

第1部(2)

recon-all -autorecon2

筑波大学医学医療系
精神医学

根本 清貴

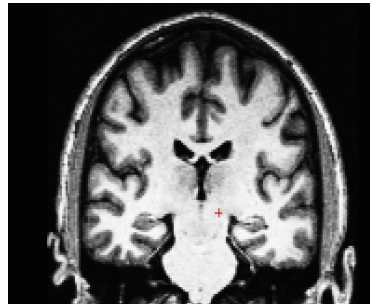
本セクションの目標

- `recon-all -autorecon2` 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

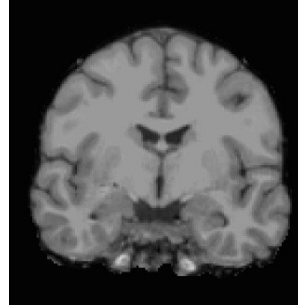
本セクションのチートシート

- `nisg-202001/docs` にある `ex2.html` をダブルクリック
- コマンドの意味も記載されている

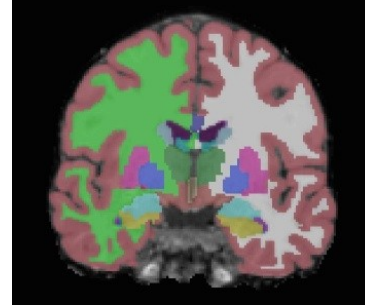
recon-allの概要



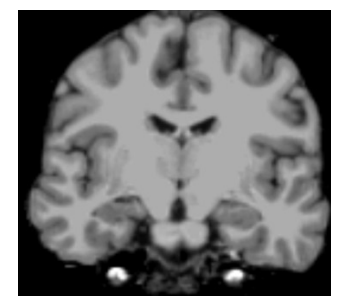
入力画像
T1強調画像



頭蓋骨除去



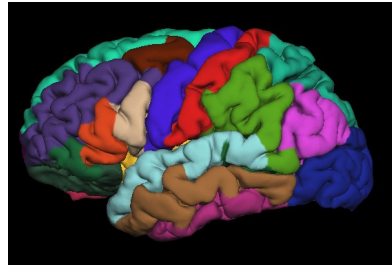
Volumeの
ラベリング



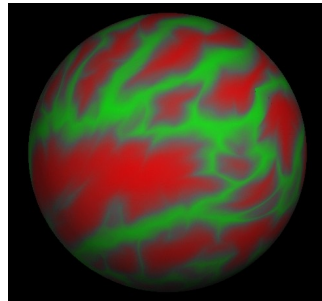
信号値の
正規化

-autorecon1

-autorecon2 (下に続く)



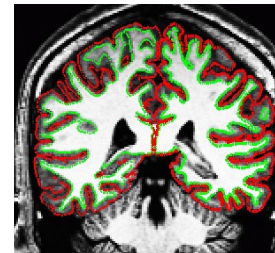
脳回のラベリング



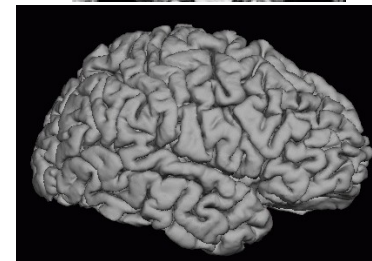
Surfaceアトラスへの
レジストレーション

-autorecon3

統計!



白質の分離



Surfaceの抽出

-autorecon2

FreeSurferスライドを改変

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 -autorecon2

- 06. GCA Registration
- 07. Canonical Intensity Normalization
- 08. Canonical Registration
- 09. SubCort Segmentation
- 10. Intensity Normalization2
- 11. Create BrainFinalSurfs
- 12. WM Segmentation
- 13. Fill
- 14. Tessellate
- 15. Smooth1
- 16. Inflate1
- 17. Qsphere
- 18. Fix Topology
- 19. Make White Surfaces
- 20. Smooth2
- 21. Inflate2
- 22. curvHK
- 23. curvstats

6. -gcareg

- アトラスへの位置合わせ GCA registration
 - GCA: Gaussian Classifier Atlas
 - `nu.mgz` を `$FREESURFER_HOME` (通常は `/usr/local/freesurfer`) の下にある `average` ディレクトリの中にある アトラスにあうように位置合わせを行う
 - 入力画像: `nu.mgz`, `brainmask.mgz`
 - 出力ファイル: `transforms/talairach.lta`

talairach.lta の確認

- 以下で talairach.lta を確認

```
cd mri
```

```
cat transforms/talairach.lta
```

```
src volume info
```

```
valid = 1 # volume info valid
```

```
filename = nu.mgz
```

```
dst volume info
```

```
valid = 1 # volume info valid
```

```
filename = /usr/local/freesurfer/average/RB_all_2016-05-10.vc700.gca
```

- nu.mgz を RB_all_2016-05-10.vc700.gca に合わせこんでいることがわかる

7. -canorm

- 標準化 CA normalize
 - CA: Canonical (標準的)
 - `nu.mgz` の アトラスへのさらなる位置合わせを行う
 - 入力画像: `nu.mgz`, `brainmask.mgz`
 - 出力画像: **`norm.mgz`**

8. -careg

- 非線形変換 CA register
 - CA: Canonical (標準的)
 - `norm.mgz` を非線形変換を用いてアトラスにさらに位置合わせを行う
 - 入力画像およびファイル: `norm.mgz`, `brainmask.mgz`, `talairach.lta`
 - 出力ファイル: `transforms/talairach.m3z`

9. -calabel

- 皮質下領域のラベリング CA label
 - 皮質下領域をラベリングする
 - 入力画像およびファイル: `norm.mgz`, `talairach.m3z`
 - 出力ファイル: `aseg.auto.mgz`, `aseg.presurf.mgz`
 - `aseg.auto.mgz` と `aseg.presurf.mgz` は同一ファイル

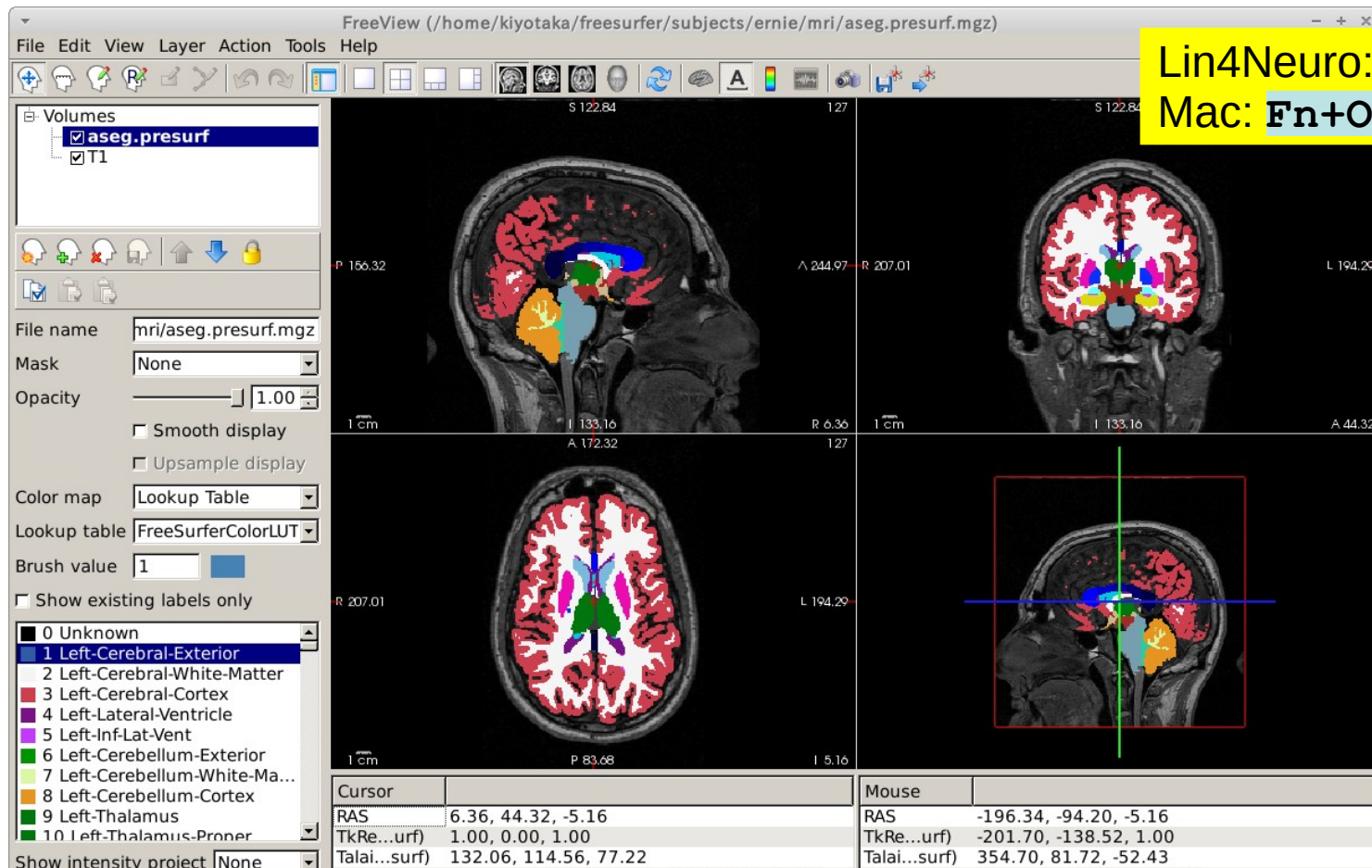
皮質下領域の確認

- 皮質下領域がどうラベリングされたかを確認

```
freeview -v T1.mgz \  
aseg.presurf.mgz:colormap=lut
```

- 画像の後ろに :colormap=lut でカラーマップにLUTを指定できる

皮質下領域の表示



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

- 上記のキーボード・ショートカットを使って表示する画像を切り替え
- 皮質下領域を確認

10. -normalization2

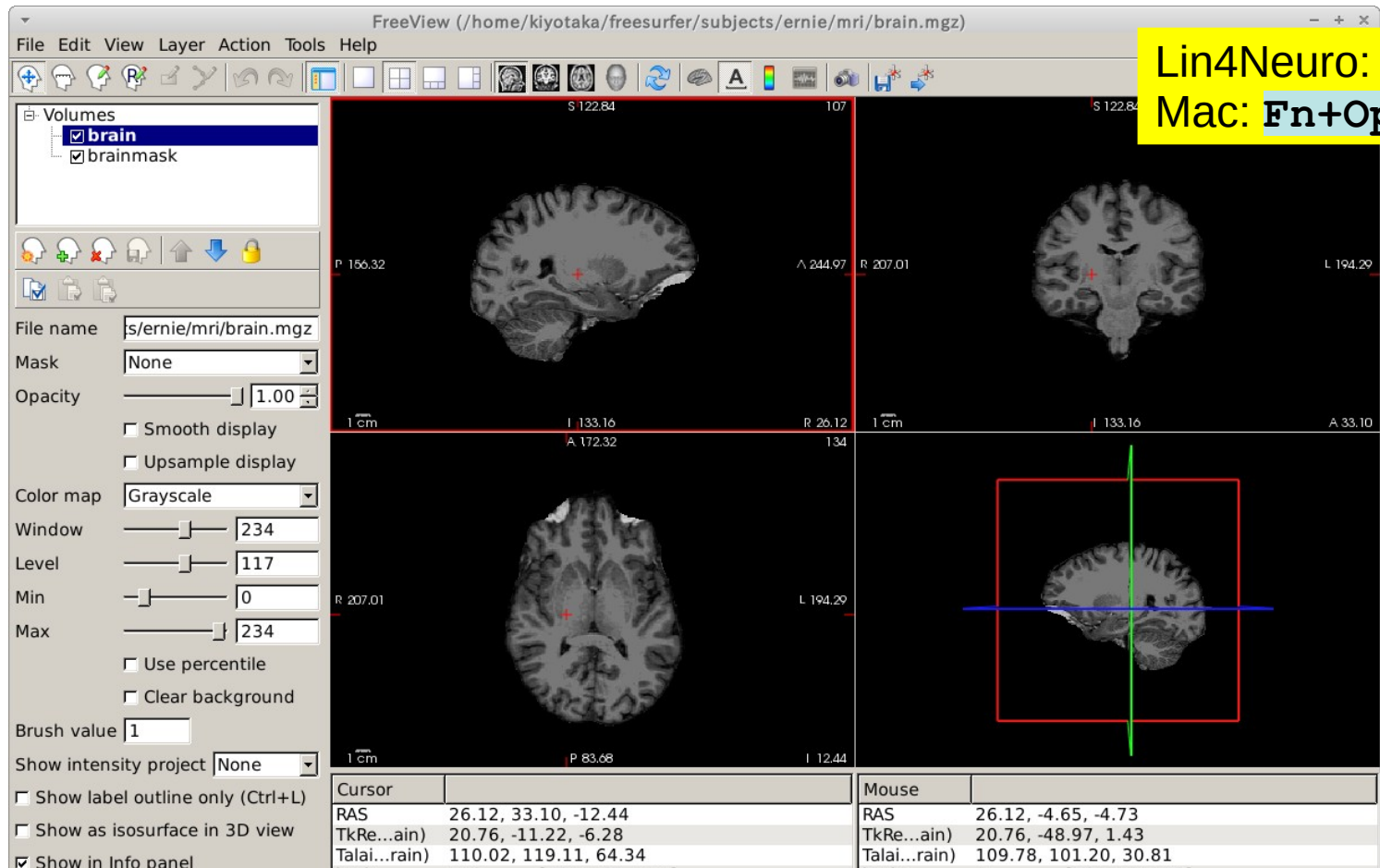
- 頭蓋骨を取り除いた後の信号値補正
 - 頭蓋骨を取り除いた後のほうが信号値補正の精度は改善するため
 - 入力画像: `brainmask.mgz`, `norm.mgz`,
`aseg.presurf.mgz`
 - 出力画像: **`brain.mgz`**

信号値補正の確認

- brainmask.mgz と brain.mgz を比較

```
freeview -v brainmask.mgz brain.mgz
```


信号値補正の効果



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

- 上記のキーボード・ショートカットを使って表示する画像を切り替え
- 白質領域がより均てん化されていることを確認

11. -maskbfs

- 脳表画像を作成するためのマスク画像を作成
 - recon-allの場合、マスク画像は2値画像ではない
 - 入力画像: `brain.mgz`, `brainmask.mgz`
 - 出力画像: `brain.finalsurfs.mgz`

12. -segmentation

- 白質領域の分離
 - 白質を脳の他の領域から区別する
 - 側脳室なども入ってくるため、正確には、白質に限っていない
 - 信号値の他、皮質下領域画像も用いる
 - 入力画像: `brain.mgz`, (`aseg.presurf.mgz`)
 - 出力画像: `wm.mgz`

wm.mgz の確認

- 白質画像を確認

```
freeview -v brain.mgz \
```

```
wm.mgz:opacity=0.8:colormap=heat
```

wm.mgz の表示

FreeView (/home/kiyotaka/git/nisg-202001-dev/subjects/ernie/mri/wm.mgz)

File Edit View Layer Action Tools Help

Volumes: ☒ wm, ☒ brain

File name: cts/ernie/mri/wm.mgz
Mask: None
Opacity: 0.80
Color map: Heat
Min: 0, Mid: 150, Max: 270, Offset: 0
Truncate: ☐, Invert: ☐, Autoset Mid: ☒, Clear higher values: ☐
Brush value: 1
Show intensity project: None
Show label outline only (Ctrl+L): ☐
Show as isosurface in 3D view: ☐
Show in Info panel: ☒

Cursor: RAS (6.36, 44.32, -5.16), TkRe...(wm) (1.00, 0.00, 1.00), Talai... (wm) (132.06, 114.56, 77.22), wm (87, [127, 127, 128]), brain (87, [127, 127, 128])

Mouse: RAS (6.36, -171.92, -20.35), TkRe...(wm) (1.00, -216.25, -14.19), Talai... (wm) (129.81, 74.94, -131.98), wm (0, [127, 142, -88]), brain (0, [127, 142, -88])

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

- 上記のキーボード・ショートカットを使って表示する画像を切り替え
- 信号値が110のみならず、複数から構成されている

13. -fill

- white surface画像を作成するための皮質下領域をまとめた塊を作成
 - 脳幹を除去
 - 空白は全て埋める
 - 左右の半球を分離し、2値画像にする（左半球：255、右半球：127）
 - ここで穴があると、次のtessellateに支障が出る
 - 入力画像：wm.mgz
 - 出力画像：filled.mgz

filled.mgz の確認

- wm.mgz と filled.mgz を比較することにより、皮質下領域をまとめた塊を確認

```
freeview -v wm.mgz filled.mgz
```

filled.mgz と wm.mgz の比較

FreeView (/home/kiyotaka/freesurfer/subjects/ernie/mri/filled.mgz)

File Edit View Layer Action Tools Help

Volumes: ☒ filled, ☒ wm

File name: /ts/ernie/mri/filled.mgz
Mask: None
Opacity: 1.00
☐ Smooth display
☐ Upsample display
Color map: Grayscale
Window: 306
Level: 153
Min: 0
Max: 306
☐ Use percentile
☐ Clear background
Brush value: 1
Show intensity project: None
☐ Show label outline only (Ctrl+L)
☐ Show as isosurface in 3D view
☒ Show in Info panel

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

- 上記のキーボード・ショートカットを使って表示する画像を切り替え
- 脳幹が切り離され、空白が塗りつぶされ、左右の信号値が違ってくる

| Cursor | | Mouse | |
|-----------------|-----------------------|-----------------|------------------------|
| RAS | 6.36, 44.32, -5.16 | RAS | 6.36, -105.19, 132.38 |
| TkRe...led) | 1.00, 0.00, 1.00 | TkRe...led) | 1.00, -149.51, 138.55 |
| Talair...illed) | 132.06, 114.56, 77.22 | Talair...illed) | 132.64, -69.92, -24.63 |
| filled | 127 [127, 127, 128] | filled | 0 [127, -11, -22] |
| wm | 87 [127, 127, 128] | wm | 0 [127, -11, -22] |

14. -tessalate

- white surface画像を作成する第一歩
 - filled.mgz を用いて各半球の表面に三角形を敷き詰める
 - このときの三角形はボクセルの四隅を頂点とした三角形なので、ギザギザ
 - 入力画像: filled.mgz, norm.mgz
 - 出力画像: **?h.orig.nofix**
 - ?hはlhかrhのどちらかの意味

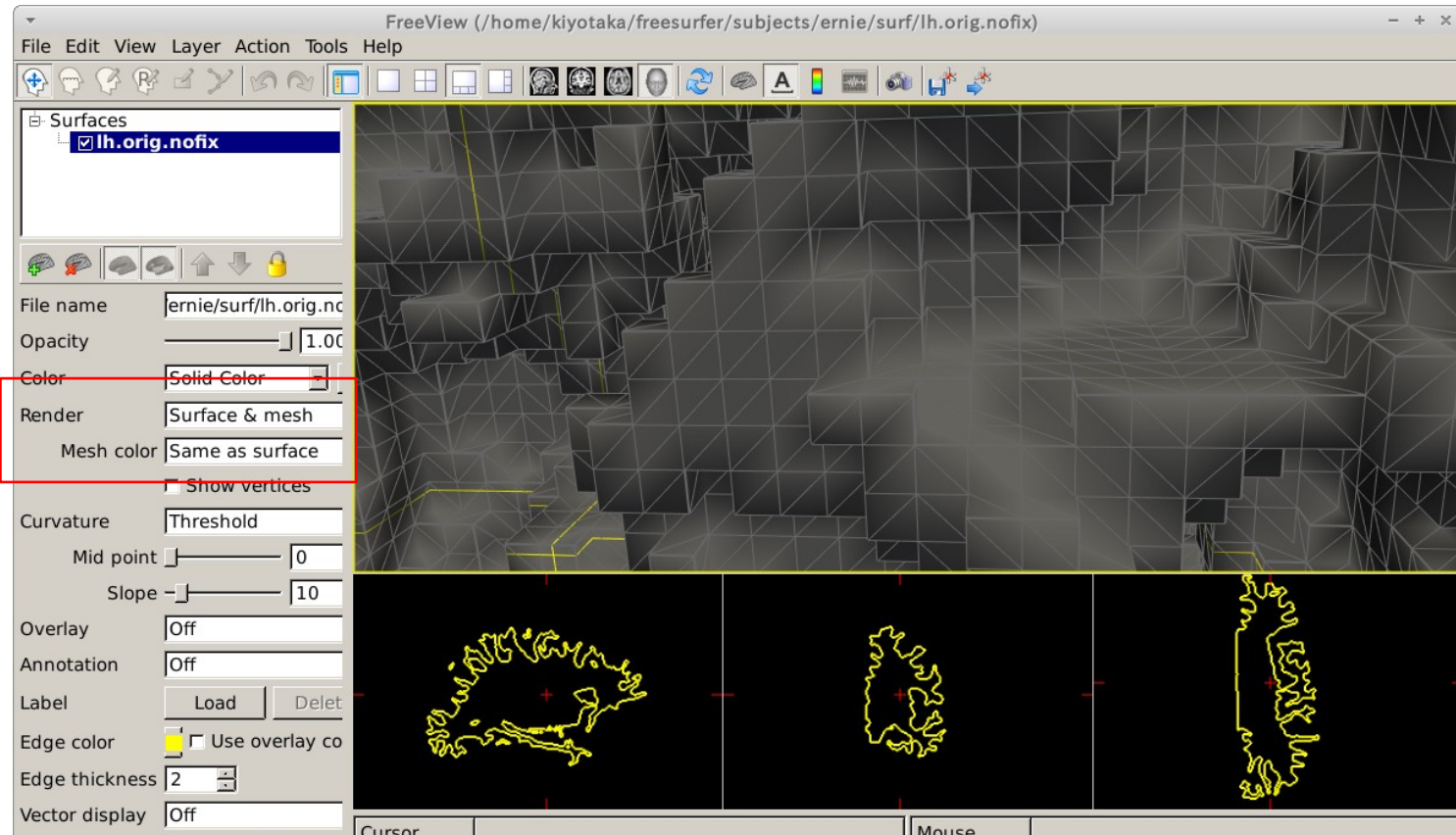
white surface画像の確認

- filled.mgzから作成された lh.orig.nofix を確認
 - 脳表画像は、freeview で -f をつけて指定する

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

```
freeview -f surf/lh.orig.nofix \  
-layout 3 -viewport 3d
```

lh.orig.nofix: surface & mesh



- 左の画面で、Renderを **Surface & mesh** に変更
- 大きく拡大すると、ボクセルと直角二等辺三角形が見える

15. -smooth1

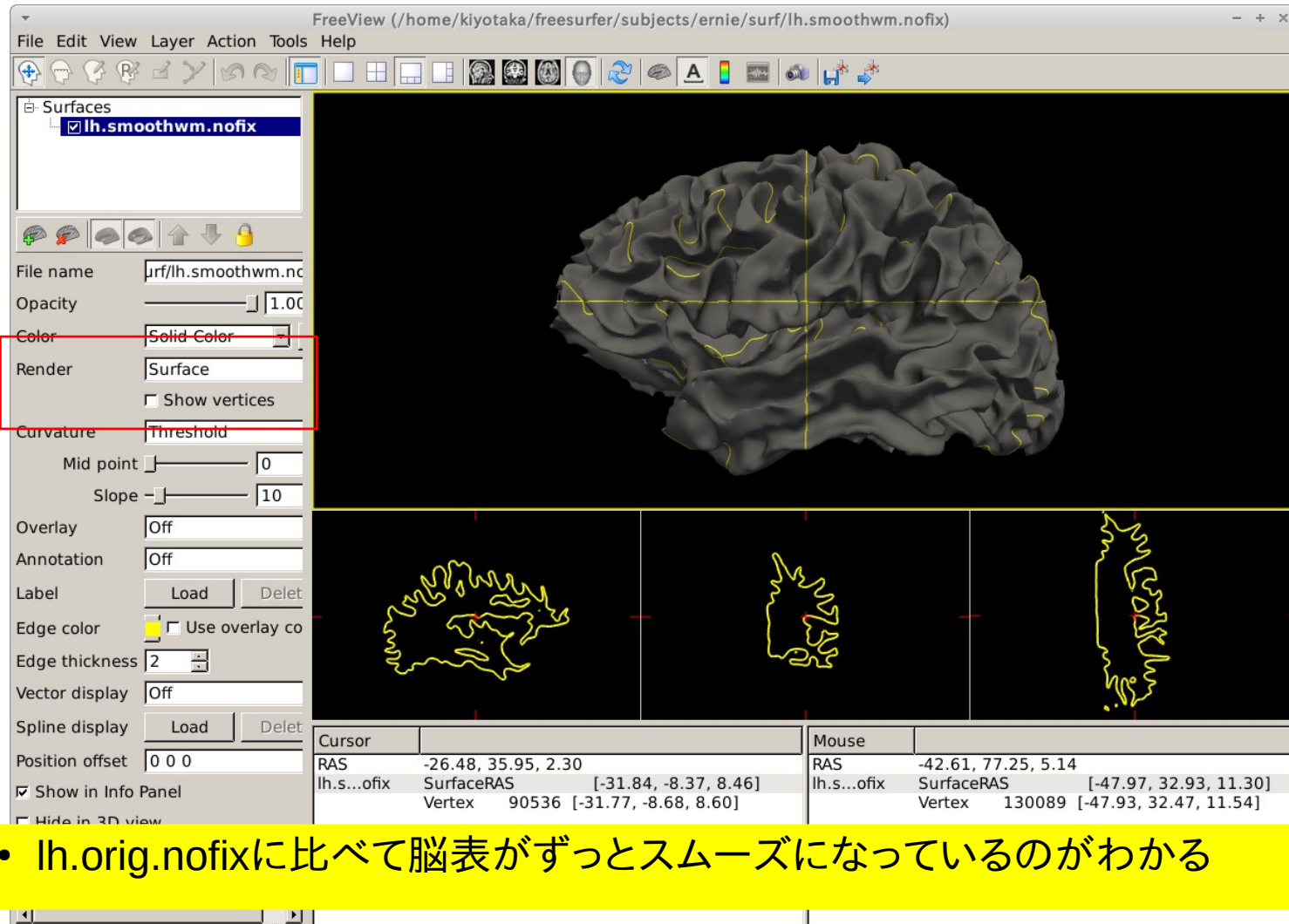
- white surfaceの表面をなめらかにする
 - 表面に敷き詰められた三角形の角度を調整する
 - 入力画像: ?h.orig.nofix
 - 出力画像: **?h.smoothwm.nofix**

平滑化された white surface 画像の確認

- `lh.smoothwm.nofix` を確認

```
freeview -f surf/lh.smoothwm.nofix \  
-layout 3 -viewport 3d
```

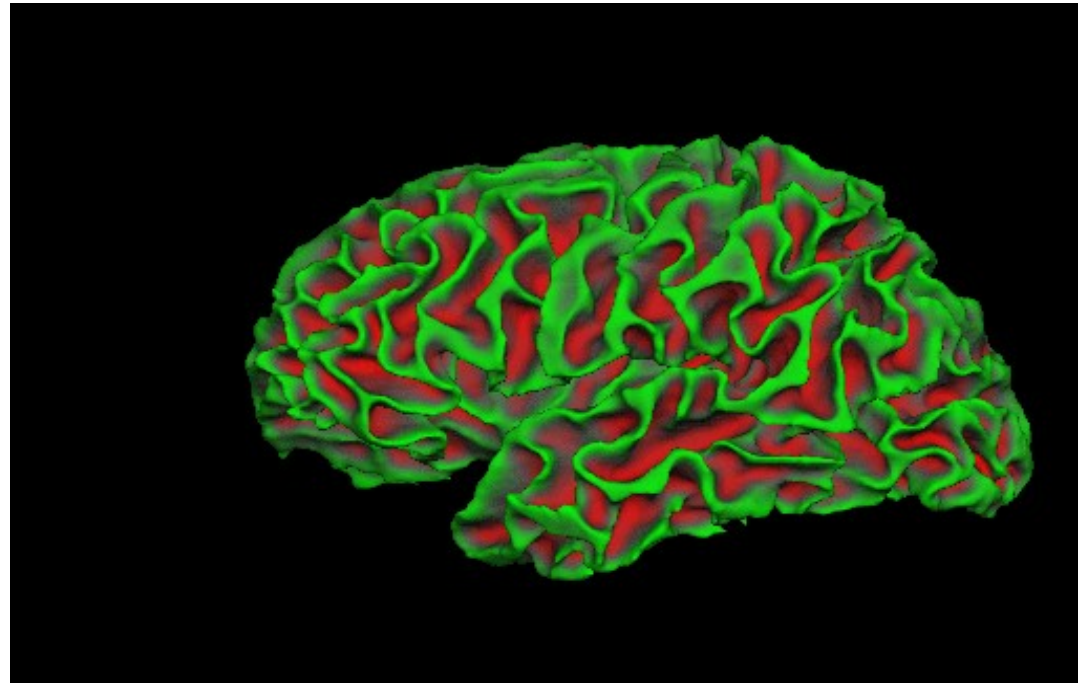
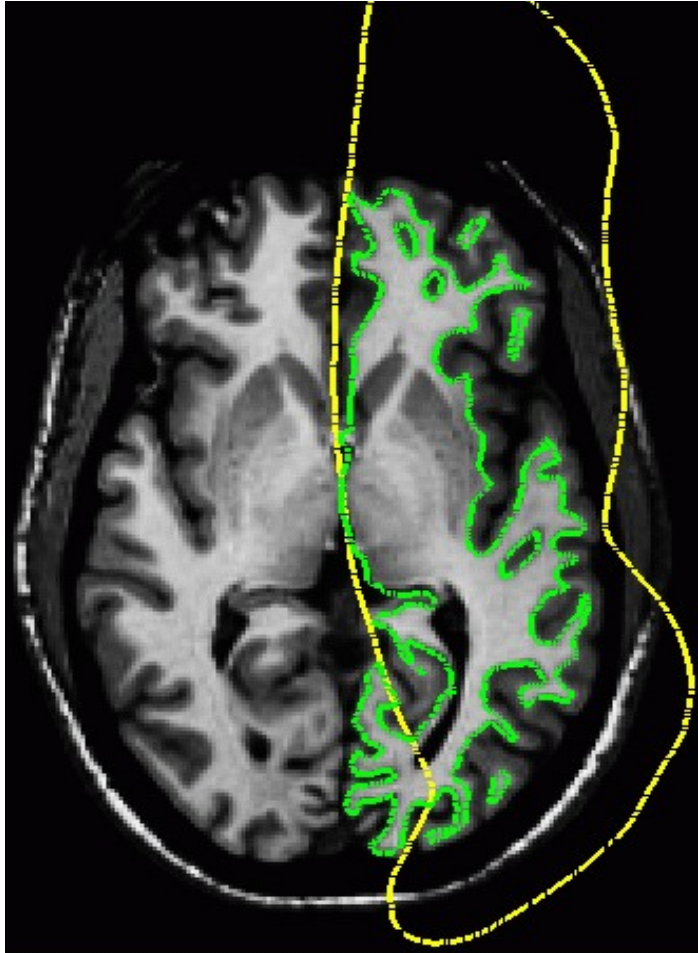
lh.smoothwm.nofix



16. -inflate1

- white surface のしわを引き伸ばす
 - 三角形の頂点同士の距離、そして面積の合計はできるだけ変わらないようにする
 - 風船のように引き伸ばしているわけではない
 - しわくちの紙袋をのばすようなイメージ
 - 入力画像: `?h.smoothwm.nofix`
 - 出力画像: `?h.inflated.nofix`

inflateの実際

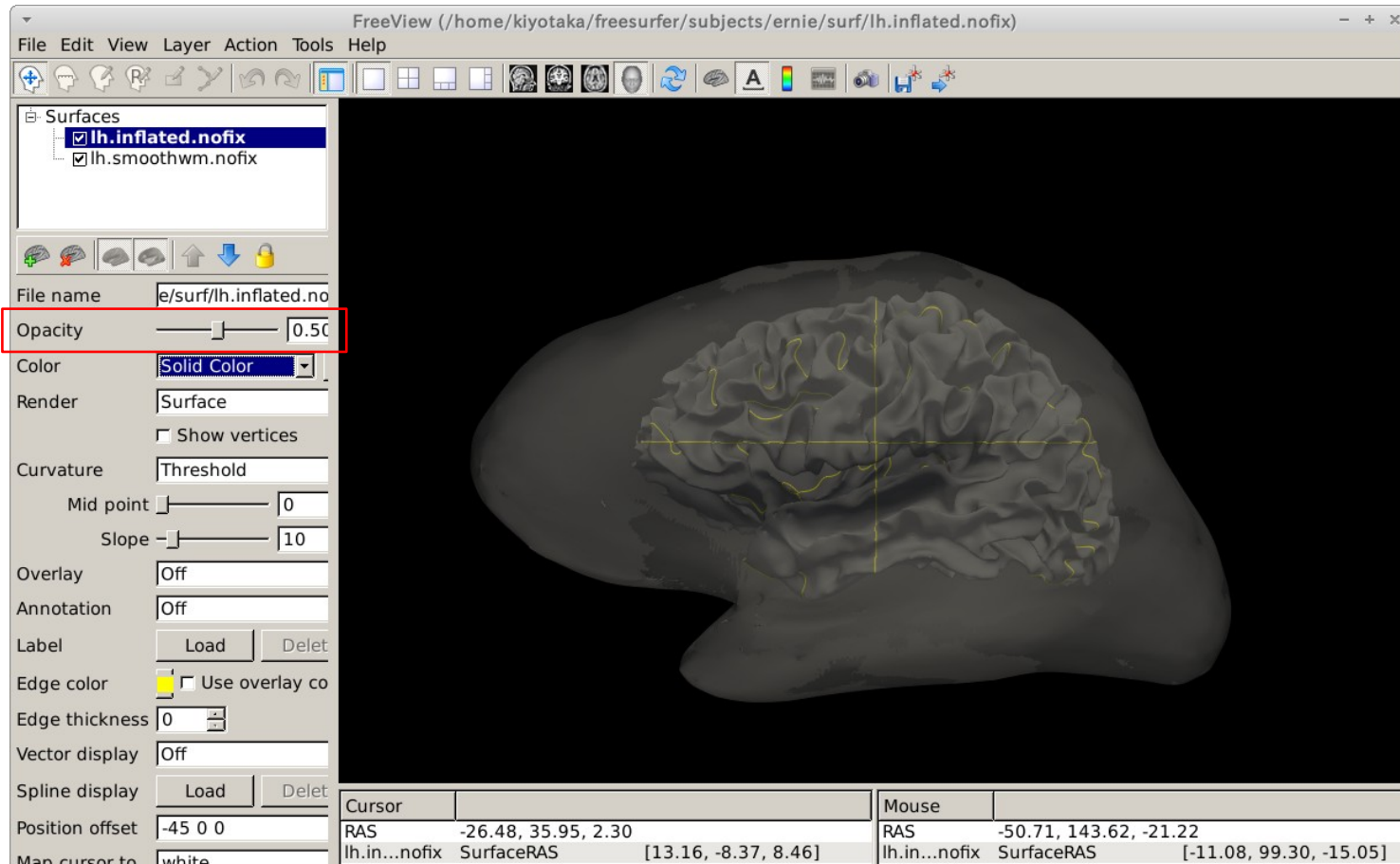


どれだけ inflate されるか確認

- lh.smoothwm.nofix と
lh.inflated.nofix を比較

```
freeview -f surf/lh.smoothwm.nofix \  
surf/lh.inflated.nofix \  
-layout 1 -viewport 3d
```

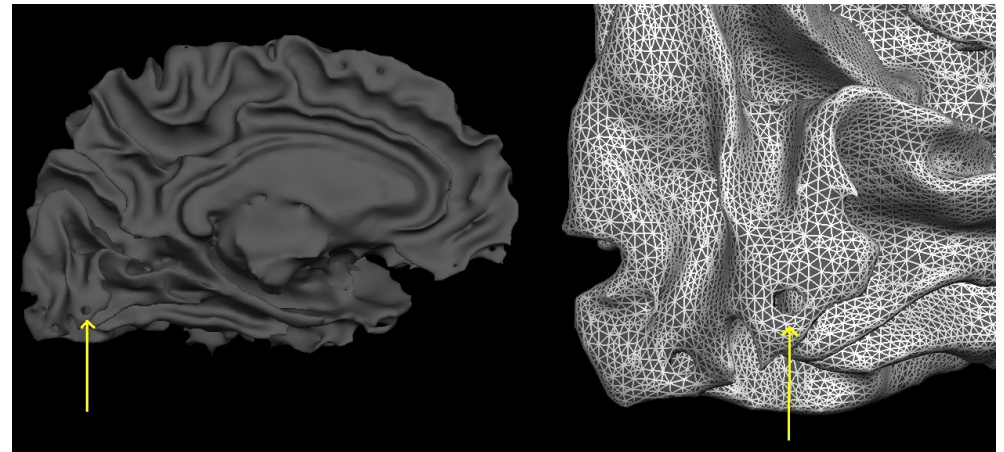
lh.inflated.nofix



- 左の画面から、Opacityを **0.5** に設定
- lh.smoothwm.nofixに比べてlh.inflated.nofixはずっと大きくなっていることがわかる

17. -qsphere

- filled.mgz にある穴を塞ぐ topology fixing の最初のステップ



- 入力画像: ?h.inflated.nofix
- 出力画像: **?h.qsphere.nofix**

18. -fix

- topological defects を塞ぐ
 - ?h.orig.nofix を修正し、?h.orig として保存される
 - 頂点の数が変わる
 - 入力画像: ?h.qsphere.nofix
 - 出力画像: **?h.orig**

頂点数などの確認

- `mriss_euler_number` を使うと、white surface の穴の数、頂点の数などを確認できる
- `lh.orig.nofix` (修正前) と `lh.orig` (修正後) を比較
- `v`: vertex 頂点の数 `f`: faces 面の数 `e`: edges 辺の数

`mriss_euler_number surf/lh.orig.nofix`

`mriss_euler_number surf/lh.orig`

`mrisc_euler_number`

`mrisc_euler_number surf/lh.orig.nofix`

`euler# =`

`v-e+f = 2g-2: 154212-462750+308500 = -38`

`--> 20 holes`

`mrisc_euler_number surf/lh.orig`

`euler# =`

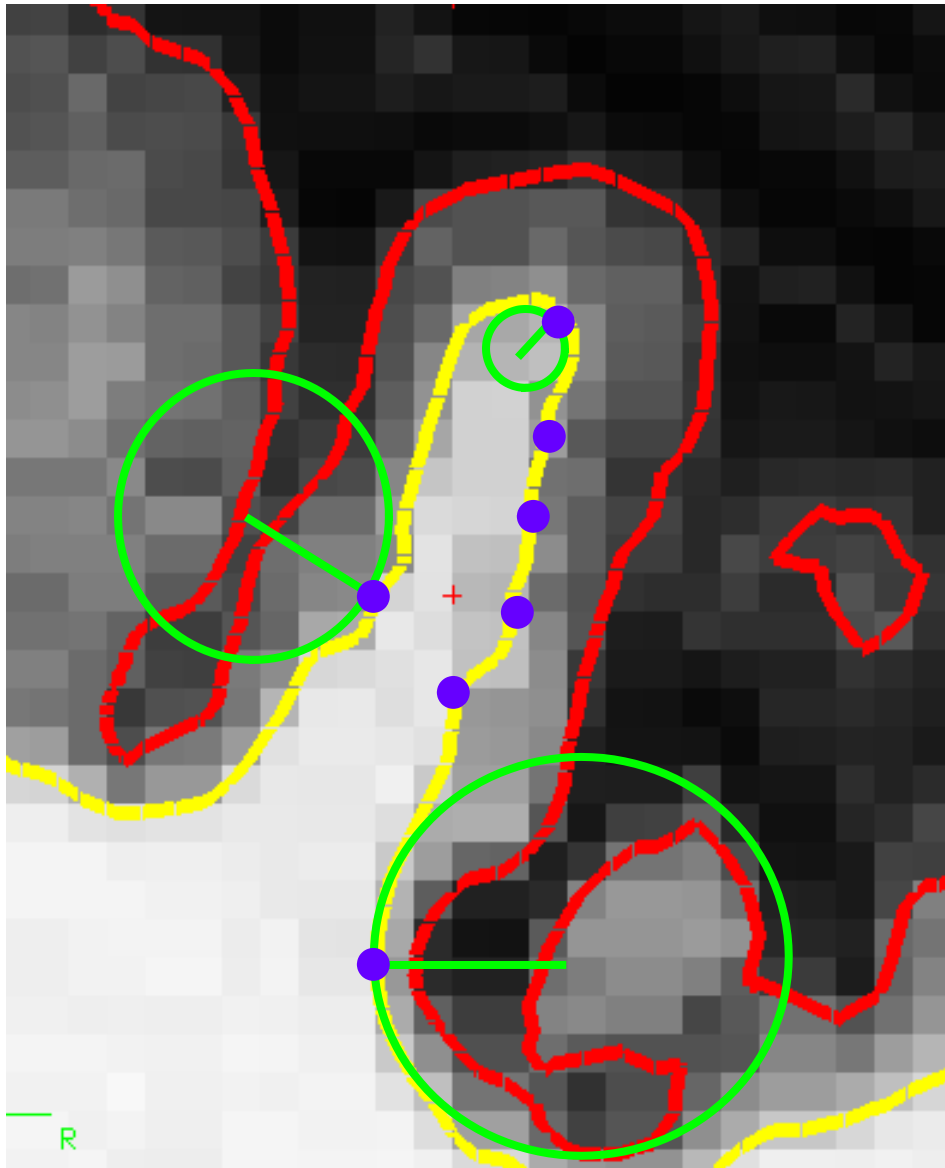
`v-e+f = 2g-2: 153341-460017+306678 = 2`

`--> 0 holes`

19. -white

- white surface のラベリングを行うための画像を作成
 - 同時に白質の曲率、面積なども算出する
 - 入力画像: `aseg.presurf.mgz`,
`brain.finalsurfs.mgz`, `wm.mgz`, `filled.mgz`
 - 出力画像: `?h.white.preaparc`,
`?h.curv`, `?h.area`, `label/?h.cortex.label`

曲率 Curvature



- 曲線を局所的に円弧とみなしたときの円の半径が曲率半径（緩やかなカーブで大きくなり急カーブで小さくなる）
- 曲率半径の逆数が曲率

20. -smooth2

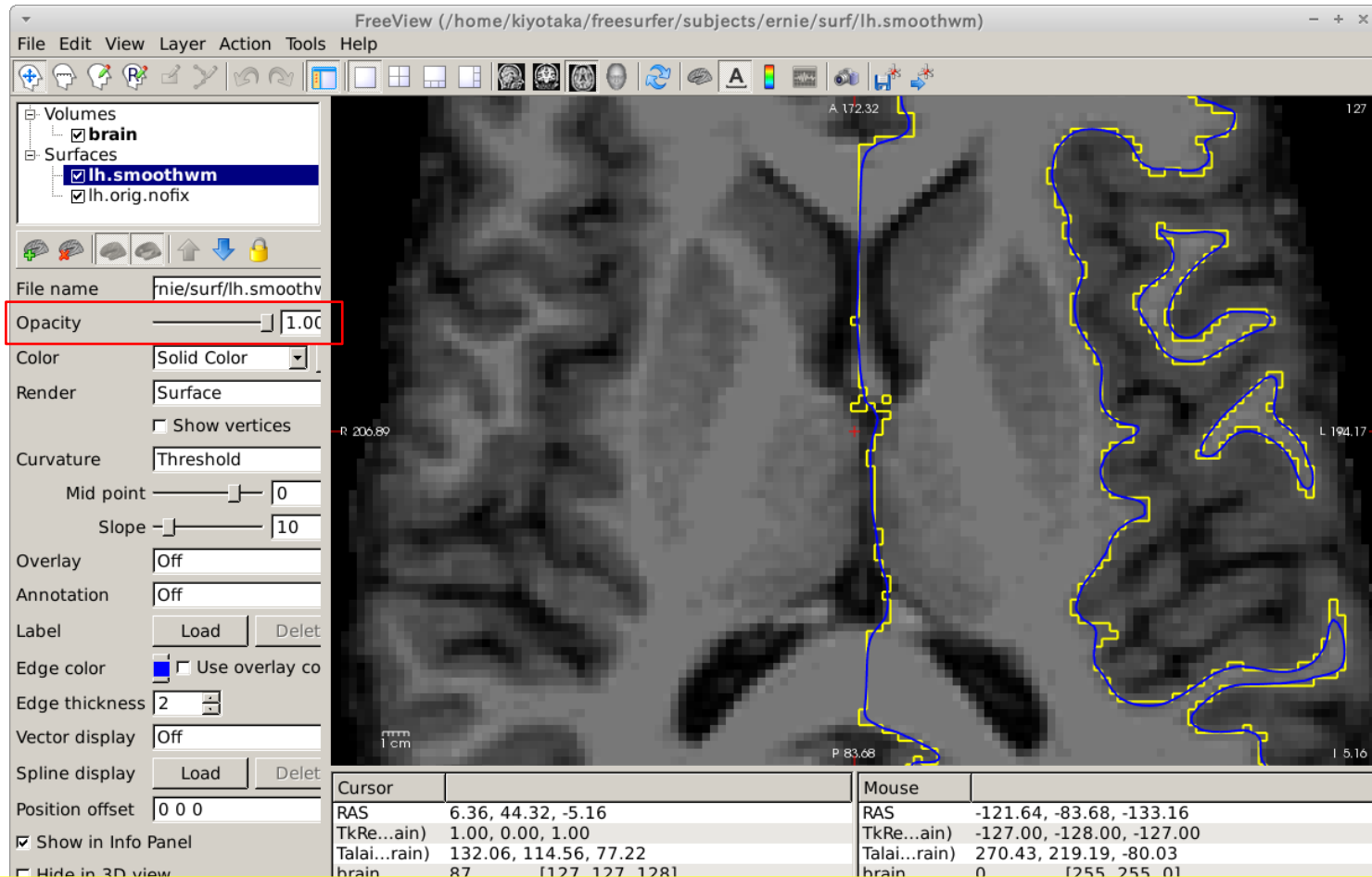
- white surface 画像をより滑らかにする
 - 入力画像: ?h.white.preaparc
 - 出力画像: ?h.smoothwm

平滑化の効果確認

- `lh.orig.nofix` と `lh.smoothwm` の違いを確認する

```
freeview -v mri/brain.mgz -f \  
surf/lh.orig.nofix:edgecolor=yellow \  
surf/lh.smoothwm:edgecolor=blue \  
-layout 1 -viewport axial -zoom 3
```

lh.orig.nofix と lh.smoothwm



- lh.orig.nofix の境界を黄色、lh.smoothwm の境界を青で表示
- lh.smoothwm は滑らかになっていることがわかる

21. -inflate2

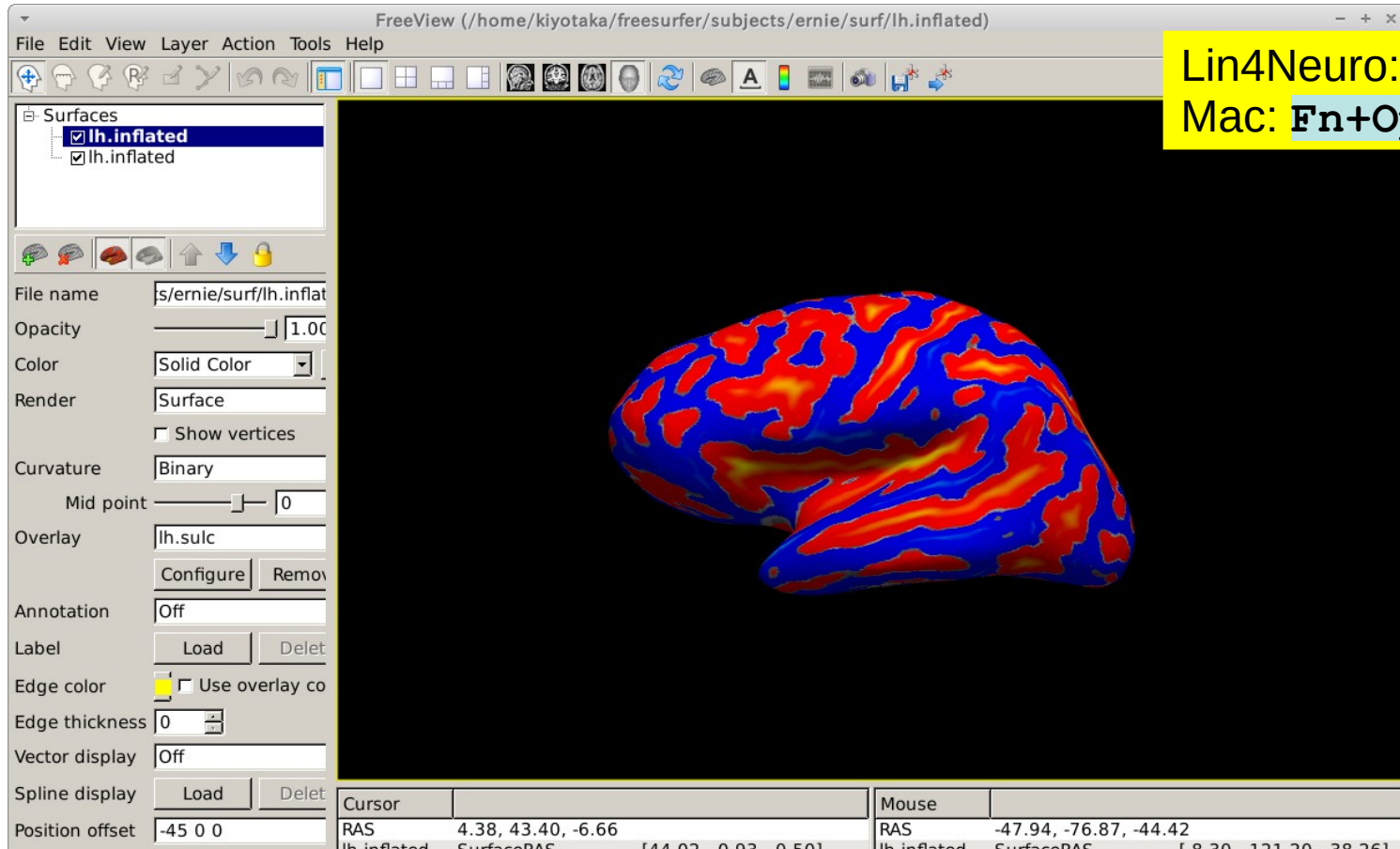
- lh.smoothwm を用いて inflate を行う
 - sulcは脳溝がどれだけ移動したかを視覚化
 - 移動距離が大きい(=深いところにある)ものほど数値が大きい
 - 入力画像: ?h.smoothwm
 - 出力画像: ?h.inflated, ?h.sulc

lh.inflated と lh.sulc の確認

- lh.inflated と lh.sulc を確認する

```
freeview -f surf/lh.inflated \  
surf/lh.inflated:overlay=lh.sulc \  
-layout 1 -viewport 3d
```

lh.inflated と lh.sulc の比較



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

- 上記のキーボード・ショートカットを使って表示する画像を切り替え
- lh.sulc は脳溝が赤く表示される
- より明るい色はより「深い」ところにあることを意味する

22. -curvHK

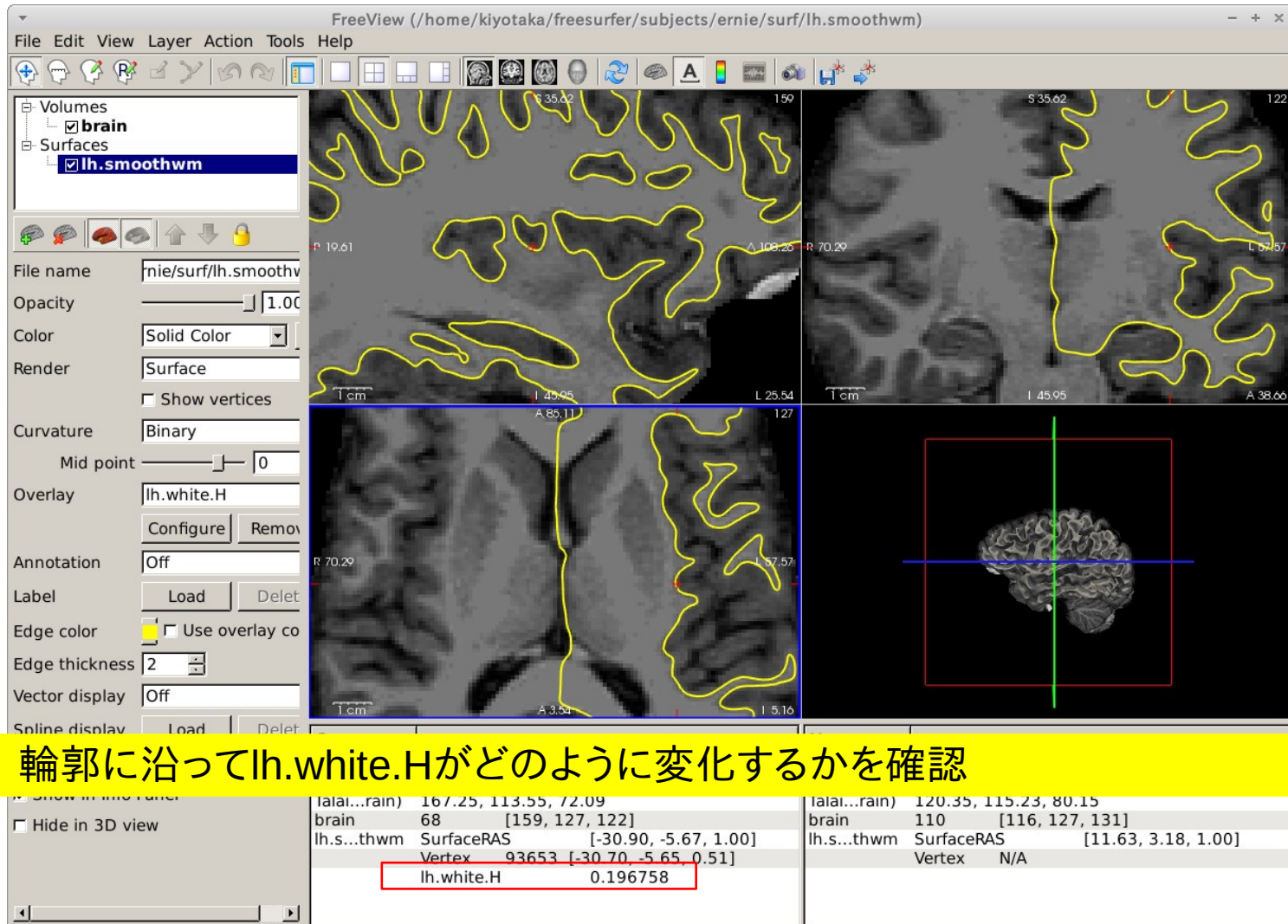
- 曲率に関する情報
 - .H: mean curvature
 - 平均曲率
 - .K: gaussian curvature
 - ガウス曲率
 - 凸の場合は正の値、凹の場合は負の値をとる
 - 入力画像: ?h.white.preaparc, ?h.inflated
 - 出力画像: ?h.white.H, ?h.white.K,
?h.inflated,H, ?h.inflated.K

lh.white.H の確認

- lh.white.H を確認する

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


lh.white.H の確認



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

File Edit View Layer Action Tools Help

Volumes
 ☒ brain
Surfaces
 ☒ lh.smoothwm

File name: /home/kiyotaka/freesurfer/subjects/ernie/surf/lh.smoothwm
Opacity: 1.00
Color: Solid Color
Render: Surface
☐ Show vertices
Curvature: Binary
Mid point: 0
Overlay: lh.white.H
Annotation: Off
Label: Load Delete
Edge color: ☒ Use overlay color
Edge thickness: 2
Vector display: Off
Spline display: Load Delete

lh.white.H 0.196758

- 輪郭に沿ってlh.white.Hがどのように変化するかを確認

23. -curvstats

- 曲率の統計値情報を計算
 - statsディレクトリに `?h.curv.stats` を生成
 - 入力画像: `?h.smoothwm`, `?h.curv`, `?h.sulc`
 - 出力ファイル: `stats/?h.curv.stats`

lh.curv.stats の確認

- `lh.curv.stats` を確認

`less stats/lh.curv.stats`

```
Raw <mean> +- <std> (using 'lh.curv'):          -0.03268 +- 0.2279
Raw Min:                                         -17.68809 at vertex 90769
Raw Max:                                         5.53396 at vertex 103850
Raw Curvature Calculation Type:                 discrete
Raw Mean Vertex Separation (Whole Surface):     0.84 +- 0.23 mm
Raw Total Surface Area:                         91794.71094 mm^2
Raw Total Number of Vertices:                   153341
Raw Average Vertex Area (Whole Surface):        0.59863 mm^2
```

- `less` コマンドは、`q` をタイプすると終了

recon-all -autorecon2 のまとめ

- filled.mgz の生成がハイライト
- これがうまくいかないと、topological defects ができる → white surface が正しく抽出されなくなる
- そのための方策
 - コントロールポイントを設定する→どこまで白質があるかを教えてあげる
 - filled.mgz を編集する→Topogogical defects をなくすことができる
- この編集を行った後に、以下を実行

recon-all -autorecon2-wm

autorecon2-wm と -cp

- FreeSurferのホームページには、以下の記載がある
 - after autorecon2, check final surfaces:
 - a. if wm edit was required, then run -autorecon2-wm
 - b. if control points added, then run -autorecon2-cp
 - c. if edits made to correct pial, then run -autorecon2-pial
- しかし、今回ひとつひとつ見たことで以下が判明
 - autorecon2-pial は、Ver.6ではなくなった (autorecon2では pial画像は生成されない)
 - autorecon2-wm と autorecon2-cp で行われることは同じ!
 - recon-allの6800行～6802行を見ると、どちらを選んでも同じことをすることがわかる

Questions?