Package ‘BayesfMRI’

March 8, 2025

Type Package

Title Spatial Bayesian Methods for Task Functional MRI Studies

Version 0.10.1

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>.

License GPL-3

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

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

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

Depends R (>= 3.6.0)

Imports car, ciftiTools (>= 0.17.2), excursions, foreach, fMRItools (>= 0.5.3), MASS, Matrix, matrixStats, methods, Rcpp, stats, sp, utils, viridisLite

Suggests covr, abind, grDevices, hrf, INLA (>= 0.0-1468840039), knitr, MatrixModels, parallel, purrr, rmarkdown, SQUAREM, testthat (>= 3.0.0), spelling

Encoding UTF-8

RoxygenNote 7.3.2

LinkingTo RcppEigen, Rcpp

Language en-US

NeedsCompilation yes

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

Repository CRAN

Date/Publication 2025-03-07 23:50:02 UTC

Contents

`BayesfMRI-package` *BayesfMRI: Spatial Bayesian Methods for Task Functional MRI Studies*

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.

Author(s)

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

Authors:

- Daniel Spencer <`danieladamspencer@gmail.com`> (ORCID)

Other contributors:

- Damon Pham <`damondpham@gmail.com`> (ORCID) [contributor]
- David Bolin <`david.bolin@kaust.edu.sa`> [contributor]
- Yu (Ryan) Yue <`yybaruch@gmail.com`> [contributor]
- Sarah Ryan <`sarahryan320@gmail.com`> [contributor]

See Also

Useful links:

- `https://github.com/mandymejia/BayesfMRI`
- Report bugs at `https://github.com/mandymejia/BayesfMRI/issues`

`.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_coefs, nTime, avg_var)
```

Arguments

`AR_coefs`	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

`activations` *Identify field activations*

Description

Identify areas of activation for each field from the result of `BayesGLM` or `fit_bayesglm`.

Usage

```
activations(
  x,
  Bayes = TRUE,
  gamma = NULL,
  alpha = 0.05,
  correction = c("FWER", "FDR", "none"),
  fields = NULL,
  sessions = NULL,
  verbose = 1
)

id_activations(
  x,
  Bayes = TRUE,
  gamma = NULL,
  alpha = 0.05,
  correction = c("FWER", "FDR", "none"),
  fields = NULL,
  sessions = NULL,
  verbose = 1
)
```

Arguments

`x`	Result of `BayesGLM` or `fit_bayesglm` model call, of class `"BGLM"` or `"fit_bglm"`.
`Bayes`	Use spatial Bayesian modeling to identify activations based on the joint posterior distribution? Default: `TRUE`. If `FALSE`, activations will be based on classical (massive univariate) GLM model, with multiple comparisons correction (see `correction`). Note that `TRUE` is only applicable if `x` includes Bayesian results (i.e. `x <- BayesGLM(..., Bayes = TRUE)` was run.)
`gamma`	Activation threshold, for example `1` for 1 percent signal 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.
`alpha`	Significance level for inference. Default: `0.05`.
`correction`	For the classical method only: Type of multiple comparisons correction: `"FWER"` (Bonferroni correction, the default), `"FDR"` (Benjamini Hochberg), or `"none"`.
`fields`	The field(s) to identify activations for. Give either the name(s) as a character vector, or the numerical indices. If `NULL` (default), analyze all fields.
`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.
`verbose`	`1` (default) to print occasional updates during model computation; `2` for occasional updates as well as running INLA in verbose mode (if `Bayes`), or `0` for no printed updates.

Value

An `"act_BGLM"` or `"act_fit_bglm"` object, a list which indicates the activated locations along with related information.

`aic_Param` *aic*

Description

aic

Arguments

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

`ar_order_Param` *ar_order*

Description

ar_order

Arguments

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

`ar_smooth_Param` *ar_smooth*

Description

ar_smooth

Arguments

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

`BayesGLM` *BayesGLM for CIFTI*

Description

Performs spatial Bayesian GLM for task fMRI activation with CIFTI-format data. The cortex is modeled as a surface mesh, and subcortical structures are modeled as distinct volumetric regions. Includes the pre-processing steps of nuisance regression, prewhitening, scaling, and variance normalization. Supports both single- and multi-session analysis. Can also compute just the classical (spatially-independent)

Usage

```
BayesGLM(
  BOLD,
  brainstructures = c("left", "right"),
 subROI = c("Amygdala-L", "Amygdala-R", "Caudate-L", "Caudate-R", "Hippocampus-L",
    "Hippocampus-R", "Thalamus-L", "Thalamus-R"),
  design,
  nuisance = NULL,
  scrub = NULL,
  hpf = NULL,
  TR = NULL,
  surfL = NULL,
  surfR = NULL,
  resamp_res = 10000,
  nbhd_order = 1,
  buffer = c(1, 1, 3, 4, 4),
  session_names = NULL,
  scale_BOLD = c("mean", "sd", "none"),
  Bayes = TRUE,
  hyperpriors = c("informative", "default"),
  ar_order = 6,
  ar_smooth = 5,
  aic = FALSE,
  n_threads = 4,
  return_INLA = c("trimmed", "full", "minimal"),
  verbose = 1,
  meanTol = 1e-06,
  varTol = 1e-06
)
```

Arguments

`BOLD`	fMRI timeseries data in CIFTI format ("*.dtseries.nii"). For single-session analysis this can be a file path to a CIFTI file or a `"xifti"` object from the `ciftiTools` package. For multi-session analysis this can be a vector of file paths or a list of `"xifti"` objects. If `BOLD` is a `"xifti"` object(s), the surfaces, if any, will be used for the spatial model. However, if `surfL` and `surfR` are provided, they will override any surfaces in `BOLD`.
`brainstructures`	Character vector indicating which brain structure(s) of `BOLD` to analyze: `"left"` cortex; `"right"` cortex; and/or `"subcortical"` structures. Or `"all"` to model all three. Default: `c("left","right")` (cortex only).
`subROI`	Which subcortical ROIs should be analyzed? Can be `"all"` to analyze all subcortex ROIs. See the `ciftiTools_Name` column of `ciftiTools:::substructure_table()` for a list of possible subcortical ROIs.
`design`	A numeric matrix or `data.frame`, or a `"BayesfMRI_design"` object from `make_design`.

	Can also be an array where the third dimension is the same length as the number of data locations, to model each location with its own design.
`nuisance`	(Optional) A $T \times N_{nuis}$ matrix of nuisance signals, where T is the number of timepoints and N is the number of nuisance signals, or a list of these for multi-session analysis. Nuisance signals are regressed from the fMRI data and design matrix prior to GLM computation. Nuisance signals can include motion regressors, HRF derivatives not being modeled as tasks, and other sources of noise. Detrending/high-pass filtering is accomplished by adding DCT bases to the nuisance matrix; see the parameters `hpf` and `DCT`. Do not add spike regressors for scrubbing to the `nuisance` matrix. Rather, provide these in `scrub` so that their corresponding timepoints are also removed from the BOLD data after nuisance regression.
`scrub`	(Optional) A $T \times N_{scrub}$ matrix of spike regressors (one 1 value at the timepoint to scrub, and 0 for all other values), or a logical vector indicating the timepoints to scrub (`TRUE` to scrub, and `FALSE` to keep). For multi-session data, a session-length list of such matrices or logical vectors. The spike regressors will be included in the nuisance regression, and afterwards the timepoints indicated in `scrub` will be removed from the BOLD data and design matrix.
`hpf`	Add DCT bases to `nuisance` to apply a temporal high-pass filter to the data, for detrending? `hpf` is the filter frequency. Use `NULL` to skip detrending. Detrending is strongly recommended for fMRI data, to help reduce the autocorrelation in the residuals, so `NULL` will induce a warning. Use `"already"` to disable the warning while skipping highpass filtering. Using at least two DCT bases is as sufficient for detrending as using linear and quadratic drift terms in the nuisance matrix. So if DCT detrending is being used here, there is no need to add linear and quadratic drift terms to `nuisance`.
`TR`	Temporal resolution of the data, in seconds.
`surfL`, `surfR`	For cortex spatial model. Left and right cortex surface geometry in GIFTI format ("*.surf.gii"). These can be a file path to a GIFTI file or a `"surf"` object from `ciftiTools`. Surfaces can alternatively be provided through the `$surf` metadata in `BOLD` if it is `"xifti"` data. If neither are provided, by default the HCP group-average fs_LR inflated surfaces included in `ciftiTools` will be used for the cortex spatial model.
`resamp_res`	For cortex spatial model. The number of vertices to which each cortical surface should be resampled, or `NULL` to not resample. For computational feasibility, a value of `10000` (default) or lower is recommended for Bayesian spatial modeling. If `Bayes=FALSE`, `resamp_res` can be set to `NULL` for full-resolution classical modeling.
`nbhd_order`	For volumetric model. What order neighborhood around data locations to keep? `0` for no neighbors, `1` for 1st-order neighbors, `2` for 1st- and 2nd-order neighbors, etc. Smaller values will provide greater computational efficiency at the cost of higher variance around the edge of the data.

`buffer`	For volumetric model. The number of extra voxel layers around the bounding box. Set to `NULL` for no buffer. (We recommend not changing `buffer` unless you know what you're doing. Instead, to reduce the number of boundary voxels, adjust `nbhd_order`).
`session_names`	The names of the task-fMRI `BOLD` sessions, for multi-session analysis. If not provided here, will be inferred from `names(BOLD)`, inferred from `names(design)`, or generated automatically, in that order.
`scale_BOLD`	Controls scaling the BOLD response at each location. **"mean":** Scale the data to percent local signal change. **"sd":** Scale the data by local standard deviation. **"none":** Center the data but do not scale it.
`Bayes`	Perform spatial Bayesian modeling? Default: `TRUE`. If `FALSE`, only perform classical (massive univariate) modeling. (The classical GLM result is always returned, whether `Bayes` is `TRUE` or `FALSE`.)
`hyperpriors`	Should informative or default non-informative hyperpriors be assumed on SPDE hyperparameters?
`ar_order`	(For prewhitening) The order of the autoregressive (AR) model to use for prewhitening. If `0`, do not prewhiten. Default: `6`. For multi-session modeling, note that a single AR model is used; its coefficients will be the average estimate from each session.
`ar_smooth`	(For prewhitening) The FWHM parameter for spatially smoothing the coefficient estimates for the AR model to use for prewhitening. Recall that $\sigma = \frac{FWHM}{2*sqrt(2*log(2))}$. Set to `0` to not smooth the estimates. Default: `5`.
`aic`	(For prewhitening) Use the Akaike information criterion (AIC) to select AR model orders between `0` and `ar_order`? Default: `FALSE`.
`n_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) returns the results sufficient for `activations` and `BayesGLM2`; `"minimal"` returns enough for `BayesGLM2` but not `activations`; `"full"` returns the full `inla` output.
`verbose`	`1` (default) to print occasional updates during model computation; `2` for occasional updates as well as running INLA in verbose mode (if `Bayes`), or `0` for no printed updates.
`meanTol`, `varTol`	Tolerance for mean and variance of each data location. Locations which do not meet these thresholds are masked out of the analysis. Default: `1e-6` for both.

Details

To use `BayesGLM`, the design matrix must first be constructed with `make_design`.

Value

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

betas_Bayesian The field coefficients for the Bayesian model.

betas_classical The field 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.

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

sessions `data.frame` with the `name` and `nTime` of each BOLD session.

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

Connectome Workbench Requirement

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

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.

INLA Latent Fields Limit

INLA computation times increase greatly when the number of columns in the design matrix exceeds five: when there are more than five tasks, or more than three tasks each with a temporal derivative modeled as a field. In cases like the latter, we recommend modeling the temporal derivatives as nuisance signals using the option `dHRF_as="nuisance"`, rather than modeling the temporal derivatives as fields.

`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
)
```

Arguments

`results`	Either (1) a length N list of `"BGLM"` objects, or (2) a length N character vector of files storing `"BGLM"` objects saved with `saveRDS`. `"fit_bglm"` objects also are accepted.
`contrasts`	(Optional) A list of contrast vectors that specify the group-level summaries of interest. If `NULL` (DEFAULT), use contrasts that compute the average of each field (field HRF) across all subjects/sessions. Each contrast vector is length KSN specifying a group-level summary of interest, where K is the number of fields in the first-level design matrices, S is the number of sessions, and N is the number of subjects. The vector is grouped by fields, then sessions, then subjects. For a single session/subject, the contrast vector for the first field would be: `c0 <- c(1, rep(0, K-1)) #indexes the first field for a single session` so the full contrast vector for the group *average over all sessions/subjects for the first field* would be: `contrasts = rep(c0, S*N) /(S*N)`. To obtain the group average for the first field, for *just the first session*, input zeros for the remaining sessions: `c2 <- c(c0, rep(0, K*(S-1))) contrasts = rep(c2, N) /N`. To obtain the group mean *difference between two sessions* ($S = 2$) for the first field: `c3 <- c(c0, -c0) contrasts = rep(c3, N) / N`. To obtain the *mean over sessions* of the first field, just for the first subject: `c4 <- rep(c0, S) c(c4, 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`	`1` (default) to print occasional updates during model computation; `2` for occasional updates as well as running INLA in verbose mode (if `Bayes`), or `0` for no printed updates.

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.

`Bayes_Param` *Bayes*

Description

Bayes

Arguments

`Bayes`	Perform spatial Bayesian modeling? Default: `TRUE`. If `FALSE`, only perform classical (massive univariate) modeling. (The classical GLM result is always returned, whether `Bayes` is `TRUE` or `FALSE`.)

`BOLD_Param_BayesGLM` *BOLD*

Description

BOLD

Arguments

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

`brainstructures_Param_BayesGLM` *brainstructures*

Description

brainstructures

Arguments

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

`buffer_Param` *buffer*

Description

buffer

Arguments

`buffer`	For volumetric model. The number of extra voxel layers around the bounding box. Set to `NULL` for no buffer. (We recommend not changing `buffer` unless you know what you're doing. Instead, to reduce the number of boundary voxels, adjust `nbhd_order`).

`Connectome_Workbench_Description` *Connectome Workbench*

Description

Connectome Workbench

Connectome Workbench Requirement

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

`contrasts_Param` *contrasts*

Description

contrasts

Arguments

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

`design_Param_BayesGLM` *design*

Description

design

Arguments

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

`do_QC` *Mask out invalid data*

Description

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

Usage

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

Arguments

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

Value

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

Examples

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

`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.

`field_names_Param` *field_names*

Description

field_names

Arguments

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

`fit_bayesglm` *fit_bayesglm*

Description

Performs spatial Bayesian GLM for task fMRI activation

Usage

```
fit_bayesglm(
  BOLD,
  design,
  nuisance = NULL,
  scrub = NULL,
  spatial,
  scale_BOLD = c("mean", "sd", "none"),
  Bayes = TRUE,
  hyperpriors = c("informative", "default"),
  ar_order = 6,
  ar_smooth = 5,
  aic = FALSE,
  n_threads = 4,
  return_INLA = c("trimmed", "full", "minimal"),
  verbose = 1,
  meanTol = 1e-06,
  varTol = 1e-06
)
```

Arguments

`BOLD`, `design`, `nuisance`
: Session-length list of numeric matrices/arrays, each with volumes along the first dimension.

`scrub`
: Session-length list of spike regressors: numeric matrices, with volumes along the first dimension, valued at 1 for scrubbed volumes and 0 otherwise.

  Scrubbing is performed by incorporating spike regressors in the nuisance matrix during nuisance regression (in a simultaneous framework), and then removing the scrubbed timepoints from the resulting BOLD and design.

`spatial`
: Gives the spatial information:

  surf A list of two: `vertices` $V \times 3$ numeric matrix of vertex locations in XYZ coordinate space, and `faces`, $F \times 3$ matrix of positive integers defining the triangular faces.

  mask Mask of locations with valid data.

  For voxel data, a list of six:

  label 3D array of labeled locations to include in the model.

	trans_mat Projection matrix to convert voxel indices to XYZ position. Can be `NULL`. **trans_units** XYZ units. Can be `NULL`. **nbhd_order** See documentation for `BayesGLM`. **buffer** See documentation for `BayesGLM`.
`scale_BOLD`	Controls scaling the BOLD response at each location. **"mean":** Scale the data to percent local signal change. **"sd":** Scale the data by local standard deviation. **"none":** Center the data but do not scale it.
`Bayes`	Perform spatial Bayesian modeling? Default: `TRUE`. If `FALSE`, only perform classical (massive univariate) modeling. (The classical GLM result is always returned, whether `Bayes` is `TRUE` or `FALSE`.)
`hyperpriors`	Should informative or default non-informative hyperpriors be assumed on SPDE hyperparameters?
`ar_order`	(For prewhitening) The order of the autoregressive (AR) model to use for prewhitening. If `0`, do not prewhiten. Default: `6`. For multi-session modeling, note that a single AR model is used; its coefficients will be the average estimate from each session.
`ar_smooth`	(For prewhitening) The FWHM parameter for spatially smoothing the coefficient estimates for the AR model to use for prewhitening. Recall that $\sigma = \frac{FWHM}{2*sqrt(2*log(2))}$. Set to `0` to not smooth the estimates. Default: `5`.
`aic`	(For prewhitening) Use the Akaike information criterion (AIC) to select AR model orders between `0` and `ar_order`? Default: `FALSE`.
`n_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) returns the results sufficient for `activations` and `BayesGLM2`; `"minimal"` returns enough for `BayesGLM2` but not `activations`; `"full"` returns the full `inla` output.
`verbose`	`1` (default) to print occasional updates during model computation; `2` for occasional updates as well as running INLA in verbose mode (if `Bayes`), or `0` for no printed updates.
`meanTol`, `varTol`	Tolerance for mean, variance and SNR of each data location. Locations which do not meet these thresholds are masked out of the analysis. Default: `1e-6` for mean and variance, `50` for SNR.

Value

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

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

field_estimates The estimated coefficients for the Bayesian model.

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

mesh The model mesh.

mask A mask of `mesh` indicating the locations inside `mesh`.

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

field_names The names of the fields.

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.

`hpf_Param_BayesGLM` *hpf*

Description

hpf

Arguments

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

`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.

`INLA_Latent_Fields_Limit_Description` *INLA Latent Fields*

Description

INLA Latent Fields

INLA Latent Fields Limit

INLA computation times increase greatly when the number of columns in the design matrix exceeds five: when there are more than five tasks, or more than three tasks each with a temporal derivative modeled as a field. In cases like the latter, we recommend modeling the temporal derivatives as nuisance signals using the option `dHRF_as="nuisance"`, rather than modeling the temporal derivatives as fields.

`make_mesh` *Make Mesh*

Description

Make INLA triangular mesh from faces and vertices

Usage

```
make_mesh(vertices, faces)
```

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.

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.

`mean_var_Tol_Param` *mean and variance tolerance*

Description

mean and variance tolerance

Arguments

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

`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).

`nbhd_order_Param` *nbhd_order*

Description

nbhd_order

Arguments

`nbhd_order`	For volumetric model. What order neighborhood around data locations to keep? `0` for no neighbors, `1` for 1st-order neighbors, `2` for 1st- and 2nd-order neighbors, etc. Smaller values will provide greater computational efficiency at the cost of higher variance around the edge of the data.

`nuisance_Param_BayesGLM` *nuisance*

Description

nuisance

Arguments

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

`n_threads_Param` *n_threads*

Description

n_threads

Arguments

`n_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_BGLM` *S3 method: use* `view_xifti` *to plot a* `"act_BGLM"` *object*

Description

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

Usage

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

Arguments

`x`	An object of class "act_BGLM"
`idx`	Which field should be plotted? Give the numeric indices or the names. `NULL` (default) will show all fields. This argument overrides the `idx` argument to `view_xifti`.
`title`	If NULL, the field names associated with idx will be used.
`session`	Which session should be plotted? `NULL` (default) will use the first.
`...`	Additional arguments to `view_xifti`

Value

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

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

Description

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

Usage

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

Arguments

`x`	An object of class "BGLM"
`Bayes`	`TRUE` for plotting Bayesian results, `FALSE` for plotting classical GLM results. Default: `NULL`, which will use the Bayesian results if available and the classical results if not.
`idx`	Which field should be plotted? Give the numeric indices or the names. `NULL` (default) will show all fields. This argument overrides the `idx` argument to `view_xifti`.
`title`	If NULL, the field names associated with idx will be used.
`session`	Which session should be plotted? `NULL` (default) will use the first.
`zlim`	Overrides the `zlim` argument for `view_xifti`. Default: `c(-1, 1)`.
`...`	Additional arguments to `view_xifti`

Value

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

`plot.BGLM2` *S3 method: use* `view_xifti` *to plot a* `"BGLM2"` *object*

Description

S3 method: use `view_xifti` to plot a `"BGLM2"` object

Usage

```
## S3 method for class 'BGLM2'
plot(x, idx = NULL, stat = c("contrasts", "activations"), zlim = c(-1, 1), ...)
```

Arguments

`x`	An object of class "BGLM2"
`idx`	Which contrast should be plotted? Give the numeric indices or the names. `NULL` (default) will show all contrasts. This argument overrides the `idx` argument to `view_xifti`.
`stat`	Estimates of the `"contrasts"` (default), or their thresholded `"activations"`.
`zlim`	Overrides the `zlim` argument for `view_xifti`. Default: `c(-1, 1)`.
`...`	Additional arguments to `view_xifti`

Value

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

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

Description

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

Usage

```
## S3 method for class 'prev_BGLM'
plot(
  x,
  idx = NULL,
  session = NULL,
  drop_zeros = NULL,
  colors = "plasma",
  zlim = c(0, 1),
  ...
)
```

Arguments

`x`	An object of class "prev_BGLM"
`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`.
`...`	Additional arguments to `view_xifti`

Value

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

`prevalence` *Activations prevalence.*

Description

Activations prevalence.

Usage

```
prevalence(
  act_list,
  gamma_idx = 1,
  p_test = NULL,
  alpha = 0.05,
  correction = c("FWER", "FDR", "none")
)
```

Arguments

`act_list`	List of activations from `activations`. All should have the same sessions, fields, and brainstructures.
`gamma_idx`	If activations at multiple thresholds were computed, which threshold should be used for prevalence? Default: the first (lowest).
`p_test`	For inference: the expected baseline rate of activation for all data locations, under the null hypothesis. Default: `NULL` (do not perform hypothesis testing).
`alpha`	Significance level for inference. Default: `.05`.
`correction`	For the classical method only: Type of multiple comparisons correction: `"FWER"` (Bonferroni correction, the default), `"FDR"` (Benjamini Hochberg), or `"none"`.

Value

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

`resamp_res_Param_BayesGLM` *resamp_res*

Description

resamp_res

Arguments

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

`return_INLA_Param` *return_INLA*

Description

return_INLA

Arguments

`return_INLA`	Return the INLA model object? (It can be large.) Use `"trimmed"` (default) returns the results sufficient for `activations` and `BayesGLM2`; `"minimal"` returns enough for `BayesGLM2` but not `activations`; `"full"` returns the full `inla` output.

`scale_BOLD` *Scale the BOLD timeseries*

Description

Scale the BOLD timeseries

Usage

```
scale_BOLD(BOLD, scale = c("mean", "sd", "none"), v_means = NULL)
```

Arguments

`BOLD`	fMRI data as a locations by time ($V \times T$) numeric matrix.
`scale`	Option for scaling the BOLD response.
`v_means`	Original means of the BOLD data. ONLY provide if data has already been centered.

```
\code{"mean"} scaling will scale the data to percent local signal change.

\code{"sd"} scaling will scale the data by local standard deviation.

\code{"none"} will only center the data, not scale it.
```

Value

Scale to units of percent local signal change and centers

`scale_BOLD_Param` *scale_BOLD*

Description

scale_BOLD

Arguments

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

`scrub_Param_BayesGLM` *scrub*

Description

scrub

Arguments

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

`seed_Param` *seed*

Description

seed

Arguments

`seed`	Random seed (optional). Default: `NULL`.

`session_names_Param` *session_names*

Description

session_names

Arguments

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

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

Description

Summary method for class "`act_BGLM`"

Usage

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

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

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

Arguments

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

Value

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

`NULL`, invisibly.

`NULL`, invisibly.

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

Description

Summary method for class `"act_fit_bglm"`

Usage

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

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

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

Arguments

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

Value

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

`NULL`, invisibly.

`NULL`, invisibly.

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

Description

Summary method for class `"BGLM"`

Usage

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

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

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

Arguments

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

Value

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

`NULL`, invisibly.

`NULL`, invisibly.

`summary.BGLM2` *Summarize a* `"BGLM2"` *object*

Description

Summary method for class `"BGLM2"`

Usage

```
## S3 method for class 'BGLM2'
summary(object, ...)

## S3 method for class 'summary.BGLM2'
print(x, ...)

## S3 method for class 'BGLM2'
print(x, ...)
```

Arguments

`object`	Object of class `"BGLM2"`.
`...`	further arguments passed to or from other methods.
`x`	Object of class `"summary.BGLM2"`.

Value

A `"summary.BGLM2"` object, a list summarizing the properties of `object`.

`NULL`, invisibly.

`NULL`, invisibly.

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

Description

Summary method for class "`fit_bglm`"

Usage

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

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

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

Arguments

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

Value

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

`NULL`, invisibly.

`NULL`, invisibly.

`summary.fit_bglm2` *Summarize a* "`fit_bglm2`" *object*

Description

Summary method for class "`fit_bglm2`"

Usage

```
## S3 method for class 'fit_bglm2'
summary(object, ...)

## S3 method for class 'summary.fit_bglm2'
print(x, ...)

## S3 method for class 'fit_bglm2'
print(x, ...)
```

Arguments

`object`	Object of class "`fit_bglm2`".
`...`	further arguments passed to or from other methods.
`x`	Object of class "`summary.fit_bglm2`".

Value

A "`summary.fit_bglm2`" object, a list summarizing the properties of `object`.

`NULL`, invisibly.

`NULL`, invisibly.

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

Description

Summary method for class "`prev_BGLM`"

Summary method for class "`prev_BGLM`"

Usage

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

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

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

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

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

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

Arguments

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

Value

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

`NULL`, invisibly.

`NULL`, invisibly.

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

`NULL`, invisibly.

`NULL`, invisibly.

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

Description

Summary method for class `"prev_fit_bglm"`

Summary method for class `"prev_fit_bglm"`

Usage

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

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

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

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

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

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

Arguments

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

Value

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

`NULL`, invisibly.

`NULL`, invisibly.

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

`NULL`, invisibly.

`NULL`, invisibly.

`surfaces_Param_BayesGLM` *surfaces*

Description

surfaces

Arguments

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

`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 `activations`? Default: `TRUE`.

`TR_Param_BayesGLM` *TR*

Description

TR

Arguments

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

`verbose_Param` *verbose*

Description

verbose

Arguments

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

`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

---

`vol2spde` *Construct a triangular mesh from a 3D volumetric mask*

Description

Construct a triangular mesh from a 3D volumetric mask

Usage

```
vol2spde(mask, res, nbhd_order = 1, buffer = c(1, 1, 3, 4, 4))
```

Arguments

`mask`	An array of 0s and 1s representing a volumetric mask
`res`	The spatial resolution in each direction, in mm. For example, c(2,2,2) indicates 2mm isotropic voxels.
`nbhd_order`	For volumetric data, what order neighborhood around data locations to keep? (0 = no neighbors, 1 = 1st-order neighbors, 2 = 1st- and 2nd-order neighbors, etc.). Smaller values will provide greater computational efficiency at the cost of higher variance around the edge of the data.
`buffer`	For volumetric data, size of extra voxels layers around the bounding box, in terms of voxels. Set to NULL for no buffer.

Value

An inla.spde2 object.

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.

Index