Center Scripts (center_scripts_v1.0) Walk-Through
July 31, 2007

Introduction

Center scripts are used to do SPM batch analysis. This version of center scripts is compatible with SPM5 version. The following things must be done before running the analysis:

- Add center_scripts_v1.0 directory and its sub-directories on MATLAB path.
- Data must be organized such that each subject has run directories.
- All the required parameters for the analysis must be entered in a preference file. A detailed explanation of the preference file is given in the next section. Example preference files like 'cs_prefs_aod_analyze.m', 'cs_prefs_aod_nifti.m' and 'cs_prefs_structural.m' are located in the directory center_scripts_v1.0/prefs.
- Create your own 'spm_defaults.m' file. Example 'spm_defaults.m' file is located in the directory center_scripts_v1.0.

Note:

1. Preference and spm_defaults files must be added to the MATLAB path.
2. Use 'spm_defaults.m' file in directory center_scripts_v1.0 instead of the standard 'spm_defaults.m' file as it contains additional fields that are required for proper running of the center scripts.

You can run the analysis using two functions like 'cs_run_all' and 'cs_run_sub'. The syntax for each function is given below:

- 'cs_run_all' – This function is used for analyzing specified subject directories. The syntax for running this file is as follows:
    - prefs_file='cs_prefs_aod_nifti';subject_directory='J:\AOD_Data\2subjects-unprocessed\s01\';
    - cs_run_all(prefs_file, subject_directory);

- 'cs_run_sub' – This function is used to automatically get the subject directories based on field scandir_regexp in structure csprefs (Global variable in preference file). It can also be used if you have several tasks to be run at once. Place each task in a preference file and call at the MATLAB command prompt as follows:
    - prefs_files = {'cs_prefs_aod_nifti', 'cs_prefs_ structural'};
    - cs_run_sub(prefs_files);

Preferences file

Preferences file contains the parameters required to do pre-processing or stats. To run a particular step use a value of 1 and a value of 0 means don't run that step. The following are the steps involved:

Processing Steps

The following are the main steps involved:

a. Dicom Converter (csprefs.run_dicom_convert)
b. Realign (csprefs.run_realign)
c. Co-register (csprefs.run_coregister)
d. Slice Time correction (csprefs.run_slicetime)
e. Normalize (csprefs.run_normalize)
f. Smooth (csprefs.run_smooth)
g. Filter (csprefs.run_filter)
h. Stats (csprefs.run_stats)
i. Segment (csprefs.run_segment)

Data directory

Data directory and log file information is described below:

a. csprefs.exp_dir – Experimental directory or the root directory where all the data is located.
b. csprefs.logfile – Log file containing information whether center scripts ran successfully or not.
c. csprefs.errorlog – Error information is stored in this log file.
d. csprefs.spm_defaults_dir – Enter full file path of the copy of 'spm_defaults.m' file.
e. csprefs.scandir_regexp – Regular expression for subject directories. This field will be used by the 'cs_run_sub' function to get the subject directories. Some of the standard regular expressions are described in the Regular Expressions section.
f. csprefs.rundir_regexp - Regular expression for run directories.
g. csprefs.scandir_postpend - Regular expression used to add an additional path to a subject directory. For example if 'C:\sub01\' is the subject directory and you have run directories inside 'Study001' directory, regular expression will be 'Study\d{3}'. Leave csprefs.scandir_postpend as empty if you have run directories immediately inside a subject directory.
h. csprefs.rundir_postpend - Regular expression used to add an additional path to a run directory. For example if 'C\sub01\Study001\aod_v1_001' is the run directory and you have images inside 'Original\Nifti' directory, regular expression will be 'Original\\Nifti' ('Original\/Nifti under Unix). Leave csprefs.rundir_postpend as empty if you have images immediately inside a run directory.
i. csprefs.dummyscans – No. of dummy scans. All the dummy scans will be moved to directory 'dummies'. Leave this as 0 if you want to use all the scans. In case of

a 4D Nifti file, all the dummy scans will be written with the same file name in ‘dummies’ directory.

j. csprefs.tr – TR for your experiment.

Dicom Converter

Variables contained in dicom converter as follows:

a. csprefs.dicom.file_pattern – File pattern of dicom files.
b. csprefs.dicom.format - Options for converting dicom files.
    a. ‘3d_analyze’ – Dicom files will be converted to 3D analyze images.
    b. ‘3d_nifti’ – Dicom files will be converted to 3D Nifti images.
    c. ‘4d_nifti’ – Dicom files will be converted to a 4D Nifti image.
c. csprefs.dicom.write_file_prefix – Prefix for the image files that are converted from dicom files.
d. csprefs.dicom.outputDir – Output directory to write analyze or Nifti images.

Realign

Variables contained in realign are as follows:

a. csprefs.coregister – Coregister images to the subject’s first scan first session. If you have already performed this step, you need to set this field as 0.
b. csprefs.reslice – Reslice images using ‘spm_reslice’ function.
c. csprefs.use_inrialign – Uses INRIAlign function to realign. A value of 0 means ‘spm_realign’ function is used.
d. csprefs.realign_pattern – File pattern of the images that need to be realigned.
e. INRIAlign defaults:
    a. csprefs.inrialign_rho – Rho function for INRIAlign function. Default value is ‘geman’ (Geman-McClure function). Other options are as follows:
        i. 'absolute' - Quite slow and not very robust
        ii. 'huber' - Huber function
        iii. 'cauchy' - Cauchy function
        iv. 'leclerc' - Leclerc-Welsch function
        v. 'tukey' - Tukey's biweight function
    b. csprefs.inrialign_cutoff – Cut off distance. Default is 2.5.
    c. csprefs.inrialign_quality – Quality. Default is 1 (slow and high quality).
f. csprefs.realign_fwhm – Size of smoothing kernel. Default is 8.
g. csprefs.realign_rtm – Realign to the mean image. Default is 0. A value of 1 does not work for INRIAlign.
h. csprefs.reslice_write_imgs – Option for writing re-sliced images. Default is 0.
i. csprefs.reslice_write_mean – Option for writing mean image. Default is 1.

Co-register

Variables contained in co-register step are as follows:

a. csprefs.run_coreg – Run co-register step. Options are 1 and 0. A value of 1 will estimate the registration parameters using the reference image (csprefs.coreg.ref) and the source image (csprefs.coreg.source).
b. csprefs.run_reslice – Run re-slice step. Options are 1 and 0. 1 means images will be re-sliced using the reference image (csprefs.write.ref). The images that will be re-sliced are source image (csprefs.coreg.source) and the other images (images obtained using file pattern from variable csprefs.coreg.other_pattern). The new set of images will have prefix 'r'.
c. csprefs.coreg.ref – Reference or template image used for registration.
d. csprefs.coreg.source – Source image or file pattern.
e. csprefs.coreg.other_pattern – You can specify additional images to apply the registration parameters.
f. csprefs.coreg.write.ref – Reference image used for re-slicing source and other images.

Note: After the co-register step, the headers of the images will be modified to incorporate the registration parameters.

Slice Time Correction

Variables contained in slice time are as follows:

a. csprefs.slicetime_pattern – File pattern of the images that need to be slice time corrected.
b. csprefs.sliceorder – Slice time order. Include all the slices.
c. csprefs.refslice – Reference slice. Default is middle slice.
d. csprefs.ta – Time of acquisition. Leave for now this variable on 'default'.

Normalize

Variables contained in normalize are as follows:

a. csprefs.determine_params – Determine normalization parameters. If you have already determined normalization parameters, set this value to 0 and specify the MAT file in field csprefs.writenorm_matname.
b. csprefs.write_normalized – Option for writing normalized images. Set this value to 1 for now.
c. csprefs.params_template – Template image used for parameter estimation. Default is 'EPI.nii'.
d. csprefs.params_pattern – Name of image to be used for parameter estimation. Usually this is the mean image created during realignment.
e. csprefs.writenorm_pattern – File pattern of the images that require the parameters to be applied.
f. csprefs.writenorm_matname - File or filter pattern containing the normalized parameters. To use this file you need to set field csprefs.determine_params to 0.

Smooth

Variables contained in smoothing are as follows:

a. csprefs.smooth_kernel – Smoothness kernel to be applied. Default is [10,10,10].
b. csprefs.smooth_pattern – File pattern of the images that need to be smoothed.

Filter

Variables contained in filtering are as follows:

a. csprefs.filter_pattern – File pattern of the images that need to be filtered.
b. csprefs.cutoff_freq – Normalized cut-off frequency. Default is 0.25.

Stats

Variables in stats are as follows:

a. csprefs.stats_make_asciis – For now leave this as 0.
b. csprefs.stats_ascii_script – Full file path of the ascii script if variable csprefs.stats_make_asciis is set to 1.
c. csprefs.stats_beh_dir_name – Behavioral directory name. This will be used only when make ascii script is used.
d. csprefs.stats_files_relative_path_sub – Option is provided to find the onset and duration files relative to a subject directory or corresponding run directory. A value of 1 means onset files will be searched relative to a subject directory and a value of 0 means files will be searched relative to corresponding run directory.
e. csprefs.stats_dir_name – Center scripts will create a directory using this name for storing the stats results.
f. csprefs.stats_pattern – Stats will be done using this image file pattern.
g. sprefs.stats_beh_units – Behavioral units. Options are 'scans' and 'secs'.
h. csprefs.stats_volterra – Model interactions. For now leave it as 0.
i. csprefs.stats_basis_func – Basis function to use. Options are 1 and 2.
    a. 1 – 'hrf'
    b. 2 – 'hrf (with time derivative)'
j. csprefs.stats_onset_files – Specify onset files location in a cell array like {'beh/TRG_PR_1_noslice.asc','beh/NOV_OM_1_noslice.asc','beh/STD_OM_1_noslice.asc';'beh/TRG_PR_2_noslice.asc','beh/NOV_OM_2_noslice.asc','beh/STD_OM_2_noslice.asc'} where rows indicate onset files for that session. You can enter full file path if the onsets information doesn't vary between the subjects.
k. csprefs.stats_duration_files – Number of files must match the number of onset files. For short events enter 0.
l. csprefs.stats_time_modulation – To run time modulation you need to specify a cell array of size number of sessions by number of conditions. If you are not running time modulation, set variable csprefs.stats_time_modulation as 0 or empty cell array. In the example below we are using first order time modulation for two sessions and three conditions:

    csprefs.stats_time_modulation = {1, 1, 1; 1, 1, 1};

m. csprefs.stats_parametric_modulation - Parametric modulation can be specified for each condition each session. If you are not running parametric modulation, set variable csprefs.stats_ parametric_modulation as 0 or empty cell array. In the example below we are using a first order polynomial for two sessions and three conditions:

   csprefs.stats_parametric_modulation = {{'Targets', [1:23], 1}, {'Novels', [1:23], 1}, {'Standards', [1:184], 1}; {'Targets', [1:24], 1}, {'Novels', [1:23], 1}, {'Standards', [1:185], 1}};

n. csprefs.stats_global_fx – Remove global effects. Options are 1 and 0. For now leave it as 0.
o. csprefs.stats_highpass_cutoff - Number of seconds for high-pass filter. Default is 128. Put Inf (without quotes) for no filtering.
p. csprefs.stats_serial_corr – Correct for serial corrections. Options are 1 and 0. For now leave it as 0.
q. csprefs.stats_tcontrasts – Contrasts are specified in a matrix whose dimensions are number of contrasts by number of regressors. Number of regressors is equal to (number of sessions * number of basis functions * number of conditions * (1 + order of time modulation + polynomial order of parametric modulation)) + number of sessions.
r. csprefs.stats_tcontrast_names – Contrast names must be entered in a cell array whose length must equal the number of rows in variable csprefs.stats_tcontrasts.

Segment

Variables in segmentation step are as follows:

a. csprefs.segment.pattern – File pattern of the images that need to be segmented.
b. The options for writing the output grey matter, white matter and CSF images are as follows:
    a. [0 0 0] - None
    b. [0 0 1] - Native Space
    c. [0 1 0] - Unmodulated Normalised
    d. [1 0 0] - Modulated Normalised
    e. [0 1 1] - Native + Unmodulated Normalised
    f. [1 0 1] - Native + Modulated Normalised
    g. [1 1 1] - Native + Modulated + Unmodulated
    h. [1 1 0] - Modulated + Unmodulated Normalised
c. csprefs.segment.output.GM – Option for writing the grey matter. Default value is [1, 1, 1].
d. csprefs.segment.output.WM – Option for writing the white matter. Default value is [0, 0, 1].
e. csprefs.segment.output.CSF - Option for writing the CSF. Default is [0, 0, 0].

f. csprefs.segment.output.biascor – Bias correction. A value of 1 means save bias corrected.
g. csprefs.segment.output.cleanup - Clean up any partitions. Options are 0, 1 and 2. Each option is explained below.
    a. 0 - Dont do cleanup
    b. 1 - Light Clean
    c. 2 - Thorough Clean

Regular Expressions

Regular expressions are used to do pattern matching. Some of the standard regular expressions are as follows:

a. '\<\w+\>' – String containing alphabets, numerals or underscore characters like 'ab_12_dd', 'sub01_vis', 'sub1', etc.

b. '\<\d\>' – String containing only numerals like '1', '2', etc.

c. '\<\d{8}_\d{6}_\d{8}\>' – String containing 8 decimals followed by underscore character followed by 6 decimals followed by underscore character followed by 8 decimals like '20040331_111557_04030569', '20040604_104632_04030708', etc.