MethodsCore
MATLAB Preprocessing Script Documentation
02/04/2013

Table of Contents

Introduction

The Preprocess script lets allows easy batch preprocessing of subjects. The script is flexible in allowing you to specify exactly what preprocessing steps are done, and especially allowing different normalization methods to be used. Batching in this manner can save a lot of time during preprocessing. There are five major sections to fill out:

1. Path and subject information
2. General Preprocessing options
3. Slicetime correction information
4. Coregistration and Normalization information
5. Smoothing information

Additionally, there is an Advanced Section to the script, which will not be covered in this documentation. There is a brief description of the advanced features in the script itself. If you want to use any of these features, we recommend you contact MethodsCoreHelp@umich.edu for information. Filling in these sections in the script is mostly self-explanatory, and there is extensive help in the script itself. Below we'll walk you through the steps that require more explanation.

Visual Tutorials

There are a series of video tutorials available online. It may be useful to view these while following along in this document.

1 Introduction	http://youtu.be/YNl3H8sXgwI?hd=1
2 Subjects & Runs	http://youtu.be/TfdgtRzezYY?hd=1
3 Slicetime Correction and Realignment	http://youtu.be/X-jkITWrado?hd=1
4 Coregistration, Normalization, and Smoothing	http://youtu.be/PcW_IvqY3jM?hd=1
5 Conclusion	http://youtu.be/BD0mn273wXY?hd=1

Getting Started

Copy the Preprocess_mc_template script from your Methods Core directory into your local experiment directory. This is the file you'll be editing.

Setting up the Preprocess_mc_template Script

You specify all of your preprocessing information in the Preprocess_mc_template script. Most of this should be self-explanatory and there is help in the script itself.

One important issue is setting up paths. You set up the paths to a lot of things (path to functional images, output path, etc…). Setting up paths is done through a standardized 'Path Template' system. See the **Path Template Documentation** in the root directory of the Methods Core repository for help on this. We'll be using this method for setting up paths in pretty much all Methods Core software.

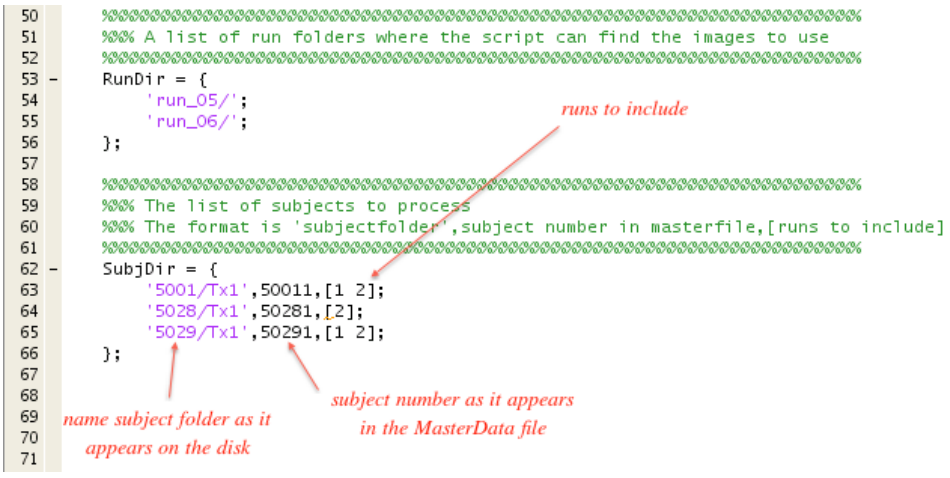
The first few variables set up your experiment and logging directories.

Exp	- The main folder containing your experiment.
LogTemplate	- A path to save logfiles from the preprocessing jobs.

Following these variables are a number of variables related to subjects, runs, and images.

ImageTemplate	- The path where your original data to be preprocessed resides.
basefile	- The prefix of each functional file (i.e. run for typical UofM scanner data).
imagetype	- Either 'nii' or 'img'. 4D NIFTI data is currently preferred, but preprocessing should currently function equally well with either.
RunDir	- A cell array of run folder names for this task.
SubjDir	- A cell array of subject directories, subject numbers, and runs to include for this task (described below).

Setting up Subjects and Runs



RunDir lists the run names as they appear on the disk. SubjDir lists the subjects to include in the analysis. It has three columns that are explained above. The second column is not used for preprocessing as it is only relevant for FirstLevel analysis – it is included here so that lists of subjects can easily be copy/pasted between Preprocess_mc_template and FirstLevel_mc_template. Notice that 'runs to include' (i.e., column 3 of SubjDir) indexes the rows of RunDir. So in the above example, we are omitting run_05 for subject 5028 Tx1.

General Preprocessing Options

This section includes variables that control what preprocessing steps will be done, as well as what prefixes to apply to the preprocessed data at each step.

SliceTimePrefix	- prefix for slicetime corrected data (typically 'a')
RealignPrefix	- prefix for realigned data (typically 'r')
CoregOverlayPrefix	- prefix for coregistered overlay image (typically none, though see note below in coregistration section)
CoregHiResPrefix	- prefix for coregistered high resolution image (same as above)
NormalizePrefix	- prefix for normalized data (typically 'w' or 'wN_' where N is the voxel size used to reslice the normalized data
SmoothPrefix	- prefix for smoothed data (typically 's' or 'sN_' where N is the filter width of the smoothing kernel)
StepsToDo	- A vector of 0's and 1's indicating which steps of preprocessing to perform in this order: [Slicetime Realign CoregOverlay CoregHiRes Normalize Smooth] **NOTE: Currently, CoregOverlay and CoregHiRes will not be performed during 'func' normalization even if they're set to 1 here. This may be changed in a future update.**
AlreadyDone	- A vector of 0's and 1's indicating that some steps of preprocessing have already been performed on the data. This is included so that future reprocessing, or different settings of processing can occur from a common base. The order is the same as the StepsToDo variable: [Slicetime Realign CoregOverlay CoregHiRes Normalize Smooth].

Slicetime Correction

The variables for slicetime correction include: the TR your data was collected at, the number of slices collected, the order of the slices, and the reference slice to correct data to. These variables are presented below:

TR - The TR of your data as collected at the scanner

NumSlices - The number of slices that were collected at each timepoint

SliceOrder - The order the slices were collected in. This can be directly entered as a vector of numbers, or you can use a variety of MATLAB expressions to have it create most common order acquisitions. Several examples of this might be:

Sequential Ascending	- [1:1:NumSlices]
Sequential Descending	- [NumSlices:-1:1]
Interleaved Ascending (starting on 1)	- [1:2:NumSlices 2:2:NumSlices]
Interleaved Ascending (starting on 2)	- [2:2:NumSlices 1:2:NumSlices]

RefSlice - The reference slice to correct your data to. If left blank, it will default to the (approximate) middle slice (i.e. floor(NumSlices/2)).

Realignment

Realignment currently has only one option included in this script. Realignment will proceed based on the default SPM procedure – if necessary this behavior can be modified by changing SPM defaults in the Advanced section of the template script. Initially the first image of each run is realigned to the reference image of the first run. Then each image in a run is realigned to the (now-realigned) first image of that run. A mean image is written out and will be used as the subsequent target for Coregistration and/or Normalization (if using simple functional normalization).

RefImage - The reference image (in the first run) to realign your data to. If left blank ([]) it will default to the first image.

Coregistration and Normalization

There are several variables that define the coregistration and normalization steps. All paths will use the same Path Template variable substitution described above. The options are:

OverlayTemplate - The path and name of a low resolution structural image, typically collected in the same orientation as the functional images.

HiResTemplate - The path and name of a high resolution structural image.

AnatTemplate - The path to place copies of your original anatomical images. This is done so that the original images are not changed by the coregistration step.
NOTE: If you specify the same folder location as your original images, make sure you include an appropriate prefix in the general section above, or your original data will be modified.

NormMethod - The method of normalization to use. Possible values include:
'func' - Normalize the functional images to a functional template
'anat' - Coregister anatomical and functional images, then normalize the anatomical image to an anatomical template and apply the same warping to the functional images.
'seg' - Coregister anatomical and functional images, then segment the anatomical image with VBM and use the GM/WM segments to normalize the anatomical data via DARTEL warping. Then apply this deformation to the functional images.

WarpTemplate - If performing 'func' or 'anat' normalization, you must provide a WarpTemplate. This is the image that will be used as a target for the normalization (typically a template image in MNI space). Custom WarpTemplates for segmentation-based normalization are not currently supported.

VoxelSize - The voxel size to use when reslicing your functional data after normalization.

Smoothing

The only option necessary for smoothing is to specify the filter width of your smoothing kernel.

SmoothingKernel - The filter width (in mm) of your Gaussian smoothing kernel. This value can be provided as a single number – in which case it will be used as an isotropic kernel with the same width in each dimension – or it can be entered as a vector of 3 values – only necessary if you want to specify a non-isotropic smoothing kernel.

Advanced Features

There is also an Advanced Section, with other features. If you want to use any of these features, we recommend you read the advanced section of the template script directly, and then contact MethodsCoreHelp@umich.edu for additional help.