Package 'BayesfMRI'

December 18, 2023

Type Package

Title Spatial Bayesian Methods for Task Functional MRI Studies

Version 0.3.11

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

Description Performs a spatial Bayesian general linear model (GLM) for task
functional magnetic resonance imaging (fMRI) data on the cortical surface.
Additional models include group analysis and inference to detect thresholded
areas of activation. Includes direct support for the 'CIFTI' neuroimaging
file format. For more information see A. F. Mejia, Y. R. Yue, D. Bolin, F.
Lindgren, M. A. Lindquist (2020) <doi:10.1080/01621459.2019.1611582> and D.
Spencer, Y. R. Yue, D. Bolin, S. Ryan, A. F. Mejia (2022)
<doi:10.1016/j.neuroimage.2022.118908>.

Depends R (>= 3.6.0)

License GPL-3

Additional_repositories https://inla.r-inla-download.org/R/testing

Encoding UTF-8

Imports ciftiTools (>= 0.8.0), excursions, foreach, fMRItools, MASS,
Matrix, matrixStats, methods, Rcpp, stats, sp, utils

Suggests covr, abind, dplyr, geometry, ggplot2, grDevices, INLA (>=
0.0-1468840039), knitr, MatrixModels, parallel, purrr, rdist,
rmarkdown, SQUAREM, testthat (>= 3.0.0)

RoxygenNote 7.2.3

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

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

LinkingTo RcppEigen, Rcpp

NeedsCompilation yes

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

        Sarah Ryan [ctb],
        Yu (Ryan) Yue [ctb]

R **topics documented:**

.findTheta   *Perform the EM algorithm of the Bayesian GLM fitting*

Description

Perform the EM algorithm of the Bayesian GLM fitting

Usage

```
.findTheta(theta, spde, y, X, QK, Psi, A, Ns, tol, verbose = FALSE)
```

Arguments

theta	the vector of initial values for theta
spde	a list containing the sparse matrix elements Cmat, Gmat, and GtCinvG
y	the vector of response values
X	the sparse matrix of the data values
QK	a sparse matrix of the prior precision found using the initial values of the hyperparameters
Psi	a sparse matrix representation of the basis function mapping the data locations to the mesh vertices
A	a precomputed matrix crossprod(X%*%Psi)
Ns	the number of columns for the random matrix used in the Hutchinson estimator
tol	a value for the tolerance used for a stopping rule (compared to the squared norm of the differences between theta(s) and theta(s-1))
verbose	(logical) Should intermediate output be displayed?

.getSqrtInvCpp	*Get the prewhitening matrix for a single data location*

Description

Get the prewhitening matrix for a single data location

Usage

```
.getSqrtInvCpp(AR_coeffs, nTime, avg_var)
```

Arguments

AR_coeffs	a length-p vector where p is the AR order
nTime	(integer) the length of the time series that is being prewhitened
avg_var	a scalar value of the residual variances of the AR model

---

.initialKP	*Find the initial values of kappa2 and phi*

Description

Find the initial values of kappa2 and phi

Usage

```
.initialKP(theta, spde, w, n_sess, tol, verbose)
```

Arguments

theta	a vector of length two containing the range and scale parameters kappa2 and phi, in that order
spde	a list containing the sparse matrix elements Cmat, Gmat, and GtCinvG
w	the beta_hat estimates for a single task
n_sess	the number of sessions
tol	the stopping rule tolerance
verbose	(logical) Should intermediate output be displayed?

.logDetQt *Find the log of the determinant of Q_tilde*

Description

Find the log of the determinant of Q_tilde

Usage

```
.logDetQt(kappa2, in_list, n_sess)
```

Arguments

kappa2   a scalar

in_list   a list with elements Cmat, Gmat, and GtCinvG

n_sess   the integer number of sessions

---

act_prevalance *Activations prevalence.*

Description

Activations prevalence.

Usage

```
act_prevalance(act_list)
```

Arguments

act_list   List of activations from [id_activations](). All should have the same sessions,
                fields, and brainstructures.

Value

A list containing the prevalances of activation, as a proportion of the results from act_list.

aic_Param	*aic*

Description

aic

Arguments

aic   Use the AIC to select AR model order between 0 and `ar_order`? Default: `FALSE`.

ar_order_Param	*ar_order*

Description

ar_order

Arguments

ar_order   (numeric) Controls prewhitening. If greater than zero, this should be a number indicating the order of the autoregressive model to use for prewhitening. If zero, do not prewhiten. Default: 6. For multi-session models, note that a single AR model is used; the parameters are estimated by averaging the estimates from each session.

ar_smooth_Param	*ar_smooth*

Description

ar_smooth

Arguments

ar_smooth   (numeric) FWHM parameter for smoothing the AR model coefficient estimates for prewhitening. Remember that $\sigma = \frac{FWHM}{2*sqrt(2*log(2))}$. Set to 0 or `NULL` to not do any smoothing. Default: 5.

BayesGLM   *BayesGLM*

Description

Performs spatial Bayesian GLM for fMRI task activation

Usage

```
BayesGLM(
  data,
  vertices = NULL,
  faces = NULL,
  mesh = NULL,
  mask = NULL,
  task_names = NULL,
  session_names = NULL,
  combine_sessions = TRUE,
  scale_BOLD = c("auto", "mean", "sd", "none"),
  scale_design = TRUE,
  Bayes = TRUE,
  ar_order = 6,
  ar_smooth = 5,
  aic = FALSE,
  num.threads = 4,
  return_INLA = c("trimmed", "full", "minimal"),
  verbose = 1,
  meanTol = 1e-06,
  varTol = 1e-06
)
```

Arguments

data
: A list of sessions in the "BfMRI.sess" object format. Each session is a list with elements "BOLD", "design", and optionally "nuisance". Each element should be a numeric matrix with T rows. The name of each element in data is the name of that session. See ?is.BfMRI.sess for details.

  Note that the argument session_names can be used instead of providing the session names as the names of the elements in data.

vertices, faces
: If Bayes, the geometry data can be provided with either both the vertices and faces arguments, or with the mesh argument.

  vertices is a $V \times 3$ matrix, where each row contains the Euclidean coordinates at which a given vertex in the mesh is located. V is the number of vertices in the mesh.

  faces is a $F \times 3$ 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.

mesh	If Bayes, the geometry data can be provided with either both the vertices and faces arguments, or with the mesh argument.
	mesh is an "inla.mesh" object. This can be created for surface data using make_mesh.
mask	(Optional) A length V logical vector indicating the vertices to include.
task_names	(Optional) Names of tasks represented in design matrix.
session_names	(Optional, and only relevant for multi-session modeling) Names of each session. Default: NULL. In BayesGLM this argument will overwrite the names of the list entries in data, if both exist.
combine_sessions	
	If multiple sessions are provided, should their data be combined and analyzed as a single session?
	If TRUE (default), the multiple sessions will be concatenated along time after scaling and nuisance regression, but before prewhitening. If FALSE, each session will be analyzed separately, except that a single estimate of the AR model coefficients for prewhitening is used, estimated across all sessions.
scale_BOLD	Option for scaling the BOLD response.
	"auto" (default) will use "mean" scaling except if demeaned data is detected (if any mean is less than one), in which case "sd" scaling will be used instead.
	"mean" scaling will scale the data to percent local signal change.
	"sd" scaling will scale the data by local standard deviation.
	"none" will only center the data, not scale it.
scale_design	Scale the design matrix by dividing each column by its maximum and then subtracting the mean? Default: TRUE. If FALSE, the design matrix is centered but not scaled.
Bayes	If TRUE (default), will fit a spatial Bayesian GLM in addition to the classical GLM. (The classical GLM is always returned.)
ar_order	(numeric) Controls prewhitening. If greater than zero, this should be a number indicating the order of the autoregressive model to use for prewhitening. If zero, do not prewhiten. Default: 6. For multi-session models, note that a single AR model is used; the parameters are estimated by averaging the estimates from each session.
ar_smooth	(numeric) FWHM parameter for smoothing the AR model coefficient estimates for prewhitening. Remember that $\sigma = \frac{FWHM}{2*sqrt(2*log(2))}$. Set to 0 or NULL to not do any smoothing. Default: 5.
aic	Use the AIC to select AR model order between 0 and ar_order? Default: FALSE.
num.threads	The maximum number of threads to use for parallel computations: prewhitening parameter estimation, and the inla-program model estimation. Default: 4. Note that parallel prewhitening requires the parallel package.
return_INLA	Return the INLA model object? (It can be large.) Use "trimmed" (default) to return only the more relevant results, which is enough for both id_activations and BayesGLM2, "minimal" to return just enough for BayesGLM2 but not id_activations, or "full" to return the full output of inla.

| verbose | Should updates be printed? Use 1 (default) for occasional updates, 2 for occasional updates as well as running INLA in verbose mode (if applicable), or 0 for no 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

A "BayesGLM" object: a list with elements

INLA_model_obj The full result of the call to INLA::inla.

task_estimates The task coefficients for the Bayesian model.

result_classical Results from the classical model: task estimates, task standard error estimates, residuals, degrees of freedom, and the mask.

mesh The model mesh including only the locations analyzed, i.e. within mask, without missing values, and meeting meanTol and varTol.

mesh_orig The original mesh provided.

mask A mask of mesh_orig indicating the locations inside mesh.

design The design matrix, after centering and scaling, but before any nuisance regression or prewhitening.

task_names The names of the tasks.

session_names The names of the sessions.

hyperpar_posteriors Hyperparameter posterior densities.

theta_estimates Theta estimates from the Bayesian model.

posterior_Sig_inv For joint group modeling.

mu_theta For joint group modeling.

Q_theta For joint group modeling.

y For joint group modeling: The BOLD data after any centering, scaling, nuisance regression, or prewhitening.

X For joint group modeling: The design matrix after any centering, scaling, nuisance regression, or prewhitening.

prewhiten_info Vectors of values across locations: phi (AR coefficients averaged across sessions), sigma_sq (residual variance averaged across sessions), and AIC (the maximum across sessions).

call match.call() for this function call.

INLA Requirement

This function requires the INLA package, which is not a CRAN package. See https://www.r-inla.org/download-install for easy installation instructions.

BayesGLM2   *Group-level Bayesian GLM*

Description

Performs group-level Bayesian GLM estimation and inference using the joint approach described in Mejia et al. (2020).

Usage

```
BayesGLM2(
  results,
  contrasts = NULL,
  quantiles = NULL,
  excursion_type = NULL,
  contrast_names = NULL,
  gamma = 0,
  alpha = 0.05,
  nsamp_theta = 50,
  nsamp_beta = 100,
  num_cores = NULL,
  verbose = 1
)

BayesGLM_group(
  results,
  contrasts = NULL,
  quantiles = NULL,
  excursion_type = NULL,
  gamma = 0,
  alpha = 0.05,
  nsamp_theta = 50,
  nsamp_beta = 100,
  num_cores = NULL,
  verbose = 1
)
```

Arguments

results   Either (1) a length N list of "BayesGLM" objects, or (2) a length N character
               vector of files storing "BayesGLM" objects saved with saveRDS.

contrasts   (Optional) A list of contrast vectors that specify the group-level summaries of
               interest. If NULL, use contrasts that compute the average of each field (task HRF)
               across subjects and sessions.

               Each contrast vector is length $K*S*N$ vector specifying a group-level summary
               of interest, where K is the number of fields (task HRFs), S is number of

sessions, and N is the number of subjects. For a single subject-session the contrast for the first field would be:

`contrast1 <- c(1, rep(0, K-1))`

and so the full contrast vector representing the group average across sessions and subjects for the first task would be:

`rep(rep(contrast1, S), N) /S /N.`

To obtain the group average for the first task, for just the first sessions from each subject:

`rep(c(contrast1, rep(0, K*(S-1))), N) /N.`

To obtain the mean difference between the first and second sessions, for the first task:

`rep(c(contrast1, -contrast1, rep(0, K-2)), N) /N.`

To obtain the mean across sessions of the first task, just for the first subject:

`c(rep(contrast1, S-1), rep(0, K*S*(N-1)) /S.`

quantiles	(Optional) Vector of posterior quantiles to return in addition to the posterior mean.
excursion_type	(For inference only) The type of excursion function for the contrast (">", "<", "!="), or a vector thereof (each element corresponding to one contrast). If NULL, no inference performed.
contrast_names	(Optional) Names of contrasts.
gamma	(For inference only) Activation threshold for the excursion set, or a vector thereof (each element corresponding to one contrast). Default: 0.
alpha	(For inference only) Significance level for activation for the excursion set, or a vector thereof (each element corresponding to one contrast). Default: .05.
nsamp_theta	Number of theta values to sample from posterior. Default: 50.
nsamp_beta	Number of beta vectors to sample conditional on each theta value sampled. Default: 100.
num_cores	The number of cores to use for sampling betas in parallel. If NULL (default), do not run in parallel.
verbose	Should updates be printed? Use 1 (default) for occasional updates, 2 for occasional updates as well as running INLA in verbose mode (if applicable), or 0 for no updates.

Value

A list containing the estimates, PPMs and areas of activation for each contrast.

INLA Requirement

This function requires the INLA package, which is not a CRAN package. See https://www.r-inla.org/download-install for easy installation instructions.

BayesGLM_cifti   *BayesGLM for CIFTI*

Description

Performs spatial Bayesian GLM on the cortical surface for fMRI task activation

Usage

```
BayesGLM_cifti(
  cifti_fname,
  surfL_fname = NULL,
  surfR_fname = NULL,
  brainstructures = c("left", "right"),
  design = NULL,
  onsets = NULL,
  TR = NULL,
  nuisance = NULL,
  dHRF = c(0, 1, 2),
  dHRF_as = c("auto", "nuisance", "task"),
  hpf = NULL,
  DCT = if (is.null(hpf)) {
      4
  } else {
      NULL
  },
  resamp_res = 10000,
  task_names = NULL,
  session_names = NULL,
  combine_sessions = TRUE,
  scale_BOLD = c("auto", "mean", "sd", "none"),
  scale_design = TRUE,
  Bayes = TRUE,
  ar_order = 6,
  ar_smooth = 5,
  aic = FALSE,
  num.threads = 4,
  return_INLA = c("trimmed", "full", "minimal"),
  verbose = 1,
  meanTol = 1e-06,
  varTol = 1e-06
)
```

Arguments

cifti_fname   fMRI timeseries data in CIFTI format ("*.dtseries.nii"). For single-session anal-
                 ysis 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.

surfL_fname   Left cortex surface geometry in GIFTI format ("*.surf.gii"). This can be a file path to a GIFTI file or a "surf" object from the ciftiTools package. This argument is only used if brainstructures includes "left" and Bayes==TRUE. If it's not provided, the HCP group-average inflated surface included in the ciftiTools package will be used.

surfR_fname   Right cortex surface geometry in GIFTI format ("*.surf.gii"). This can be a file path to a GIFTI file or a "surf" object from the ciftiTools package. This argument is only used if brainstructures includes "right" and Bayes==TRUE. If it's not provided, the HCP group-average inflated surface included in the ciftiTools package will be used.

brainstructures

Character vector indicating which brain structure(s) to analyze: "left" (left cortical surface) and/or "right" (right cortical surface). Default: c("left","right") (both hemispheres). Note that the subcortical models have not yet been implemented.

design, onsets, TR

Either provide design directly, or provide both onsets and TR from which the design matrix or matrices will be constructed.

design is a $T \times K$ task design matrix. Each column represents the expected BOLD response due to each task, a convolution of the hemodynamic response function (HRF) and the task stimulus. Note that the scale of the regressors will affect the scale and interpretation of the beta coefficients, so imposing a proper scale is recommended; see the scale_design argument, which by default is TRUE. Task names should be the column names, if not provided by the task_names argument. For multi-session modeling, this argument should be a list of such matrices. To model HRF derivatives, calculate the derivatives of the task columns beforehand (see the helper function [cderiv](#) which computes the discrete central derivative) and either add them to design to model them as tasks, or nuisance to model them as nuisance signals; it's recommended to then drop the first and last timepoints because the discrete central derivative doesn't exist at the time series boundaries. Do note that INLA computation times increase greatly if the design matrix has more than five columns, so it might be required to add these derivatives to nuisance rather than design.

onsets is an L-length list in which the name of each element is the name of the corresponding task, and the value of each element is a matrix of onsets (first column) and durations (second column) for each stimuli (each row) of the corresponding task. The units of both columns is seconds. For multi-session modeling, this argument should be a list of such lists. To model HRF derivatives, use the arguments dHRF and dHRF_as. If dHRF==0 or dHRF_as=="nuisance", the total number of columns in the design matrix, K, will equal L. If dHRF_as=="task", K will equal L times dHRF+1.

TR is the temporal resolution of the data, in seconds.

nuisance   (Optional) A $T \times J$ matrix of nuisance signals. These are regressed from the fMRI data and the design matrix prior to the GLM computation. For multi-session modeling, this argument should be a list of such matrices.

dHRF, dHRF_as	Only applicable if onsets and TR are provided. These arguments enable the modeling of HRF derivatives.
	Set dHRF to 1 to model the temporal derivatives of each task, 2 to add the second derivatives too, or 0 to not model the derivatives. Default: 1.
	If dHRF > 0, dHRF_as controls whether the derivatives are modeled as "nuisance" signals to regress out, "tasks", or "auto" (default) to treat them as tasks unless the total number of columns in the design matrix would exceed five.
hpf, DCT	Add DCT bases to nuisance to apply a temporal high-pass filter to the data? Only one of these arguments should be provided. hpf should be the filter frequency; if it is provided, TR must be provided too. The number of DCT bases to include will be computed to yield a filter with as close a frequency to hpf as possible. Alternatively, DCT can be provided to directly specify the number of DCT bases to include.
	Default: DCT=4. For typical TR, four DCT bases amounts to a lower frequency cutoff than the approximately .01 Hz used in most studies. We selected this default to err on the side of retaining more low-frequency information, but we recommend setting these arguments to values most appropriate for the data analysis at hand.
	Using at least two DCT bases is as sufficient as using linear and quadratic drift terms in the design matrix. So if DCT detrending is being used, there is no need to add linear and quadratic drift terms to nuisance.
resamp_res	The number of vertices to which each cortical surface should be resampled, or NULL to not resample. For computational feasibility, a value of 10000 or lower is recommended.
task_names	(Optional) Names of tasks represented in design matrix.
session_names	(Optional, and only relevant for multi-session modeling) Names of each session. Default: NULL. In [BayesGLM](#) this argument will overwrite the names of the list entries in data, if both exist.
combine_sessions	
	If multiple sessions are provided, should their data be combined and analyzed as a single session?
	If TRUE (default), the multiple sessions will be concatenated along time after scaling and nuisance regression, but before prewhitening. If FALSE, each session will be analyzed separately, except that a single estimate of the AR model coefficients for prewhitening is used, estimated across all sessions.
scale_BOLD	Option for scaling the BOLD response.
	"auto" (default) will use "mean" scaling except if demeaned data is detected (if any mean is less than one), in which case "sd" scaling will be used instead.
	"mean" scaling will scale the data to percent local signal change.
	"sd" scaling will scale the data by local standard deviation.
	"none" will only center the data, not scale it.
scale_design	Scale the design matrix by dividing each column by its maximum and then subtracting the mean? Default: TRUE. If FALSE, the design matrix is centered but not scaled.
Bayes	If TRUE (default), will fit a spatial Bayesian GLM in addition to the classical GLM. (The classical GLM is always returned.)

ar_order	(numeric) Controls prewhitening. If greater than zero, this should be a number indicating the order of the autoregressive model to use for prewhitening. If zero, do not prewhiten. Default: 6. For multi-session models, note that a single AR model is used; the parameters are estimated by averaging the estimates from each session.
ar_smooth	(numeric) FWHM parameter for smoothing the AR model coefficient estimates for prewhitening. Remember that $\sigma = \frac{FWHM}{2*sqrt(2*log(2))}$. Set to 0 or NULL to not do any smoothing. Default: 5.
aic	Use the AIC to select AR model order between 0 and ar_order? Default: FALSE.
num.threads	The maximum number of threads to use for parallel computations: prewhitening parameter estimation, and the inla-program model estimation. Default: 4. Note that parallel prewhitening requires the parallel package.
return_INLA	Return the INLA model object? (It can be large.) Use "trimmed" (default) to return only the more relevant results, which is enough for both id_activations and BayesGLM2, "minimal" to return just enough for BayesGLM2 but not id_activations, or "full" to return the full output of inla.
verbose	Should updates be printed? Use 1 (default) for occasional updates, 2 for occasional updates as well as running INLA in verbose mode (if applicable), or 0 for no 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 "BayesGLM_cifti": a list with elements

betas_Bayesian The task coefficients for the Bayesian model.

betas_classical The task coefficients for the classical model.

GLMs_Bayesian The entire list of GLM results, except for parameters estimated for the classical model.

GLMs_classical Parameters estimated for the classical model from the GLM.

session_names The names of the sessions.

n_sess_orig The number of sessions (before averaging, if applicable).

task_names The task part of the design matrix, after centering and scaling, but before any nuisance regression or prewhitening.

INLA latent fields limit

INLA computation times increase greatly when the number of columns in the design matrix exceeds five. So if there are more than five tasks, or three or more tasks each with its temporal derivative being modeled as a task, BayesGLM will raise a warning. In cases like the latter, we recommend modeling the temporal derivatives as nuisance signals using the nuisance argument, rather than modeling them as tasks.

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 .

INLA Requirement

This function requires the INLA package, which is not a CRAN package. See https://www.r-inla.org/download-install for easy installation instructions.

Bayes_Param   *Bayes*

Description

Bayes

Arguments

Bayes   If TRUE (default), will fit a spatial Bayesian GLM in addition to the classical
                       GLM. (The classical GLM is always returned.)

---

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)))
```

combine_sessions_Param

combine_sessions

Description

combine_sessions

Arguments

combine_sessions

If multiple sessions are provided, should their data be combined and analyzed as a single session?

If TRUE (default), the multiple sessions will be concatenated along time after scaling and nuisance regression, but before prewhitening. If FALSE, each session will be analyzed separately, except that a single estimate of the AR model coefficients for prewhitening is used, estimated across all sessions.

contrasts_Param   *contrasts*

Description

contrasts

Arguments

contrasts   List of contrast vectors to be passed to inla::inla.

emTol_Param   *emTol*

Description

emTol

Arguments

emTol   The stopping tolerance for the EM algorithm. Default: 1e-3.

EM_Param	*EM*

Description

EM

Arguments

EM	(logical) Should the EM implementation of the Bayesian GLM be used? Default: `FALSE`. This method is still in development.

---

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.

HRF	*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.

Usage

```
HRF(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

```
downsample <- 100
HRF(seq(0, 30, by=1/downsample))
```

---

id_activations   *Identify task activations*

Description

Identify areas of activation for each task from the result of BayesGLM or BayesGLM_cifti.

Usage

```
id_activations(
  model_obj,
  tasks = NULL,
  sessions = NULL,
  method = c("Bayesian", "classical"),
  alpha = 0.05,
  gamma = NULL,
  correction = c("FWER", "FDR", "none"),
  verbose = 1
)
```

Arguments

model_obj	Result of BayesGLM or BayesGLM_cifti model call, of class "BayesGLM" or "BayesGLM_cifti".
tasks	The task(s) to identify activations for. Give either the name(s) as a character vector, or the numerical indices. If NULL (default), analyze all tasks.
sessions	The session(s) to identify activations for. Give either the name(s) as a character vector, or the numerical indices. If NULL (default), analyze the first session.
Currently, if multiple sessions are provided, activations are identified separately for each session. (Information is not combined between the different sessions.)
method	"Bayesian" (default) or "classical". If model_obj does not have Bayesian results because Bayes was set to FALSE, only the "classical" method can be used.
alpha	Significance level. Default: 0.05.
gamma	Activation threshold, for example 1 for 1\ change if scale_BOLD=="mean" during model estimation. Setting a gamma is required for the Bayesian method; NULL (default) will use a gamma of zero for the classical method.

correction	For the classical method only: Type of multiple comparisons correction: ″FWER″ (Bonferroni correction, the default), ″FDR″ (Benjamini Hochberg), or ″none″.
verbose	Should updates be printed? Use 1 (default) for occasional updates, 2 for occasional updates as well as running INLA in verbose mode (if applicable), or 0 for no updates.

Value

An ″act_BayesGLM″ or ″act_BayesGLM_cifti″ object, a list which indicates the activated locations along with related information.

  INLA_Description   *INLA*

Description

INLA

INLA Requirement

This function requires the INLA package, which is not a CRAN package. See https://www.r-inla.org/download-install for easy installation instructions.

---

  is.BfMRI.sess   *Validate a* ″BfMRI.sess″ *object.*

Description

Check if object is valid for a ″BfMRI.sess″ object.

Usage

```
is.BfMRI.sess(x)
```

Arguments

x	The putative ″BfMRI.sess″ object.

Details

A ″BfMRI.sess″ object is a list of length S, where S is the number of sessions in the analysis. Each list entry corresponds to a separate session, and should itself be a list with these named fields:

BOLD a $T \times V$ BOLD matrix. Rows are time points; columns are data locations (vertices/voxels).

design a $T \times K$ matrix containing the K task regressors. See make_HRFs.

nuisance an optional argument. $T \times J$ matrix containing the L nuisance regressors.

In addition, all sessions must have the same number of data locations, V, and tasks, K.

Value

Logical. Is x a valid "BfMRI.sess" object?

Examples

```
nT <- 180
nV <- 700
BOLD1 <- matrix(rnorm(nT*nV), nrow=nT)
BOLD2 <- matrix(rnorm(nT*nV), nrow=nT)
onsets1 <- list(taskA=cbind(c(2,17,23),4)) # one task, 3 four sec-long stimuli
onsets2 <- list(taskA=cbind(c(1,18,25),4))
TR <- .72 # .72 seconds per volume, or (1/.72) Hz
duration <- nT # session is 180 volumes long (180*.72 seconds long)
design1 <- make_HRFs(onsets1, TR, duration)$design
design2 <- make_HRFs(onsets2, TR, duration)$design
x <- list(
 sessionOne = list(BOLD=BOLD1, design=design1),
 sessionTwo = list(BOLD=BOLD2, design=design2)
)
stopifnot(is.BfMRI.sess(x))
```

---

make_HRFs   *Make HRFs*

Description

Create HRF design matrix columns from onsets and durations

Usage

```
make_HRFs(
  onsets,
  TR,
  duration,
  dHRF = c(0, 1, 2),
  dHRF_as = c("auto", "nuisance", "task"),
  downsample = 100,
  verbose = FALSE
)
```

Arguments

onsets	*L*-length list in which the name of each element is the name of the corresponding task, and the value of each element is a matrix of onsets (first column) and durations (second column) for each stimuli (each row) of the corresponding task.
TR	Temporal resolution of the data, in seconds.
duration	The number of volumes in the fMRI data.

dHRF	Set to 1 to add the temporal derivative of each column in the design matrix, 2 to add the second derivatives too, or 0 to not add any columns. Default: 1.
dHRF_as	Only applies if dHRF > 0. Model the temporal derivatives as "nuisance" signals to regress out, "tasks", or "auto" to treat them as tasks unless the total number of columns in the design matrix (i.e. the total number of tasks, times dHRF+1), would be >=10, the limit for INLA.
downsample	Downsample factor for convolving stimulus boxcar or stick function with canonical HRF. Default: 100.
verbose	If applicable, print a message saying how the HRF derivatives will be modeled? Default: FALSE.

Value

List with the design matrix and/or the nuisance matrix containing the HRF-convolved stimuli as columns, depending on dHRF_as.

Examples

```
onsets <- list(taskA=cbind(c(2,17,23),4)) # one task, 3 four sec-long stimuli
TR <- .72 # .72 seconds per volume, or (1/.72) Hz
duration <- 300 # session is 300 volumes long (300*.72 seconds long)
make_HRFs(onsets, TR, duration)
```

---

make_mask	*Mask out invalid data*

Description

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

Usage

```
make_mask(data, meanTol = 1e-06, varTol = 1e-06, verbose = TRUE)
```

Arguments

data	A list of sessions, where each session is a list with elements BOLD, design, and optionally nuisance. See ?is.BfMRI.sess for details.
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
data <- list(sess1=list(BOLD=BOLD1, design=NULL), sess2=list(BOLD=BOLD2, design=NULL))
make_mask(data)
```

---

make_mesh   *Make Mesh*

Description

Make INLA triangular mesh from faces and vertices

Usage

```
make_mesh(vertices, faces, use_INLA = TRUE)
```

Arguments

vertices	A $V \times 3$ matrix, where each row contains the Euclidean coordinates at which a given vertex in the mesh is located. V is the number of vertices in the mesh
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.
use_INLA	(logical) Use the INLA package to make the mesh? Default: `TRUE`. Otherwise, mesh construction is based on an internal function, `galerkin_db`.

Value

INLA triangular mesh

INLA Requirement

This function requires the `INLA` package, which is not a CRAN package. See https://www.r-inla.org/download-install for easy installation instructions.

---

mask_Param_vertices   *mask: vertices*

Description

mask: vertices

Arguments

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

---

max.threads_Param   *max.threads*

Description

max.threads

Arguments

max.threads   The maximum number of threads to use in the inla-program for model estima-
                tion. 0 (default) will use the maximum number of threads allowed by the system.

---

mesh_Param_either   *mesh: either*

Description

mesh: either

Arguments

mesh   An "inla.mesh" object (see [make_mesh](#) for surface data)

---

mesh_Param_inla   *mesh: INLA only*

Description

mesh: INLA only

Arguments

mesh   An "inla.mesh" object (see [make_mesh](#) for surface data).

---

num.threads_Param *num.threads*

Description

num.threads

Arguments

num.threads   The maximum number of threads to use for parallel computations: prewhitening parameter estimation, and the inla-program model estimation. Default: 4. Note that parallel prewhitening requires the `parallel` package.

---

plot.act_BayesGLM_cifti

S3   method: use [view_xifti_surface](#) *to   plot   a*
`"act_BayesGLM_cifti"` *object*

Description

S3 method: use [view_xifti_surface](#) to plot a `"act_BayesGLM_cifti"` object

Usage

```
## S3 method for class 'act_BayesGLM_cifti'
plot(x, idx = NULL, session = NULL, ...)
```

Arguments

x   An object of class "act_BayesGLM_cifti"

idx   Which task should be plotted? Give the numeric indices or the names. `NULL` (default) will show all tasks. This argument overrides the `idx` argument to [view_xifti_surface](#).

session   Which session should be plotted? `NULL` (default) will use the first.

...   Additional arguments to [view_xifti_surface](#)

Value

Result of the call to `ciftiTools::view_cifti_surface`.

---

plot.BayesGLM2_cifti   *S3 method: use* `view_xifti_surface` *to plot a* `"BayesGLM2_cifti"`
                         object

Description

S3 method: use `view_xifti_surface` to plot a `"BayesGLM2_cifti"` object

Usage

```
## S3 method for class 'BayesGLM2_cifti'
plot(x, idx = NULL, what = c("contrasts", "activations"), ...)
```

Arguments

x	An object of class "BayesGLM2_cifti"
idx	Which contrast should be plotted? Give the numeric index. NULL (default) will show all contrasts. This argument overrides the idx argument to `view_xifti_surface`.
what	Estimates of the "contrasts" (default), or their thresholded "activations".
...	Additional arguments to `view_xifti_surface`

Value

Result of the call to `ciftiTools::view_cifti_surface`.

---

plot.BayesGLM_cifti   *S3 method: use* `view_xifti_surface` *to plot a* `"BayesGLM_cifti"`
                         object

Description

S3 method: use `view_xifti_surface` to plot a `"BayesGLM_cifti"` object

Usage

```
## S3 method for class 'BayesGLM_cifti'
plot(x, idx = NULL, session = NULL, method = NULL, zlim = c(-1, 1), ...)
```

Arguments

x	An object of class "BayesGLM_cifti"
idx	Which task should be plotted? Give the numeric indices or the names. NULL (default) will show all tasks. This argument overrides the idx argument to `view_xifti_surface`.
session	Which session should be plotted? NULL (default) will use the first.
method	"Bayes" or "classical". NULL (default) will use the Bayesian results if available, and the classical results if not.
zlim	Overrides the zlim argument for `view_xifti_surface`. Default: c(-1, 1).
...	Additional arguments to `view_xifti_surface`

Value

Result of the call to ciftiTools::view_cifti_surface.

---

plot.prev_BayesGLM_cifti

S3 method: use `view_xifti` to plot a "prev_BayesGLM_cifti" *object*

Description

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

Usage

```
## S3 method for class 'prev_BayesGLM_cifti'
plot(
  x,
  idx = NULL,
  session = NULL,
  drop_zeros = NULL,
  colors = "plasma",
  zlim = c(round(1/x$n_results - 0.005, 2), 1),
  ...
)
```

Arguments

x	An object of class "prev_BayesGLM_cifti"
idx	Which task should be plotted? Give the numeric indices or the names. NULL (default) will show all tasks. This argument overrides the idx argument to `view_xifti`.
session	Which session should be plotted? NULL (default) will use the first.

drop_zeros   Color locations without any activation across all results (zero prevalence) the
              same color as the medial wall? Default: NULL to drop the zeros if only one idx
              is being plotted.

colors, zlim See [view_xifti](). Here, the defaults are overrided to use the Viridis "plasma"
              color scale between 1/nA and 1, where nA is the number of results in x.

...   Additional arguments to [view_xifti]()

Value

Result of the call to ciftiTools::view_cifti_surface.

---

pw_estimate   *Estimate residual autocorrelation for prewhitening*

Description

Estimate residual autocorrelation for prewhitening

Usage

```
pw_estimate(resids, ar_order, aic = FALSE)
```

Arguments

resids   Estimated residuals

ar_order, aic   Order of the AR model used to prewhiten the data at each location. If !aic
                (default), the order will be exactly ar_order. If aic, the order will be between
                zero and ar_order, as determined by the AIC.

Value

Estimated AR coefficients and residual variance at every vertex

---

pw_smooth   *Smooth AR coefficients and white noise variance*

Description

Smooth AR coefficients and white noise variance

Usage

```
pw_smooth(vertices, faces, mask = NULL, AR, var, FWHM = 5)
```

Arguments

vertices	A $V \times 3$ matrix, where each row contains the Euclidean coordinates at which a given vertex in the mesh is located. V is the number of vertices in the mesh
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.
mask	A logical vector indicating, for each vertex, whether to include it in smoothing. NULL (default) will use a vector of all TRUE, meaning that no vertex is masked out; all are used for smoothing.
AR	A Vxp matrix of estimated AR coefficients, where V is the number of vertices and p is the AR model order
var	A vector length V containing the white noise variance estimates from the AR model
FWHM	FWHM parameter for smoothing. Remember that $\sigma = \frac{FWHM}{2*sqrt(2*log(2))}$. Set to 0 or NULL to not do any smoothing. Default: 5.#'

Value

Smoothed AR coefficients and residual variance at every vertex

---

return_INLA_Param	*return_INLA*

Description

return_INLA

Arguments

return_INLA	Return the INLA model object? (It can be large.) Use "trimmed" (default) to return only the more relevant results, which is enough for both id_activations and BayesGLM2, "minimal" to return just enough for BayesGLM2 but not id_activations, or "full" to return the full output of inla.

---

scale_BOLD_Param	*scale_BOLD*

Description

scale_BOLD

Arguments

scale_BOLD   Option for scaling the BOLD response.

"auto" (default) will use "mean" scaling except if demeaned data is detected (if any mean is less than one), in which case "sd" scaling will be used instead.

"mean" scaling will scale the data to percent local signal change.

"sd" scaling will scale the data by local standard deviation.

"none" will only center the data, not scale it.

---

scale_design_Param   *scale_design*

Description

scale_design

Arguments

scale_design   Scale the design matrix by dividing each column by its maximum and then subtracting the mean? Default: TRUE. If FALSE, the design matrix is centered but not scaled.

---

seed_Param   *seed*

Description

seed

Arguments

seed   Random seed (optional). Default: NULL.

---

session_names_Param   *session_names*

Description

session_names

Arguments

session_names   (Optional, and only relevant for multi-session modeling) Names of each session. Default: NULL. In [BayesGLM](#) this argument will overwrite the names of the list entries in data, if both exist.

summary.act_BayesGLM   *Summarize a* "act_BayesGLM" *object*

Description

Summary method for class "act_BayesGLM"

Usage

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

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

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

Arguments

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

Value

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

NULL, invisibly.

NULL, invisibly.

summary.act_BayesGLM_cifti
                              Summarize a "act_BayesGLM_cifti" *object*

Description

Summary method for class "act_BayesGLM_cifti"

Usage

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

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

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

Arguments

object	Object of class ″act_BayesGLM_cifti″.
...	further arguments passed to or from other methods.
x	Object of class ″summary.act_BayesGLM_cifti″.

Value

A ″summary.act_BayesGLM_cifti″ object, a list summarizing the properties of object.

NULL, invisibly.

NULL, invisibly.

---

summary.BayesGLM   *Summarize a* ″BayesGLM″ *object*

Description

Summary method for class ″BayesGLM″

Usage

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

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

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

Arguments

object	Object of class ″BayesGLM″.
...	further arguments passed to or from other methods.
x	Object of class ″summary.BayesGLM″.

Value

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

NULL, invisibly.

NULL, invisibly.

---

summary.BayesGLM2   *Summarize a* "BayesGLM2" *object*

Description

Summary method for class "BayesGLM2"

Usage

```
## S3 method for class 'BayesGLM2'
summary(object, ...)

## S3 method for class 'summary.BayesGLM2'
print(x, ...)

## S3 method for class 'BayesGLM2'
print(x, ...)
```

Arguments

object	Object of class "BayesGLM2".
...	further arguments passed to or from other methods.
x	Object of class "summary.BayesGLM2".

Value

A "summary.BayesGLM2" object, a list summarizing the properties of object.

NULL, invisibly.

NULL, invisibly.

summary.BayesGLM2_cifti

Summarize a "BayesGLM2_cifti" *object*

Description

Summary method for class "BayesGLM2_cifti"

Usage

```
## S3 method for class 'BayesGLM2_cifti'
summary(object, ...)

## S3 method for class 'summary.BayesGLM2_cifti'
print(x, ...)

## S3 method for class 'BayesGLM2_cifti'
print(x, ...)
```

Arguments

object	Object of class "BayesGLM2_cifti".
...	further arguments passed to or from other methods.
x	Object of class "summary.BayesGLM2_cifti".

Value

A "summary.BayesGLM2_cifti" object, a list summarizing the properties of object.

NULL, invisibly.

NULL, invisibly.

---

summary.BayesGLM_cifti

Summarize a "BayesGLM_cifti" *object*

Description

Summary method for class "BayesGLM_cifti"

Usage

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

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

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

Arguments

object	Object of class ″BayesGLM_cifti″.
...	further arguments passed to or from other methods.
x	Object of class ″summary.BayesGLM_cifti″.

Value

A ″summary.BayesGLM_cifti″ object, a list summarizing the properties of object.

NULL, invisibly.

NULL, invisibly.

---

summary.prev_BayesGLM *Summarize a* ″prev_BayesGLM″ *object*

Description

Summary method for class ″prev_BayesGLM″

Usage

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

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

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

Arguments

object	Object of class ″prev_BayesGLM″.
...	further arguments passed to or from other methods.
x	Object of class ″summary.prev_BayesGLM″.

Value

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

NULL, invisibly.

NULL, invisibly.

---

summary.prev_BayesGLM_cifti

Summarize a "prev_BayesGLM_cifti" *object*

Description

Summary method for class "prev_BayesGLM_cifti"

Usage

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

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

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

Arguments

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

Value

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

NULL, invisibly.

NULL, invisibly.

---

task_names_Param   *task_names*

Description

task_names

Arguments

task_names	(Optional) Names of tasks represented in design matrix.

---

trim_INLA_Param *trim_INLA*

Description

trim_INLA

Arguments

trim_INLA   (logical) should the INLA_model_obj within the result be trimmed to only what is necessary to use id_activations? Default: TRUE.

---

verbose_Param *verbose*

Description

verbose

Arguments

verbose   Should updates be printed? Use 1 (default) for occasional updates, 2 for occasional updates as well as running INLA in verbose mode (if applicable), or 0 for no updates.

---

vertex_areas *Surface area of each vertex*

Description

Compute surface areas of each vertex in a triangular mesh.

Usage

```
vertex_areas(mesh)
```

Arguments

mesh   An "inla.mesh" object (see [make_mesh](#) for surface data).

Value

Vector of areas

INLA Requirement

This function requires the INLA package, which is not a CRAN package. See https://www.r-inla.org/download-install for easy installation instructions.

---

vertices_Param   *vertices*

Description

vertices

Arguments

vertices   A $V \times 3$ matrix, where each row contains the Euclidean coordinates at which a given vertex in the mesh is located. V is the number of vertices in the mesh

Index