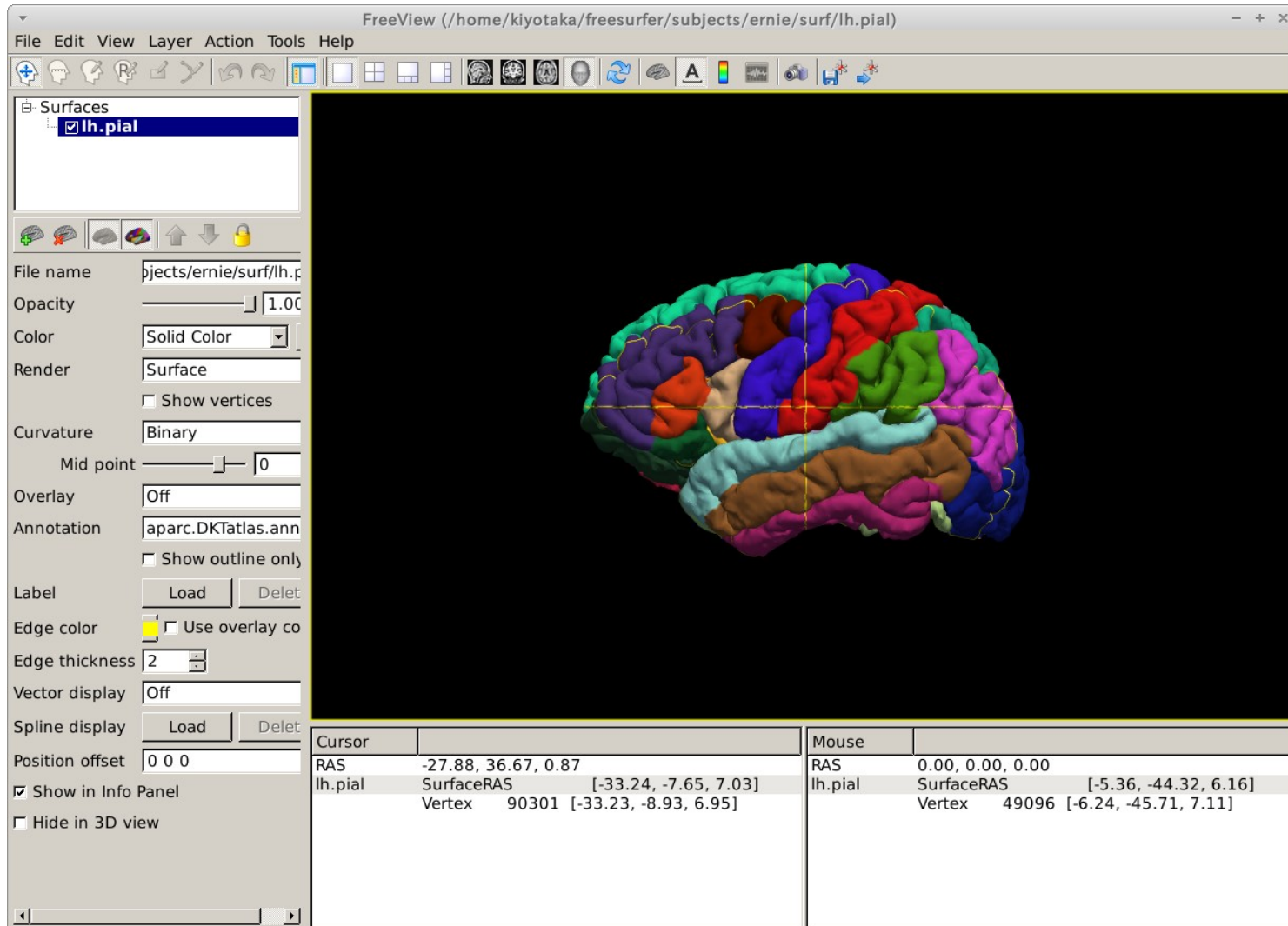
DKT アトラスの確認

- DKT アトラスのラベルを確認
- annotationファイルは、指定する際に lh, rh はつけなくてよい
 - 指定されたファイルから左右を自動で判定している
- lh.pialで見る

freeview -f \

lh.pial:annot=aparc.DKTatlas.annot

DKT atlas on pial



35. -parcstats3

- 皮質の統計値をDKTアトラスに基づいて計算
 - 1. 領域名 2. 頂点数 3. 全脳表面積 (mm²) 4. 全灰白質容積 (mm³) 5. 平均皮質厚 (mm) 6. 皮質厚の標準誤差 (mm) 7. 平均曲率 8. ガウス曲率 9. folding index 10. intrinsic curvature index
 - 入力画像: ?haparc.DKTatlas.annot, wm.mgz, ribbon.mgz, ?h.white, ?h.pial, ?h.thickness
 - 出力ファイル: stats/?h.aparc.DKTatlas.stats, label/aparc.annot.DKTatlas.ctab
- aparc.DKTatlas.stats はROI解析のセクションで扱う

36. -pctsurfcon

- 灰白質/白質の信号値比率を計算
 - $((\text{wm}-\text{gm})/((\text{wm}+\text{gm})/2))*100$
- 入力画像: rawavg.mgz, orig.mgz,
?h.cortex.label, ?h.white
- 出力ファイル: ?h.w-g.pct.mgh,
stats/?h.w-g.pct.stats

37. -hyporelabel

- 画像の信号値低値の領域を修正
 - 皮質下領域の番号が間違っている場合に修正
 - 入力画像: `aseg.presurf.mgz`, `?h.white`
 - 出力ファイル: `aseg.presurf.hypos.mgz`

38. -aparc2aseg

- 皮質領域と皮質下領域を合体させたvolume画像を生成
 - アトラスごとに3種類生成される
 - 入力画像: `aseg.presurf.mgz`, `?h.ribbon.mgz`,
`ribbon.mgz`,
`label/?h.aparc{,.a2009s,.DKTatlas}.annot`
 - 出力画像:
`aparc+aseg.mgz`
`aparc.a2009s+aseg.mgz`
`aparc.DKTatlas+aseg.mgz`

皮質+aseg.mgz の確認

- 3種類のアトラス+aseg.mgz を確認

```
cd ../mri
```

```
freeview -v \
```

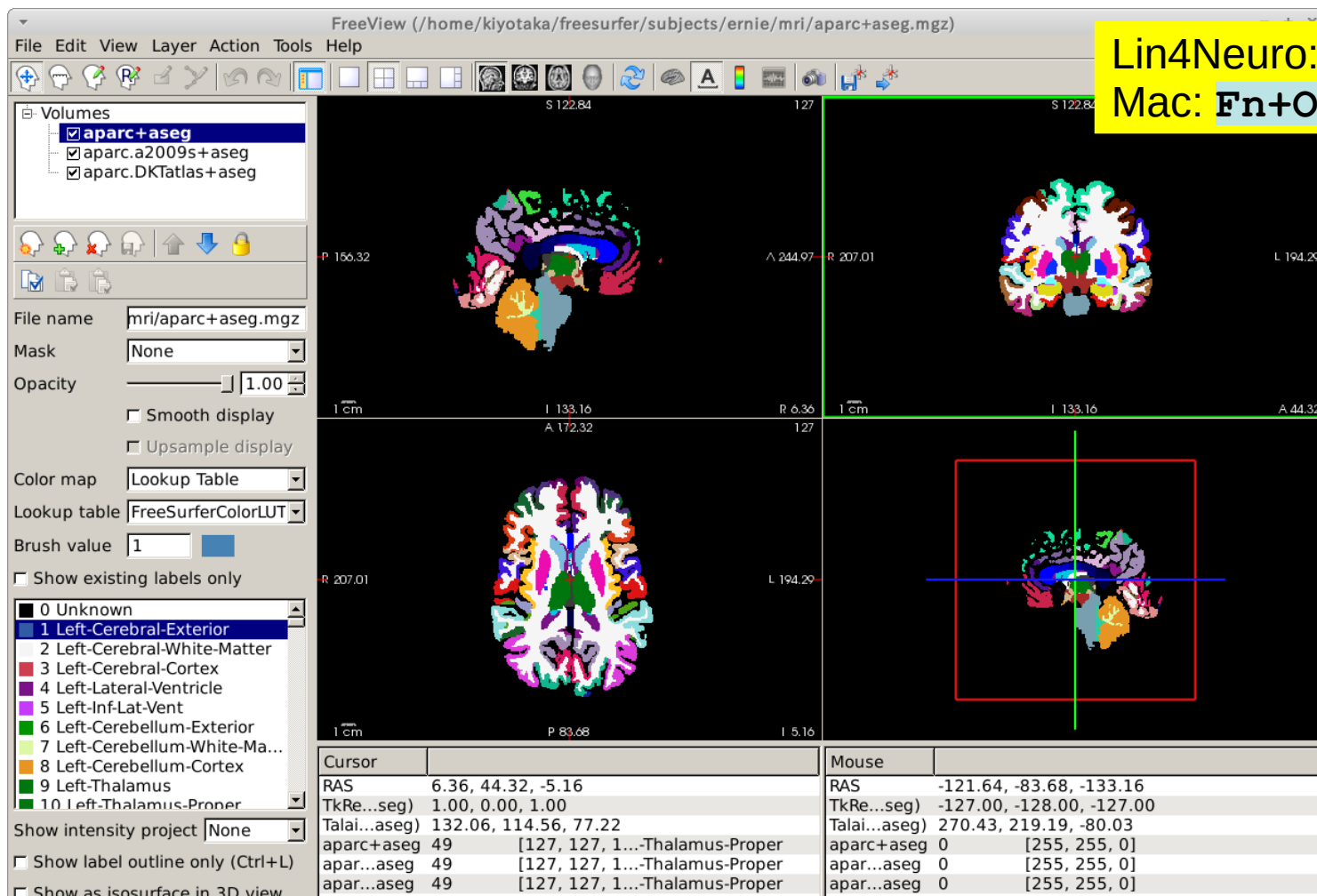
```
  aparca.DKTatlas+aseg.mgz:colormap=lut \
```

```
  aparca.a2009s+aseg.mgz:colormap=lut \
```

```
  aparca+aseg.mgz:colormap=lut \
```

```
-layout 2 -viewport sagittal
```

aparc+aseg.mgz



Lin4Neuro: **Alt+C**
Mac: **Fn+Option+C**

- ・ レイヤーを入れ替えてアトラスの違いを確認
- ・ 様々な皮質領域をクリックしながらアトラスにより領域が異なることを確認

39. -apas2aseg

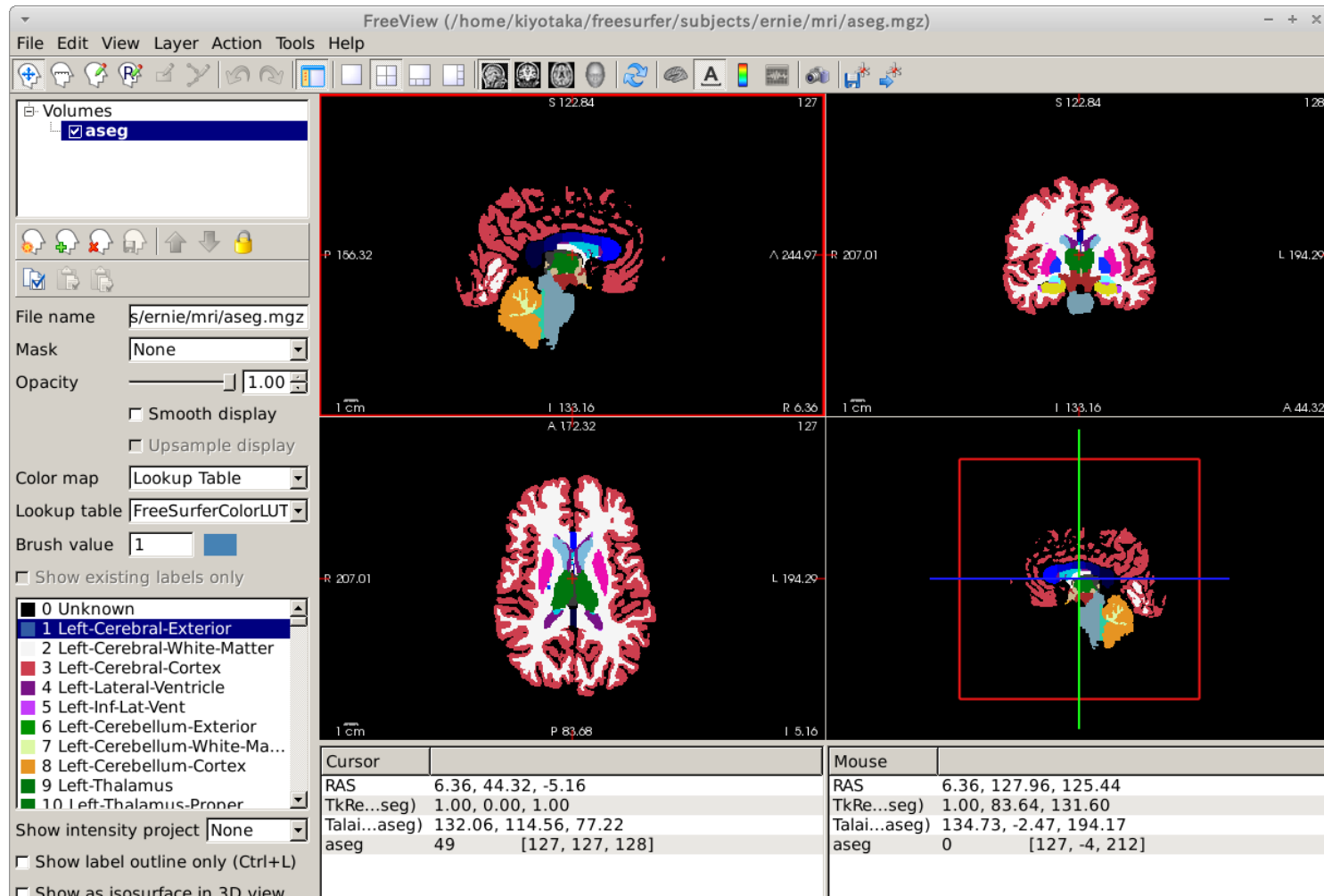
- 皮質領域のアトラス情報を削除
 - 皮質下領域のみの `aseg.mgz` を生成
 - 入力画像: `aparc+aseg.mgz`
 - 出力画像: **`aseg.mgz`**

aseg.mgz の確認

- aseg.mgz を確認

```
freeview -v aseg.mgz:colormap=lut
```

aseg.mgz



- 皮質領域の値は ribbon.mgz と同じく左3,右42になっている

40. -segstats

- aseg.mgz を用いて、皮質下領域の統計値を算出
 - 皮質下領域のみの aseg.mgz を生成
 - 入力画像: brainmask.mgz, norm.mgz, aseg.mgz, aseg.presurf.mgz, ribbon.mgz, ?h.orig.nofix, ?h.white, ?h.pial
 - 出力ファイル: stats/aseg.stats

41. -wmparc

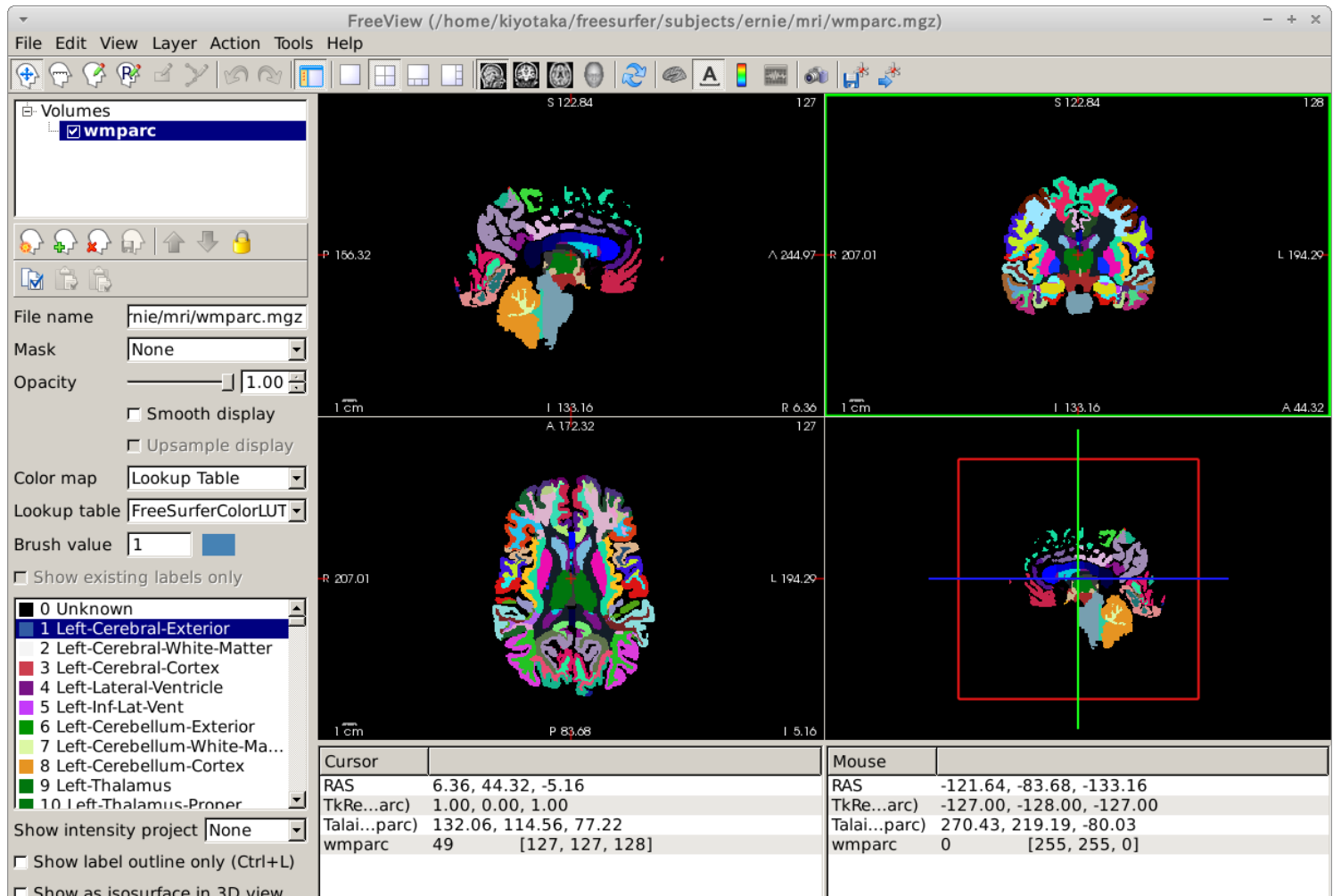
- `aparc+aseg.mgz` を用いて、白質の区域情報を追加
 - `wmparc.mgz` が生成され、さらに統計値を算出
 - 入力画像: `aparc+aseg.mgz`
 - 出力画像: **`wmparc.mgz`**
 - 入力ファイル: `talairach.xfm`, `brainmask.mgz`,
`norm.mgz`, `ribbon.mgz`, `wmparc.mgz`,
`aseg.presurf.mgz`, `?h.white`, `?h.pial`
 - 出力ファイル: **`stats/wmparc.stats`**

wmparc.mgz の確認

- wmparc.mgz を確認

```
freeview -v wmparc.mgz:colormap=lut
```

wmparc.mgz



- 白質が細分化されていることがわかる

42. -balabels

- Brodmann area のラベルを作成
 - すべてではない
 - 詳細は、下記リンクを参照
 - <https://surfer.nmr.mgh.harvard.edu/fswiki/BrodmannAreaMaps?highlight=%28Brodmann%29>
 - 入力画像: `?h.sphere.reg`
 - 出力画像: `label/?h.BA*_exvivo.label`
`label/?h.perirhinal_exvivo.label`
`label/?h.entorhinal_exvivo.label`

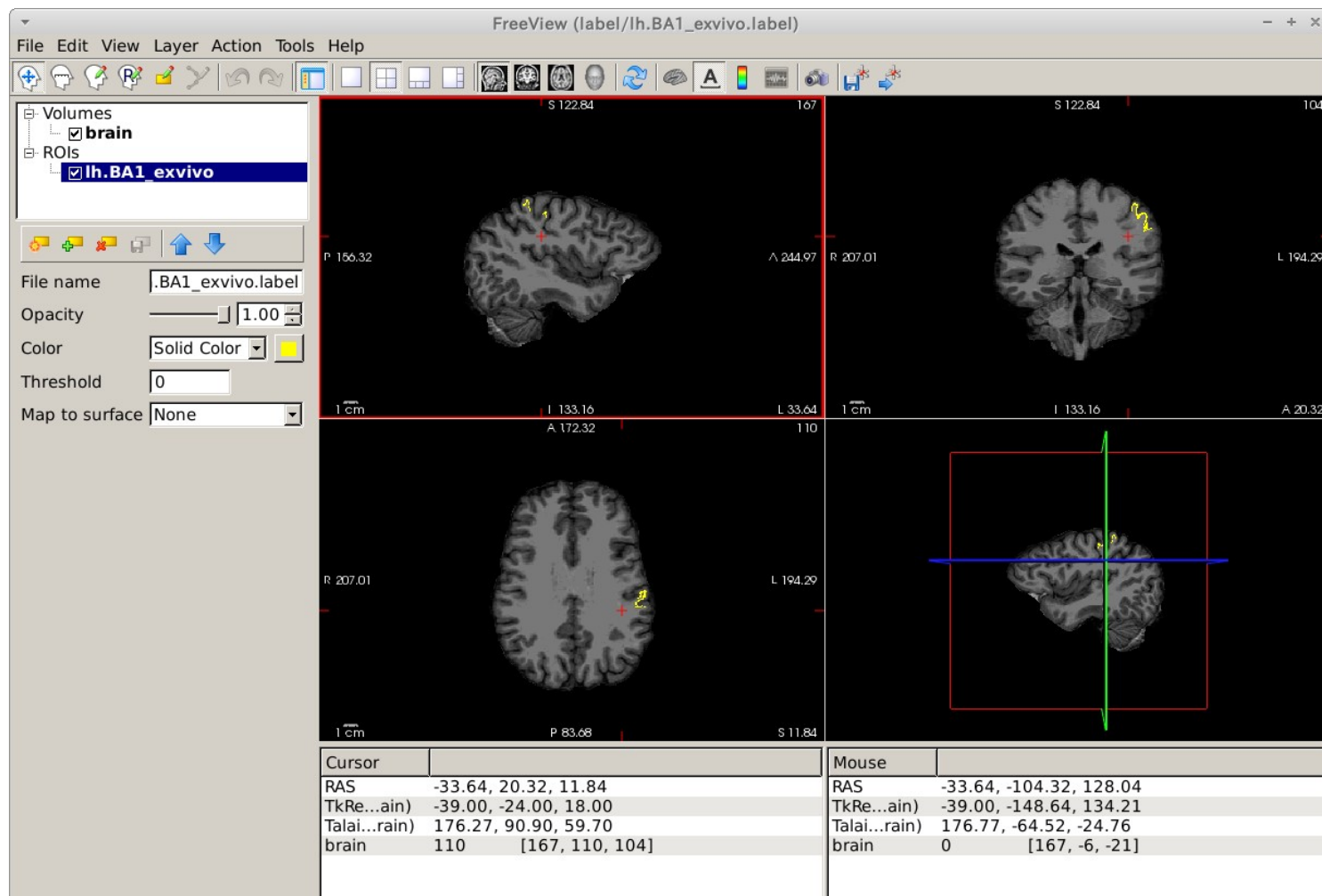
ラベルの表示法(1)

- volume データとともに -l オプションで表示

```
cd .. # $SUBJECTS_DIR/ernie
```

```
freeview -v mri/brain.mgz -l \
  label/lh.BA1_exvivo.label
```

Volumeへのラベル表示



- 脳表に沿ってラベルが存在することがわかる

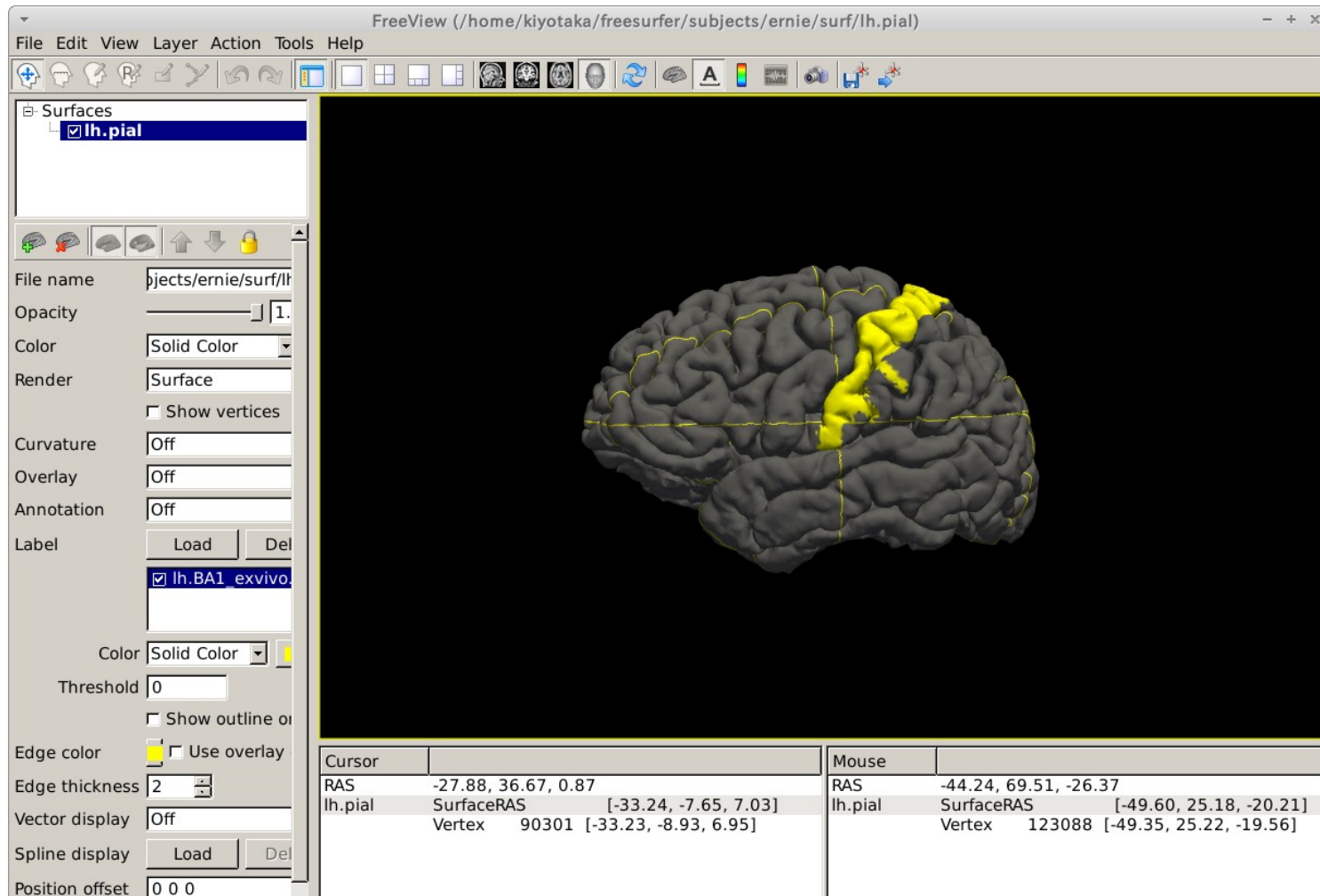
ラベルの表示法(2)

- surface データに :label で表示する

```
freeview -f \  
  surf/lh.pial:curvature_method=off  
  :label=label/lh.BA1_exvivo.label \  
  -layout 1 -viewport 3d
```

- :curvature_method=off で curvature を表示させない

Surfaceへのラベル表示



- 脳表に沿ってラベルが存在することがわかる

recon-all -autorecon3 のまとめ

- pial 画像の生成
- 最終的な皮質下画像の生成
- 統計値の算出
- QCが必要なものは、autorecon2 まで
- FreeSurfer 6.0では、ウェブサイトはまだ反映されていない変化がいくつもある

Questions?