Package ‘hrf’

January 30, 2025

Type Package

Title Hemodynamic Response Function

Version 0.1.3

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

Description Computes the hemodynamic response function (HRF) for task functional magnetic resonance imaging (fMRI) data. Also includes functions for constructing a design matrix from task fMRI event timings, and for comparing multiple design matrices in a general linear model (GLM). A wrapper function is provided for GLM analysis of CIFTI-format data. Lastly, there are supporting functions which provide visual summaries of the HRFs and design matrices.

License GPL-3

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

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

Depends R (>= 3.6.0)

Imports car, ciftiTools (>= 0.15.0), fMRItools, Matrix, matrixStats, stats

Suggests covr, ggplot2, knitr, rmarkdown, testthat (>= 3.0.0), tidyr, spelling

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.3.2

Language en-US

NeedsCompilation no

Author Amanda Mejia [aut, cre],
Damon Pham [ctb] (<https://orcid.org/0000-0001-7563-4727>),
David Bolin [ctb],
Yu (Ryan) Yue [ctb],
Daniel Spencer [aut] (<https://orcid.org/0000-0002-9705-3605>),
Sarah Ryan [ctb]

Repository CRAN

Date/Publication 2025-01-30 19:10:02 UTC

Contents

`aic_Param` *aic*

Description

aic

Arguments

`aic`	(For prewhitening) Use the Akaike information criterion (AIC) to select AR model orders between `0` and `ar_order`? Default: `FALSE`.

`ar_order_Param` *ar_order*

Description

ar_order

Arguments

`ar_order`	(For prewhitening) The order of the autoregressive (AR) model to use for prewhitening. If `0`, do not prewhiten. Default: `6`. For multi-session modeling, note that a single AR model is used; its coefficients will be the average estimate from each session.

`ar_smooth_Param` *ar_smooth*

Description

ar_smooth

Arguments

`ar_smooth`	(For prewhitening) The FWHM parameter for spatially smoothing the coefficient estimates for the AR model to use for prewhitening. Recall that $\sigma = \frac{FWHM}{2*sqrt(2*log(2)}$. Set to `0` to not smooth the estimates. Default: `5`.

`BOLD_Param_BayesGLM` *BOLD*

Description

BOLD

Arguments

`BOLD`	fMRI timeseries data in CIFTI format ("*.dtseries.nii"). For single-session analysis this can be a file path to a CIFTI file or a `"xifti"` object from the `ciftiTools` package. For multi-session analysis this can be a vector of file paths or a list of `"xifti"` objects. If `BOLD` is a `"xifti"` object(s), the surfaces, if any, will be used for the spatial model. However, if `surfL` and `surfR` are provided, they will override any surfaces in `BOLD`.

`brainstructures_Param_BayesGLM` *brainstructures*

Description

brainstructures

Arguments

`brainstructures`	Character vector indicating which brain structure(s) of `BOLD` to analyze: `"left"` cortex; `"right"` cortex; and/or `"subcortical"` structures. Or `"all"` to model all three. Default: `c("left","right")` (cortex only).

`cderiv` *Central derivative*

Description

Take the central derivative of numeric vectors by averaging the forward and backward differences.

Usage

```
cderiv(x)
```

Arguments

`x`	A numeric matrix, or a vector which will be converted to a single-column matrix.

Value

A matrix or vector the same dimensions as `x`, with the derivative taken for each column of `x`. The first and last rows may need to be deleted, depending on the application.

Examples

```
x <- cderiv(seq(5))
stopifnot(all(x == c(.5, 1, 1, 1, .5)))
```

`Connectome_Workbench_Description` *Connectome Workbench*

Description

Connectome Workbench

Connectome Workbench Requirement

This function uses a system wrapper for the 'wb_command' executable. The user must first download and install the Connectome Workbench, available from https://www.humanconnectome.org/software/get-connectome-workbench .

`design_Param_BayesGLM` *design*

Description

design

Arguments

`design`	A numeric matrix or `data.frame`, or a `"BayesfMRI_design"` object from `make_design`. Can also be an array where the third dimension is the same length as the number of data locations, to model each location with its own design.

`do_QC` *Mask out invalid data*

Description

Mask out data locations that are invalid (missing data, low mean, or low variance) for any session.

Usage

```
do_QC(BOLD, meanTol = 1e-06, varTol = 1e-06, verbose = TRUE)
```

Arguments

`BOLD`	A session-length list of $T \times V$ numeric BOLD data.
`meanTol`, `varTol`	Tolerance for mean and variance of each data location. Locations which do not meet these thresholds are masked out of the analysis. Defaults: `1e-6`.
`verbose`	Print messages counting how many locations are removed? Default: `TRUE`.

Value

A logical vector indicating locations that are valid across all sessions.

Examples

```
nT <- 30
nV <- 400
BOLD1 <- matrix(rnorm(nT*nV), nrow=nT)
BOLD1[,seq(30,50)] <- NA
BOLD2 <- matrix(rnorm(nT*nV), nrow=nT)
BOLD2[,65] <- BOLD2[,65] / 1e10
BOLD <- list(sess1=BOLD1, sess2=BOLD2)
do_QC(BOLD)
```

`faces_Param` *faces*

Description

faces

Arguments

`faces`	An $F \times 3$ matrix, where each row contains the vertex indices for a given triangular face in the mesh. F is the number of faces in the mesh.

`field_names_Param` *field_names*

Description

field_names

Arguments

`field_names`	(Optional) Names of fields represented in design matrix.

`hpf_Param_BayesGLM` *hpf*

Description

hpf

Arguments

`hpf`	Add DCT bases to `nuisance` to apply a temporal high-pass filter to the data, for detrending? `hpf` is the filter frequency. Use `NULL` to skip detrending. Detrending is strongly recommended for fMRI data, to help reduce the autocorrelation in the residuals, so `NULL` will induce a warning. Use `"already"` to disable the warning while skipping highpass filtering. Using at least two DCT bases is as sufficient for detrending as using linear and quadratic drift terms in the nuisance matrix. So if DCT detrending is being used here, there is no need to add linear and quadratic drift terms to `nuisance`.

`HRF96` *Canonical (double-gamma) HRF (old one from SPM96, Glover)*

Description

Calculate the HRF from a time vector and parameters. Optionally compute the first or second derivative of the HRF instead.

Usage

```
HRF96(t, deriv = 0, a1 = 6, b1 = 0.9, a2 = 12, b2 = 0.9, c = 0.35)
```

Arguments

`t`	time vector
`deriv`	`0` (default) for the HRF, `1` for the first derivative of the HRF, or `2` for the second derivative of the HRF.
`a1`	delay of response. Default: `6`
`b1`	response dispersion. Default: `0.9`
`a2`	delay of undershoot. Default: `12`
`b2`	dispersion of undershoot. Default: `0.9`
`c`	scale of undershoot. Default: `0.35`

Value

HRF vector (or dHRF, or d2HRF) corresponding to time

Examples

```
upsample <- 100
HRF96(seq(0, 30, by=1/upsample))
```

`HRF_calc` *Canonical HRF and Derivatives*

Description

Calculate the HRF from a time vector and parameters, or its derivative with respect to delay or dispersion.

Usage

```
HRF_calc(
  t,
  deriv = 0,
  a1 = 6,
  b1 = 1,
  a2 = 16/6 * a1 * sqrt(b1),
  b2 = b1,
  c = 1/6,
  o = 0
)
```

Arguments

`t`	time vector (in units of seconds)
`deriv`	`0` (default) for the HRF, `1` for the delay derivative of the HRF, or `2` for the dispersion derivative of the HRF.
`a1`	delay of response. Default: `6`
`b1`	response dispersion. Default: `1`
`a2`	delay of undershoot. Default: `16/6 * a1 * sqrt(b1) = 16`
`b2`	dispersion of undershoot. Default: `b1 = 1`
`c`	scale of undershoot. Default: `1/6`
`o`	onset of response. Default: `0`

Value

HRF vector (or dHRF, or d2HRF) corresponding to time vector t

Examples

```
samples_per_sec <- 200
nsec <- 50
HRF_calc(seq(nsec*samples_per_sec)/samples_per_sec)
```

`HRF_main` *Canonical (double-gamma) HRF*

Description

Calculate the HRF from a time vector and parameters. Optionally compute the first or second derivative of the HRF instead. Form of HRF is similar to SPM but here the response and undershoot are scaled so the difference of the HRFs peaks at 1 and -c

Usage

```
HRF_main(t, a1 = 6, b1 = 1, a2 = NULL, b2 = NULL, c = 1/6, o = 0)
```

Arguments

`t`	time vector (in seconds). Must be equally spaced.
`a1`	delay of response. Default: `6`
`b1`	response dispersion. Default: `1`
`a2`	delay of undershoot. Default: `16/6*a1 = 16`
`b2`	dispersion of undershoot. Default: `b1 = 1`
`c`	scale of undershoot. Default: `1/6`
`o`	onset of response (in seconds). Default: `0`

Value

HRF vector corresponding to time vector t

Examples

```
upsample <- 100
HRF_main(seq(0, 30, by=1/upsample))
```

`make_design` *Make design matrix*

Description

Make the design matrix for the GLM, from the task information.

Usage

```
make_design(
  EVs,
  nTime,
  TR,
  dHRF = 0,
  upsample = 100,
  onset = NULL,
  offset = NULL,
  scale_design = TRUE,
  onsets_sep = FALSE,
  offsets_sep = FALSE,
  verbose = TRUE,
  ...
)
```

Arguments

`EVs`	The explanatory variables i.e. the task stimulus information, from which a design matrix will be constructed. This is a list where each entry represents a task as a matrix of onsets (first column) and durations (second column) for each stimuli (each row) of the task, in seconds. List names should be the task names. `nTime` and `TR` are required. An example of a properly-formatted `EVs` is: `on_s1 <- list(taskA=cbind(on=c(1,9,17), dr=rep(1,3)), taskB=cbind(on=c(3,27), dr=rep(5,2)))`. In this example, there are two tasks: the first has three 1s-long stimuli, while the second has two 5s-long stimuli.
`nTime`	the number of timepoints (volumes) in the task fMRI data.
`TR`	the temporal resolution of the data, in seconds.
`dHRF`	Controls the extent of HRF derivatives modeling. Set to `0` to only model the main HRF regressor (default), and not include its derivatives; set to `1` to model the temporal derivative too; or, set to `2` to model both the temporal and dispersion derivatives. If `dHRF==0`, there is one design column (field) per task. If `dHRF==1`, there are two fields per task. And if `dHRF==2`, there are three fields per task. If there are several tasks and `dHRF>0`, the total number of design matrix columns may exceed five, which may require large computation times with INLA. The analysis can be adjusted by modeling the derivatives as nuisance signals rather than as fields. To do so, move the corresponding columns from the design matrix to the `nuisance` argument for `BayesGLM`.
`upsample`	Upsample factor for convolving stimulus boxcar or stick function with canonical HRF. Default: `100`.
`onset`, `offset`	Add task regressors indicating the onset and/or offset of each event block? Provide the names of the tasks as a character vector. All onsets (or offsets) across the specified tasks will be represented by one additional column in the design matrix. The task names must match the names of `EVs`. Can also be `"all"` to use all tasks.

	Onsets/offset modeling is only compatible with a block design experiment. An error will be raised if the events in `EVs` do not have duration greater than one second.
`scale_design`	Scale the columns of the design matrix? Default: `TRUE`.
`onsets_sep`, `offsets_sep`	Model the onsets (`onsets_sep`) or offsets (`offsets_sep`) separately for each task? Default: `FALSE`, to model all onsets together, or all offsets together, as a single field in the design.
`verbose`	Print diagnostic messages? Default: `TRUE`.
`...`	Additional arguments to `HRF_calc`.

Value

A `"BfMRI_design"` object: a list with elements

design The volumes by fields design matrix. Column names are field names.

field_names The name of each task from the provided onsets.

dHRF The input `dHRF` parameter.

HRF_info Additional HRF modeling results.

Examples

```
EVs <- list(taskA=cbind(on=c(1,9,17), dr=rep(1,3)), taskB=cbind(on=c(3,27), dr=rep(5,2)))
TR <- .72
nTime <- ceiling(65/TR)
make_design(EVs, nTime, TR)
```

`mask_Param_vertices` *mask: vertices*

Description

mask: vertices

Arguments

`mask`	A length V logical vector indicating if each vertex is within the input mask.

`mean_var_Tol_Param` *mean and variance tolerance*

Description

mean and variance tolerance

Arguments

`meanTol`, `varTol`	Tolerance for mean and variance of each data location. Locations which do not meet these thresholds are masked out of the analysis. Default: `1e-6` for both.

`multiGLM` *multiGLM for CIFTI*

Description

Performs classical Bayesian GLM for task fMRI activation with CIFTI-format data, evaluating multiple design matrices. Includes the pre-processing steps of nuisance regression. Supports single-session analysis only.

Usage

```
multiGLM(
  BOLD,
  design,
  brainstructures = c("left", "right"),
  TR = NULL,
  resamp_res = 10000,
  hpf = NULL,
  nuisance = NULL,
  design_canonical = NULL,
  verbose = 1,
  meanTol = 1e-06,
  varTol = 1e-06
)
```

Arguments

`BOLD`	fMRI timeseries data in CIFTI format ("*.dtseries.nii"). For single-session analysis this can be a file path to a CIFTI file or a `"xifti"` object from the `ciftiTools` package. For multi-session analysis this can be a vector of file paths or a list of `"xifti"` objects. If `BOLD` is a `"xifti"` object(s), the surfaces, if any, will be used for the spatial model. However, if `surfL` and `surfR` are provided, they will override any surfaces in `BOLD`.

`design`	A 3D numeric array that is locations by fields by designs.
`brainstructures`	Character vector indicating which brain structure(s) of `BOLD` to analyze: `"left"` cortex; `"right"` cortex; and/or `"subcortical"` structures. Or `"all"` to model all three. Default: `c("left","right")` (cortex only).
`TR`	Temporal resolution of the data, in seconds.
`resamp_res`	For cortex spatial model. The number of vertices to which each cortical surface should be resampled, or `NULL` to not resample. For computational feasibility, a value of `10000` (default) or lower is recommended for Bayesian spatial modeling. If `Bayes=FALSE`, `resamp_res` can be set to `NULL` for full-resolution classical modeling.
`hpf`	Add DCT bases to `nuisance` to apply a temporal high-pass filter to the data, for detrending? `hpf` is the filter frequency. Use `NULL` to skip detrending. Detrending is strongly recommended for fMRI data, to help reduce the autocorrelation in the residuals, so `NULL` will induce a warning. Use `"already"` to disable the warning while skipping highpass filtering. Using at least two DCT bases is as sufficient for detrending as using linear and quadratic drift terms in the nuisance matrix. So if DCT detrending is being used here, there is no need to add linear and quadratic drift terms to `nuisance`.
`nuisance`	(Optional) A $T \times N_{nuis}$ matrix of nuisance signals, where T is the number of timepoints and N is the number of nuisance signals, or a list of these for multi-session analysis. Nuisance signals are regressed from the fMRI data and design matrix prior to GLM computation. Nuisance signals can include motion regressors, HRF derivatives not being modeled as tasks, and other sources of noise. Detrending/high-pass filtering is accomplished by adding DCT bases to the nuisance matrix; see the parameters `hpf` and `DCT`. Do not add spike regressors for scrubbing to the `nuisance` matrix. Rather, provide these in `scrub` so that their corresponding timepoints are also removed from the BOLD data after nuisance regression.
`design_canonical`	TO DO
`verbose`	`1` (default) to print occasional updates during model computation; `2` for occasional updates as well as running INLA in verbose mode (if `Bayes`), or `0` for no printed updates.
`meanTol`, `varTol`	Tolerance for mean and variance of each data location. Locations which do not meet these thresholds are masked out of the analysis. Default: `1e-6` for both.

Value

An object of class `"mGLM"`: a list with elements

brainstructures `data.frame` summarizing the spatial features of each brain structure modeled.

fields `data.frame` with the `name`, related `task`, and `HRF_order` of each field.

Connectome Workbench Requirement

This function uses a system wrapper for the 'wb_command' executable. The user must first download and install the Connectome Workbench, available from https://www.humanconnectome.org/software/get-connectome-workbench .

`multiGLM_fun` *multiGLM0*

Description

Performs classical GLM for task fMRI activation, comparing multiple designs

Usage

```
multiGLM_fun(
  BOLD,
  design,
  nuisance = NULL,
  design_canonical = NULL,
  verbose = 1,
  meanTol = 1e-06,
  varTol = 1e-06
)
```

Arguments

`BOLD`, `design`, `nuisance`	Session-length list of numeric matrices/arrays, each with volumes along the first dimension.
`design_canonical`	TO DO
`verbose`	`1` (default) to print occasional updates during model computation; `2` for occasional updates as well as running INLA in verbose mode (if `Bayes`), or `0` for no printed updates.
`meanTol`, `varTol`	Tolerance for mean, variance and SNR of each data location. Locations which do not meet these thresholds are masked out of the analysis. Default: `1e-6` for mean and variance, `50` for SNR.

Value

A list with elements

bestmodel ...

Fstat ...

pvalF ...

`nuisance_Param_BayesGLM` *nuisance*

Description

nuisance

Arguments

`nuisance`	(Optional) A $T \times N_{nuis}$ matrix of nuisance signals, where T is the number of timepoints and N is the number of nuisance signals, or a list of these for multi-session analysis. Nuisance signals are regressed from the fMRI data and design matrix prior to GLM computation. Nuisance signals can include motion regressors, HRF derivatives not being modeled as tasks, and other sources of noise. Detrending/high-pass filtering is accomplished by adding DCT bases to the nuisance matrix; see the parameters `hpf` and `DCT`. Do not add spike regressors for scrubbing to the `nuisance` matrix. Rather, provide these in `scrub` so that their corresponding timepoints are also removed from the BOLD data after nuisance regression.

`plot.BfMRI_design` *S3 method: use `view_xifti` to plot a `"BGLM"` object*

Description

S3 method: use `view_xifti` to plot a `"BGLM"` object

Usage

```
## S3 method for class 'BfMRI_design'
plot(x, ...)
```

Arguments

`x`	An object of class "BfMRI_design".
`...`	Additional arguments to `plot_design`.

Value

Result of the call to `plot_design`

`plot_design` *Plot design matrix*

Description

Plot design matrix

Plot design with lineplot

Plot design with imageplot

Usage

```
plot_design(design, method = c("lineplot", "imageplot"), ...)

plot_design_line(
  design,
  colors = "Set1",
  linetype = "solid",
  linewidth = 0.7,
  alpha = 0.8
)

plot_design_image(design)
```

Arguments

`design`	The timepoints by fields design matrix or data.frame.
`method`	`"lineplot"` (default) or `"imageplot"`.
`...`	Additional arguments to `plot_design_line` or `plot_design_image`.
`colors`	The name of a ColorBrewer palette (see RColorBrewer::brewer.pal.info and colorbrewer2.org), the name of a viridisLite palette, or a character vector of colors. Default: `"Set1"`.
`linetype`, `linewidth`, `alpha`	Parameters for `ggplot2::geom_line`. Defaults: `"solid"` linetype, `0.7` linewidth and `0.8` alpha. `linetype` can also be a vector of options with length matching the number of fields in `design`.

Value

A ggplot

A ggplot

A ggplot

`resamp_res_Param_BayesGLM` *resamp_res*

Description

resamp_res

Arguments

`resamp_res`	For cortex spatial model. The number of vertices to which each cortical surface should be resampled, or `NULL` to not resample. For computational feasibility, a value of `10000` (default) or lower is recommended for Bayesian spatial modeling. If `Bayes=FALSE`, `resamp_res` can be set to `NULL` for full-resolution classical modeling.

`scale_BOLD_Param` *scale_BOLD*

Description

scale_BOLD

Arguments

`scale_BOLD`	Controls scaling the BOLD response at each location. **"mean":** Scale the data to percent local signal change. **"sd":** Scale the data by local standard deviation. **"none":** Center the data but do not scale it.

`scrub_Param_BayesGLM` *scrub*

Description

scrub

Arguments

`scrub`	(Optional) A $T \times N_{scrub}$ matrix of spike regressors (one 1 value at the timepoint to scrub, and 0 for all other values), or a logical vector indicating the timepoints to scrub (`TRUE` to scrub, and `FALSE` to keep). For multi-session data, a session-length list of such matrices or logical vectors. The spike regressors will be included in the nuisance regression, and afterwards the timepoints indicated in `scrub` will be removed from the BOLD data and design matrix.

`session_names_Param` *session_names*

Description

session_names

Arguments

`session_names`	The names of the task-fMRI `BOLD` sessions, for multi-session analysis. If not provided here, will be inferred from `names(BOLD)`, inferred from `names(design)`, or generated automatically, in that order.

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

Description

Summary method for class `"BfMRI_design"`

Usage

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

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

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

Arguments

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

Value

A `"summary.BfMRI_design"` object, a list summarizing the properties of `object`.

`NULL`, invisibly.

`NULL`, invisibly.

`surfaces_Param_BayesGLM` *surfaces*

Description

surfaces

Arguments

`surfL`, `surfR`	For cortex spatial model. Left and right cortex surface geometry in GIFTI format ("*.surf.gii"). These can be a file path to a GIFTI file or a `"surf"` object from `ciftiTools`. Surfaces can alternatively be provided through the `$surf` metadata in `BOLD` if it is `"xifti"` data. If neither are provided, by default the HCP group-average fs_LR inflated surfaces included in `ciftiTools` will be used for the cortex spatial model.

`TR_Param_BayesGLM` *TR*

Description

TR

Arguments

`TR`	Temporal resolution of the data, in seconds.

`verbose_Param` *verbose*

Description

verbose

Arguments

`verbose`	`1` (default) to print occasional updates during model computation; `2` for occasional updates as well as running INLA in verbose mode (if `Bayes`), or `0` for no printed updates.

Index