Package ‘fMRIscrub’

July 8, 2022

Type Package

Title Scrubbing and Other Data Cleaning Routines for fMRI

Version 0.11.2

Maintainer Amanda Mejia <mandy.mejia@gmail.com>

Description Data-driven fMRI denoising with projection scrubbing (Pham et al (2022) <arXiv:2108.00319>). Also includes routines for DVARS (Derivatives VARianceS) (Afyouni and Nichols (2018) <doi:10.1016/j.neuroimage.2017.12.098>), motion scrubbing (Power et al (2012) <doi:10.1016/j.neuroimage.2011.10.018>), aCompCor (anatomical Components Correction) (Muschelli et al (2014) <doi:10.1016/j.neuroimage.2014.03.028>), detrending, and nuisance regression. Projection scrubbing and DVARS are also applicable to other outlier detection tasks involving high-dimensional data.

Depends R (>= 3.5.0)

License GPL-3

Encoding UTF-8

Imports MASS, e1071, pesel, robustbase, stats, utils

Suggests corpcor, cowplot, ciftiTools, gifti, knitr, rmarkdown, RNifti, ggplot2, gsignal, fastICA, oro.nifti, testthat (>= 3.0.0), covr

RoxygenNote 7.2.0

URL https://github.com/mandymejia/fMRIscrub

BugReports https://github.com/mandymejia/fMRIscrub/issues

LazyData true

NeedsCompilation no

Author Amanda Mejia [aut, cre],
John Muschelli [aut] (<https://orcid.org/0000-0001-6469-1750>),
Damon Pham [aut] (<https://orcid.org/0000-0001-7563-4727>),
Daniel McDonald [ctb]

Repository CRAN

Date/Publication 2022-07-08 14:30:02 UTC

R topics documented:

`artifact_images` *Artifact images*

Description

Visualize artifact patterns from the results of `pscrub`. Requires `pscrub(..., get_dirs=TRUE)`.

Usage

```
artifact_images(psx, idx = NULL, use_dt = TRUE)
```

Arguments

`psx` A `"scrub_projection"` object containing projection scrubbing results.

`idx` The timepoints or column indexes for which to compute artifact images. If `NULL` (default), use the outlying timepoints.

`use_dt` If detrended components are available (the "U" matrix of PCA or "M" matrix of ICA), should they be used to compute the artifact images? Otherwise, use the non-detrended components. Default: `TRUE`.

Details

Computes two types: "mean" artifact images based on a weighted sum of the projection directions, with weights determined by the scores for each component at the flagged timepoint, and "top" artifact images based on the projection direction with the greatest score at the flagged timepoint.

Value

A list of three: `idx`, the timepoints for which the artifact images were computed; `mean`, the "mean" artifact images; and `top`, the "top" artifact images. The row names of the `top` artifact images matrix give the index of the top component ("V" in PCA and "S" in ICA) at each timepoint.

`bandstop_filter` *Bandstop filter*

Description

Filter out frequencies within a given range using a Chebyshev Type II stopband. Basically a convenience wrapper for the `gsignal::cheby2` function.

Usage

```
bandstop_filter(x, TR, f1, f2, Rs = 20)
```

Arguments

`x` A numeric matrix, with each column being a timeseries to apply the stopband filter

`TR` the time difference between rows in x, in seconds.

`f1, f2` The frequency limits for the filter, in Hz. `f1` < `f2`.

`Rs` The amount of attenuation of the stopband ripple, in dB

Value

The filtered data

Examples

```
if (requireNamespace("gsignal", quietly = TRUE)) {
 n_voxels = 1e4
 n_timepoints = 100
 X = cbind(arima.sim(n=100, list(ar=.6)), arima.sim(n=100, list(ar=.6)))
 Y = bandstop_filter(X, .72, .31, .43)
}
```

`carpetplot` *Carpetplot*

Description

Plot a matrix with `graphics::image`. For fMRI data, this is the "carpetplot" or grayplot coined by (Power, 2017). The `graphics` package is required.

Usage

```
carpetplot(
  x,
  qcut = 0.1,
  fname = NULL,
  center = TRUE,
  scale = FALSE,
  colors = "gray255",
  sortSub = TRUE,
  ...
)
```

Arguments

`x` The TxV numeric data matrix, or a `"xifti"` object. In the plot, the T index will increase from left to right, and the V will increase from top to bottom.

`qcut` Sets blackpoint at the `qcut` quantile, and the whitepoint at the `1-qcut` quantile. Default: `.1`. This is equivalent to setting the color range between the 10% and 90% quantiles. The quantiles are computed across the entire data matrix after any centering or scaling.

Must be between 0 and .49. If `0` or `NULL` (default), do not clamp the data values.

`fname` A `.pdf` (highly recommended) or `.png` file path to write the carpetplot to. If `NULL` (default), return the plot directly instead of writing a file.

`center, scale` Center and scale the data? If `x` is fMRI data which has not otherwise been centered or scaled, it is recommended to center but not scale it (default).

`colors` `"gray255"` (default) will use a grayscale color ramp from black to white. Otherwise, this should be a character vector of color names to use.

Colors will be assigned from the lowest to the highest data value, after any clamping of the data values by `qcut`.

`sortSub` If `x` is a `"xifti"` object with subcortical data, should the voxels be sorted by structure alphabetically? Default: `TRUE`.

`...` Additional arguments to `pdf` or `png`, such as width and height.

Value

The image or `NULL, invisibly` if a file was written.

References

- Power, J. D. A simple but useful way to assess fMRI scan qualities. NeuroImage 154, 150-158 (2017).

`carpetplot_stack` *Stacked carpetplot*

Description

Stacks carpetplots on top of one anotherr by rbinding the matrices.

Usage

```
carpetplot_stack(
  x_list,
  center = TRUE,
  scale = FALSE,
  qcut = 0.1,
  match_scale = TRUE,
  nsep = 0,
  ...
)
```

Arguments

`x_list` List of data matrices

`center, scale` Center and scale the data? If `x` is fMRI data which has not otherwise been centered or scaled, it is recommended to center but not scale it (default).

`qcut` Sets blackpoint at the `qcut` quantile, and the whitepoint at the `1-qcut` quantile. Default: `.1`. This is equivalent to setting the color range between the 10% and 90% quantiles. The quantiles are computed across the entire data matrix after any centering or scaling.

Must be between 0 and .49. If `0` or `NULL` (default), do not clamp the data values.

`match_scale` Match the scales of the carpetplots? Default: `TRUE`.

`nsep` Equivalent number of data locations for size of gap between carpetplots. Default: zero (no gap).

`...` Additional arguments to `carpetplot`

Value

NULL, invisibly

CompCor *Anatomical CompCor*

Description

The aCompCor algorithm for denoising fMRI data using noise ROIs data

Usage

```
CompCor(
  X,
  ROI_data = "infer",
  ROI_noise = NULL,
  noise_nPC = 5,
  noise_erosion = NULL,
  center = TRUE,
  scale = TRUE,
  nuisance = NULL
)
```

Arguments

X
: Wide numeric data matrix ($Tobservations$ by $Vvariables$, $T << V$). For example, if X represents an fMRI run, T should be the number of timepoints and V should be the number of brainordinate vertices/voxels.

Or, a 4D array or NIFTI or file path to a NIFTI (I by J by K by T observations), in which case ROI_data must be provided. (The vectorized data will be $Ttimepoints$ by $V_{in-mask}voxels$)

Or, a ciftiTools "xifti" object or a file path to a CIFTI (The vectorized data will be $Ttimepoints$ by $V_{left+right+sub}greyordinates$).

ROI_data
: Indicates the data ROI. Allowed arguments depend on X:

If X is a matrix, this must be a length V logical vector, where the data ROI is indicated by TRUE values. If "infer" (default), all columns of X will be included in the data ROI (rep(TRUE, V)).

If X is an array or NIFTI, this must be either a vector of values to expect for out-of-mask voxels in X, or a (file path to a) 3D NIFTI. In the latter case, each of the volume dimensions should match the first three dimensions of X. Voxels in the data ROI should be indicated by TRUE and all other voxels by FALSE. If "infer" (default), will be set to c(0, NA, NaN) (include all voxels which are not constant 0, NA, or NaN).

If X is a "xifti" this must be the brainstructures argument to read_cifti. If "infer" (default), brainstructures will be set to "all" (use both left and right cortex vertices, and subcortical voxels).

If `NULL`, the data ROI will be empty. This is useful for obtaining just the noise ROI, if the data and noise are located in separate files.

`ROI_noise` Indicates the noise ROIs for aCompCor. Should be a list where each entry corresponds to a distinct noise ROI. The names of the list should be the ROI names, e.g. `"white_matter"` and `"csf"`. The expected formats of the list entries depends on `X`:

For all types of `X`, `ROI_noise` entries can be a matrix of noise ROI data. The matrix should have T rows, with each column being a data location's timeseries.

If `X` is a matrix, entries can also indicate a noise ROI within `X`. These entries must be a length V logical vector with `TRUE` values indicating locations in `X` within that noise ROI. Since the ROIs must not overlap, the masks must be mutually exclusive with each other, and with `ROI_data`.

If `X` is an array or NIFTI, entries can also indicate a noise ROI within `X`. These entries must be a logical array or (file path to) a 3D NIFTI with the same spatial dimensions as `X`, and with `TRUE` values indicating voxels inside the noise ROI. Since the ROIs must not overlap, the masks must be mutually exclusive with each other, and with `ROI_data`.

(If `X` is a `"xifti"`, entries must be data matrices, since no greyordinate locations in `X` are appropriate noise ROIs).

`noise_nPC` The number of principal components to compute for each noise ROI. Alternatively, values between 0 and 1, in which case they will represent the minimum proportion of variance explained by the PCs used for each noise ROI. The smallest number of PCs will be used to achieve this proportion of variance explained.

Should be a list or numeric vector with the same length as `ROI_noise`. It will be matched to each ROI based on the name of each entry, or if the names are missing, the order of entries. If it is an unnamed vector, its elements will be recycled. Default: `5` (compute the top 5 PCs for each noise ROI).

`noise_erosion` The number of voxel layers to erode the noise ROIs by. Should be a list or numeric vector with the same length as `ROI_noise`. It will be matched to each ROI based on the name of each entry, or if the names are missing, the order of entries. If it is an unnamed vector, its elements will be recycled. Default: `NULL`, which will use a value of 0 (do not erode the noise ROIs). Note that noise erosion can only be performed if the noise ROIs are volumetric.

`center, scale` Center the columns of the noise ROI data by their medians, and scale by their MADs? Default: `TRUE` for both. Note that this argument affects the noise ROI data and not the data that is being cleaned with aCompCor. Centering and scaling of the data being cleaned can be done after this function call.

`nuisance` Nuisance signals to regress from each data column in addition to the noise ROI PCs. Should be a T by N numeric matrix where N represents the number of nuisance signals. To not perform any nuisance regression set this argument to `NULL`, `0`, or `FALSE`. Default: `NULL`.

Details

First, the principal components (PCs) of each noise region of interest (ROI) are calculated. For each ROI, voxels are centered and scaled (can be disabled with the arguments `center` and `scale`), and then the PCs are calculated via the singular value decomposition.

Next, aCompCor is performed to remove the shared variation between the noise ROI PCs and each location in the data. This is accomplished by a nuisance regression using a design matrix with the noise ROI PCs, any additional regressors specified by `nuisance`, and an intercept term. (To detrend the data and perform aCompCor in the same regression, `nuisance` can be set to DCT bases obtained with the function `dct_bases`.)

Value

A list with entries `"data"`, `"noise"`, and potentially `"ROI_data"`.

The entry `"data"` will be a `V x T` matrix where each row corresponds to a data location (if it was originally an array, the locations will be voxels in spatial order). Each row will be a time series with each noise PC regressed from it. This entry will be `NULL` if there was no data.

The entry `"noise"` is a list of noise PC scores, their corresponding variance, and their ROI mask, for each noise ROI.

If the data ROI is not all `TRUE`, the entry `"ROI_data"` will have the ROI mask for the data.

References

- Behzadi, Y., Restom, K., Liau, J. & Liu, T. T. A component based noise correction method (CompCor) for BOLD and perfusion based fMRI. NeuroImage 37, 90-101 (2007).
- Muschelli, J. et al. Reduction of motion-related artifacts in resting state fMRI using aCompCor. NeuroImage 96, 22-35 (2014).

See Also

CompCor_HCP

`CompCor_HCP` *Anatomical CompCor for HCP NIFTI and CIFTI data*

Description

Wrapper to `CompCor` for HCP-format data. Can be used to clean the surface-based CIFTI data with aCompCor using the noise PCs and ROIs calculated from the NIFTI fMRI data and NIFTI mask. Can also be used to just obtain the noise PCs and ROIs without performing aCompCor, if the CIFTI data is not provided.

Usage

```
CompCor_HCP(
  nii,
  nii_labels,
  ROI_noise = c("wm_cort", "csf"),
  noise_nPC = 5,
  noise_erosion = NULL,
  idx = NULL,
  cii = NULL,
  brainstructures = c("left", "right"),
  center = TRUE,
  scale = TRUE,
  DCT = 0,
  nuisance_too = NULL,
  verbose = FALSE
)
```

Arguments

`nii` I by J by K by T NIFTI object or array (or file path to the NIFTI) which contains whole-brain data, including the noise ROIs. In the HCP, the corresponding file is e.g. "../Results/rfMRI_REST1_LR/rfMRI_REST1_LR.nii.gz"

`nii_labels` I by J by K NIFTI object or array (or file path to the NIFTI) which contains the corresponding labels to each voxel in `nii`. Values should be according to this table: https://surfer.nmr.mgh.harvard.edu/fswiki/FsTutorial/AnatomicalROI/FreeSurferColorLUT . In the HCP, the corresponding file is "ROIs/Atlas_wmparc.2.nii.gz".

`ROI_noise` A list of numeric vectors. Each entry should represent labels in `nii_labels` belonging to a single noise ROI, named by that entry's name. Or, this can be a character vector of at least one of the following: `"wm_cort"` (cortical white matter), `"wm_cblm"` (cerebellar white matter), `"csf"` (cerebrospinal fluid). In the latter case, these labels will be used:

`"wm_cort"` `c(3000:4035, 5001, 5002)`

`"wm_cblm"` `c(7, 46)`

`"csf"` `c(4, 5, 14, 15, 24, 31, 43, 44, 63, 250, 251, 252, 253, 254, 255))`

These default ROIs are based on this forum post: https://www.mail-archive.com/hcp-users@humanconnectome.org/msg00931.html

Default: `c("wm_cort", "csf")`

`noise_nPC` The number of principal components to compute for each noise ROI. Alternatively, values between 0 and 1, in which case they will represent the minimum proportion of variance explained by the PCs used for each noise ROI. The smallest number of PCs will be used to achieve this proportion of variance explained.

Should be a list or numeric vector with the same length as `ROI_noise`. It will be matched to each ROI based on the name of each entry, or if the names are missing, the order of entries. If it is an unnamed vector, its elements will be recycled. Default: `5` (compute the top 5 PCs for each noise ROI).

`noise_erosion` The number of voxel layers to erode the noise ROIs by. Should be a list or numeric vector with the same length as `ROI_noise`. It will be matched to each ROI based on the name of each entry, or if the names are missing, the order of entries. If it is an unnamed vector, its elements will be recycled. Default: `NULL`, which will use a value of 0 (do not erode the noise ROIs).

`idx` A numeric vector indicating the timepoints to use, or `NULL` (default) to use all idx. (Indexing begins with 1, so the first timepoint has index 1 and the last has the same index as the length of the scan.)

`cii` "xifti" (or file path to the CIFTI) from which the noise ROI components will be regressed. In the HCP, the corresponding file is e.g. "../Results/rfMRI_REST1_LR/rfMRI_REST1_LR_
If not provided, only the noise components will be returned (no data will be cleaned).

`brainstructures`
Choose among "left", "right", and "subcortical". Default: `c("left", "right")` (cortical data only)

`center, scale` Center the columns of the data by median, and scale the columns of the data by MAD? Default: `TRUE` for both. Affects both `X` and the noise data. `center` also applies to `nuisance_too` so if it is `FALSE`, `nuisance_too` must already be centered.

`DCT` Add DCT bases to the nuisance regression? Use an integer to indicate the number of cosine bases. Use `0` (default) to forgo detrending.
The data must be centered, either before input or with `center`.

`nuisance_too` A matrix of nuisance signals to add to the nuisance regression. Should have T rows. `NULL` to not add additional nuisance regressors (default).

`verbose` Should occasional updates be printed? Default: `FALSE`.

Value

The noise components, and if `cii` is provided, the cleaned surface-based data as a `"xifti"` object.

References

- Behzadi, Y., Restom, K., Liau, J. & Liu, T. T. A component based noise correction method (CompCor) for BOLD and perfusion based fMRI. NeuroImage 37, 90-101 (2007).
- Muschelli, J. et al. Reduction of motion-related artifacts in resting state fMRI using aCompCor. NeuroImage 96, 22-35 (2014).

See Also

CompCor

`Dat1` *First Example Time Series from the ABIDE*

Description

A sagittal slice from the fMRI time series for subject 0050048. The scan was obtained at the University of Pittsburgh School of Medicine. The scan has been pre-processed with slice time correction, rigid body realignment estimation, spatial normalization to MNI space, and linear detrending. Subject 0050048 was a typically-developing 11-year-old male. The scan has many artifacts. A mask was applied to vectorize the spatial dimensions.

Usage

```
Dat1
```

Format

A numeric matrix of 193 time points by 4675 voxels

Details

Source: http://fcon_1000.projects.nitrc.org/indi/abide/abide_I.html

References

- 1.Di Martino, A. et al. The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. Mol Psychiatry 19, 659–667 (2014).

`Dat2` *Second Example Time Series from the ABIDE*

Description

A sagittal slice from the fMRI time series for subject 0051479. The scan was obtained at the California Institute of Technology. The scan has been pre-processed with slice time correction, rigid body realignment estimation, spatial normalization to MNI space, and linear detrending. Subject 0051479 was a typically-developing 20-year-old female. The scan has few visible artifacts. A mask was applied to vectorize the spatial dimensions.

Usage

```
Dat2
```

Format

A numeric matrix of 145 time points by 4679 voxels

Details

Source: http://fcon_1000.projects.nitrc.org/indi/abide/abide_I.html

References

- 1.Di Martino, A. et al. The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. Mol Psychiatry 19, 659–667 (2014).

dct_bases *Generate cosine bases for the DCT*

Description

Generate cosine bases for the DCT

Usage

```
dct_bases(T_, n)
```

Arguments

T_	Length of timeseries
n	Number of cosine bases

Value

Matrix with cosine bases along columns

dct_convert *DCT and frequency conversion*

Description

Convert between number of DCT bases and Hz of highpass filter

Usage

```
dct_convert(T_, TR, n = NULL, f = NULL)

dct2Hz(T_, TR, n)

Hz2dct(T_, TR, f)
```

Arguments

T_	Length of timeseries (number of timepoints)
TR	TR of the fMRI scan, in seconds (the time between timepoints)
n	Number of cosine bases
f	Hz of highpass filter

Details

Provide either n or f to calculate the other.

If only the total length of the scan is known, you can set that to TR and use T_=1.

$f = n/(2 * T_* TR)$

Value

If n was provided, the highpass filter cutoff (Hz) is returned. Otherwise, if f was provided, the number of cosine bases is returned. The result should be rounded before passing to dct_bases

---

detrend *Detrending with DCT or FFT*

Description

Detrending with DCT or FFT

Usage

```
detrend(X, TR, f = 0.008, method = c("DCT", "FFT"))
```

Arguments

X	$T \times V$ numeric matrix. Each column is a voxel or vertex time series.
TR	TR of the fMRI scan, in seconds (the time between timepoints)
f	Hz of highpass filter. Default: .008
method	"DCT" (default) or "FFT".

Value

Detrended X

DVARS *DVARS*

Description

Computes the DSE decomposition and DVARS-related statistics. Based on code from github.com/asoroosh/DVARS
.

Usage

```
DVARS(
  X,
  normalize = TRUE,
  cutoff_DPD = 5,
  cutoff_ZD = qnorm(1 - 0.05/nrow(as.matrix2(X))),
  verbose = FALSE
)
```

Arguments

`X` a T by N numeric matrix representing an fMRI run. There should not be any missing data (`NA` or `NaN`).

`normalize` Normalize the data? Default: `TRUE`. Normalization removes constant-zero voxels, scales by 100 / the median of the mean image, and then centers each voxel on its mean.

To replicate Afyouni and Nichols' procedure for the HCP MPP data, since the HCP scans are already normalized to 10,000, just divide the data by 100 and center the voxels on their means:

`Y <- Y/100; DVARS(t(Y - apply(Y, 1, mean)))` where `Y` is the V by T data matrix.

Note that while voxel centering doesn't affect DVARS, it does affect DPD and ZD.

`cutoff_DPD, cutoff_ZD` Numeric outlier cutoffs. Timepoints exceeding these cutoffs will be flagged as outliers.

`verbose` Should occasional updates be printed? Default is `FALSE`.

Value

A list with components

measure A data.frame with T rows, each column being a different variant of DVARS.

measure_info "DVARS"

outlier_cutoff The outlier cutoff value(s).

outlier_flag A logical data.frame with T rows, where `TRUE` indicates suspected outlier presence.

References

- Afyouni, S. & Nichols, T. E. Insight and inference for DVARS. NeuroImage 172, 291-312 (2018).

erode_vol *Erode volumetric mask*

Description

Erode a volumetric mask by a certain number of voxel layers. For each layer, any in-mask voxel adjacent to at least one out-of-mask voxel is removed from the mask.

Usage

```
erode_vol(vol, n_erosion = 1, out_of_mask_val = NA)
```

Arguments

vol	The volume to erode. Out-of-mask voxels should be indicated by a value in `out_of_mask_val`.
n_erosion	The number of layers to erode the mask by.
out_of_mask_val	A voxel is not included in the mask if and only if its value is in this vector. The first value in this vector will be used to replace the eroded voxels. Default: `NA`.

Details

Diagonal voxels are not considered adjacent, i.e. the voxel at (0,0,0) is not adjacent to the voxel at (1,1,0) or (1,1,1), although it is adjacent to (1,0,0).

Value

The eroded `vol`. It is the same as `vol` but with eroded voxels replaced with the value `out_of_mask_val[1]`.

FD *Framewise Displacement*

Description

Calculate Framewise Displacement (FD)

Usage

```
FD(
  X,
  trans_units = c("mm", "cm", "in"),
  rot_units = c("deg", "rad", "mm", "cm", "in"),
  brain_radius = NULL,
  detrend = FALSE,
  lag = 1,
  cutoff = 0.3
)
```

Arguments

X
: An N by 6 matrix in which the first three columns represent the translational RPs (trans_units), and the second three columns represent the rotational RPs (rot_units). If rot_units measures an angle, it will be converted to trans_units by measuring displacement on a sphere of radius brain_radius trans_units.

  Alternatively, this can be the file path to an N by 6 matrix which can be read with read.table (fields separated by white-space; no header).

trans_units
: "mm" for millimeters (default), "cm" for centimeters, or "in" for inches.

rot_units
: "deg" for degrees (default), "rad" for radians, or one of the trans_units options.

brain_radius
: If rot_units measures an angle, the rotational RPs are transformed to a spatial measurement representing the displacement on a sphere of radius brain_radius trans_units.

  If brain_radius is NULL (default), it will be set to 50 mm.

detrend
: Detrend each RP with the DCT before computing FD? Default: FALSE. Can be a number of DCT bases to use, or TRUE to use 4.

lag
: The difference of indices between which to calculate change in position. Default: 1 (the previous timepoint). Changing this argument sets $\Delta x_i = x_{i-lag} - x_i$ (and similarly for the other RPs).

cutoff
: FD values higher than this will be flagged. Default: .3.

Details

The FD formula is taken from Power et. al. (2012):

$$FD_i = |\Delta x_i| + |\Delta y_i| + |\Delta z_i| + |\Delta \alpha_i| + |\Delta \beta_i| + |\Delta \gamma_i|$$

where i is the timepoint; x, y and z are the translational realignment parameters (RPs); α, β and γ are the rotational RPs; and $\Delta x_i = x_{i-1} - x_i$ (and similarly for the other RPs).

Value

A list with components

measure A length N vector of FD values in trans_units.

measure_info "FD"

outlier_cutoff cutoff

outlier_flag A length-N logical vector, where TRUE indicates suspected outlier presence.

References

- Power, J. D., Barnes, K. A., Snyder, A. Z., Schlaggar, B. L. & Petersen, S. E. Spurious but systematic correlations in functional connectivity MRI networks arise from subject motion. Neuroimage 59, 2142-2154 (2012).

fMRIscrub — *fMRIscrub: fMRI scrubbing and other data cleaning routines*

Description

See `help(package="fMRIscrub")` for a list of functions.

fsl_bptf — `bptf` *function from FSL*

Description

Copy of highpass filter as implemented by `bptf` in FSL. The results are very similar but not identical.

Usage

```
fsl_bptf(orig_data, HP_sigma = 2000)
```

Arguments

`orig_data`	T by V data matrix whose columns will be detrended
`HP_sigma`	The frequency parameter for the highpass filter

Details

Sources: https://cpb-us-w2.wpmucdn.com/sites.udel.edu/dist/7/4542/files/2016/09/fsl_temporal_filt-15sywxn.m https://github.com/rordenlab/niimath/blob/master/src/core32.c

Value

The data with detrended columns

References

- Jenkinson, M., Beckmann, C. F., Behrens, T. E. J., Woolrich, M. W. & Smith, S. M. FSL. NeuroImage 62, 782-790 (2012).

high_kurtosis *Which components have high kurtosis?*

Description

The kurtosis cutoff is a high quantile (default 0.99) of the sampling distribution of kurtosis for Normal iid data of the same length as the components; it is estimated by simulation or calculated from the theoretical asymptotic distribution if the components are long enough.

Usage

```
high_kurtosis(Comps, kurt_quantile = 0.99, n_sim = 5000, min_1 = FALSE)
```

Arguments

`Comps`	A matrix; each column is a component. For PCA, this is the U matrix. For ICA, this is the M matrix.
`kurt_quantile`	components with kurtosis of at least this quantile are kept.
`n_sim`	The number of simulation data to use for estimating the sampling distribution of kurtosis. Only used if a new simulation is performed. (If $n < 1000$ and the quantile is 90%, a pre-computed value is used instead. If $n > 1000$, the theoretical asymptotic distribution is used instead.)
`min_1`	Require at least one component to be selected? In other words, if no components meet the quantile cutoff, should the component with the highest kurtosis be returned? Default: `FALSE`.

Details

The components should not have any strong low-frequency trends, because trends can affect kurtosis in unpredictable ways unrelated to outlier presence.

Value

A logical vector indicating whether each component has high kurtosis.

leverage *Leverage*

Description

Computes the leverage of each observation in the PC score (U) or IC mixing (M) matrix for projection scrubbing. Can threshold to flag potential outliers.

Usage

```
leverage(Comps, are_norm = FALSE, median_cutoff = NULL)
```

Arguments

`Comps` The n by Q PC score matrix/IC mixing matrix.

`are_norm` Assume the columns of `Comps` are orthogonal and have 2-norms equal to 1? Speeds up the computation.

`median_cutoff` The outlier cutoff, in multiples of the median leverage. Default: `NULL` (do not compute outliers).

Value

A list with entries `"meas"` (the leverage values), `"cut"` (the leverage cutoff value) and `"flag"` (logical vector indicating the outliers). If `!is.null(median_cutoff)`, `"cut"` and `"flag"` are omitted.

`nuisance_regression` *Nuisance regression*

Description

Performs nuisance regression. The data and design matrix must both be centered, or an intercept must be included in the design matrix!

Usage

```
nuisance_regression(Y, design)
```

Arguments

`Y` The T by V or V by T data.

`design` The T by Q matrix of nuisance regressors.

Value

The data after nuisance regression.

`pct_sig` *Convert data values to percent signal.*

Description

Convert data values to percent signal.

Usage

```
pct_sig(X, center = median, by = c("column", "all"))
```

Arguments

X	a T by N numeric matrix. The columns will be normalized to percent signal.
center	A function that computes the center of a numeric vector. Default: `median`. Other common options include `mean` and `mode`.
by	Should the center be measured individually for each `"column"` (default), or should the center be the same across `"all"` columns?

Value

`X` with its columns normalized to percent signal. (A value of 85 will represent a -15% signal change.)

`plot.scrub_DVARS` *Plot a* `"scrub_DVARS"` *object*

Description

Plot a `"scrub_DVARS"` object

Usage

```
## S3 method for class 'scrub_DVARS'
plot(x, title = NULL, ...)
```

Arguments

x	The `"scrub_DVARS"` object
title	(Optional) If provided, will add a title to the plot.
...	Additional arguments to ggplot, e.g. `main`, `sub`, `xlab`, `ylab`, `legend.position`

Value

A ggplot

plot.scrub_FD *Plot a "scrub_FD" object*

Description

Plot a "scrub_FD" object

Usage

```
## S3 method for class 'scrub_FD'
plot(x, title = NULL, ...)
```

Arguments

x	The "scrub_FD" object
title	(Optional) If provided, will add a title to the plot.
...	Additional arguments to ggplot, e.g. main, sub, xlab, ylab, legend.position

Value

A ggplot

plot.scrub_projection *Plot a "scrub_projection" object*

Description

Plot a "scrub_projection" object

Usage

```
## S3 method for class 'scrub_projection'
plot(x, title = NULL, ...)
```

Arguments

x	The "scrub_projection" object
title	(Optional) If provided, will add a title to the plot.
...	Additional arguments to ggplot, e.g. main, sub, xlab, ylab, legend.position

Value

A ggplot

pscrub *Projection scrubbing*

Description

Projection scrubbing is a data-driven method for identifying artifact-contaminated volumes in fMRI. It works by identifying component directions in the data likely to represent patterns of burst noise, and then computing a composite measure of outlyingness based on leverage within these directions, at each volume. The projection can be PCA, ICA, or "fused PCA." Projection scrubbing can also be used for other outlier detection tasks involving high-dimensional data.

Usage

```
pscrub(
  X,
  projection = c("ICA", "PCA"),
  nuisance = "DCT4",
  center = TRUE,
  scale = TRUE,
  comps_mean_dt = FALSE,
  comps_var_dt = FALSE,
  PESEL = TRUE,
  kurt_quantile = 0.99,
  get_dirs = FALSE,
  full_PCA = FALSE,
  get_outliers = TRUE,
  cutoff = 4,
  seed = 0,
  verbose = FALSE
)
```

Arguments

`X` Wide numeric data matrix (T observations by V variables, $T << V$). If `X` represents an fMRI run, T should be the number of timepoints and V should be the number of vertices/voxels. Projection scrubbing will measure the outlyingness of each row in `X`.

`projection` One of the following: `"ICA"` (default) or `"PCA"`.

`nuisance` Nuisance signals to regress from each column of `X`. Should be specified as a design matrix: a T by N numeric matrix where N represents the number of nuisance signals. Or can be "DCT4" (default), which will create a matrix with a constant column (the intercept term) and four DCT bases. This default nuisance regression will have the effect of demeaning and detrending the data by removing low-frequency components. To not perform any nuisance regression set this argument to `NULL`, `0`, or `FALSE`.

Detrending is highly recommended for time-series data, especially if there are many time points or evolving circumstances affecting the data. Additionally, if kurtosis is being used to select the projection directions, trends can induce positive or negative kurtosis, contaminating the connection between high kurtosis and outlier presence. Detrending should not be used with non-time-series data because the observations are not temporally related.

Additional nuisance regressors can be specified like so: `cbind(1, dct_bases(nrow(x), 4), more_nuisance)`.

`center, scale` Center the columns of the data by their medians, and scale the columns of the data by their median absolute deviations (MADs)? Default: `TRUE`. Centering is necessary for computing the projections, so if `center` is `FALSE`, the data must already be centered.

Note that centering and scaling occur after nuisance regression, so even if `center` is `FALSE`, the data will be centered on the means if the nuisance regression included an intercept term, as it does by default.

`comps_mean_dt, comps_var_dt`

Stabilize the mean and variance of each projection component's timecourse prior to computing kurtosis and leverage? These arguments should be `TRUE`, `FALSE` (default), or the number of DCT bases to use for detrending (`TRUE` will use 4). Note that these arguments affect the projection components and not the data itself. Also, if variance-stabilizing but not mean-stabilizing, the components must already be expected to be mean-stabilized, for example if the data was rigorously detrended; otherwise, the results will be invalid.

Slow-moving mean and variance patterns in the components will interfere with the roles of kurtosis and leverage in identifying outliers. While `nuisance` can be used to detrend the data, this nuisance regression is estimated *non-robustly*, since a robust model takes too long to estimate at each data location. On the other hand, `comps_mean_dt` and `comps_var_dt` can be used to apply a *robust* nuisance regression at each component, since there are much fewer components than original data locations. Thus, even if the data has been detrended with `nuisance` it may be helpful to detrend the components with `comps_mean_dt`; furthermore, the data nuisance regression does not address the potential existence of variance patterns in the components.

Overall, for fMRI we recommend enabling `comps_mean_dt` and `comps_var_dt` unless the data has been cleaned not only with a low-pass filter like DCT nuisance regression, but also with anatomical CompCor, ICA-FIX, or a similar data-driven strategy that takes into account common sources of artifactual mean and variance trends such as motion and physiological cycles.

`PESEL` Use `pesel` to select the number of components? Default: `TRUE`. Otherwise, use the number of principal components with above-average variance.

`kurt_quantile` What quantile cutoff should be used to select the components? Default: `0.99`. Use `0` to select all high-variance components regardless of kurtosis value.

We model each component as a length T vector of Normal iid random variables, for which the distribution of kurtosis values can be approximated. The quantile is estimated based on this distribution.

`get_dirs` Should the projection directions be returned? This is the V matrix in PCA and S matrix in ICA. The default is `FALSE` to save memory. However, `get_dirs==TRUE` is required for `artifact_images`.

`full_PCA`	Only applies to the PCA projection. Return the full SVD? Default: `FALSE` (return only the high-variance components).
`get_outliers`	Should outliers be flagged based on `cutoff`? Default: `TRUE`.
`cutoff`	Median leverage cutoff value. Default: `4`.
`seed`	Set a seed right before the call to `pesel::pesel` or `ica::icaimax`? If `NULL`, do not set a seed. If numeric (default: `0`), will use as the seed.
`verbose`	Should occasional updates be printed? Default: `FALSE`.

Details

Refer to the projection scrubbing vignette for a demonstration and an outline of the algorithm: `vignette("projection_scrubbing", package="fMRIscrub")`

Value

A `"pscrub"` object, i.e. a list with components

measure A numeric vector of leverage values.

outlier_cutoff The numeric outlier cutoff value (`cutoff` times the median leverage).

outlier_flag A logical vector where `TRUE` indicates where leverage exceeds the cutoff, signaling suspected outlier presence.

mask A length P numeric vector corresponding to the data locations in `X`. Each value indicates whether the location was masked:

- **0** The data location was not masked out.
- **-1** The data location was masked out, because it had at least one `NA` or `NaN` value.
- **-2** The data location was masked out, because it was constant.

PCA This will be a list with components:

- **U** The T by Q PC score matrix.
- **D** The standard deviation of each PC.
- **V** The P by Q PC directions matrix. Included only if `get_dirs`.
- **highkurt** The length `Q` logical vector indicating scores of high kurtosis.
- **U_dt** Detrended components of `U`. Included only if components were mean- or variance-detrended.
- **highkurt** The length `Q` logical vector indicating detrended scores of high kurtosis.
- **nPCs_PESEL** The number of PCs selected by PESEL.
- **nPCs_avgvar** The number of above-average variance PCs.

where `Q` is the number of PCs selected by PESEL or of above-average variance (or the greater of the two if both were used). If PCA was not used, all entries except `nPCs_PESEL` and/or `nPCs_avgvar` will not be included, depending on which method(s) was used to select the number of PCs.

ICA If ICA was used, this will be a list with components:

- **S** The P by Q source signals matrix. Included only if `get_dirs`
- **M** The T by Q mixing matrix.
- **highkurt** The length `Q` logical vector indicating mixing scores of high kurtosis.

M_dt Detrended components of M. Included only if components were mean- or variance-detrended.

highkurt The length Q logical vector indicating detrended mixing scores of high kurtosis. Included only if components were mean- or variance-detrended.

References

- Mejia, A. F., Nebel, M. B., Eloyan, A., Caffo, B. & Lindquist, M. A. PCA leverage: outlier detection for high-dimensional functional magnetic resonance imaging data. Biostatistics 18, 521-536 (2017).
- Pham, D., McDonald, D., Ding, L., Nebel, M. B. & Mejia, A. Less is more: balancing noise reduction and data retention in fMRI with projection scrubbing. (2022).

Examples

```
if (requireNamespace("fastICA", quietly = TRUE)) {
 n_voxels = 5e3
 n_timepoints = 70
 X = matrix(rnorm(n_timepoints*n_voxels), ncol = n_voxels)
 psx = pscrub(X)
}
```

---

rob_stabilize *Stabilize the center and scale of a timeseries robustly*

Description

Stabilize the center and scale of a timeseries using robust regression of DCT bases on the first and second moments.

Usage

```
rob_stabilize(
  x,
  center = TRUE,
  scale = TRUE,
  lmrob_method = "MM",
  rescale = TRUE
)
```

Arguments

x	The timeseries to stabilize.
center, scale	Center and scale? Default: TRUE for both. If scaling but not centering, the data must already be centered; otherwise, the results will be invalid. Can also be the number of DCT bases to use for robust stabilization of center/scale; TRUE will use 4.

`lmrob_method` The `lmrob_method` argument to `robustbase::lmrob`.

`rescale` After stabilizing `x`, re-center and re-scale to the original mean and variance? Default: `TRUE`.

Value

the timeseries with its center and scale stabilized

`scale_med` *Robust scaling*

Description

Centers and scales the columns of a matrix robustly

Usage

```
scale_med(mat, TOL = 1e-08, drop_const = TRUE)
```

Arguments

`mat` A numerical matrix.

`TOL` minimum MAD to consider a column non-constant. Default: `1e-8`

`drop_const` Drop

Details

Centers each column on its median, and scales each column by its median absolute deviation (MAD). Constant-valued columns are set to `NA` (or removed if `drop_const`) and a warning is raised. If all MADs are zero, an error is raised.

Value

The input matrix with its columns centered and scaled.

`scrub` *Data-driven scrubbing*

Description

Performs projection scrubbing or DVARS scrubbing, and optionally thresholds to identify artifactual time points.

Usage

```
scrub(X, method = c("projection", "DVARS"), ...)
```

Arguments

`X`	A T by V numeric matrix representing an fMRI run. There should not be any missing data (`NA` or `NaN`).
`method`	`"projection"` (default) or `"DVARS"`
`...`	Additional arguments to the specific scrubbing function: see `pscrub` or `DVARS`.

Value

A list with components

measure A length T vector or data.frame with T rows, giving the outlyingness measure(s)

measure_info Describes the outlyingness measure(s)

outlier_cutoff The outlier cutoff value(s).

outlier_flag A length T vector or data.frame with T rows, where `TRUE` indicates suspected outlier presence.

`scrub_xifti` *Scrub fMRI data in CIFTI format*

Description

Performs projection scrubbing or DVARS scrubbing, and optionally thresholds to identify artifactual time points. Requires `ciftiTools` and the Connectome Workbench.

Usage

```
scrub_xifti(
  X,
  method = c("projection", "DVARS"),
  brainstructures = c("left", "right"),
  ...
)
```

Arguments

`X`	Path to a CIFTI file, or a `"xifti"` object.
`method`	`"projection"` or `"DVARS"`
`brainstructures`	Character vector indicating which brain structure(s) to use: `"left"` (left cortical surface), `"right"` (right cortical surface) and/or `"subcortical"` (subcortical and cerebellar gray matter). Can also be `"all"` (obtain all three brain structures). Default: `c("left", "right")` (excludes the subcortex).
`...`	Additional arguments to each specific scrubbing function: `pscrub` or `DVARS`.

Value

A list with components

measure A length T vector or data.frame with T rows, giving the outlyingness measure(s)

measure_info Describes the outlyingness measure(s)

outlier_cutoff The outlier cutoff value(s).

outlier_flag A length T vector or data.frame with T rows, where `TRUE` indicates suspected outlier presence.

`summary.scrub_DVARS` *Summarize a* `"scrub_DVARS"` *object*

Description

Summary method for class `"scrub_DVARS"`

Usage

```
## S3 method for class 'scrub_DVARS'
summary(object, ...)

## S3 method for class 'summary.scrub_DVARS'
print(x, ...)

## S3 method for class 'scrub_DVARS'
print(x, ...)
```

Arguments

`object`	Object of class `"scrub_DVARS"`.
`...`	further arguments passed to or from other methods.
`x`	Object of class `"scrub_DVARS"`.

Value

A plot of the scrubbing results

summary.scrub_FD *Summarize a "scrub_FD" object*

Description

Summary method for class "scrub_FD"

Usage

```
## S3 method for class 'scrub_FD'
summary(object, ...)

## S3 method for class 'summary.scrub_FD'
print(x, ...)

## S3 method for class 'scrub_FD'
print(x, ...)
```

Arguments

object	Object of class "scrub_FD".
...	further arguments passed to or from other methods.
x	Object of class "scrub_FD".

Value

A plot of the scrubbing results

summary.scrub_projection *Summarize a "scrub_projection" object*

Description

Summary method for class "scrub_projection"

Usage

```
## S3 method for class 'scrub_projection'
summary(object, ...)

## S3 method for class 'summary.scrub_projection'
print(x, ...)

## S3 method for class 'scrub_projection'
print(x, ...)
```

Arguments

`object` Object of class `"scrub_projection"`.

`...` further arguments passed to or from other methods.

`x` Object of class `"scrub_projection"`.

Value

A plot of the scrubbing results

`unmask_vol` *Undo a volumetric mask*

Description

Un-applies a mask to vectorized data to yield its volumetric representation. The mask and data should have compatible dimensions: the number of rows in `dat` should equal the number of locations within the `mask`.

Usage

```
unmask_vol(dat, mask, fill = NA)
```

Arguments

`dat` Data matrix with locations along the rows and measurements along the columns. If only one set of measurements were made, this may be a vector.

`mask` Volumetric binary mask. `TRUE` indicates voxels inside the mask.

`fill` The value for locations outside the mask. Default: `NA`.

Value

The 3D or 4D unflattened volume array

Index