This is a MATLAB software package to accompany the article:

Motion denoising of multiband resting state functional connectivity MRI data: An improved volume censoring method.

John C. Williams, MS and Jared X. Van Snellenberg, PhD.

Multi-Modal Translational Imaging Laboratory

Department of Psychiatry and Behavioral Health, Renaissance School of Medicine at Stony Brook University

Stony Brook, NY, USA.

This software allows a user to get optimal LPF-FD and GEV-DV volume censoring parameters based on a given dataset size, and, using either these parameters or other, user-specified parameters, obtain a vector of volumes censored using LPF-FD and GEV-DV methods using a given matrix of motion parameters (MPs) and a resting-state fMRI time series (either a voxel or greyordinate time series) and brain mask, respectively. These methods may be used separately or in tandem, and using images that are either pre-loaded into MATLAB or loaded using user-specified file names.

This software requires MATLAB R2017b or later running on any operating system. This was created and tested using MATLAB R2019b. This software does not require any non-standard software.

Installation of this software consists of downloading, copying, or moving all functions into a directory accessable to your MATLAB installation. If downloaded as a zip file, all functions should be unzipped to a single folder before use. Unzipping this software is expected to take less than one minute on a normal desktop computer.

**1. [optimal\_LPFFD\_threshold,optimal\_GEVDV\_d] = …   
getOptimalStudyThresholds(runLength,numRuns,usingGSR)**

This is a function that can be used to obtain the optimal LPF-FD and GEV-DV thresholds given a specified number of volumes per run, the number of runs per subject, and whether the data are additionally processed using global signal regression (GSR).

Inputs

1. **runLength:** The number of volumes per run in the dataset (e.g., 1000).
2. **numRuns:** The number of runs per subject in the dataset (e.g., 4).
3. **usingGSR:** True or 1 if global signal regression (GSR) is being used; false or 0 if GSR is now being used.

Outputs

1. **optimal\_LPFFD\_threshold:** The optimal LPF-FD threshold for the study given the inputs.
2. **optimal\_GEVDV\_d:** The optimal GEV-DV d for the study given the inputs.

**2. [censoredVolumes,lpffdCensoredVolumes,lpfdvCensoredVolumes] = …   
getCensoredVolumes(timeSeries,brainMask,MPs,TR,LPF\_FD\_Threshold,DV\_GEV\_d)**

This function returns a boolean (logical) vector of censored volumes using combined LPF-FD and GEV-DV based volume censoring, given a time series (either as a 4D matrix of voxel time series, or a 2D matrix of greyordinate time series), a gray matter mask (3D matrix, or 1D with columns corresponding to greyordinates), a matrix of motion parameters (MPs), the TR (required for low-pass filtering), and the desired LPF-FD threshold and DV-GEV d. It calculates the censored volumes using LPF-FD by calling **lpfFDcensoredVolumes** and the censored volumes using GEV-DV by calling **lpfDVcensoredvolumes**.

Inputs

1. **timeSeries:** The time series data to be used in order to calculate LPF-DV. This should be either in the form of a 4D matrix of voxel time series or as a 2D matrix of greyordinate time series. If voxel time series, then time should be the 4th dimension, as is the default when reading a nifti using niftiread. If greyordinate time series, then time should be the 1st non-singleton dimension (e.g., it can be either 1 x 1 x 1 x 1 x time x greyordinate, or time x greyordinate).
2. **brainMask:** The grey matter mask to be used when calculating LPF-DV. If **timeSeries** was passed in as voxel time series, then **brainMask** should be a boolean 3D matrix of voxels in the same space as **timeSeries**. If **timeSeries** was passed in as greyordinate time series, then **brainMask** should be a boolean row vector of greyordinates to use.
3. **MPs:** The motion parameters (MPs) to use when calculating LPF-FD. Dimension 1 (across rows) should be time x MP, where time is rows and MP is columns. The first 3 MPs (i.e., MP(:,1:3) ) should be translation in units of mm. The last 3 MPs (i.e., MP(:,4:6) ) should be rotation in units of degrees.
4. **TR:** The repetition time (TR) of the time series data, in units of seconds (e.g., 0.72).
5. **LPF\_FD\_Threshold:** The LPF-FD threshold desired by the user for LPF-FD volume censoring.
6. **DV\_GEV\_d:** The GEV-DV d parameter desired by the user for run-adaptive volume censoring of LPF-DV.

Outputs

1. **censoredVolumes:** A boolean column vector denoting whether a frame is targeted for removal (true) or not targeted for removal (false) by either LPF-FD or GEV-DV. It is produced by a logical “or” operation on **lpffdCensoredVolumes** and **lpfdvCensoredVolumes**.
2. **lpffdCensoredVolumes:** A boolean column vector whether a frame is targeted for removal (true) or not targeted for removal (false) by LPF-FD volume censoring only.
3. **lpfdvCensoredVolumes:** A boolean column vector whether a frame is targeted for removal (true) or not targeted for removal (false) by GEV-DV volume censoring only.

**3. [censoredVolumes,lpffdCensoredVolumes,lpfdvCensoredVolumes] = …  
getCensoredVolumesFromNifti(fileNameDataNifti,fileNameBrainMask3D,MPs,LPF\_FD\_Threshold,DV\_GEV\_d)**

This is a wrapper for **getCensoredVolumes** that opens a nifti or cifti time series using niftiread. It is identical, except that instead of taking in time series and a gray matter mask, it takes in the file name of the time series and the file name of the gray matter mask. The repetition time (TR) is obtained from the nifti header. It opens them using niftiread and calls **getCensoredVolumes;** the outputs are identical.

Inputs

1. **fileNameDataNifti4D:** The file name of the nifti RSFC time series.
2. **fileNameBrainMask3D:** The file name of the nifti gray matter mask.
3. **LPF\_FD\_Threshold:** The LPF-FD threshold desired by the user for LPF-FD volume censoring.
4. **DV\_GEV\_d:** The GEV-DV d parameter desired by the user for run-adaptive volume censoring of LPF-DV.

Outputs

1. **censoredVolumes:** A boolean column vector denoting whether a frame is targeted for removal (true) or not targeted for removal (false) by either LPF-FD or GEV-DV. It is produced by a logical “or” operation on **lpffdCensoredVolumes** and **lpfdvCensoredVolumes**.
2. **lpffdCensoredVolumes:** A boolean column vector whether a frame is targeted for removal (true) or not targeted for removal (false) by LPF-FD volume censoring only.
3. **lpfdvCensoredVolumes:** A boolean column vector whether a frame is targeted for removal (true) or not targeted for removal (false) by GEV-DV volume censoring only.

4. **[lpffdCensoredVolumes] = lpfFDcensoredVolumes(MPs,TR,LPF\_FD\_Threshold)**

This function returns a logical vector denoting which volumes are targeted for removal using LPF-FD based on input motion parameters, the TR (for low-pass filtering), and an LPF-FD threshold desired by the user.

Inputs

1. **MPs:** The motion parameters (MPs) to use when calculating LPF-FD. Dimension 1 (across rows) should be time x MP, where time is rows and MP is column. The first 3 MPs (i.e., MP(:,1:3) ) should be translation in units of mm. The last 3 MPs (i.e., MP(:,4:6) ) should be rotation in units of degrees.
2. **TR:** The repetition time (TR) of the time series data, in units of seconds (e.g., 0.72).
3. **LPF\_FD\_Threshold:** The LPF-FD threshold desired by the user for LPF-FD volume censoring.

Outputs

1. **lpffdCensoredVolumes:** A boolean column vector whether a frame is targeted for removal (true) or not targeted for removal (false) by LPF-FD volume censoring only.

5. **[lpfdvCensoredVolumes] = lpfDVcensoredvolumes(timeSeries,brainMask,TR,DV\_GEV\_d)**

This function returns a logical vector denoting which volumes are targeted for removal using GEV-DV based on input time series, a gray matter mask in the same space as the input time series, the TR (for low-pass filtering), and a GEV-DV d parameter desired by the user.

Inputs

1. **timeSeries:** The time series data to be used to calculate LPF-DV. This should be either in the form of a 4D matrix of voxel time series or as a 2D matrix of greyordinate time series. If voxel time series, then time should be the 4th dimension, as is the default when reading a nifti using niftiread. If greyordinate time series, then time should be the 1st non-singleton dimension (e.g., it can be either 1 x 1 x 1 x 1 x time x greyordinate, or time x greyordinate).
2. **brainMask:** The grey matter mask to be used when calculating LPF-DV. If **timeSeries** was passed in as voxel time series, then **brainMask** should be a boolean 3D matrix of voxels in the same space as **timeSeries**. If **timeSeries** was passed in as greyordinate time series, then **brainMask** should be a boolean row vector of greyordinates to use.
3. **TR:** The repetition time (TR) of the time series data, in units of seconds (e.g., 0.72).
4. **DV\_GEV\_d:** The GEV-DV d parameter desired by the user for run-adaptive volume censoring of LPF-DV.

Outputs

1. **lpfdvCensoredVolumes:** A boolean column vector whether a frame is targeted for removal (true) or not targeted for removal (false) by GEV-DV volume censoring only.

This software is released with the GNU General Public License Version 3.