



run_cpm.m

Written by Suyeon Ju (suyeon.ju@yale.edu) on 1.10.22

Updated by Suyeon Ju on 4.1.22

Script locations

- MRRC server:

`/data23` → `mri_researchers` → `fredericks_data` → `shared` → `hcp_aging_analyses` →
`hcp-a_cpm` → `run_cpm.m`

- Box:

`Box` → `CPM_HCP-A` → `run_cpm.m`

How to run this script

Inputs:

- `subj_list` = list of all subject IDs (should be file name of .txt file that has list of all subject IDs [`'HCA#####'`])
 - i.e., `'all_subjs_ravlt_hcp-a_2.txt'`
- `behav_param` = behavioral parameter to be tested (should match with the file name of the .txt file from HCP-A dataset with that parameter's data)
 - i.e., `'ravlt'`
- `scan_type` = one of the following:
 - `'rfMRI_REST1_AP'`, `'rfMRI_REST1_PA'`, `'rfMRI_REST2_AP'`, `'rfMRI_REST2_PA'`,
`'tfMRI_CARIT'`, `'tfMRI_FACENAME'`, `'tfMRI_VISMOTOR'`

Outputs:

- For now, the code just prints the resulting correlation between actual and predicted y (R and p value), but I will revise the code later to collect all CPM outputs (`y_predict` and `performance`)

Example command line:

```
run_cpm('all_subjs_ravlt_hcp-a_2.txt', 'ravlt', 'tfMRI_CARIT')
```

Pseudocode (what the script does, in a nutshell)

step 1: create string array with all subj IDs from `subj_list`

step 2: create for loop where:

- iterative variable = all subj IDs
- `conn_mat_single` = connectivity matrix for each subj ID for specified scan_type; each `conn_mat_single` matrix is added in 3rd dimension to `conn_mat` (holds all conn mats across all subjs)
- `conn_mat` = compilation of all conn mats across all subjs
- `conn_subj_array` = collect subj IDs of all subjs in `conn_mat`

step 3: create matrix listing all subj IDs in `conn_subj_array` (in col 1) and corresponding behavioral parameter data pulled from `behav_param` (in col 2)

step 4: call `cpm_main` function from constable's CPM matlab code

Example output

```

>> run_cpm('all_subjs_psm_hcp-a_2.txt', 'psm', 'tfMRI_VISMOTOR')
Warning: The DATETIME data was created using format 'MM/dd/yyyy' but also matched 'dd/MM/yyyy'.
To avoid ambiguity, supply a datetime format using SETVARTOPTS, e.g.
    opts = setvartopts(opts,varname,'InputFormat','MM/dd/yyyy');
> In matlab.io.internal.text.TableParser/readData (line 82)
   In matlab.io.internal.functions.ReadTableWithImportOptionsText/executeImpl (line 76)
   In matlab.io.internal.functions.ReadTableWithImportOptions/executeImpl (line 18)
   In matlab.io.internal.functions.ReadTableWithImportOptionsText/execute (line 122)
   In matlab.io.internal.functions.ReadTableWithImportOptions/execute (line 25)
   In matlab.io.internal.functions.ExecutableFunction/validateAndExecute (line 98)
   In matlab.io.ImportOptions/readtable (line 490)
   In run\_cpm (line 71)

ans =

    523     2

Warning: Data: 1 subjects have missing nodes. Please check your data.
> In cpm check errors (line 45)
   In cpm main (line 50)
   In run\_cpm (line 106)

# Running over 2 Folds.
Performing fold no. 1 2
Done.
    0.0301    0.4924

```