

# Package: templateICAr (via r-universe)

November 25, 2024

**Type** Package

**Title** Estimate Brain Networks and Connectivity with ICA and Empirical Priors

**Version** 0.9.1

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

**Description** Implements the template ICA (independent components analysis) model proposed in Mejia et al. (2020) <[doi:10.1080/01621459.2019.1679638](https://doi.org/10.1080/01621459.2019.1679638)> and the spatial template ICA model proposed in proposed in Mejia et al. (2022) <[doi:10.1080/10618600.2022.2104289](https://doi.org/10.1080/10618600.2022.2104289)>. Both models estimate subject-level brain as deviations from known population-level networks, which are estimated using standard ICA algorithms. Both models employ an expectation-maximization algorithm for estimation of the latent brain networks and unknown model parameters. Includes direct support for 'CIFTI', 'GIFTI', and 'NIFTI' neuroimaging file formats.

**Depends** R (>= 3.6.0)

**License** GPL-3

**Additional\_repositories** <https://inla.r-inla-download.org/R/testing>

**Encoding** UTF-8

**Imports** abind, fMRITools (>= 0.4.4), fMRIscrub, foreach, ica, Matrix, matrixStats, methods, pesel, SQUAREM, stats, utils

**Suggests** ciftiTools (>= 0.13.2), excursions, RNifti, oro.nifti, gifti, covr, parallel, doParallel, knitr, rmarkdown, INLA, testthat (>= 3.0.0)

**RoxygenNote** 7.3.2

**URL** <https://github.com/mandymejia/templateICAr>

**BugReports** <https://github.com/mandymejia/templateICAr/issues>

**NeedsCompilation** yes

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

**Repository** CRAN

**Date/Publication** 2024-11-21 16:50:02 UTC

**Config/pak/sysreqs** libicu-dev

## Contents

activations . . . . .	3
bdiag_m2 . . . . .	4
Chol_samp_fun . . . . .	5
dim_reduce . . . . .	5
EM_templateICA . . . . .	6
estimate.ESS . . . . .	8
estimate_nu . . . . .	8
estimate_nu_matrix . . . . .	9
estimate_template . . . . .	9
estimate_template_FC . . . . .	15
estimate_template_from_DR . . . . .	15
export_template . . . . .	16
getInvCovAR . . . . .	17
groupICA.cifti . . . . .	17
IW_var . . . . .	19
IW_var_cor . . . . .	20
lik . . . . .	20
make_mesh . . . . .	21
make_mesh_2D . . . . .	21
norm_BOLD . . . . .	22
orthonorm . . . . .	23
plot.template.cifti . . . . .	24
plot.template.gifti . . . . .	24
plot.template.matrix . . . . .	25
plot.template.nifti . . . . .	26
plot.tICA.cifti . . . . .	27
plot.tICA.matrix . . . . .	27
plot.tICA.nifti . . . . .	28
plot.tICA_act.cifti . . . . .	28
pw_estimate . . . . .	29
resample_template . . . . .	29
sqrt_XtX . . . . .	30
summary.template.cifti . . . . .	30
summary.template.gifti . . . . .	31
summary.template.matrix . . . . .	32
summary.template.nifti . . . . .	32
summary.tICA.cifti . . . . .	33

summary.tICA.matrix . . . . . 34  
summary.tICA.nifti . . . . . 35  
summary.tICA\_act.cifti . . . . . 35  
summary.tICA\_act.matrix . . . . . 36  
summary.tICA\_act.nifti . . . . . 37  
templateICA . . . . . 38  
UpdateTheta\_templateICA . . . . . 42  
UT2mat . . . . . 43  
var\_sq\_err . . . . . 44  
var\_sq\_err\_constrained . . . . . 44

**Index** **46**

activations *Activations of (spatial) template ICA*

**Description**

Identify areas of activation in each independent component map from the result of (spatial) template ICA.

**Usage**

```

activations(
  tICA,
  u = NULL,
  z = NULL,
  alpha = 0.01,
  type = c(">", "abs >", "<", "!="),
  method_p = "BH",
  verbose = FALSE,
  which.ICs = NULL,
  deviation = FALSE
)

```

**Arguments**

tICA Fitted (spatial) template ICA object from [templateICA](#).

u, z Set a threshold value for activation? A threshold value can be specified directly with u, or a z-score-like threshold in terms of standard deviations (the SD of values in the mean template) can be specified with z. Only one type of threshold can be used. Default: NULL (do not use a threshold). Either argument can also be a vector to test multiple thresholds at once, as long as type is not "!=" (to ensure the activation regions are successive subsets).

alpha Significance level for hypothesis testing. Default: 0.01.

type Type of region: ">" (default), "abs >", "<", or "!=". "abs >" tests for magnitude by taking the absolute value and then testing if they are greater than...

method_p	If the input is a "tICA.[format]" model object, the type of multiple comparisons correction to use for p-values, or NULL for no correction. See help(p.adjust). Default: "BH" (Benjamini & Hochberg, i.e. the false discovery rate). Note that multiple comparisons will account for data locations, but not ICs.
verbose	If TRUE, display progress of algorithm. Default: FALSE.
which.ICs	Indices of ICs for which to identify activations. If NULL (default), use all ICs.
deviation	If TRUE identify significant deviations from the template mean, rather than significant areas of engagement. Default: FALSE.

### Value

A list containing activation maps for each IC, the joint and marginal PPMs for each IC, and the parameters used for computing activation. If the input represented CIFTI- or NIFTI-format data, then the activations maps will be formatted accordingly.

Use summary to obtain information about the activations results. For CIFTI-format activations, use plot to visualize the activation maps.

### Examples

```
## Not run:
  activations(tICA_result, alpha=.05, deviation=TRUE)

## End(Not run)
```

---

bdiag\_m2

*Bdiag m2*

---

### Description

Bdiag m2

### Usage

```
bdiag_m2(mat, N)
```

### Arguments

mat	a k x k 'matrix'
N	how many times to repeat mat

### Value

a sparse ( $Nk \times Nk$ ) matrix of class "dgCMatrix".

---

Chol_samp_fun	<i>Cholesky-based FC sampling</i>
---------------	-----------------------------------

---

**Description**

Cholesky-based FC sampling

**Usage**

```
Chol_samp_fun(Chol_vals, p, M, chol_diag, chol_offdiag, Chol_mat_blank)
```

**Arguments**

Chol_vals	Matrix of Cholesky factorizations (upper triangular values) for all template sessions (nN*nM x nChol)
p	Pivot/reordering applied to FC matrices prior to Cholesky factorization
M	Number of samples to draw
chol_diag	Indices of diagonal upper triangular elements
chol_offdiag	Indices of off-diagonal upper triangular elements
Chol_mat_blank	A nLxnL matrix indexing the upper triangular elements

---

dim_reduce	<i>PCA-based Dimension Reduction and Prewhitening</i>
------------	---

---

**Description**

Performs dimension reduction and prewhitening based on probabilistic PCA using SVD. If dimensionality is not specified, it is estimated using the method described in Minka (2008).

**Usage**

```
dim_reduce(X, Q = NULL, Q_max = 100)
```

**Arguments**

X	A numeric matrix, with each column being a centered timeseries. For fMRI data, X should be T timepoints by V brain locations.
Q	Number of latent dimensions to estimate. If NULL (default), estimated using PESEL (Sobczyk et al. 2020).
Q_max	Maximal number of principal components for automatic dimensionality selection with PESEL. Default: 100.

**Value**

A list containing the dimension-reduced data (`data_reduced`, a  $V \times Q$  matrix), prewhitening/dimension reduction matrix (`H`, a  $Q \times T$  matrix) and its (pseudo-)inverse (`Hinv`, a  $T \times Q$  matrix), the noise variance (`sigma_sq`), the correlation matrix of the dimension-reduced data (`C_diag`, a  $Q \times Q$  matrix), and the dimensionality (`Q`).

**Examples**

```
nT <- 30
nV <- 400
nQ <- 7
X <- matrix(rnorm(nV*nQ), nrow=nV) %*% diag(seq(nQ, 1)) %*% matrix(rnorm(nQ*nT), nrow=nQ)
dim_reduce(X, Q=nQ)
```

---

EM\_templateICA

*EM Algorithms for Template ICA Models*


---

**Description**

EM Algorithms for Template ICA Models

**Usage**

```
EM_templateICA.spatial(
  template_mean,
  template_var,
  meshes,
  BOLD,
  theta0,
  C_diag,
  H,
  Hinv,
  maxiter = 100,
  usePar = FALSE,
  epsilon = 0.001,
  reduce_dim = TRUE,
  verbose = FALSE
)

EM_templateICA.independent(
  template_mean,
  template_var,
  BOLD,
  theta0,
  C_diag,
  H,
```

```

    Hinv,
    maxiter = 100,
    epsilon = 0.001,
    reduce_dim = FALSE,
    usePar = FALSE,
    verbose
)

```

### Arguments

template_mean	( $V \times Q$ matrix) mean maps for each IC in template, where $Q$ is the number of ICs, $V = nvox$ is the number of data locations.
template_var	( $V \times Q$ matrix) between-subject variance maps for each IC in template
meshes	NULL for spatial independence model, otherwise a list of objects of class "templateICA_mesh" containing the triangular mesh (see <a href="#">make_mesh</a> ) for each brain structure.
BOLD	( $V \times Q$ matrix) dimension-reduced fMRI data
theta0	(list) initial guess at parameter values: A ( $Q \times Q$ mixing matrix), nu0_sq (residual variance from first level) and (for spatial model only) kappa (SPDE smoothness parameter for each IC map)
C_diag	( $Q \times 1$ ) diagonal elements of matrix proportional to residual variance.
H, Hinv	For dimension reduction of the spatial template ICA model, which assumes that all IC's have the same smoothness parameter, $\kappa$
maxiter	Maximum number of EM iterations. Default: 100.
usePar	Parallelize the computation? Default: FALSE. Can be the number of cores to use or TRUE, which will use the number available minus two. Not yet implemented for spatial template ICA.
epsilon	Smallest proportion change between iterations. Default: 0.001.
reduce_dim	Reduce the temporal dimension of the data using PCA? Default: TRUE for the spatial EM algorithm, and FALSE for the independent EM algorithm.
verbose	If TRUE, display progress of algorithm. Default: FALSE.

### Details

EM\_templateICA.spatial implements the expectation-maximization (EM) algorithm described in Mejia et al. (2019+) for estimating the subject-level ICs and unknown parameters in the template ICA model with spatial priors on subject effects.

In both models, if original fMRI timeseries has covariance  $\sigma^2 I_T$ , the prewhitened timeseries achieved by premultiplying by ( $Q \times T$ ) matrix  $H$  from PCA has diagonal covariance  $\sigma^2 H H'$ , so C\_diag is  $diag(H H')$ .

### Value

A list: theta (list of final parameter estimates), subICmean (estimates of subject-level ICs), subICvar (variance of subject-level ICs, for non-spatial model) or subjICcov (covariance matrix of subject-level ICs, for spatial model – note that only diagonal and values for neighbors are computed), and success (flag indicating convergence (TRUE) or not (FALSE))

---

estimate.ESS	<i>Estimation of effective sample size</i>
--------------	--

---

**Description**

Estimation of effective sample size

**Usage**

```
estimate.ESS(mesh, Y, ind = NULL, trace = FALSE)
```

**Arguments**

mesh	INLA mesh
Y	data
ind	index of the data locations in the mesh
trace	If TRUE, a formula based on the trace is used, otherwise the inverse correlation is used

**Details**

The functions computes the effective sample size as  $trace(Q^{-1})/trace(Q * Q)$  as in Bretherton et al. (1999), Journal of Climate.

**Value**

Estimate of the effective sample size

---

estimate_nu	<i>Universally estimate IW dof parameter nu based on method of moments, so that no empirical variance is under-estimated</i>
-------------	--

---

**Description**

Universally estimate IW dof parameter nu based on method of moments, so that no empirical variance is under-estimated

**Usage**

```
estimate_nu(var_FC, mean_FC)
```

**Arguments**

var_FC	Empirical between-subject variance of covariance matrices (QxQ)
mean_FC	Empirical mean of covariance matrices (QxQ)



**Value**

estimate for nu

---

estimate\_nu\_matrix     *Estimate IW dof parameter nu based on method of moments*

---

**Description**

Estimate IW dof parameter nu based on method of moments

**Usage**

```
estimate_nu_matrix(var_FC, mean_FC)
```

**Arguments**

var\_FC            Empirical between-subject variance of covariance matrices (QxQ)  
mean\_FC           Empirical mean of covariance matrices (QxQ)

**Value**

QxQ matrix of estimates for nu

---

estimate\_template     *Estimate template*

---

**Description**

Estimate template for Template ICA based on fMRI data

**Usage**

```
estimate_template(  
  BOLD,  
  BOLD2 = NULL,  
  GICA,  
  mask = NULL,  
  inds = NULL,  
  scale = c("local", "global", "none"),  
  scale_sm_surfl = NULL,  
  scale_sm_surfr = NULL,  
  scale_sm_fwhm = 2,  
  TR = NULL,  
  hpf = 0.01,  
  GSR = FALSE,
```

```
Q2 = 0,
Q2_max = NULL,
brainstructures = "all",
resamp_res = NULL,
keep_S = FALSE,
keep_FC = FALSE,
FC = TRUE,
FC_nPivots = 100,
FC_nSamp = 50000,
varTol = 1e-06,
maskTol = 0.1,
missingTol = 0.1,
usePar = FALSE,
wb_path = NULL,
verbose = TRUE
)

estimate_template.cifti(
  BOLD,
  BOLD2 = NULL,
  GICA,
  inds = NULL,
  scale = c("local", "global", "none"),
  scale_sm_surfl = NULL,
  scale_sm_surfr = NULL,
  scale_sm_FWHM = 2,
  TR = NULL,
  hpf = 0.01,
  GSR = FALSE,
  Q2 = 0,
  Q2_max = NULL,
  brainstructures = "all",
  resamp_res = resamp_res,
  keep_S = FALSE,
  keep_FC = FALSE,
  FC = FALSE,
  varTol = 1e-06,
  maskTol = 0.1,
  missingTol = 0.1,
  usePar = FALSE,
  wb_path = NULL,
  verbose = TRUE
)

estimate_template.gifti(
  BOLD,
  BOLD2 = NULL,
  GICA,
```

```

inds = NULL,
scale = c("local", "global", "none"),
scale_sm_surfL = NULL,
scale_sm_surfR = NULL,
scale_sm_FWHM = 2,
TR = NULL,
hpf = 0.01,
GSR = FALSE,
Q2 = 0,
Q2_max = NULL,
brainstructures = "all",
keep_S = FALSE,
keep_FC = FALSE,
FC = FALSE,
varTol = 1e-06,
maskTol = 0.1,
missingTol = 0.1,
usePar = FALSE,
wb_path = NULL,
verbose = TRUE
)

estimate_template.nifti(
  BOLD,
  BOLD2 = NULL,
  GICA,
  inds = NULL,
  scale = c("local", "global", "none"),
  TR = NULL,
  hpf = 0.01,
  GSR = FALSE,
  Q2 = 0,
  Q2_max = NULL,
  mask = NULL,
  keep_S = FALSE,
  keep_FC = FALSE,
  FC = FALSE,
  varTol = 1e-06,
  maskTol = 0.1,
  missingTol = 0.1,
  usePar = FALSE,
  wb_path = NULL,
  verbose = TRUE
)

```

### Arguments

**BOLD, BOLD2**      Vector of subject-level fMRI data in one of the following formats: CIFTI file paths, "xifti" objects, GIFTI file paths, "gifti" objects, NIFTI file paths,

"nifti" objects, or  $V \times T$  numeric matrices, where  $V$  is the number of data locations and  $T$  is the number of timepoints.

If BOLD2 is provided it must be in the same format as BOLD; BOLD will be the test data and BOLD2 will be the retest data. BOLD2 should be the same length as BOLD and have the same subjects in the same order. If BOLD2 is not provided, BOLD will be split in half; the first half will be the test data and the second half will be the retest data.

GICA	<p>Group ICA maps in a format compatible with BOLD. Can also be a (vectorized) numeric matrix (<math>V \times Q</math>) no matter the format of BOLD. Its columns will be centered.</p> <p>New: can also be a parcellation in CIFTI format (other formats to be implemented in the future). The parcellation should have the same locations and one column, with integer values indicating the parcel to which each location belongs to. Each parcel is modeled as a brain map; instead of the first step of dual regression, the medial timecourse of each parcel is used.</p>
mask	<p>Required if BOLD are NIFTI file paths or "nifti" objects, and optional for other formats. For NIFTI data, this is a brain map formatted as a logical array of the same spatial dimensions as the fMRI data, with TRUE corresponding to in-mask voxels. For other data, this is a logical vector with the same length as the number of locations in GICA, with TRUE corresponding to in-mask locations.</p>
inds	<p>Numeric indices of the group ICs to include in the template. If NULL, use all group ICs (default).</p> <p>If inds is provided, the ICs not included will be removed after calculating dual regression, not before. This is because removing the ICs prior to dual regression would leave unmodelled signals in the data, which could bias the templates.</p>
scale	<p>"local" (default), "global", or "none". Local scaling will divide each data location's time series by its estimated standard deviation. Global scaling will divide the entire data matrix by the mean image standard deviation (<math>\text{mean}(\text{sqrt}(\text{rowVars}(\text{BOLD})))</math>).</p>
scale_sm_surfL, scale_sm_surfR, scale_sm_FWHM	<p>Only applies if <code>scale=="local"</code> and BOLD represents surface data (CIFTI or GIFTI). To smooth the standard deviation estimates used for local scaling, provide the surface geometries along which to smooth as GIFTI geometry files or "surf" objects, as well as the smoothing FWHM (default: 2).</p> <p>If <code>scale_sm_FWHM==0</code>, no smoothing of the local standard deviation estimates will be performed.</p> <p>If <code>scale_sm_FWHM&gt;0</code> but <code>scale_sm_surfL</code> and <code>scale_sm_surfR</code> are not provided, the default inflated surfaces from the HCP will be used.</p> <p>To create a "surf" object from data, see <a href="#">make_surf</a>. The surfaces must be in the same resolution as the BOLD data.</p>
TR	<p>The temporal resolution of the data, i.e. the time between volumes, in seconds. TR is required for detrending with <code>hpf</code>.</p>
hpf	<p>The frequency at which to apply a highpass filter to the data during pre-processing, in Hertz. Default: 0.01 Hertz. Set to 0 to disable the highpass filter.</p> <p>The highpass filter serves to detrend the data, since low-frequency variance is associated with noise. Highpass filtering is accomplished by nuisance regression of discrete cosine transform (DCT) bases.</p>

	Note the TR argument is required for highpass filtering. If TR is not provided, hpf will be ignored.
GSR	Center BOLD across columns (each image)? This is equivalent to performing global signal regression. Default: FALSE.
Q2, Q2_max	Obtain dual regression estimates after denoising? Denoising is based on modeling and removing nuisance ICs. It may result in a cleaner estimate for smaller datasets, but it may be unnecessary (and time-consuming) for larger datasets. Set Q2 to control denoising: use a positive integer to specify the number of nuisance ICs, NULL to have the number of nuisance ICs estimated by PESEL, or zero (default) to skip denoising. If <code>is.null(Q2)</code> , use Q2_max to specify the maximum number of nuisance ICs that should be estimated by PESEL. Q2_max must be less than $T * .75 - Q$ where $T$ is the minimum number of timepoints in each fMRI scan and $Q$ is the number of group ICs. If NULL (default), Q2_max will be set to $T * .50 - Q$ , rounded.
brainstructures	Only applies if the entries of BOLD are CIFTI file paths. This is a character vector indicating which brain structure(s) to obtain: "left" (left cortical surface), "right" (right cortical surface) and/or "subcortical" (subcortical and cerebellar gray matter). Can also be "all" (obtain all three brain structures). Default: <code>c("all")</code> .
resamp_res	Only applies if the entries of BOLD are CIFTI file paths. Resample the data upon reading it in? Default: NULL (no resampling).
keep_S	Keep the DR estimates of S? If FALSE (default), do not save the DR estimates and only return the templates. If TRUE, the DR estimates of S are returned too. If a single file path, save the DR estimates as an RDS file at that location rather than returning them.
keep_FC	Keep the DR estimates of the FC $\text{cor}(A)$ ? If FALSE (default), do not save the DR estimates and only return the templates. If TRUE, the DR estimates of $\text{cor}(A)$ and its Cholesky factor are returned too. If a single file path, save the DR estimates as an RDS file at that location rather than returning them.
FC	Include the functional connectivity template? Default: TRUE.
FC_nPivots	Number of pivots to use in Cholesky-based FC template estimation. Set to zero to skip Cholesky-based FC template estimation. Default: 100.
FC_nSamp	Number of FC matrix samples to generate across all pivots. This should be a multiple of FC_nPivots.
varTol	Tolerance for variance of each data location. For each scan, locations which do not meet this threshold are masked out of the analysis. Default: $1e-6$ . Variance is calculated on the original data, before any normalization.
maskTol	For computing the dual regression results for each subject: tolerance for number of locations masked out due to low variance or missing values. If more than this many locations are masked out, a subject is skipped without calculating dual regression. maskTol can be specified either as a proportion of the number of locations (between zero and one), or as a number of locations (integers greater than one). Default: .1, i.e. up to 10 percent of locations can be masked out. If BOLD2 is provided, masks are calculated for both scans and then the intersection of the masks is used, for each subject.

missingTol	For computing the variance decomposition across all subjects: tolerance for number of subjects masked out due to low variance or missing values at a given location. If more than this many subjects are masked out, the location's value will be NA in the templates. missingTol can be specified either as a proportion of the number of locations (between zero and one), or as a number of locations (integers greater than one). Default: .1, i.e. up to 10 percent of subjects can be masked out at a given location.
usePar, wb_path	Parallelize the DR computations over subjects? Default: FALSE. Can be the number of cores to use or TRUE, which will use the number on the PC minus two. If the input data is in CIFTI format, the wb_path must also be provided.
verbose	Display progress updates? Default: TRUE.

### Details

All fMRI data (entries in BOLD and BOLD2, and GICA) must be in the same spatial resolution.

### Value

A list: the template and var\_decomp with entries in matrix format; the mask of locations without template values due to too many low variance or missing values; the function params such as the type of scaling and detrending performed; the dat\_struct which can be used to convert template and var\_decomp to "xifti" or "nifti" objects if the BOLD format was CIFTI or NIFTI data; and DR results if isTRUE(keep\_S) and/or isTRUE(keep\_FC).

Use summary to print a description of the template results, and for CIFTI-format data use plot to plot the template mean and variance estimates. Use [export\\_template](#) to save the templates to individual RDS, CIFTI, or NIFTI files (depending on the BOLD format).

### Examples

```
nT <- 21
nV <- 140
nQ <- 6
mU <- matrix(rnorm(nV*nQ), nrow=nV)
mS <- mU %%% diag(seq(nQ, 1)) %%% matrix(rnorm(nQ*nT), nrow=nQ)
BOLD <- list(B1=mS, B2=mS, B3=mS)
BOLD <- lapply(BOLD, function(x){x + rnorm(nV*nT, sd=.05)})
GICA <- mU
estimate_template(BOLD=BOLD, GICA=mU, FC_nSamp=2000)

## Not run:
estimate_template(
  run1_cifti_fnames, run2_cifti_fnames,
  gICA_cifti_fname, brainstructures="all",
  scale="global", TR=0.71, Q2=NULL, varTol=10
)

## End(Not run)
```

---

estimate\_template\_FC *Estimate FC template*

---

### Description

Estimate FC template

### Usage

```
estimate_template_FC(FC0, nu_adjust = 1)
```

### Arguments

FC0	The FC estimates from <a href="#">estimate_template</a> .
nu_adjust	Factor by which to adjust estimate of nu. Values < 1 will inflate the template variance to avoid an over-informative prior on FC.

---

estimate\_template\_from\_DR  
*Estimate template from DR*

---

### Description

Estimate variance decomposition and templates from DR estimates.

### Usage

```
estimate_template_from_DR(DR, LV = NULL)
```

### Arguments

DR	the test/retest dual regression estimates, as an array with dimensions $M \times N \times (L \times V)$ , where $M$ is the number of visits (2), $N$ is the number of subjects, $L$ is the number of IC networks, and $V$ is the number of data locations. ( $L$ and $V$ are collapsed because they are treated equivalently in the context of calculating the variance decomposition and templates).
LV	A length-two integer vector giving the dimensions $L$ and $V$ to reshape the result. Default: NULL (do not reshape the result).

### Value

List of two elements: the templates and the variance decomposition.

There are two version of the variance template: varUB gives the unbiased variance estimate, and varNN gives the upwardly-biased non-negative variance estimate. Values in varUB will need to be clamped above zero before using in templateICA.

---

export_template	<i>Export template</i>
-----------------	------------------------

---

### Description

Export the templates (mean and variance) as separate files for visualization or processing outside of templateICAR.

### Usage

```
export_template(
  x,
  out_fname = NULL,
  var_method = c("non-negative", "unbiased")
)
```

### Arguments

x	The result of estimate_template
out_fname	Use NULL (default) to just return the template objects directly. Otherwise, use a character vector of length 3 or 4 of file path(s) to save the output to: the mean template, the variance template, the variance decomposition, and the FC template if present, in that order. If one file name is provided, it will be appended with "_mean.[file_ext]" for the template mean map, "_var.[file_ext]" for the template variance map, "_varDecomp.rds" for the variance decomposition, and "_FC.rds" where [file_ext] will be "dscalar.nii" for CIFTI input, "nii" for NIFTI input, and "rds" for data input.
var_method	"non-negative" (default) or "unbiased"

### Value

If is.null(out\_fname), the templates in data matrix, "xiffti", or "nifti" format, to match the format of the original BOLD data. Otherwise, the paths to the new files specified by out\_fname. If template includes functional connectivity components, the FC template and its mean and variance will be included.

### Examples

```
## Not run:
tm <- estimate_template(cii1_fnames, cii2_fnames, gICA_fname)
export_template(tm, out_fname="my_template", var_method="unbiased")

## End(Not run)
```



---

getInvCovAR	<i>Compute inverse covariance matrix for AR process (up to a constant scaling factor)</i>
-------------	---

---

**Description**

Compute inverse covariance matrix for AR process (up to a constant scaling factor)

**Usage**

```
getInvCovAR(ar, ntime)
```

**Arguments**

ar	vector of p AR parameters
ntime	number of time points in timeseries

**Value**

inverse covariance matrix for AR process (up to a constant scaling factor)

---

groupICA.cifti	<i>Perform group ICA based on CIFTI data</i>
----------------	--

---

**Description**

Perform group ICA based on CIFTI data

**Usage**

```
groupICA.cifti(  
  cifti_fnames,  
  subjects = NULL,  
  brainstructures = "all",  
  num_PCs = 100,  
  num_ICs,  
  max_rows_GPCA = 10000,  
  GSR = FALSE,  
  scale = c("local", "global", "none"),  
  scale_sm_FWHM = 2,  
  TR = NULL,  
  hpf = 0.01,  
  verbose = TRUE,  
  out_fname = NULL,  
  surfL = NULL,
```

```

    surfR = NULL,
    smooth = TRUE,
    smooth_surf_FWHM = 5,
    smooth_vol_FWHM = 5,
    smooth_zeroes_as_NA = FALSE,
    smooth_subcortical_merged = FALSE
)

```

## Arguments

cifti_fnames	Vector of file paths of CIFTI-format fMRI timeseries (*.dtseries.nii) for which to calculate group ICA
subjects	Use this argument if some subjects have more than one scan. This is a numeric or factor vector the same length as cifti_fnames. Scans for the same subject should have the same value. For example, if there are four subjects with two visits each, and the scans are ordered with the first subject's two scans first, then the second subject's two scans next, etc., then this argument should be rep(seq(4), each=2). If there are three subjects and four scans, with the last two scans belonging to the same subject, this argument should be c(1, 2, 3, 3). If all subjects only have one scan, keep this argument as NULL (default).
brainstructures	Character vector indicating which brain structure(s) to obtain: "left" (left cortical surface), "right" (right cortical surface) and/or "subcortical" (subcortical and cerebellar gray matter). Can also be "all" (obtain all three brain structures). Default: c("all").
num_PCs	Maximum number of PCs to retain in initial subject-level PCA
num_ICs	Number of independent components to identify.
max_rows_GPCA	The maximum number of rows for the matrix on which group level PCA will be performed. This matrix is the result of temporal concatenation and contains length(cifti_fnames) * num_PCs rows.
GSR, scale, scale_sm_FWHM, TR, hpf	Center BOLD columns, scale by the standard deviation, and detrend voxel time-courses? See <a href="#">norm_BOLD</a> . Normalization is applied separately to each scan. Defaults: Center BOLD columns, scale by the global standard deviation, and apply a .01 Hz HPF if the TR is provided.
verbose	If TRUE, display progress updates
out_fname	(Optional) If not specified, a xifti object will be returned but the GICA maps will not be written to a CIFTI file.
surfL	(Optional) File path to a surface GIFTI for the left cortex. If provided, will be part of xifti result object for visualization in R. Will also be used to perform any smoothing.
surfR	(Optional) File path to a surface GIFTI for the right cortex. If provided, will be part of xifti result object for visualization in R. Will also be used to perform any smoothing.
smooth	Smooth the CIFTI data prior to reading in each file? Default: TRUE. Use the following arguments to control the smoothing parameters.

smooth\_surf\_FWHM, smooth\_vol\_FWHM, smooth\_zeroes\_as\_NA,  
smooth\_subcortical\_merged

See [smooth\\_cifti](#). The defaults here are the same. Note that `smooth_zeroes_as_NA` will control both of the corresponding surface and volume arguments to [smooth\\_cifti](#). These arguments only apply if `smooth`.

### Value

out\_fname if a file was written, or the GICA as a "xifti" object if not.

---

IW_var	<i>Compute theoretical Inverse-Wishart variance of covariance matrix elements</i>
--------	---

---

### Description

Compute theoretical Inverse-Wishart variance of covariance matrix elements

### Usage

```
IW_var(nu, p, xbar_ij, xbar_ii, xbar_jj)
```

### Arguments

nu	Inverse Wishart degrees of freedom parameter
p	Matrix dimension for IW distribution
xbar_ij	Empirical mean of covariance matrices at element (i,j)
xbar_ii	Empirical mean of covariance matrices at the ith diagonal element
xbar_jj	Empirical mean of covariance matrices at the jth diagonal element

### Value

Theoretical IW variance for covariance element (i,j)

---

IW_var_cor	<i>Compute theoretical Inverse-Wishart variance of correlation matrix elements</i>
------------	--

---

**Description**

Compute theoretical Inverse-Wishart variance of correlation matrix elements

**Usage**

```
IW_var_cor(nu, p, xbar_ij)
```

**Arguments**

nu	Inverse Wishart degrees of freedom parameter
p	Matrix dimension for IW distribution
xbar_ij	Empirical mean of covariance matrices at element (i,j)

**Value**

Theoretical IW variance for correlation element (i,j)

---

lik	<i>Compute likelihood in SPDE model for ESS estimation</i>
-----	--

---

**Description**

Compute likelihood in SPDE model for ESS estimation

**Usage**

```
lik(theta, Y, G, C, ind = NULL)
```

**Arguments**

theta	Value of hyperparameters
Y	Data vector
G	SPDE G matrix
C	SPDE C matrix
ind	Indices of data locations in the mesh

**Value**

Log likelihood value

---

make_mesh	<i>Make INLA mesh from "surf" object</i>
-----------	--

---

**Description**

Create INLA mesh and observation weight matrix based on a "surf" object

**Usage**

```
make_mesh(surf = NULL, inds_data = NULL, inds_mesh = NULL)
```

**Arguments**

surf	Object of class "surf". See <code>make_surf</code> and <code>is.surf</code> .
inds_data	Subset of vertices to include in analysis, e.g. non-medial wall locations.
inds_mesh	Subset of vertices to retain in mesh, e.g. non-medial wall locations. Must be a superset of inds_data.

**Value**

List containing INLA mesh, observation weight matrix **A** for translating between mesh locations and original data locations, the brain mask used to create the mesh, and the number of original and mesh data locations.

---

make_mesh_2D	<i>Make 2D INLA mesh</i>
--------------	--------------------------

---

**Description**

Create INLA mesh and observation weight matrix based on a binary brain mask

**Usage**

```
make_mesh_2D(mask)
```

**Arguments**

mask	Brain mask (matrix of 0 and 1 or TRUE and FALSE).
------	---

**Details**

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

**Value**

List containing INLA mesh, observation weight matrix **A** for translating between mesh locations and original data locations, the brain mask used to create the mesh, and the number of original and mesh data locations.

---

norm_BOLD	<i>Normalize BOLD data</i>
-----------	----------------------------

---

**Description**

Center the data across space and/or time, detrend, and scale, in that order. For dual regression, row centering is required and column centering is not recommended. Scaling and detrending depend on the user preference.

**Usage**

```
norm_BOLD(
  BOLD,
  center_rows = TRUE,
  center_cols = FALSE,
  scale = c("local", "global", "none"),
  scale_sm_xifti = NULL,
  scale_sm_FWHM = 2,
  scale_sm_xifti_mask = NULL,
  TR = NULL,
  hpf = 0.01
)
```

**Arguments**

BOLD	fMRI numeric data matrix ( $V \times T$ )
center_rows, center_cols	Center BOLD data across rows (each data location's time series) or columns (each time point's image)? Default: TRUE for row centering, and FALSE for column centering.
scale	"local" (default), "global", or "none". Local scaling will divide each data location's time series by its estimated standard deviation. Global scaling will divide the entire data matrix by the mean image standard deviation ( $\text{mean}(\text{sqrt}(\text{rowVars}(\text{BOLD})))$ ).
scale_sm_xifti, scale_sm_FWHM	Only applies if <code>scale=="local"</code> and BOLD represents CIFTI-format data. To smooth the standard deviation estimates used for local scaling, provide a "xifti" object with data locations in alignment with BOLD, as well as the smoothing FWHM (default: 2). If no "xifti" object is provided (default), do not smooth.
scale_sm_xifti_mask	For local scaling with smoothing, the data must be unmasked to be mapped back to the surface. So if the data are masked, provide the mask here.

TR	The temporal resolution of the data, i.e. the time between volumes, in seconds. TR is required for detrending with hpf.
hpf	The frequency at which to apply a highpass filter to the data during pre-processing, in Hertz. Default: 0.01 Hertz. Set to 0 to disable the highpass filter.  The highpass filter serves to detrend the data, since low-frequency variance is associated with noise. Highpass filtering is accomplished by nuisance regression of discrete cosine transform (DCT) bases.  Note the TR argument is required for highpass filtering. If TR is not provided, hpf will be ignored.

**Value**

Normalized BOLD data matrix ( $V \times T$ )

---

orthonorm	<i>Orthonormalizes a square, invertible matrix</i>
-----------	--

---

**Description**

Orthonormalizes a square, invertible matrix

**Usage**

orthonorm(X)

**Arguments**

X                    A square matrix to be orthonormalized.

**Details**

Y is orthonormal if  $YY' = Y'Y = I$ . Orthonormalization of X is given by  $X (X'X)^{-.5}$ .

**Value**

X after orthonormalization

---

plot.template.cifti *Plot template*

---

### Description

Plot template

### Usage

```
## S3 method for class 'template.cifti'
plot(
  x,
  stat = c("both", "mean", "sd", "var"),
  var_method = c("non-negative", "unbiased"),
  ...
)
```

### Arguments

x	The template from estimate_template.cifti
stat	"mean", "sd", or "both" (default). By default the square root of the variance template is shown; another option is stat="var" to instead display the variance template directly.
var_method	"non-negative" (default) or "unbiased"
...	Additional arguments to view_xifti

### Value

The plot

---

plot.template.gifti *Plot template*

---

### Description

Plot template

### Usage

```
## S3 method for class 'template.gifti'
plot(
  x,
  stat = c("both", "mean", "sd", "var"),
  var_method = c("non-negative", "unbiased"),
  ...
)
```



**Arguments**

x	The template from estimate_template.gifti
stat	"mean", "sd", or "both" (default). By default the square root of the variance template is shown; another option is stat="var" to instead display the variance template directly.
var_method	"non-negative" (default) or "unbiased"
...	Additional arguments to view_xifti

**Value**

The plot

---

plot.template.matrix *Plot template*

---

**Description**

Plot template

**Usage**

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

**Arguments**

x	The template from estimate_template.matrix
...	Additional arguments

**Value**

The plot

---

plot.template.nifti *Plot template*

---

## Description

Based on `oro.nifti::image`.

## Usage

```
## S3 method for class 'template.nifti'
plot(
  x,
  stat = c("mean", "sd", "var"),
  plane = c("axial", "sagittal", "coronal"),
  n_slices = 9,
  slices = NULL,
  var_method = c("non-negative", "unbiased"),
  ...
)
```

## Arguments

<code>x</code>	The template from <code>estimate_template.nifti</code>
<code>stat</code>	"mean" (default), "sd", or "var". ("sd" will show the square root of the variance template.)
<code>plane, n_slices, slices</code>	Anatomical plane and which slice indices to show. Default: 9 axial slices.
<code>var_method</code>	"non-negative" (default) or "unbiased"
<code>...</code>	Additional arguments to <code>oro.nifti::image</code>

## Details

Consider using `struct_template` to obtain the 3D volumes to plot with a different viewer function (e.g. from `oro.nifti`) if desired.

## Value

The plot

---

plot.tICA.cifti      *Plot template*

---

**Description**

Plot template

**Usage**

```
## S3 method for class 'tICA.cifti'  
plot(x, stat = c("mean", "se", "both"), ...)
```

**Arguments**

x	The result of templateICA with CIFTI data
stat	"mean" (default), "se", or "both"
...	Additional arguments to view_xifti

**Value**

The plot

---

plot.tICA.matrix      *Plot template*

---

**Description**

This feature is not supported yet.

**Usage**

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

**Arguments**

x	The result of templateICA with NIFTI data
...	Additional arguments

**Value**

Nothing, because an error is raised.

---

plot.tICA.nifti      *Plot template*

---

### Description

Plot template

### Usage

```
## S3 method for class 'tICA.nifti'
plot(
  x,
  stat = c("mean", "se"),
  plane = c("axial", "sagittal", "coronal"),
  n_slices = 9,
  slices = NULL,
  ...
)
```

### Arguments

`x`                    The result of templateICA with NIFTI data

`stat`                 "mean" (default), "se"

`plane, n_slices, slices`  
                      Anatomical plane and which slice indices to show. Default: 9 axial slices.

`...`                 Additional arguments

### Value

The plot

---

plot.tICA\_act.cifti      *Plot activations*

---

### Description

Plot activations

### Usage

```
## S3 method for class 'tICA_act.cifti'
plot(x, stat = c("active", "pvals", "pvals_adj", "tstats", "se"), ...)
```

**Arguments**

x                    The activations from `activations.cifti`  
 stat                "active" (default), "pvals", "pvals\_adj", "tstats", or "vars".  
 ...                 Additional arguments to `view_xifti`

**Value**

The activations plot

---

pw_estimate	<i>Estimate residual autocorrelation for prewhitening</i>
-------------	---

---

**Description**

Estimate residual autocorrelation for prewhitening

**Usage**

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

**Arguments**

A                    Estimated A matrix (T x Q)  
 ar\_order, aic       Order of the AR model used to prewhiten the data at each location. If !aic (default), the order will be exactly ar\_order. If aic, the order will be between zero and ar\_order, as determined by the AIC.

**Value**

Estimated AR coefficients and residual variance at every vertex

---

resample_template	<i>Resample CIFTI template</i>
-------------------	--------------------------------

---

**Description**

Resample a CIFTI template to a new spatial resolution.

**Usage**

```
resample_template(x, resamp_res, verbose = FALSE)
```

**Arguments**

x                    The "template.cifti" object.  
 resamp\_res        The new resampling resolution.  
 verbose            Give occasional updates? Default: FALSE.

**Value**

The resampled "template.cifti" object.

---

sqrt\_XtX            *Compute matrix square root of X'X*

---

**Description**

Compute matrix square root of X'X

**Usage**

```
sqrt_XtX(X, inverse = FALSE)
```

**Arguments**

X                    A numerical matrix  
 inverse            if inverse=TRUE, compute inverse of square root

**Value**

A matrix equalling the (inverse) matrix square root of X'X

---

summary.template.cifti  
                           *Summarize a "template.cifti" object*

---

**Description**

Summary method for class "template.cifti"

**Usage**

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

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

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

**Arguments**

object	Object of class "template.cifti".
...	further arguments passed to or from other methods.
x	The template from estimate_template.cifti

**Value**

A list summarizing the template: data dimensions, options used for template estimation, etc.

Nothing, invisibly.

Nothing, invisibly.

---

```
summary.template.gifti
```

*Summarize a "template.gifti" object*

---

**Description**

Summary method for class "template.gifti"

**Usage**

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

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

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

**Arguments**

object	Object of class "template.gifti".
...	further arguments passed to or from other methods.
x	The template from estimate_template.gifti

**Value**

A list summarizing the template: data dimensions, options used for template estimation, etc.

Nothing, invisibly.

Nothing, invisibly.

---

summary.template.matrix

*Summarize a "template.matrix" object*

---

### Description

Summary method for class "template.matrix"

### Usage

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

```
## S3 method for class 'summary.template.matrix'  
print(x, ...)
```

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

### Arguments

object	Object of class "template.matrix".
...	further arguments passed to or from other methods.
x	The template from estimate_template.cifti

### Value

A list summarizing the template: data dimensions, options used for template estimation, etc.

Nothing, invisibly.

Nothing, invisibly.

---

summary.template.nifti

*Summarize a "template.nifti" object*

---

### Description

Summary method for class "template.nifti"



**Usage**

```

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

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

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

```

**Arguments**

object	Object of class "template.nifti".
...	further arguments passed to or from other methods.
x	The template from estimate_template.nifti

**Value**

A list summarizing the template: data dimensions, options used for template estimation, etc.  
Nothing, invisibly.  
Nothing, invisibly.

---

summary.tICA.cifti	<i>Summarize a "tICA.cifti" object</i>
--------------------	--

---

**Description**

Summary method for class "tICA.cifti"

**Usage**

```

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

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

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

```

**Arguments**

object	Object of class "tICA.cifti".
...	further arguments passed to or from other methods.
x	The result of templateICA with CIFTI data

**Value**

A list summarizing of the results of the templateICA analysis.

Nothing, invisibly.

Nothing, invisibly.

---

summary.tICA.matrix    *Summarize a "tICA.matrix" object*

---

**Description**

Summary method for class "tICA.matrix"

**Usage**

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

```
## S3 method for class 'summary.tICA.matrix'  
print(x, ...)
```

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

**Arguments**

object	Object of class "tICA.matrix".
...	further arguments passed to or from other methods.
x	The template from estimate_template.cifti

**Value**

A list summarizing of the results of the templateICA analysis.

Nothing, invisibly.

Nothing, invisibly.

---

summary.tICA.nifti      *Summarize a "tICA.nifti" object*

---

### Description

Summary method for class "tICA.nifti"

### Usage

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

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

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

### Arguments

object	Object of class "tICA.nifti".
...	further arguments passed to or from other methods.
x	The template from estimate_template.cifti

### Value

A list summarizing of the results of the templateICA analysis.

Nothing, invisibly.

Nothing, invisibly.

---

summary.tICA\_act.cifti  
*Summarize a "tICA\_act.cifti" object*

---

### Description

Summary method for class "tICA\_act.cifti"

**Usage**

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

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

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

**Arguments**

object	Object of class "tICA_act.cifti".
...	further arguments passed to or from other methods.
x	The activations from activations.cifti

**Value**

A list summarizing the data and results for the activations analysis.  
 Nothing, invisibly.  
 Nothing, invisibly.

---

```
summary.tICA_act.matrix
```

*Summarize a "tICA\_act.matrix" object*

---

**Description**

Summary method for class "tICA\_act.matrix"

**Usage**

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

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

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

**Arguments**

object	Object of class "tICA_act.matrix".
...	further arguments passed to or from other methods.
x	The activations from activations

**Value**

A list summarizing the data and results for the activations analysis.

Nothing, invisibly.

Nothing, invisibly.

---

```
summary.tICA_act.nifti
```

```
Summarize a "tICA_act.nifti" object
```

---

**Description**

Summary method for class "tICA\_act.nifti"

**Usage**

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

```
## S3 method for class 'summary.tICA_act.nifti'  
print(x, ...)
```

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

**Arguments**

object	Object of class "tICA_act.nifti".
...	further arguments passed to or from other methods.
x	The activations from activations

**Value**

A list summarizing the data and results for the activations analysis.

Nothing, invisibly.

Nothing, invisibly.

---

 templateICA

*Template ICA*


---

### Description

Perform template independent component analysis (ICA) using variational Bayes (VB) or expectation-maximization (EM).

### Usage

```
templateICA(
  BOLD,
  template,
  tvar_method = c("non-negative", "unbiased"),
  scale = c("template", "global", "local", "none"),
  scale_sm_surfl = NULL,
  scale_sm_surfr = NULL,
  scale_sm_FWHM = "template",
  nuisance = NULL,
  scrub = NULL,
  TR = NULL,
  hpf = "template",
  GSR = "template",
  Q2 = "template",
  Q2_max = "template",
  brainstructures = "template",
  mask = NULL,
  time_inds = NULL,
  varTol = "template",
  spatial_model = NULL,
  resamp_res = NULL,
  rm_mwall = TRUE,
  reduce_dim = FALSE,
  method_FC = c("VB1", "VB2", "none"),
  maxiter = 100,
  miniter = 3,
  epsilon = 0.001,
  kappa_init = 0.2,
  usePar = TRUE,
  PW = FALSE,
  verbose = TRUE
)
```

### Arguments

**BOLD**                      Vector of subject-level fMRI data in one of the following formats: CIFTI file paths, "xiffti" objects, NIFTI file paths, "nifti" objects, or  $V \times T$  numeric

matrices, where  $V$  is the number of data locations and  $T$  is the number of time-points.

If multiple BOLD data are provided, they will be independently centered, scaled, detrended (if applicable), and denoised (if applicable). Then they will be concatenated together followed by computing the initial dual regression estimate.

template	Template estimates in a format compatible with BOLD, from <a href="#">estimate_template</a> .
tvar_method	Which calculation of the template variance to use: "non-negative" (default) or "unbiased". The unbiased template variance is based on the assumed mixed effects/ANOVA model, whereas the non-negative template variance adds to it to account for greater potential between-subjects variation. (The template mean is the same for either choice of tvar_method.)
scale	"global", "local", or "none". Global scaling will divide the entire data matrix by the mean image standard deviation ( $\text{mean}(\text{sqrt}(\text{rowVars}(\text{BOLD})))$ ). Local scaling will divide each data location's time series by its estimated standard deviation. Default: "template", to use the same option used for estimation of the template.
scale_sm_surfL, scale_sm_surfR, scale_sm_FWHM	<p>Only applies if scale=="local" and BOLD represents CIFTI-format data. To smooth the standard deviation estimates used for local scaling, provide the surface geometries along which to smooth as GIFTI geometry files or "surf" objects, as well as the smoothing FWHM (default: "template" to use the same option used for estimation of the template).</p> <p>If scale_sm_FWHM==0, no smoothing of the local standard deviation estimates will be performed.</p> <p>If scale_sm_FWHM&gt;0 but scale_sm_surfL and scale_sm_surfR are not provided, the default inflated surfaces from the HCP will be used.</p> <p>To create a "surf" object from data, see <a href="#">make_surf</a>. The surfaces must be in the same resolution as the BOLD data.</p>
nuisance	(Optional) Signals to regress from the data, given as a numeric matrix with the same number of rows as there are volumes in the BOLD data. If multiple BOLD sessions are provided, this argument can be a list to use different nuisance regressors for different sessions. Nuisance regression is performed as a first step, before centering, scaling, and denoising. An intercept column will automatically be added to nuisance. If NULL, no extra nuisance signals will be regressed from the data, but a nuisance regression will still be used if warranted by scrub or hpf.
scrub	(Optional) A numeric vector of integers indicating the indices of volumes to scrub from the BOLD data. (List the volumes to remove, not the ones to keep.) If multiple BOLD sessions are provided, this argument can be a list to remove different volumes for different sessions. Scrubbing is performed within nuisance regression by adding a spike regressor to the nuisance design matrix for each volume to scrub. If NULL, do not scrub.
TR, hpf	These arguments control detrending. TR is the temporal resolution of the data, i.e. the time between volumes, in seconds; hpf is the frequency of the high-pass filter, in Hertz. Detrending is performed via nuisance regression of DCT bases. Default: "template" to use the values from the template. Be sure to

set the correct TR if it's different for the new data compared to the data used in `estimate_template`.

Note that if multiple BOLD sessions are provided, their TR and hpf must be the same; both arguments accept only one value.

GSR	Center BOLD across columns (each image)? This is equivalent to performing global signal regression. Default: "template", to use the same option used for estimation of the template.
Q2, Q2_max	<p>Denoise the BOLD data? Denoising is based on modeling and removing nuisance ICs. It may result in a cleaner estimate for smaller datasets, but it may be unnecessary (and time-consuming) for larger datasets.</p> <p>Set Q2 to control denoising: use a positive integer to specify the number of nuisance ICs, NULL to have the number of nuisance ICs estimated by PESEL, or zero to skip denoising.</p> <p>If <code>is.null(Q2)</code>, use <code>Q2_max</code> to specify the maximum number of nuisance ICs that should be estimated by PESEL. <code>Q2_max</code> must be less than <math>T * .75 - Q</math> where <math>T</math> is the number of timepoints in BOLD and <math>Q</math> is the number of group ICs. If NULL, <code>Q2_max</code> will be set to <math>T * .50 - Q</math>, rounded.</p> <p>The defaults for both arguments is "template", to use the same option used for estimation of the template.</p>
brainstructures	Only applies if the entries of BOLD are CIFTI file paths. This is a character vector indicating which brain structure(s) to obtain: "left" (left cortical surface), "right" (right cortical surface) and/or "subcortical" (subcortical and cerebellar gray matter). Can also be "all" (obtain all three brain structures). Default: "template" to use the same brainstructures present in the template).
mask	Required if and only if the entries of BOLD are NIFTI file paths or "nifti" objects. This is a brain map formatted as a binary array of the same spatial dimensions as the fMRI data, with TRUE corresponding to in-mask voxels.
time_inds	Subset of fMRI BOLD volumes to include in analysis, as a vector of numeric integers. If NULL (default), use all volumes. Subsetting is performed before any centering, scaling, detrending, and denoising. If multiple BOLD are provided, <code>time_inds</code> can be a list of the same length, with each element being a vector of numeric integers indicating the timepoints to include for each BOLD.
varTol	Tolerance for variance of each data location. For each scan, locations which do not meet this threshold are masked out of the analysis. Default: "template" to use the same brainstructures present in the template). Variance is calculated on the original data, before any normalization. Set to 0 to avoid removing locations due to low variance.
spatial_model	<p>Should spatial modeling be performed? If NULL, assume spatial independence. Otherwise, provide meshes specifying the spatial priors assumed on each independent component. Each should represent a brain structure, between which spatial independence can be assumed.</p> <p>If BOLD represents CIFTI-format data, <code>spatial_model</code> should give the left and right cortex surface geometries (whichever one(s) are being used) as "surf" objects or GIFTI surface geometry file paths. Spatial modeling is not yet available</p>



for the subcortex. This argument can also be TRUE, in which case spatial modeling will be performed with the surfaces included in the first entry of BOLD if it is a "xifti" object, or if those are not present available, the default inflated surfaces from `ciftiTools`.

If BOLD represents NIFTI-format data, spatial modeling is not yet available.

If BOLD is a numeric matrix, `spatial_model` should be a list of meshes (see [make\\_mesh](#)).

<code>resamp_res</code>	Only applies if BOLD represents CIFTI-format data. The target resolution for resampling (number of cortical surface vertices per hemisphere). For spatial modelling, a value less than 10000 is recommended for computational feasibility. If NULL (default), do not perform resampling.
<code>rm_mwall</code>	Only applies if BOLD represents CIFTI-format data. Should medial wall (missing data) locations be removed from the mesh? If TRUE, faster computation but less accurate estimates at the boundary of wall.
<code>reduce_dim</code>	Reduce the temporal dimension of the data using PCA? Default: TRUE. Skipping dimension reduction will slow the model estimation, but may result in more accurate results. Ignored for FC template ICA
<code>method_FC</code>	Variational Bayes (VB) method for FC template ICA model: "VB1" (default) uses a conjugate Inverse-Wishart prior for the $\text{cor}(A)$ ; "VB2" draws samples from $p(\text{cor}(A))$ to emulate the population distribution using a combination of Cholesky, SVD, and random pivoting. "none" Uses standard template ICA without FC prior
<code>maxiter</code>	Maximum number of EM or VB iterations. Default: 100.
<code>miniter</code>	Minimum number of EM or VB iterations. Default: 3.
<code>epsilon</code>	Smallest proportion change between iterations. Default: .001.
<code>kappa_init</code>	Starting value for kappa. Default: 0.2.
<code>usePar</code>	Parallelize the computation? Default: TRUE. Can be the number of cores to use or TRUE, which will use the number available minus two.
<code>PW</code>	Prewhiten to account for residual autocorrelation? Default: FALSE. Only affects FC template ICA models.
<code>verbose</code>	If TRUE, display progress of algorithm

### Value

A (spatial) template ICA object, which is a list containing: `subjICmean`, the  $V \times L$  estimated independent components **S**; `subjICse`, the standard errors of **S**; the mask of locations without template values due to too many low variance or missing values; the nuisance design matrix or matrices if applicable; and the function params such as the type of scaling and detrending performed.

If BOLD represented CIFTI or NIFTI data, `subjICmean` and `subjICse` will be formatted as "xifti" or "nifti" objects, respectively.

### Examples

```
## Not run:
tm <- estimate_template(cii1_fnames, cii2_fnames, gICA_fname)
```

```

templateICA(newcii_fname, tm, spatial_model=TRUE, resamp_res=2000)

## End(Not run)

```

---

UpdateTheta\_templateICA

*Parameter Estimates in EM Algorithm for Template ICA Model*

---

## Description

Parameter Estimates in EM Algorithm for Template ICA Model

## Usage

```

UpdateTheta_templateICA.spatial(
  template_mean,
  template_var,
  meshes,
  BOLD,
  theta,
  C_diag,
  H,
  Hinv,
  s0_vec,
  D,
  Dinv_s0,
  verbose = FALSE,
  return_MAP = FALSE,
  update = c("all", "kappa", "A")
)

UpdateTheta_templateICA.independent(
  template_mean,
  template_var,
  BOLD,
  theta,
  C_diag,
  H,
  Hinv,
  update_nu0sq = TRUE,
  return_MAP = FALSE,
  verbose = TRUE
)

```

## Arguments

`template_mean` ( $V \times Q$  matrix) mean maps for each IC in template

template_var	( $V \times Q$ matrix) between-subject variance maps for each IC in template
meshes	NULL for spatial independence model, otherwise a list of objects of class "templateICA_mesh" containing the triangular mesh (see <a href="#">make_mesh</a> ) for each brain structure.
BOLD	( $V \times Q$ matrix) dimension-reduced fMRI data
theta	(list) current parameter estimates
C_diag	( $Q \times 1$ ) diagonal elements of residual covariance after dimension reduction
H, Hinv	For dimension reduction
s0_vec	Vectorized template means
D	Sparse diagonal matrix of template standard deviations
Dinv_s0	The inverse of D times s0_vec
verbose	If TRUE, display progress of algorithm. Default: FALSE.
return_MAP	If TRUE. return the posterior mean and precision of the latent fields instead of the parameter estimates. Default: FALSE.
update	Which parameters to update. Either "all", "A" or "kappa".
update_nu0sq	For non-spatial model: updating nu0sq is recommended if dimension reduction was not performed, and is not recommended if it was.

**Value**

An updated list of parameter estimates, theta, OR if return\_MAP=TRUE, the posterior mean and precision of the latent fields

---

 UT2mat

---

*Transform upper-triangular elements to matrix form*


---

**Description**

Transform upper-triangular elements to matrix form

**Usage**

```
UT2mat(x, diag = TRUE)
```

**Arguments**

x                    Vector of upper triangular values  
 diag                Are diagonal values included in x? Default: TRUE.

---

var_sq_err	<i>Compute the error between empirical and theoretical variance of covariance matrix elements</i>
------------	---

---

**Description**

Compute the error between empirical and theoretical variance of covariance matrix elements

**Usage**

```
var_sq_err(nu, p, var_ij, xbar_ij, xbar_ii, xbar_jj)
```

**Arguments**

nu	Inverse Wishart degrees of freedom parameter
p	Matrix dimension for IW distribution
var_ij	Empirical between-subject variance of covariance matrices at element (i,j)
xbar_ij	Empirical mean of covariance matrices at element (i,j)
xbar_ii	Empirical mean of covariance matrices at the ith diagonal element
xbar_jj	Empirical mean of covariance matrices at the jth diagonal element

**Value**

Squared difference between the empirical and theoretical IW variance of covariance matrices at element (i,j)

---

var_sq_err_constrained	<i>Compute the overall error between empirical and theoretical variance of CORRELATION matrix elements</i>
------------------------	--

---

**Description**

Compute the overall error between empirical and theoretical variance of CORRELATION matrix elements

**Usage**

```
var_sq_err_constrained(nu, p, var, xbar, M = 10000)
```

**Arguments**

nu	Inverse Wishart degrees of freedom parameter
p	Matrix dimension for IW distribution
var	Empirical between-subject variances of CORRELATION matrix (upper triangle)
xbar	Empirical mean of CORRELATION matrix (upper triangle)
M	Penalty to assign if theoretical variance is smaller than empirical variance

**Value**

Sum of squared difference between the empirical and theoretical IW variance of CORRELATION matrix, but with constraint that theoretical variances must not be smaller than empirical variances

# Index

activations, [3](#)

bdiag\_m2, [4](#)

Chol\_samp\_fun, [5](#)

dim\_reduce, [5](#)

EM\_templateICA, [6](#)

estimate.ESS, [8](#)

estimate\_nu, [8](#)

estimate\_nu\_matrix, [9](#)

estimate\_template, [9](#), [15](#), [39](#)

estimate\_template\_FC, [15](#)

estimate\_template\_from\_DR, [15](#)

export\_template, [14](#), [16](#)

getInvCovAR, [17](#)

groupICA.cifti, [17](#)

is\_surf, [21](#)

IW\_var, [19](#)

IW\_var\_cor, [20](#)

lik, [20](#)

make\_mesh, [7](#), [21](#), [41](#), [43](#)

make\_mesh\_2D, [21](#)

make\_surf, [12](#), [21](#), [39](#)

norm\_BOLD, [18](#), [22](#)

orthonorm, [23](#)

plot.template.cifti, [24](#)

plot.template.gifti, [24](#)

plot.template.matrix, [25](#)

plot.template.nifti, [26](#)

plot.tICA.cifti, [27](#)

plot.tICA.matrix, [27](#)

plot.tICA.nifti, [28](#)

plot.tICA\_act.cifti, [28](#)

print.summary.template.cifti  
(summary.template.cifti), [30](#)

print.summary.template.gifti  
(summary.template.gifti), [31](#)

print.summary.template.matrix  
(summary.template.matrix), [32](#)

print.summary.template.nifti  
(summary.template.nifti), [32](#)

print.summary.tICA.cifti  
(summary.tICA.cifti), [33](#)

print.summary.tICA.matrix  
(summary.tICA.matrix), [34](#)

print.summary.tICA.nifti  
(summary.tICA.nifti), [35](#)

print.summary.tICA\_act.cifti  
(summary.tICA\_act.cifti), [35](#)

print.summary.tICA\_act.matrix  
(summary.tICA\_act.matrix), [36](#)

print.summary.tICA\_act.nifti  
(summary.tICA\_act.nifti), [37](#)

print.template.cifti  
(summary.template.cifti), [30](#)

print.template.gifti  
(summary.template.gifti), [31](#)

print.template.matrix  
(summary.template.matrix), [32](#)

print.template.nifti  
(summary.template.nifti), [32](#)

print.tICA.cifti (summary.tICA.cifti),  
[33](#)

print.tICA.matrix  
(summary.tICA.matrix), [34](#)

print.tICA.nifti (summary.tICA.nifti),  
[35](#)

print.tICA\_act.cifti  
(summary.tICA\_act.cifti), [35](#)

print.tICA\_act.matrix  
(summary.tICA\_act.matrix), [36](#)

print.tICA\_act.nifti

(summary.tICA\_act.nifti), [37](#)  
pw\_estimate, [29](#)  
resample\_template, [29](#)  
smooth\_cifti, [19](#)  
sqrt\_XtX, [30](#)  
summary.template.cifti, [30](#)  
summary.template.gifti, [31](#)  
summary.template.matrix, [32](#)  
summary.template.nifti, [32](#)  
summary.tICA.cifti, [33](#)  
summary.tICA.matrix, [34](#)  
summary.tICA.nifti, [35](#)  
summary.tICA\_act.cifti, [35](#)  
summary.tICA\_act.matrix, [36](#)  
summary.tICA\_act.nifti, [37](#)  
templateICA, [3](#), [38](#)  
UpdateTheta\_templateICA, [42](#)  
UT2mat, [43](#)  
var\_sq\_err, [44](#)  
var\_sq\_err\_constrained, [44](#)