

第1部(1) recon-all -autorecon1

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 -autorecon1` 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

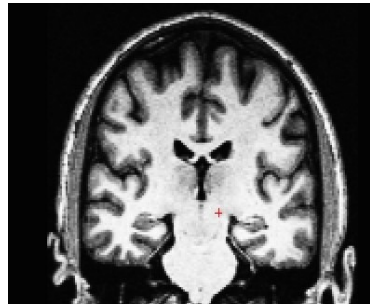
本セクションのチートシート

- `nisg-202001/docs` にある `ex1.html` をダブルクリック
- コマンドの意味も記載されている

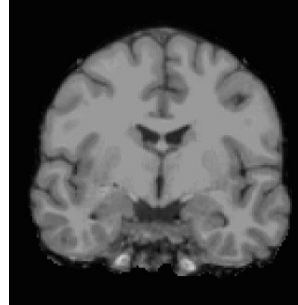
FreeSurferでの画像管理

- FreeSurferは、変数 **SUBJECTS_DIR** で指定したディレクトリに画像を保存する
- FreeSurfer ID (`fsid`) の下に同じ名前の画像が保存されていく
- `fsid` をしっかり決めておかないと混乱する

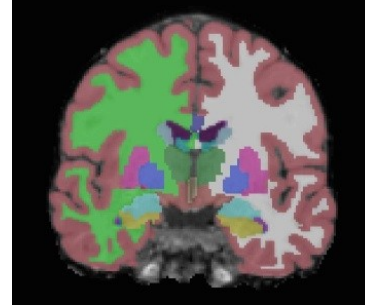
recon-allの概要



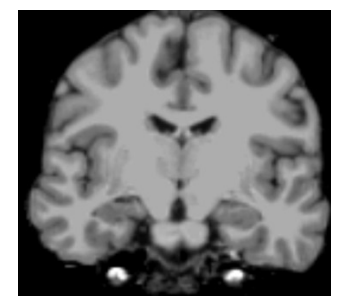
入力画像
T1強調画像



頭蓋骨除去



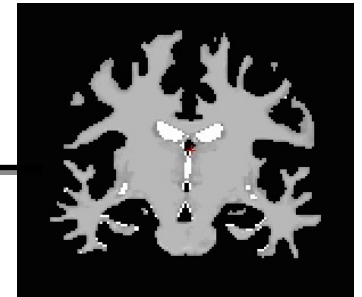
Volumeの
ラベリング



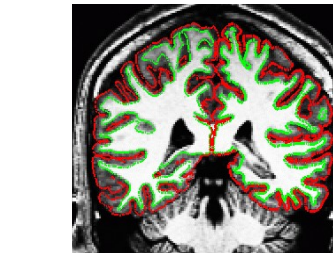
信号値の
正規化

-autorecon1

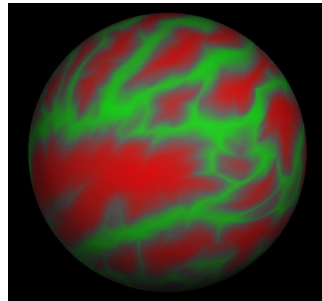
-autorecon2 (下に続く)



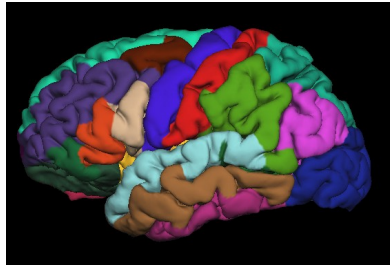
白質の分離



Surfaceの抽出



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!

-autorecon2

FreeSurferスライドを改変

recon-all

- `recon-all -s <fsid> -i <input> <directives>`
- 全自動で行いたい場合、directives は `-all`
- fsid が s01, 入力画像が 01.nii で全自動で行う場合、コマンドは以下になる

`recon-all -s s01 -i 01.nii -all`

recon-allの各ステップ

- recon-allは **directives** を指定することで、各ステップを個別に実行できる
- 具体的な **directive** は下記に記載
- <https://surfer.nmr.mgh.harvard.edu/fswiki/ReconAllTableStableV6.0>
 - <https://surfer.nmr.mgh.harvard.edu/fswiki/recon-all> は古い情報が混じっている
 - recon-allのプログラムのヘルプも古い情報が混じっている

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

サンプルデータの表示

- FreeSurferには、サンプルデータが提供されている
 - 同一人物の2回のスキャン
 - sample-001.mgz, sample-002.mgz
- サンプルデータを freeview で確認
- volumeデータは -v をつけて指定する

```
cd $SUBJECTS_DIR
```

```
freeview -v sample-001.mgz \
```

```
sample-002.mgz -layout 2 \
```

```
-viewport sagittal
```

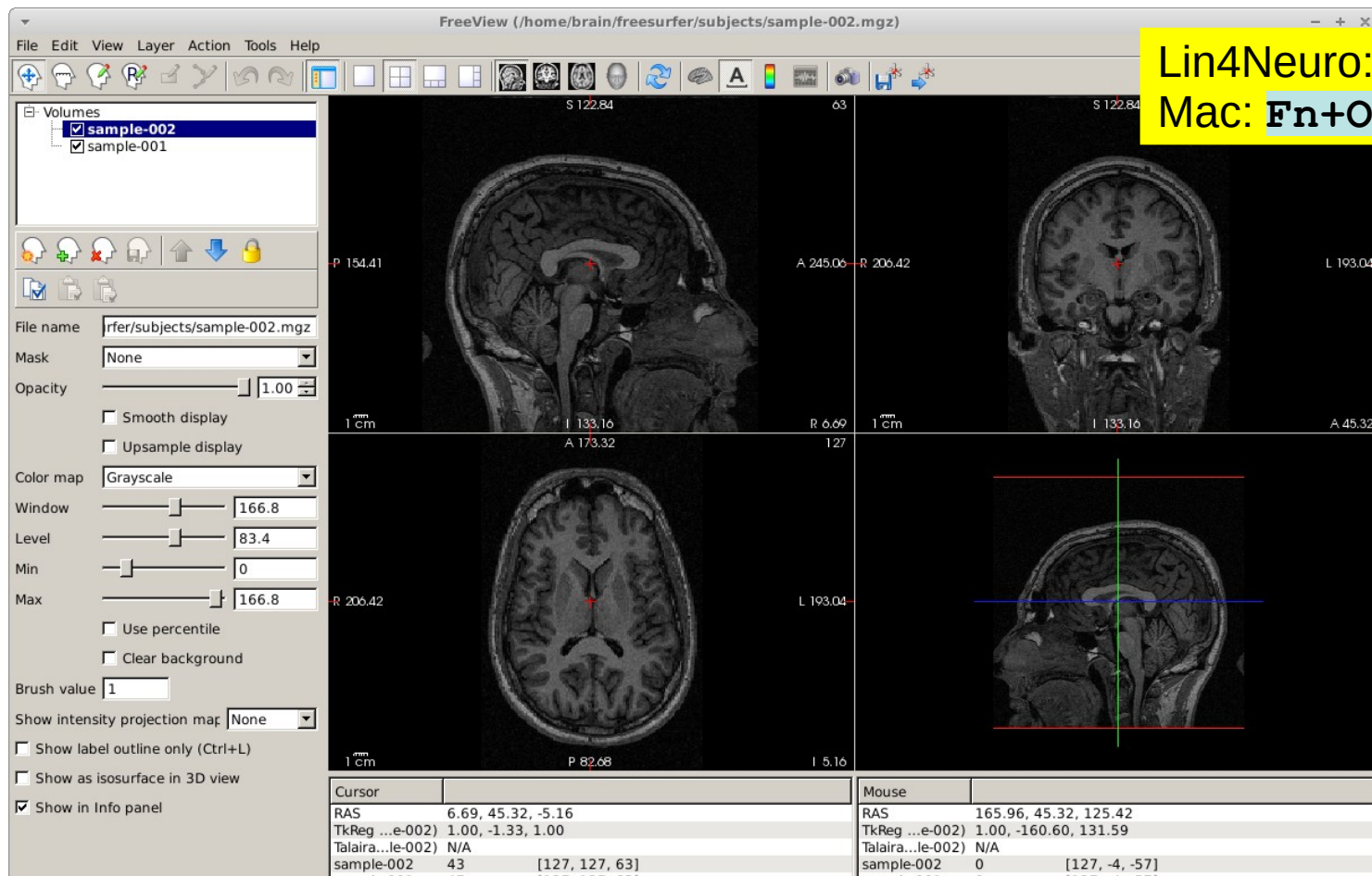
Linux/MacOS での \ の意味

- コマンドは、すべて1行で指示する
- しかし、とても長いコマンドの時は改行したくなる
- 改行したい時は、改行の前に \ をタイプすることで、「まだコマンドは続いているよ」とシェルに伝えることができる
- 日本語環境では、\ は、¥ と表示されることもある
- 以下の2つは同じ

```
freeview -v 001.mgz 002.mgz
```

```
freeview -v \  
    001.mgz \  
    002.mgz
```

sample-00[12].mgz



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

- 上記のキーボード・ショートカットを使うと、表示する画像を切り替えられる
- sample-001.mgzとsample-002.mgzが微妙に位置がずれていることを確認

sample-001.mgzのヘッダーを表示

- `mri_info`: FreeSurferに入っている画像のヘッダー情報を表示するプログラム
- DICOM, nifti, mgz など、FreeSurferが対応しているファイルのヘッダー情報を表示できる
- Freeviewを閉じてから、以下をタイプ

```
mri_info sample-001.mgz
```

mri_info sample-001.mgz

- 一部を抜粋
- dimensions: 256 x 256 x 128
 - 256 x 256 の画像が128枚並んでいる
- voxel sizes: 1.000000, 1.000000, 1.328125
 - ボクセルサイズは 1 x 1 x 1.3
- Primary Slice Direction: sagittal
 - 矢状断画像

勉強会のディレクトリの変数設定

- sample-001.mgz と sample-002.mgz を用いて前処理を行った画像を fsid を ernie として準備
- これ以降は、ernie ディレクトリの中を見ていく
- まずは、勉強会のディレクトリを nimg という変数に設定
 - nimg: neuroimaging study group
- Lin4Neuroの場合

```
nimg=/media/sf_share/nimg-202001
```
- MacOSの場合

```
nimg=~/.git/nimg-202001
```
- こうすると、勉強会ディレクトリは `$nimg` で指定できる

SUBJECTS_DIRの設定

```
SUBJECTS_DIR=$nisg/subjects
```

```
cd $SUBJECTS_DIR/ernie
```

- 勉強会ディレクトリの下にある subjects ディレクトリを SUBJECTS_DIR に設定
- SUBJECTS_DIR の下にある ernie に移動

recon-all -all で使われる directives (1)

- 01 -motioncor
- 02 -talairach
- 03 -nuintensitycor
- 04 -normalization
- 05 -skullstrip
- 06 -gcareg
- 07 -canorm
- 08 -careg
- 09 -calabel
- 10 -normalization2
- 11 -maskbfs
- 12 -segmentation
- 13 -fill
- 14 -tessellate
- 15 -smooth1
- 16 -inflate1
- 17 -qsphere
- 18 -fix
- 19 -white
- 20 -smooth2

recon-all -all で使われる directives (2)

- 21 -inflate2
- 22 -curvHK
- 23 -curvstats
- 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

`recon-all -s <fsid> -i <input>`

- 画像を指定すると、SUBJECTS_DIR の下に fsid ディレクトリが作成され、画像が mri/orig ディレクトリに保存される

`recon-all -s ernie -i sample-001.mgz -i sample-002.mgz` #既に実行済みのためタイプする必要はない

- \$SUBJECTS_DIR の下に ernie/mri/orig が作成され、入力画像が以下に配置 (ファイル名のみ変更される)
 - sample-001.mgz → mri/orig/001.mgz
 - sample-002.mgz → mri/orig/002.mgz

recon-all -autorecon1 で 使われる directives

- 01 -motioncor
- 02 -talairach
- 03 -nuintensitycor
- 04 -normalization
- 05 -skullstrip

1. -motioncor

- 動き補正と画像のリスライス
 - 2つ以上の画像がある場合、動き補正を行い、平均画像を作成
 - 入力画像: `orig/001.mgz orig/002.mgz`
 - 出力画像: `rawavg.mgz`
 - 平均画像をリスライスし、dimensionを256x256x256 に、ボクセルサイズを 1x1x1 に変更
 - 入力画像: `rawavg.mgz`
 - 出力画像: `orig.mgz`

{rawavg,orig}.mgzのヘッダーを確認

```
cd mri
```

```
mri_info rawavg.mgz
```

- dimensions: 256 x 256 x 128
- voxel sizes: 1.000000, 1.000000, 1.328125

```
mri_info orig.mgz
```

- dimensions: 256 x 256 x 256
- voxel sizes: 1.000000, 1.000000, 1.000000

リスライスすることの意義

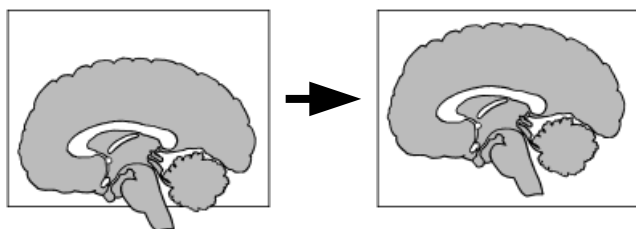
- すべての画像をdimensionとボクセルサイズを揃えることのメリット
 - 異なる画像の重ね合わせが容易になる
 - 様々なMRIで撮像された画像のdimensionとボクセルサイズを統一することにより、それ以降の処理を統一できる
- ボクセルサイズはデフォルトでは 1x1x1 になることに注意
 - 高解像度で撮像したとしても、デフォルトではこのサイズになる

2. -talairach

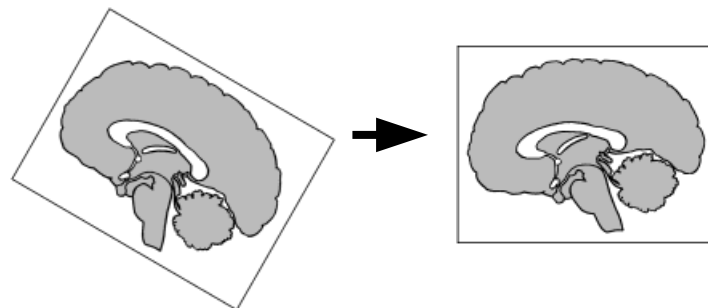
- Talairach変換
 - `orig.mgz` をMNI305アトラスにアフィン変換
 - 入力画像: `orig.mgz`
 - 出力ファイル: `talairach.xfm`

アフィン変換

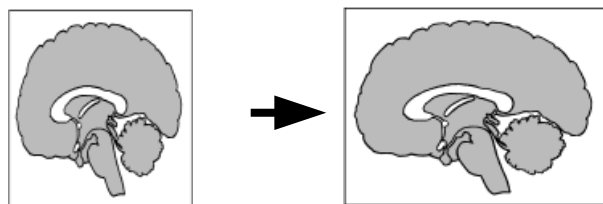
- アフィン変換: 以下の4種類の変換
- 4種類×3次元(x,y,z方向) = 12パラメータ
 - *平行移動と回転は 剛体変換 rigid body transform という



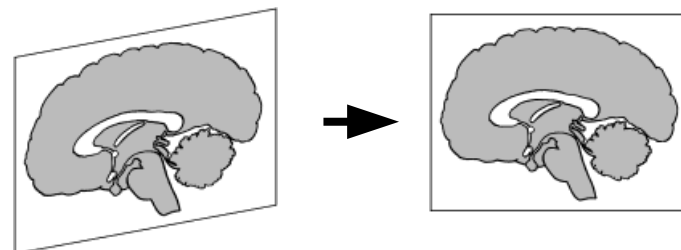
Translate 平行移動



Rotate 回転



Zoom



Shear

talairach.xfmの確認

- (ernie/mri にいる前提)

```
cd transforms
```

```
cat talairach.xfm
```

```
Transform_Type = Linear;
```

```
Linear_Transform =
```

```
1.104743 -0.009389 -0.014386 -5.386261
```

```
-0.007588 0.947063 0.289007 -46.896755
```

```
0.014128 -0.257706 1.061050 24.091187;
```

Freeviewで標準化の結果確認

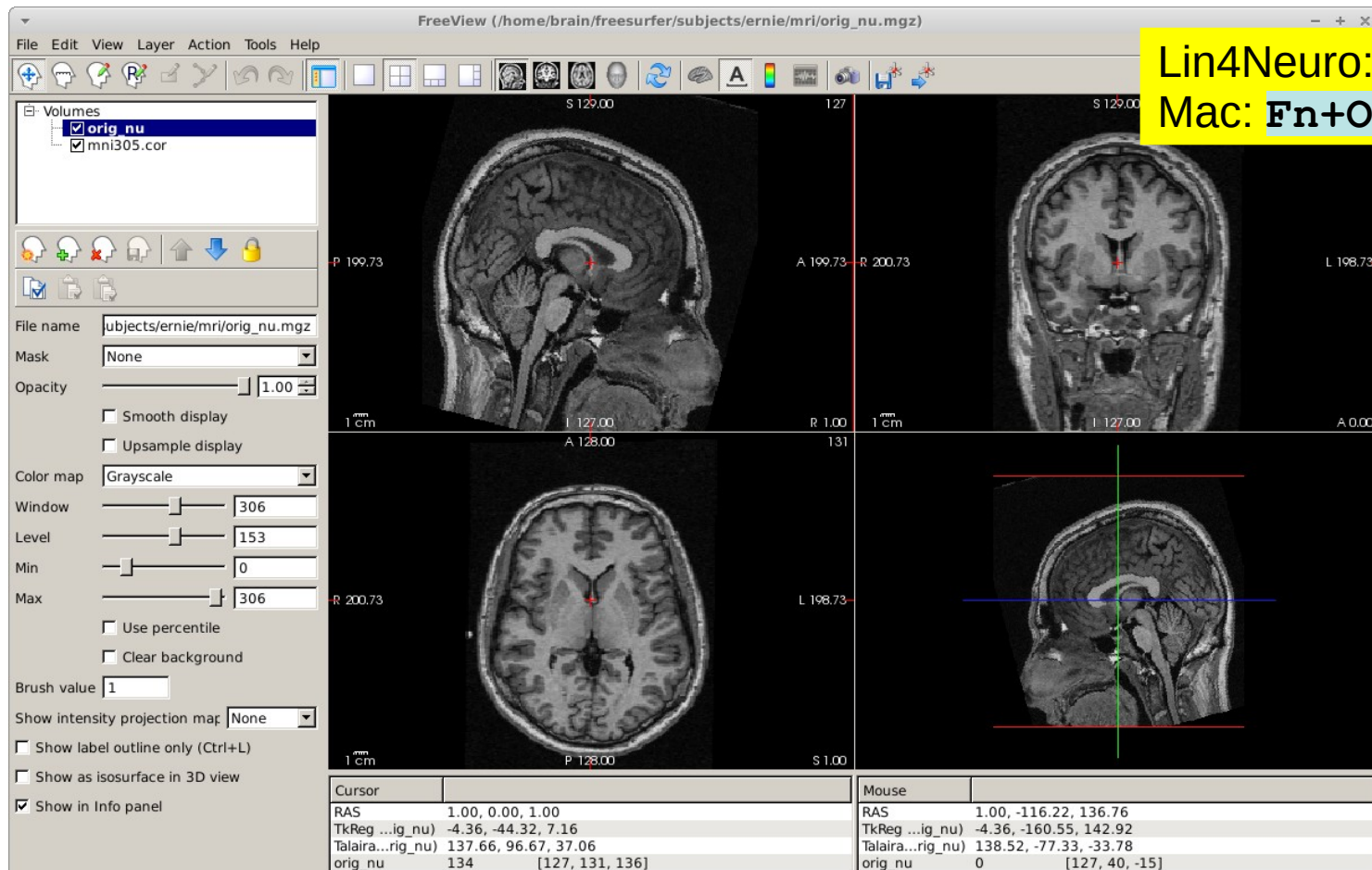
- -talairachはSPMやFSLのように、標準化画像は作成しない
- しかし、今確認したtalairach.xfmを用いることで、どのように標準化されるかを表示できる
- そのためには、FreeSurferについてくるAtlas (fsaverage/mri/mni.305.cor.mgz) を使う

```
cd $SUBJECTS_DIR
```

```
freeview -v fsaverage/mri/mni305.cor.mgz \
```

```
ernie/mri/orig_nu.mgz:reg=ernie/mri/transforms/talairach.xfm
```

標準化の結果確認



3. -nuintensitycor

- 信号値補正
 - MRIの信号ムラを補正するために、Non-parametric Non-uniform intensity Normalization (N3) を実行
 - デフォルトでは4回繰り返す
 - 入力画像: `orig.mgz`
 - 出力画像: `nu.mgz`

4. -normalization

- 信号値の正規化
 - N3の信号値補正の後、さらに信号ムラの補正を行う
 - 白質の信号値を **110** に設定する
 - ここでうまくいかない場合、コントロールポイントを設定することで、修正できる
 - 入力画像: `nu.mgz`
 - 出力画像: **T1.mgz**

信号値の確認

- 白質の信号値がどう変化したかを確認
- `orig.mgz`, `nu.mgz`, `T1.mgz` を読み込む

```
cd $SUBJECTS_DIR/ernie/mri
```

```
freeview -v orig.mgz nu.mgz T1.mgz
```

信号値の変化

FreeView (/home/brain/freesurfer/subjects/ernie/mri/T1.mgz)

File Edit View Layer Action Tools Help

Volumes: T1, nu, orig

File name: /rfer/subjects/ernie/mri/T1.mgz

Mask: None

Opacity: 1.00

Color map: Grayscale

Window: 261.6

Level: 130.8

Min: 0

Max: 261.6

Brush value: 1

Show intensity projection map: None

Show label outline only (Ctrl+L):

Show as isosurface in 3D view:

Show in Info panel: ☒

Lin4Neuro: **Alt+C**

Mac: **Fn+Option+C**

- 明らかに白質というところをクリック
- 下に表示されている信号値を確認
- T1は110となっている

Cursor	
RAS	24.44, 89.58, -5.16
TkReg RAS (T1)	19.08, 45.25, 1.00
Talairach (T1)	112.51, 125.96, 119.94
T1	110 [109, 127, 173]
nu	112 [109, 127, 173]
orig	168 [109, 127, 173]

Mouse	
RAS	24.44, -15.08, 125.42
TkReg RAS (T1)	19.08, -59.40, 131.59
Talairach (T1)	113.41, -39.57, 58.56
T1	0 [109, -4, 69]
nu	0 [109, -4, 69]
orig	0 [109, -4, 69]

5. -skullstrip

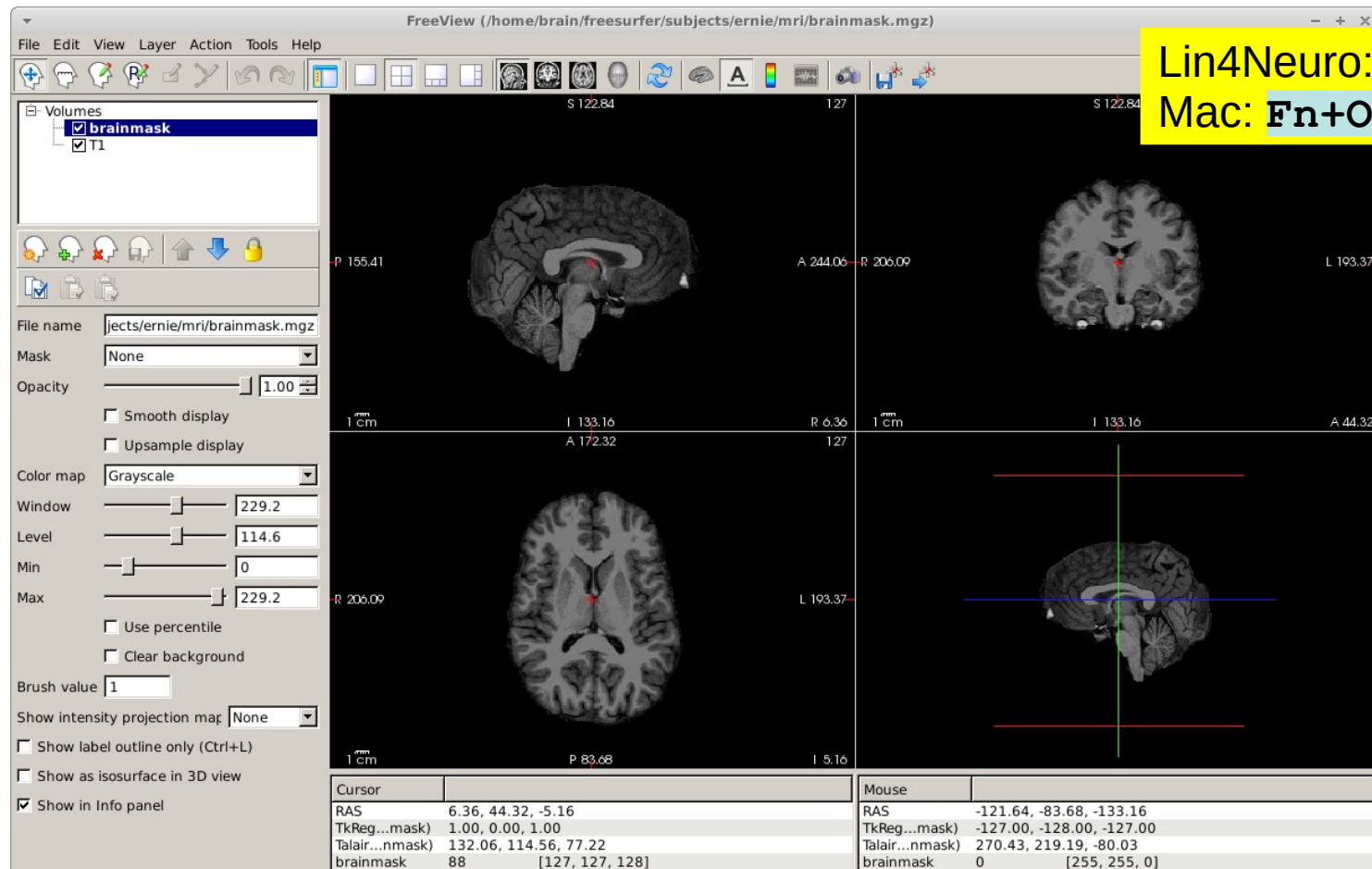
- 頭蓋骨の除去
 - T1.mgz から頭蓋骨の除去を行う
 - ここでうまくいかない場合、seed pointを追加するか、`mri_watershed` の閾値設定を変更する (`-wsthresh`, `-wsmore`, `-wsless`)
 - 入力画像: `T1.mgz`
 - 出力画像: **`brainmask.mgz`**, **`brainmask.auto.mgz`**
 - `brainmask.mgz` と `brainmask.auto.mgz` は同一のファイル
 - `mri_watershed` の出力が `brainmask.auto.mgz` で、それを `brainmask.mgz` としてコピーしている

頭蓋骨除去の確認

- 頭蓋骨がどのように除去されたかを確認
- T1.mgz, brainmask.mgz を読み込む

```
freeview -v T1.mgz brainmask.mgz
```

頭蓋骨除去の結果確認



- 上記のキーボード・ショートカットを使うと、表示する画像を切り替え
- 頭蓋骨が取り除かれていることを確認

recon-all -autorecon1のまとめ

- 脳外組織の除去
- `brainmask.mgz` を見れば、このステップがうまくいったかわかる
- Quality Checkのポイント: 脳外組織が除去されているかどうか

Questions?