

# Methods Core

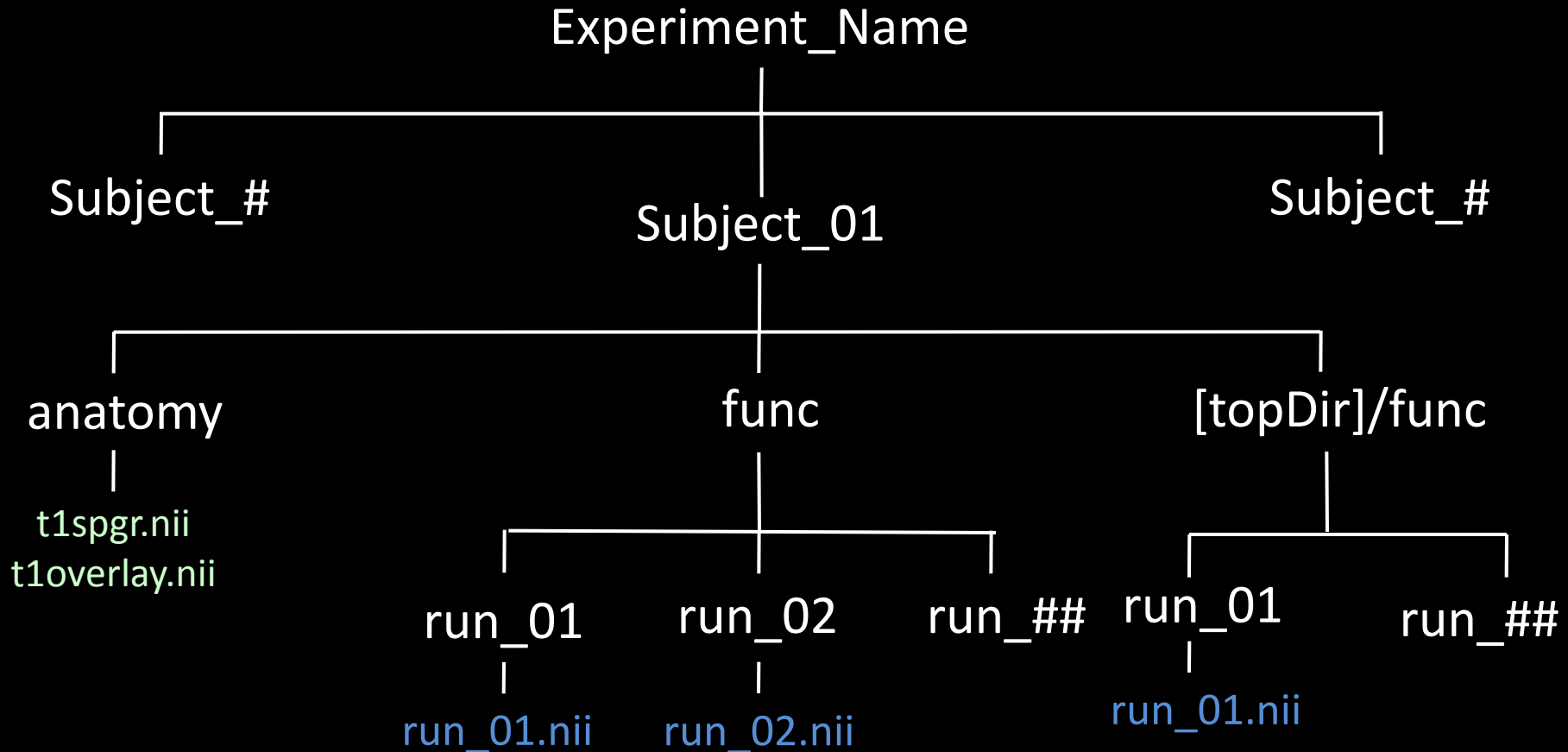


## Preprocessing Documentation

# Requirements

- SPM8 with VBM8 toolbox
- FSL 4.1.9 or higher
- Bash
- 4D NIFTI images (.nii)
- Standard directory structure

# Expected Directory Structure

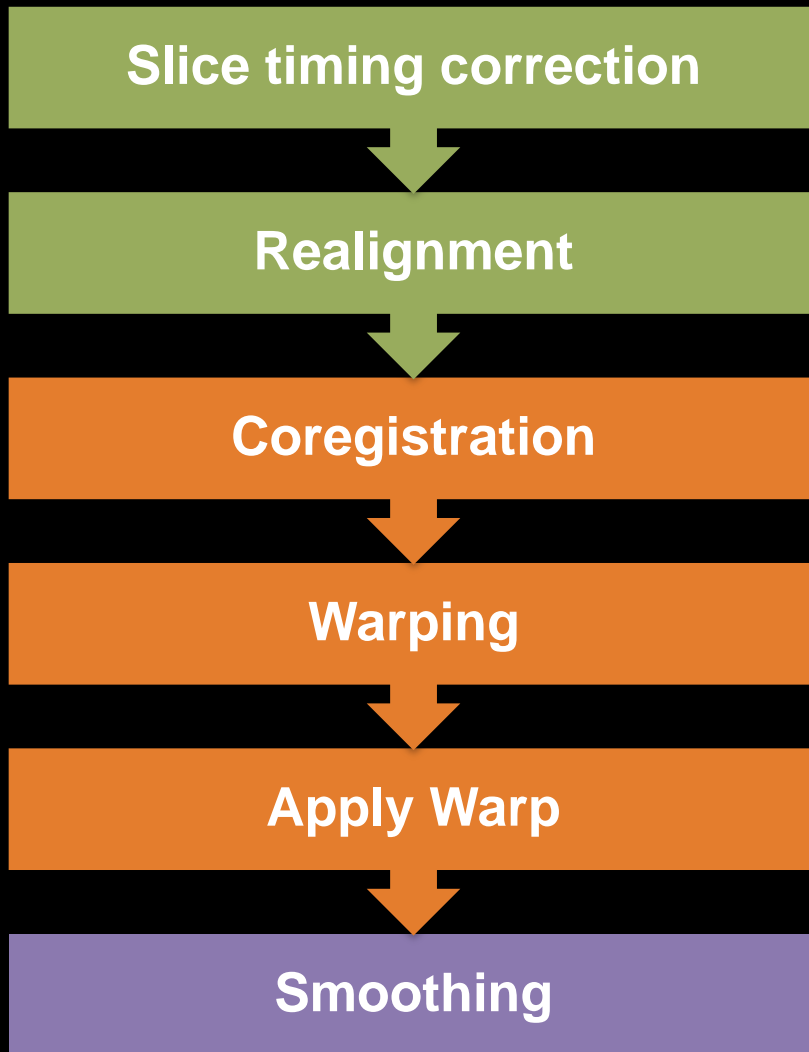


Note: All example commands will use this directory structure and are executed while at the directory `Experiment_Name` in a Unix shell.

# Command Features

- Minimize user script writing
- Extensive logging for all processing
- Allows quality control between commands
- Allows for large scale data throughput
- Built-in help (just type command name at shell)
- Launches to background
- Email/text message notification
- Optimized performance

# Processing Stream



**sliceTime8**

**realignfMRI**

**coregOverlay**

**coregHiRes**

**vbm8HiRes**

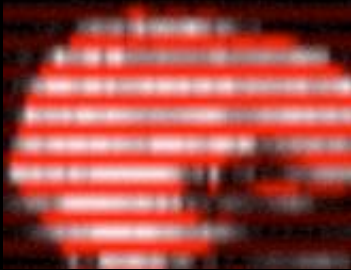
**vbm8Check**

**warpfMRI**

**vbm8Check**

**smoothfMRI**

# sliceTime8



This script uses SPM8 to perform slice time correction.

TYPE THIS COMMAND BY ITSELF:

`sliceTime8`

View information & available options. All commands have this feature.

# sliceTime8

sliceTime8 -v run -M ./ <Subjects>

-U youremail@umich.edu

## sliceTime8 Options

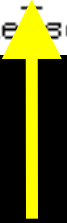
-B	run in the foreground
-D	enable super debug flag
-d	enable debug flag
-F	set the TR for volume acquisition
-f [directory]	functional directory e.g. connect/func
-G	set the value for the reference slice
-M [directory]	master subject directory
-n [name]	name prepend
-O	slice acquisition order
-t	test flag
-U [unique]	user email name/txt msg address
-v [name]	volume name (name of functional file)

# sliceTime8

If you typed everything correctly, you will get an email when sliceTime8 is complete.

List the contents of Subject\_01's run\_01 directory:

```
[chelsea@venus Experiment_Chelsea]$ ls Subject_01/func/run_01/  
arun_01.nii  run_01.nii  
[chelsea@venus Experiment_Chelsea]$
```

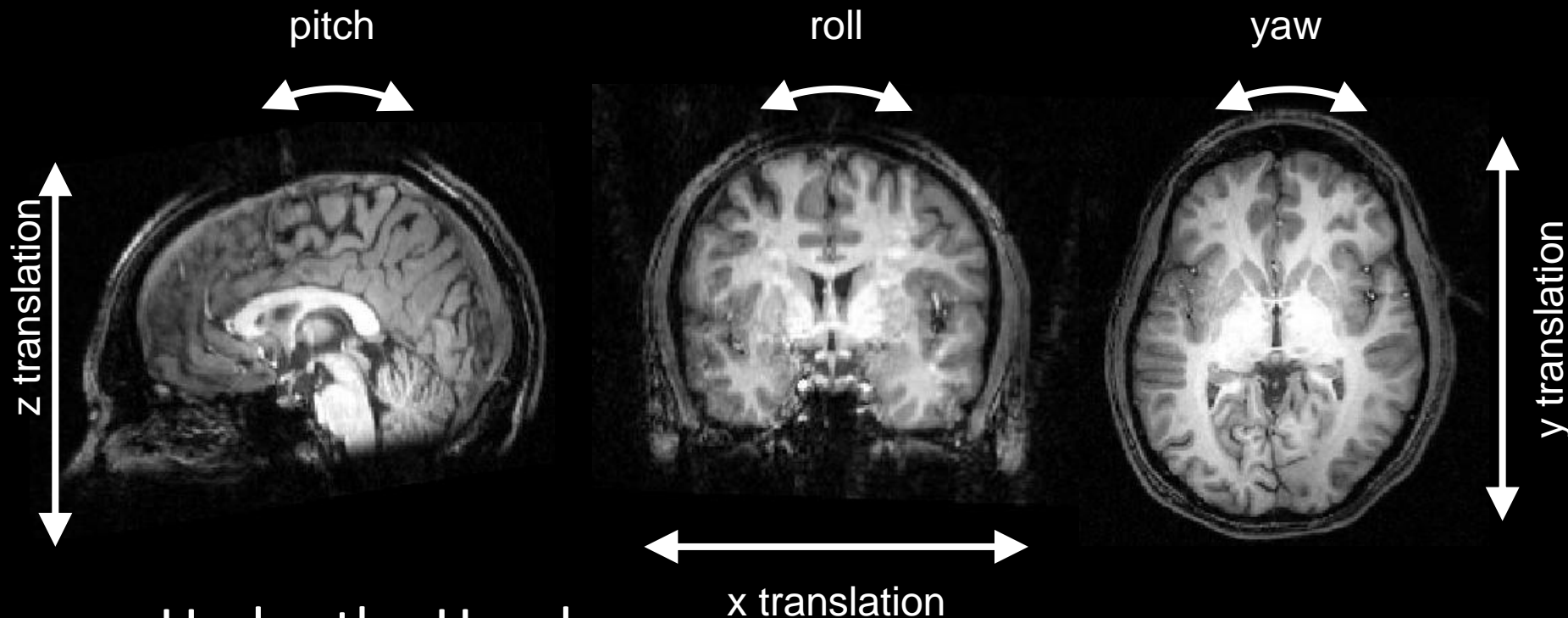


You should now see a slice timed corrected run – arun\_01.nii



# Realignment

Remember: This step is to align each volume of the brain to a target volume using six parameters: three translations and three rotations



## Under the Hood

- Builds shell script that calls FSL's *mcflirt*
- Default mcflirt options: “-cost normcorr -stats -plots”

# Realignment

realignfMRI -v arun -i 1-2 -M ./ <Subjects>  
-U youremail@umich.edu

## realignfMRI Options

-A	all runs present
-B	run in the foreground
-D	enable super debug flag
-d	enable debug flag
-f [directory]	functional directory e.g. connect/func
-i [#-#]	inclusive run list
-M [directory]	master subject directory
-m ["options"]	mcflirt options
-O	name to give realignRefVolume
-n [name]	name prepend
-S [#]	standard volume number for mcflirt
-t	test flag
-U [unique]	user email name/txt msg address
-v [name]	volume name wild card
-# [run number]	include this run number

# Realignment

Launching Script into Background = YAY!

Another email

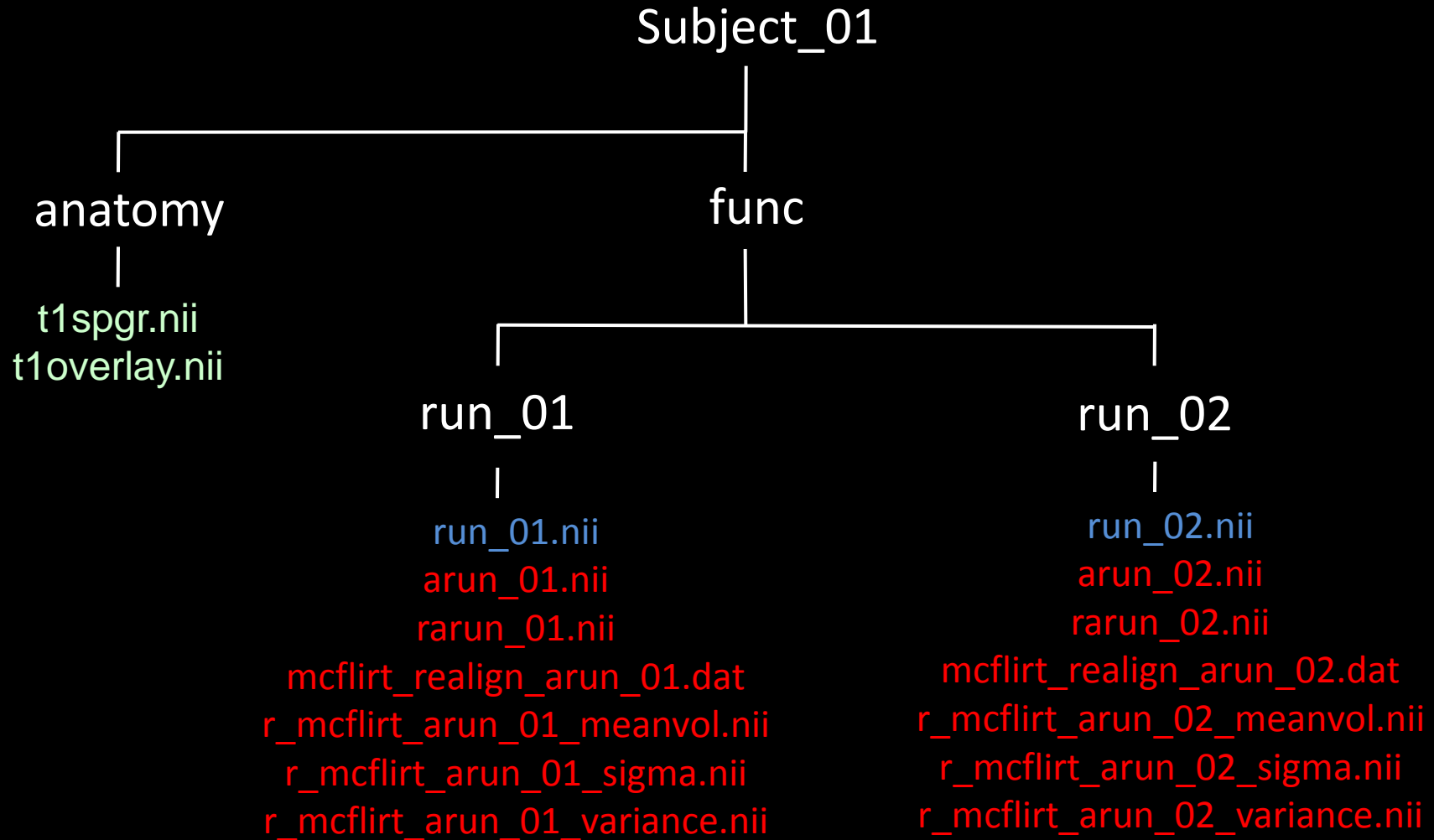
List the contents of the run directories again

```
[chelsea@venus Experiment_Chelsea]$ ls Subject_01/func/run_01/
arun_01.nii                rarun_01.nii                r_mcflirt_arun_01_sigma.nii    run_01.nii
mcflirt_realign_arun_01.dat r_mcflirt_arun_01_meanvol.nii r_mcflirt_arun_01_variance.nii
[chelsea@venus Experiment_Chelsea]$
```

## Output

- rarun\_##.nii
- mcflirt\_realign\_arun.dat - transformation parameters
- r\_mcflirt\_arun\_meanvol.nii - mean image after realign
- r\_mcflirt\_arun\_sigma.nii - std over time
- r\_mcflirt\_arun\_variance.nii - variance over time

# Pause – A look at what you have so far

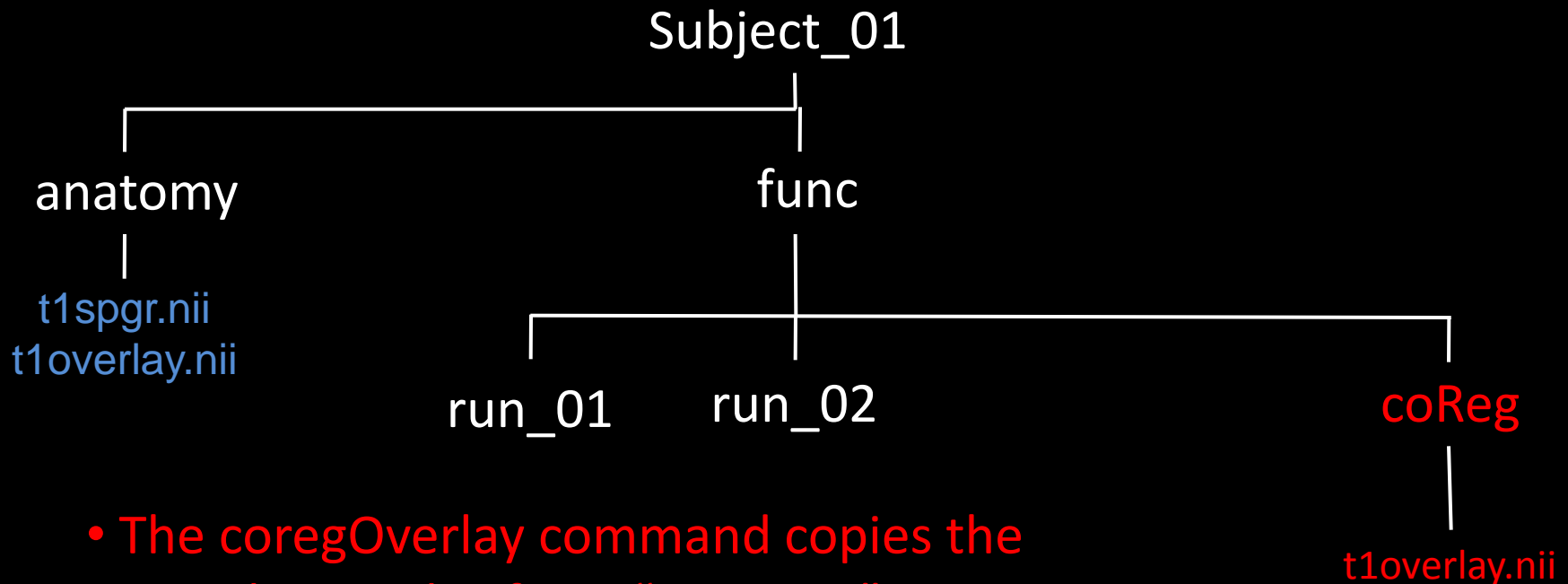


# coregOverlay

```
coregOverlay -o t1overlay -v rarun -M ./  
<Subjects> -U your_email@umich.edu
```

-a [directory]	set path for anatomy directory
-B	run in the foreground
-D	enable super debug flag
-d	enable debug flag
-f [directory]	functional directory e.g. connect/func
-M [directory]	master subject directory
-n [name]	name to add to output file name
-O	name of other files to process
-o [name]	name of overlay file
-r	set reslice flag
-s [directory]	set sub-directory under 'run_##'
-t	test flag
-U [unique]	user email name/txt msg address
-v [name]	name of functional image
-w [directory]	set output path

# coregOverlay



- The coregOverlay command copies the original t1overlay from “anatomy” into a new directory “coReg” located under “func”
- The copied t1overlay is the one that is coregistered

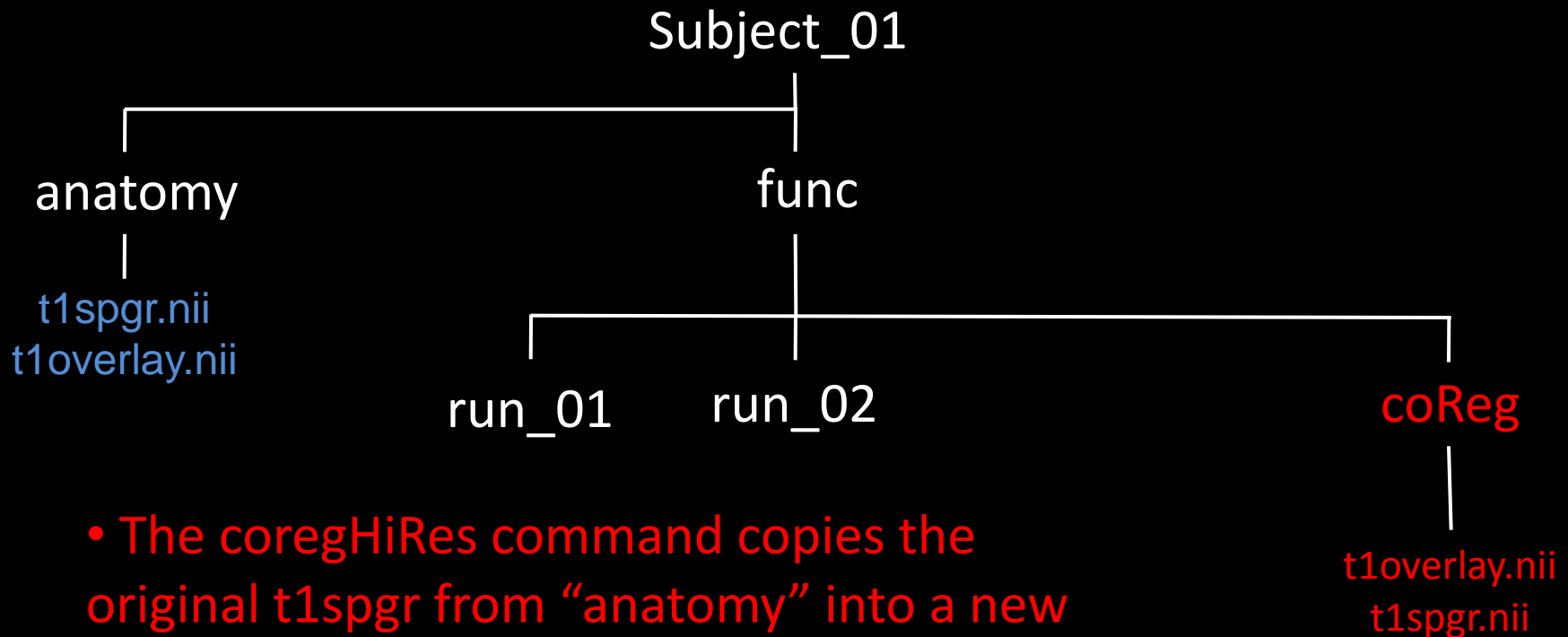
# coregHiRes

coregHiRes -h t1spgr -o t1overlay -M ./

<Subjects> -U your\_email@umich.edu

- a [directory] set path for anatomy directory
- B run in the foreground
- D enable super debug flag
- d enable debug flag
- f [directory] functional directory e.g. connect/func
- h [name] name of high resolution anatomical (t1spgr)
- M [directory] master subject directory
- n [name] name to add to output file name
- O name of other files to process
- o [name] name of overlay file
- r set reslice flag
- t test flag
- U [unique] user email name/txt msg address
- w [directory] set output path

# coregHiRes



- The coregHiRes command copies the original t1spgr from “anatomy” into a new directory “coReg” located under “func”
- The copied t1spgr is the one that is coregistered to the t1overlay in coReg (the one that is already in functional space)

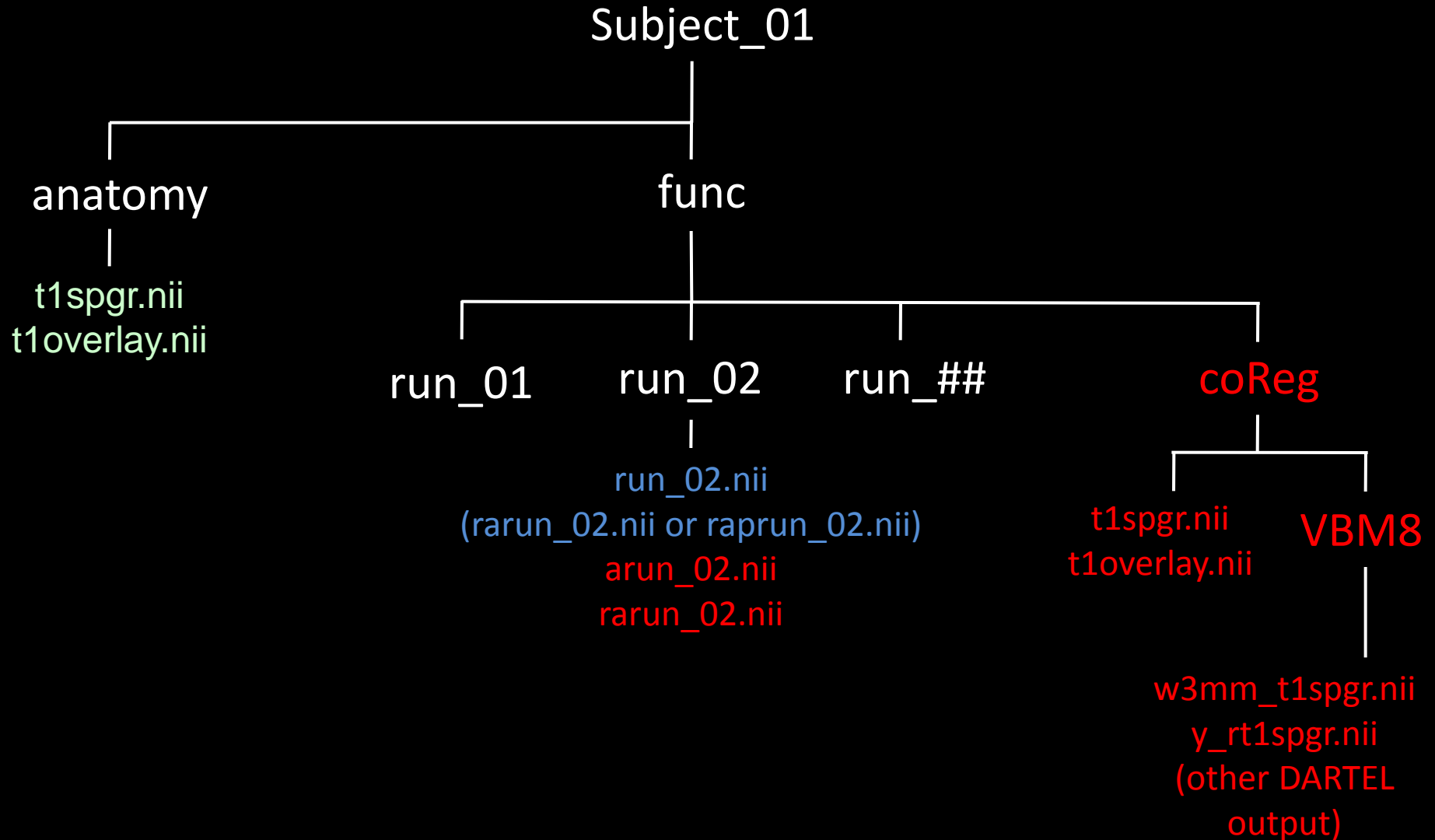


# DARTEL Warping – vbm8HiRes

vbm8HiRes -h t1spgr -a func/coReg  
-w func/coReg/VBM8 -I r3mm\_avg152T1\_BET -n w3mm\_  
-M ./ <Subjects> -U your\_email@umich.edu

-a [directory]	set path for anatomy directory
-B	run in the foreground
-D	enable super debug flag
-E	set the erosion iteration numbers
-d	enable debug flag
-H	calculate bias field corrected image
-h [name]	name of high resolution anatomical (t1spgr)
-I [Ref name]	set the reference image to use for VBM8
-L	erosion flag
-M [directory]	master subject directory
-m	fsl options
-n [name]	name to add to output file name
-O	name of other files to process
-t	test flag
-U [unique]	user email name/txt msg address
-w [directory]	set output path
-z	set the voxel size for resampling

# Directory Structure – After DARTTEL



# DARTEL Warping – vbm8HiRes

List the contents of func/coReg to see new VBM8 directory.  
List the contents of the VBM8 directory to see DARTEL output

```
[chelsea@venus Experiment_Chelsea]$ ls Subject_01/func/coReg/
tloverlay.nii  t1spgr.nii  VBM8
[chelsea@venus Experiment_Chelsea]$ ls Subject_01/func/coReg/VBM8/
bet_mt1spgr.nii  p0t1spgr.nii  t1spgr.nii  w1nrm_p1t1spgr.nii  wrp0t1spgr.nii
bet_t1spgr.nii  p1t1spgr.nii  t1spgr_seg8.mat  w1nrm_p2t1spgr.nii  wrp1t1spgr.nii
m0wrp1t1spgr.nii  p2t1spgr.nii  w1nrm_bet_t1spgr.nii  w1nrm_p3t1spgr.nii  wrp2t1spgr.nii
m0wrp2t1spgr.nii  p3t1spgr.nii  w1nrm_mt1spgr.nii  w1nrm_t1spgr.nii  wrp3t1spgr.nii
mt1spgr.nii  pt1spgr_seg8.txt  w1nrm_p0t1spgr.nii  w1nrm_t1spgr.nii  y_rt1spgr.nii
[chelsea@venus Experiment_Chelsea]$
```

# Quality Checks

A quality check too with FSL:

vbm8Check – 3 steps

0. check skull stripping
1. check registration between spgr & template
2. (later)

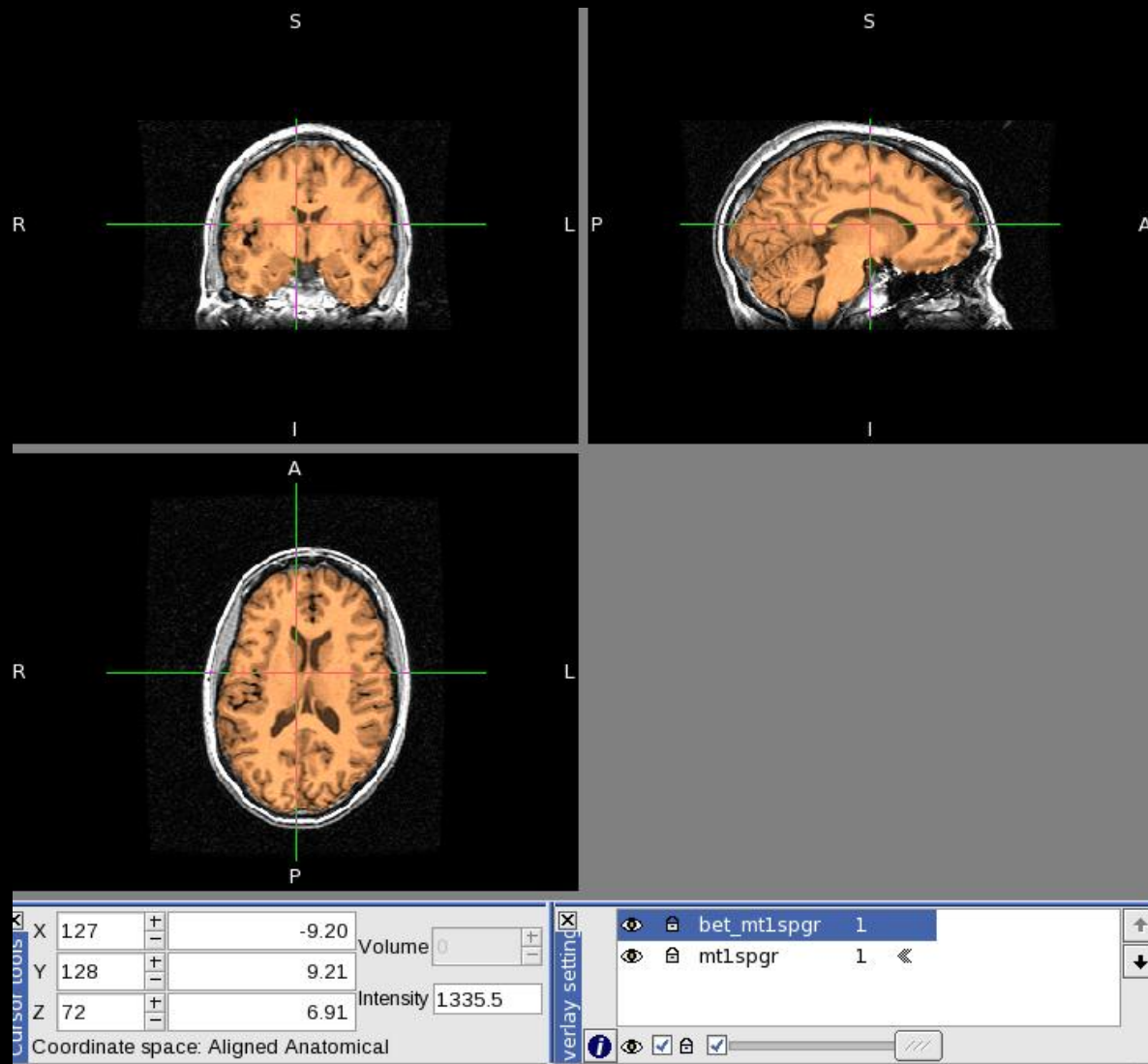
# Quality Checks

0. skull stripping of spgr

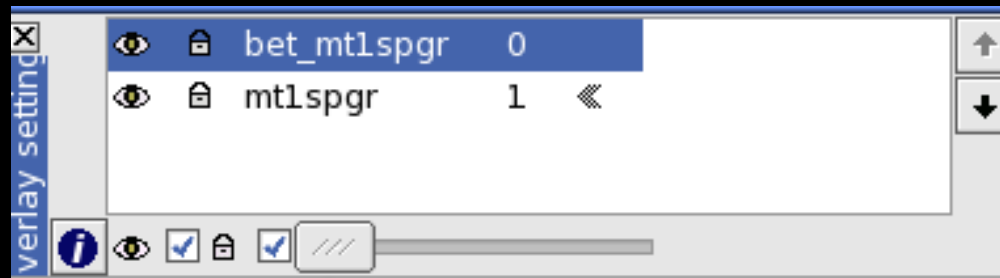
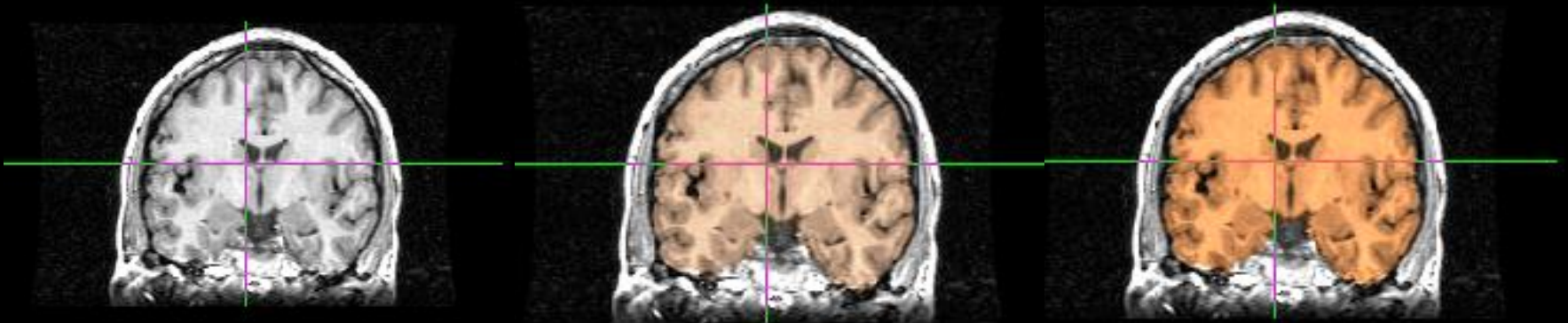
TYPE:

```
vbm8Check -a func/coReg/VBM8 -h t1spgr  
-M ./ <Subjects>
```

# Quality Checks



# Quality Checks



Scroll this button back and forth to fade between spgr and skull-stripped spgr. Click around the brain and look at different views. Close FSL

# Quality Checks

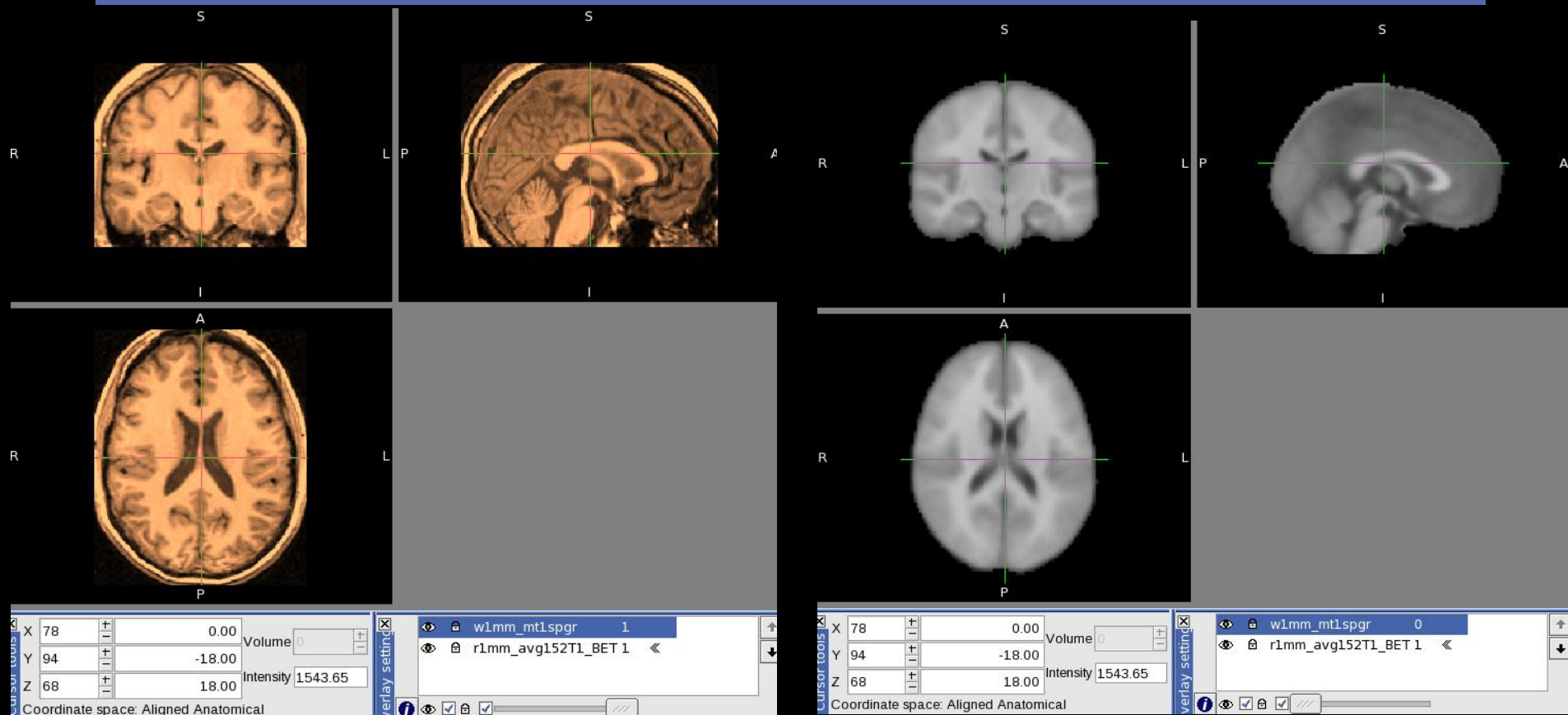
## 1. Registration between spgr & template

TYPE:

```
vbm8Check -a func/coReg/VBM8  
-h w3mm_mt1spgr -1  
-l r3mm_avg152T1_BET  
-M ./ <Subjects>
```



# Quality Checks



Use the little button again to fade back and forth between the spgr and the template checking that they match up. Close FSL

# Apply Warp – warpfMRI

```
warpfMRI -h t1spgr -w coReg/VBM8  
-l r3mm_avg152T1_BET -n w3mm_ -v rarun -W -M ./  
<Subjects> -U your_email@umich.edu
```

-B	run in the foreground
-f [directory]	set path to functional directory
-h [name]	name of high resolution anatomical (t1spgr)
-l [Ref name]	set the reference image to use for VBM8
-M [directory]	master subject directory
-n [name]	name to add to output file name
-t	test flag
-U [unique]	user email name/txt msg address
-v [name]	name of functional volume
-W	Enable VBM8 (DARTEL) warping for fMRI
-w [directory]	set output path
-z	set the voxel size for resampling

# Quality Checks

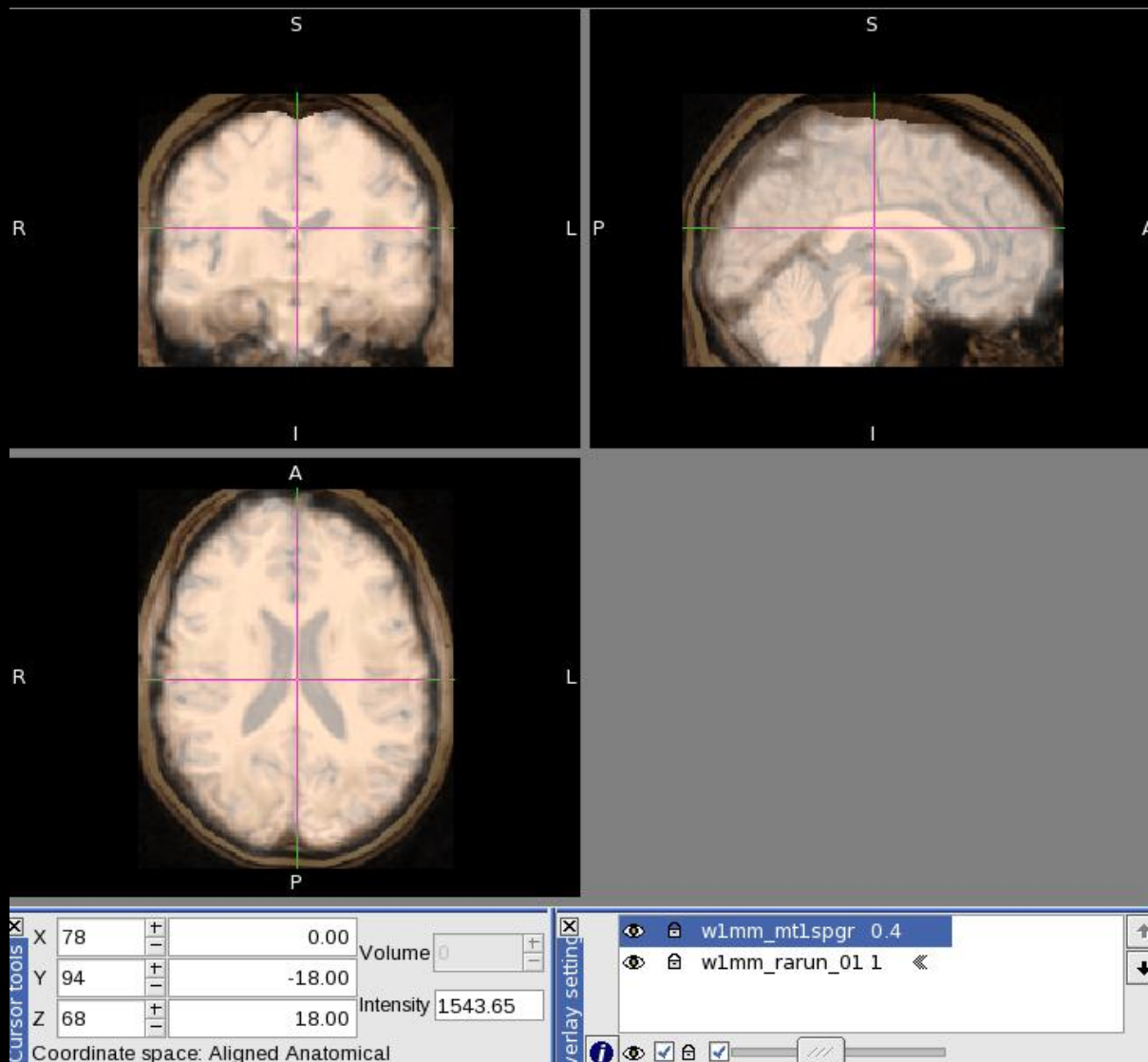
vbm8Check

2. Registration between spgr & functionals

TYPE:

```
vbm8Check -a func/coReg/VBM8  
-h w3mm_mt1spgr -2 -v w3mm_rarun  
-M ./ <Subjects>
```

# Quality Checks

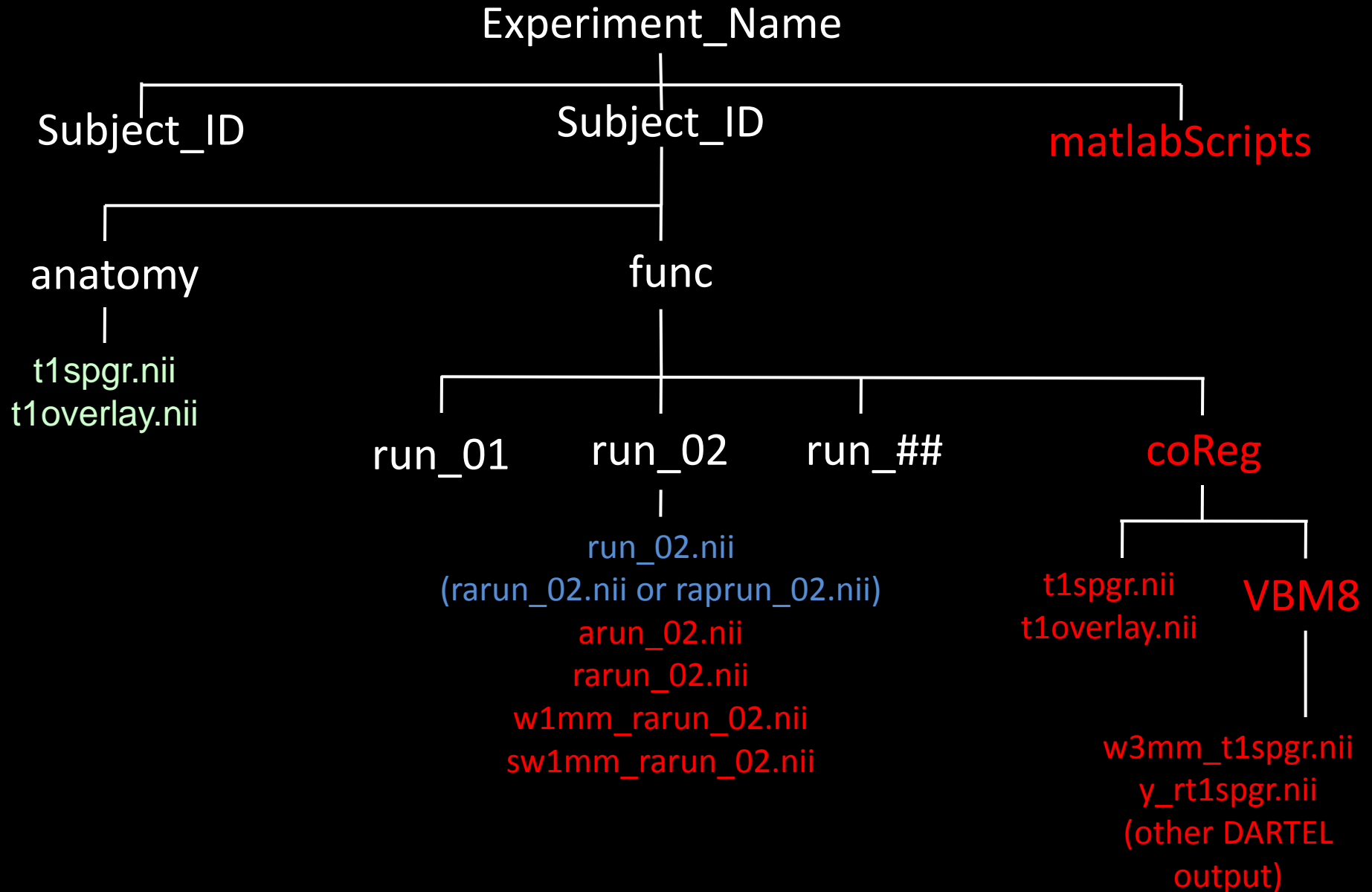


# Smoothing - smoothfMRI

```
smoothfMRI 5 5 5 -v w3mm_ -M ./ < <Subjects> -U  
your_email@umich.edu
```

-B	run in the foreground
-D	enable super debug flag
-d	enable debug flag
-e	set file extension for smoothfMRI
-f [directory]	set path to functional directory
-M [directory]	master subject directory
-n [name]	name to add to output file name
-t	test flag
-U [unique]	user email name/txt msg address
-v [name]	name of functional volume

# Directory Structure – After Processing



# Logging

When a process is run, the command will automatically configure a “job” by writing a shellscript file and as needed a matlab script file. These files will be created in a directory specified by the name for the command being issued and then further segregated by the year and month. The “job” files have names that are based on the command issued, the date,/time, the user and the computer node. Once the command completes building the process it will launch into the background for execution, releasing the interactive terminal.

Example:

matlabScripts

spm8Batch

coregOverlay

2012\_07

coregOverlay\_120722\_09\_37\_18\_heffjos.sh  
coregOverlay\_120722\_09\_37\_18\_heffjos.m  
coregOverlay\_120722\_09\_37\_18\_heffjos.log