

spm8Batch

A unix shell command interface to SPM8 to facilitate
data pre-processing for
spatial normalization to MNI space

Preprocessing with spm8Batch

There are philosophies behind the design of spm8Batch:

- minimize user script writing
- automatic and extensive logging/tracking of all processing
- interaction with SPM/MATLAB from simple shell command line
- allowing for quality control (check reg) between steps
- ensuring analysis pieces (data) exist prior to execution
- allowing for large scale data throughput
- uniformity of pre-processing (with allowing for flexibility)
- notification when process is complete
- truly running in background to be less sensitive to glitches
- “job” files are essentially archives of processing
- commands all have “built-in” help

Preprocessing with spm8Batch

Assuming data are from an imaging center that provides for *slicetiming* and *realignment* corrections the remaining steps to get functionals into MNI space prior to statistical analysis are:

- coregistration of overlay to functional
- coregistration of high-resolution to overlay
- MNI normalization of high-resolution image (VBM8 or SPM8)
- MNI normalization of functional data
- smoothing of functional data

Preprocessing with spm8Batch

Preprocessing Workflow



Preprocessing with spm8Batch

To facilitate data integrity, operations such as coregistration and MNI normalization of high-resolution image takes place in new coReg directory.

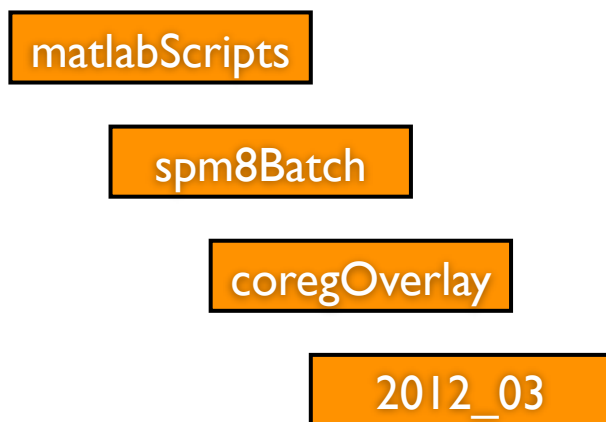
MNI normalization and smoothing of functional data take place in the run directories.

Preprocessing with spm8Batch

When a process is run, the command will automatically configure a “job” by writing a shell-script file and as needed a matlab script file. These files will be created in a directory specified by the name of 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.

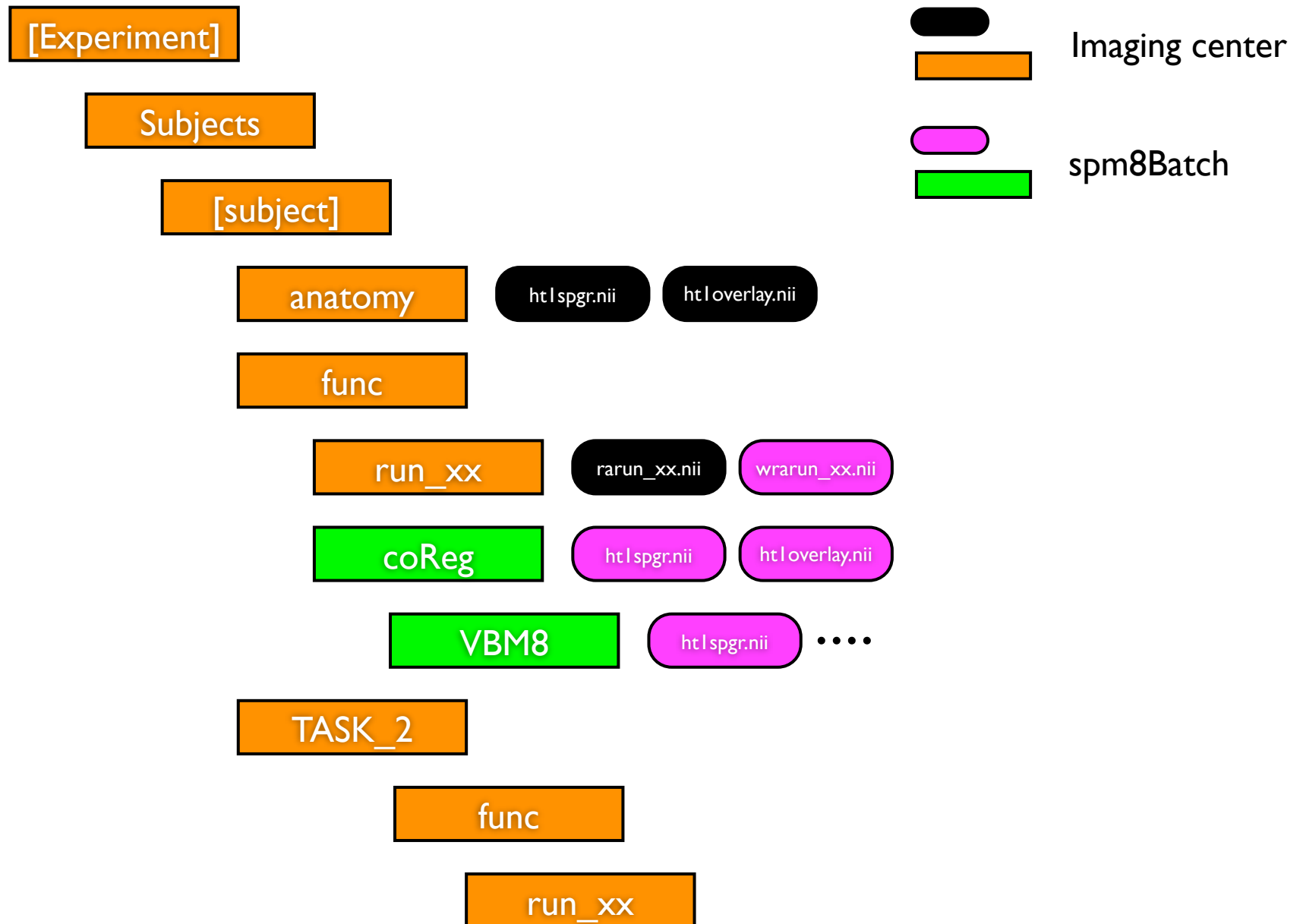
A log file is also produced.

Example:



coregOverlay_120313_09_37_18_rcwelsh_gaviero.sh
coregOverlay_120313_09_37_18_rcwelsh_gaviero.m
coregOverlay_120313_09_37_18_rcwelsh_gaviero.log

Expected Data Structure



Preprocessing with spm8Batch

Example Preprocessing Workflow



coregOverlay

```
coregOverlay -o htlooverlay -v rar -w coREG_FINAL -M ./ 080430rf
```

coregHiRes

```
coregHiRes -o htlooverlay -h ht1spgr -w coREG_FINAL -M ./ 080430rf
```

vbm8HiRes

```
vbm8HiRes -h ht1spgr -a func/coreg_FINAL \  
-w func/coreg_FINAL/VBM8 \  
-I r3mm_avg152T1 -n w3mm_ -M ./ 080430rf
```

warpfMRI

```
warpfMRI -h ht1spgr -w coREG_FINAL/VBM8 -I r3mm_avg152T1 \  
-n w3mm_ -M ./ 080430rf -v rar -W
```

smoothfMRI

```
smoothfMRI -v w3mm_rar -M ./ 080430rf
```