

# Package ‘BayesfMRI’

June 8, 2023

**Type** Package

**Title** Spatial Bayesian Methods for Task Functional MRI Studies

**Version** 0.3.5

**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](https://doi.org/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](https://doi.org/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]

**Repository** CRAN

**Date/Publication** 2023-06-08 17:52:54 UTC

## R topics documented:

.findTheta . . . . .	3
.getSqrtInvCpp . . . . .	4
.initialKP . . . . .	4
.logDetQt . . . . .	5
aic_Param . . . . .	5
ar_order_Param . . . . .	5
ar_smooth_Param . . . . .	6
BayesGLM . . . . .	6
BayesGLM2 . . . . .	9
BayesGLM_cifti . . . . .	11
Bayes_Param . . . . .	15
cderiv . . . . .	15
combine_sessions_Param . . . . .	16
contrasts_Param . . . . .	16
emTol_Param . . . . .	17
EM_Param . . . . .	17
faces_Param . . . . .	17
HRF . . . . .	18
id_activations . . . . .	18
INLA_Description . . . . .	20
is.BfMRI.sess . . . . .	20
make_HRFs . . . . .	21
make_mask . . . . .	22
make_mesh . . . . .	23
mask_Param_vertices . . . . .	24
max.threads_Param . . . . .	24
mesh_Param_either . . . . .	24
mesh_Param_inla . . . . .	24
num.threads_Param . . . . .	25
plot.act_BayesGLM_cifti . . . . .	25
plot.BayesGLM2_cifti . . . . .	26
plot.BayesGLM_cifti . . . . .	26
pw_estimate . . . . .	27
pw_smooth . . . . .	28
return_INLA_Param . . . . .	28
scale_BOLD_Param . . . . .	29
scale_design_Param . . . . .	29
seed_Param . . . . .	29
session_names_Param . . . . .	30
summary.act_BayesGLM . . . . .	30

`.findTheta` 3

summary.act_BayesGLM_cifti . . . . .	31
summary.BayesGLM . . . . .	31
summary.BayesGLM2 . . . . .	32
summary.BayesGLM2_cifti . . . . .	33
summary.BayesGLM_cifti . . . . .	34
task_names_Param . . . . .	34
trim_INLA_Param . . . . .	35
verbose_Param . . . . .	35
vertex_areas . . . . .	35
vertices_Param . . . . .	36

**Index** 37

---

<code>.findTheta</code>	<i>Perform the EM algorithm of the Bayesian GLM fitting</i>
-------------------------	---

---

**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 hyper-parameters
- `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**

<code>AR_coeffs</code>	a length-p vector where p is the AR order
<code>nTime</code>	(integer) the length of the time series that is being prewhitened
<code>avg_var</code>	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**

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

---

.logDetQt	<i>Find the log of the determinant of Q_tilde</i>
-----------	---

---

**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

---

aic_Param	<i>aic</i>
-----------	------------

---

**Description**

aic

**Arguments**

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

---

ar_order_Param	<i>ar_order</i>
----------------	-----------------

---

**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	<i>ar_smooth</i>
-----------------	------------------

---

**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	<i>BayesGLM</i>
----------	-----------------

---

**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	<p>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 <math>T</math> rows. The name of each element in data is the name of that session. See <code>?is.BfMRI.sess</code> for details.</p> <p>Note that the argument <code>session_names</code> can be used instead of providing the session names as the names of the elements in data.</p>
vertices, faces	<p>If Bayes, the geometry data can be provided with either both the vertices and faces arguments, or with the mesh argument.</p> <p>vertices is a <math>V \times 3</math> matrix, where each row contains the Euclidean coordinates at which a given vertex in the mesh is located. <math>V</math> is the number of vertices in the mesh.</p> <p>faces is a <math>F \times 3</math> matrix, where each row contains the vertex indices for a given triangular face in the mesh. <math>F</math> is the number of faces in the mesh.</p>
mesh	<p>If Bayes, the geometry data can be provided with either both the vertices and faces arguments, or with the mesh argument.</p> <p>mesh is an "inla.mesh" object. This can be created for surface data using <code>make_mesh</code>.</p>
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	<p>If multiple sessions are provided, should their data be combined and analyzed as a single session?</p> <p>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.</p>
scale_BOLD	<p>Option for scaling the BOLD response.</p> <p>"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.</p> <p>"mean" scaling will scale the data to percent local signal change.</p> <p>"sd" scaling will scale the data by local standard deviation.</p> <p>"none" will only center the data, not scale it.</p>
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.
<code>ar_smooth</code>	(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.
<code>aic</code>	Use the AIC to select AR model order between 0 and <code>ar_order</code> ? Default: FALSE.
<code>num.threads</code>	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 <code>parallel</code> package.
<code>return_INLA</code>	Return the INLA model object? (It can be large.) Use "trimmed" (default) to return only the more relevant results, which is enough for both <code>id_activations</code> and BayesGLM2, "minimal" to return just enough for BayesGLM2 but not <code>id_activations</code> , or "full" to return the full output of inla.
<code>verbose</code>	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.
<code>meanTol, varTol</code>	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 modelling.

**mu\_theta** For joint group modelling.

**Q\_theta** For joint group modelling.



- y** For joint group modelling: The BOLD data after any centering, scaling, nuisance regression, or prewhitening.
- X** For joint group modelling: 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 <a href="#">saveRDS</a> .
contrasts	<p>(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.</p> <p>Each contrast vector is length <math>K * S * N</math> vector specifying a group-level summary of interest, where <math>K</math> is the number of fields (task HRFs), <math>S</math> is the number of sessions, and <math>N</math> is the number of subjects. For a single subject-session the contrast for the first field would be:</p> <pre>contrast1 &lt;- c(1, rep(0, K-1))</pre> <p>and so the full contrast vector representing the group average across sessions and subjects for the first task would be:</p> <pre>rep(rep(contrast1, S), N) / S / N.</pre> <p>To obtain the group average for the first task, for just the first sessions from each subject:</p> <pre>rep(c(contrast1, rep(0, K*(S-1))), N) / N.</pre> <p>To obtain the mean difference between the first and second sessions, for the first task:</p> <pre>rep(c(contrast1, -contrast1, rep(0, K-2)), N) / N.</pre> <p>To obtain the mean across sessions of the first task, just for the first subject:</p> <pre>c(rep(contrast1, S-1), rep(0, K*S*(N-1))) / S.</pre>
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	<i>BayesGLM for CIFTI</i>
----------------	---------------------------

---

**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 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.

**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.

<code>nuisance</code>	(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.
<code>dHRF, dHRF_as</code>	<p>Only applicable if <code>onsets</code> and <code>TR</code> are provided. These arguments enable the modeling of HRF derivatives.</p> <p>Set <code>dHRF</code> 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.</p> <p>If <code>dHRF &gt; 0</code>, <code>dHRF_as</code> 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.</p>
<code>hpf, DCT</code>	<p>Add DCT bases to nuisance to apply a temporal high-pass filter to the data? Only one of these arguments should be provided. <code>hpf</code> should be the filter frequency; if it is provided, <code>TR</code> must be provided too. The number of DCT bases to include will be computed to yield a filter with as close a frequency to <code>hpf</code> as possible. Alternatively, <code>DCT</code> can be provided to directly specify the number of DCT bases to include.</p> <p>Default: <code>DCT=4</code>. For typical <code>TR</code>, 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.</p> <p>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.</p>
<code>resamp_res</code>	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.
<code>task_names</code>	(Optional) Names of tasks represented in design matrix.
<code>session_names</code>	(Optional, and only relevant for multi-session modeling) Names of each session. Default: NULL. In <code>BayesGLM</code> this argument will overwrite the names of the list entries in <code>data</code> , if both exist.
<code>combine_sessions</code>	<p>If multiple sessions are provided, should their data be combined and analyzed as a single session?</p> <p>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.</p>
<code>scale_BOLD</code>	<p>Option for scaling the BOLD response.</p> <p>"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.</p>

	"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 <code>id_activations</code> and <code>BayesGLM2</code> , "minimal" to return just enough for <code>BayesGLM2</code> but not <code>id_activations</code> , 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	<i>combine_sessions</i>
------------------------	-------------------------

---

**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	<i>contrasts</i>
-----------------	------------------

---

**Description**

contrasts

**Arguments**

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



---

emTol_Param	<i>emTol</i>
-------------	--------------

---

**Description**

emTol

**Arguments**

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

---

EM_Param	<i>EM</i>
----------	-----------

---

**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	<i>faces</i>
-------------	--------------

---

**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,
  threshold = 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.
threshold	Activation threshold, for example 1 for 1\ change if scale_BOLD=="mean" during model estimation. Setting a threshold is required for the Bayesian method; NULL (default) will use a threshold 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" $T \times V$  BOLD matrix. Rows are time points; columns are data locations (vertices/voxels).
- "design" $T \times K$  matrix containing the  $K$  task regressors.
- "nuisance"Optional.  $T \times J$  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	<i>L</i> -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	<i>Mask out invalid data</i>
-----------	------------------------------

---

**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 = FALSE)
```

**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: FALSE. Otherwise, mesh construction is based on an internal function, <code>galerkin_db</code> .

**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 estimation. 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	<i>num.threads</i>
-------------------	--------------------

---

**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\_xiffti\_surface to plot a "BayesGLM2_cifti"
                      object
```

---

### Description

S3 method: use [view\\_xiffti\\_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 <a href="#">view_xiffti_surface</a> .
what	Estimates of the "contrasts" (default), or their thresholded "activations".
...	Additional arguments to <a href="#">view_xiffti_surface</a>

### Value

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

---

```
plot.BayesGLM_cifti  S3 method: use view\_xiffti\_surface to plot a "BayesGLM_cifti"
                      object
```

---

### Description

S3 method: use [view\\_xiffti\\_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 <a href="#">view_xifti_surface</a> .
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 <a href="#">view_xifti_surface</a> . Default: c(-1, 1).
...	Additional arguments to <a href="#">view_xifti_surface</a>

**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 <code>!aic</code> (default), the order will be exactly <code>ar_order</code> . If <code>aic</code> , the order will be between zero and <code>ar_order</code> , 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 $V \times p$ 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 <a href="#">id_activations</a> and <a href="#">BayesGLM2</a> , "minimal" to return just enough for <a href="#">BayesGLM2</a> but not <a href="#">id_activations</a> , or "full" to return the full output of inla.
-------------	--

---

scale_BOLD_Param	<i>scale_BOLD</i>
------------------	-------------------

---

**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	<i>scale_design</i>
--------------------	---------------------

---

**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	<i>seed</i>
------------	-------------

---

**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	<i>Summarize a "BayesGLM2" object</i>
-------------------	---------------------------------------

---

**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.

---

```
task_names_Param
```

*task\_names*

---

### Description

task\_names

### Arguments

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

---

trim_INLA_Param	<i>trim_INLA</i>
-----------------	------------------

---

**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	<i>verbose</i>
---------------	----------------

---

**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	<i>Surface area of each vertex</i>
--------------	------------------------------------

---

**Description**

Compute surface areas of each vertex in a triangular mesh.

**Usage**

vertex\_areas(mesh)

**Arguments**

mesh	An "inla.mesh" object (see <a href="#">make_mesh</a> 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	<i>vertices</i>
----------------	-----------------

---

**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

.findTheta, 3  
.getSqrtInvCpp, 4  
.initialKP, 4  
.logDetQt, 5

aic\_Param, 5  
ar\_order\_Param, 5  
ar\_smooth\_Param, 6

Bayes\_Param, 15  
BayesGLM, 6, 7, 13, 30  
BayesGLM2, 8, 9, 14, 28  
BayesGLM\_cifti, 11  
BayesGLM\_group (BayesGLM2), 9

cderiv, 12, 15  
combine\_sessions\_Param, 16  
contrasts\_Param, 16

EM\_Param, 17  
emTol\_Param, 17

faces\_Param, 17

HRF, 18

id\_activations, 8, 14, 18, 28  
INLA\_Description, 20  
is.BfMRI.sess, 20

make\_HRFs, 21  
make\_mask, 22  
make\_mesh, 7, 23, 24, 35  
mask\_Param\_vertices, 24  
max\_threads\_Param, 24  
mesh\_Param\_either, 24  
mesh\_Param\_inla, 24

num\_threads\_Param, 25

plot.act\_BayesGLM\_cifti, 25  
plot.BayesGLM2\_cifti, 26  
plot.BayesGLM\_cifti, 26  
print.act\_BayesGLM  
    (summary.act\_BayesGLM), 30  
print.act\_BayesGLM\_cifti  
    (summary.act\_BayesGLM\_cifti),  
    31  
print.BayesGLM (summary.BayesGLM), 31  
print.BayesGLM2 (summary.BayesGLM2), 32  
print.BayesGLM2\_cifti  
    (summary.BayesGLM2\_cifti), 33  
print.BayesGLM\_cifti  
    (summary.BayesGLM\_cifti), 34  
print.summary.act\_BayesGLM  
    (summary.act\_BayesGLM), 30  
print.summary.act\_BayesGLM\_cifti  
    (summary.act\_BayesGLM\_cifti),  
    31  
print.summary.BayesGLM  
    (summary.BayesGLM), 31  
print.summary.BayesGLM2  
    (summary.BayesGLM2), 32  
print.summary.BayesGLM2\_cifti  
    (summary.BayesGLM2\_cifti), 33  
print.summary.BayesGLM\_cifti  
    (summary.BayesGLM\_cifti), 34  
pw\_estimate, 27  
pw\_smooth, 28

return\_INLA\_Param, 28

saveRDS, 10  
scale\_BOLD\_Param, 29  
scale\_design\_Param, 29  
seed\_Param, 29  
session\_names\_Param, 30  
summary.act\_BayesGLM, 30  
summary.act\_BayesGLM\_cifti, 31  
summary.BayesGLM, 31  
summary.BayesGLM2, 32

summary.BayesGLM2\_cifti, [33](#)  
summary.BayesGLM\_cifti, [34](#)

task\_names\_Param, [34](#)  
trim\_INLA\_Param, [35](#)

verbose\_Param, [35](#)  
vertex\_areas, [35](#)  
vertices\_Param, [36](#)  
view\_xifti\_surface, [25–27](#)