

# Package ‘templateICAr’

January 27, 2023

**Type** Package

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

**Version** 0.6.2

**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, excursions, expm, fMRItools (>= 0.2.2), ica, Matrix, matrixStats, methods, pesel, Rcpp, SQUAREM, stats

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

**RoxygenNote** 7.2.3

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

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

**LinkingTo** RcppEigen, Rcpp

**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 2023-01-27 09:10:02 UTC

## R topics documented:

activations . . . . .	3
dual_reg . . . . .	4
EM_FCtemplateICA . . . . .	5
EM_templateICA . . . . .	6
estimate_nu_matrix . . . . .	8
estimate_template . . . . .	8
estimate_template_from_DR . . . . .	13
export_template . . . . .	14
Gibbs_AS_posteriorCPP . . . . .	15
groupICA.cifti . . . . .	16
IW_var . . . . .	17
make_mesh . . . . .	18
make_mesh_2D . . . . .	19
norm_BOLD . . . . .	19
PCA . . . . .	20
plot.template.cifti . . . . .	21
plot.template.gifti . . . . .	21
plot.template.matrix . . . . .	22
plot.template.nifti . . . . .	23
plot.tICA.cifti . . . . .	24
plot.tICA.matrix . . . . .	24
plot.tICA.nifti . . . . .	25
plot.tICA_act.cifti . . . . .	25
resample_template . . . . .	26
summary.template.cifti . . . . .	26
summary.template.gifti . . . . .	27
summary.template.matrix . . . . .	28
summary.template.nifti . . . . .	29
summary.tICA.cifti . . . . .	29
summary.tICA.matrix . . . . .	30
summary.tICA.nifti . . . . .	31
summary.tICA_act.cifti . . . . .	32
summary.tICA_act.matrix . . . . .	32
templateICA . . . . .	33
UpdateTheta_FCtemplateICA . . . . .	36
UpdateTheta_FCtemplateICAcpp . . . . .	37
UpdateTheta_templateICA . . . . .	38
var_sq_err . . . . .	39

Index

41

---

activations                      *Activations of (s)tICA*

---

**Description**

Identify areas of activation in each independent component map

**Usage**

```

activations(
  tICA,
  u = 0,
  alpha = 0.01,
  type = ">",
  method_p = "BH",
  verbose = FALSE,
  which.ICs = NULL,
  deviation = FALSE
)

```

**Arguments**

tICA	Fitted (spatial) template ICA object from <a href="#">templateICA</a> .
u	Activation threshold. Default: 0.
alpha	Significance level for joint PPM. Default: 0.01.
type	Type of region. Default: ">" (positive excursion region).
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 <code>help(p.adjust)</code> . Default: "BH" (Benjamini & Hochberg, i.e. the false discovery rate).
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 and the joint and marginal PPMs for each IC. 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)
```

dual\_reg

*Dual Regression***Description**

Dual Regression

**Usage**

```
dual_reg(
  BOLD,
  GICA,
  scale = c("global", "local", "none"),
  scale_sm_xifti = NULL,
  scale_sm_FWHM = 2,
  detrend_DCT = 0,
  center_Bcols = FALSE,
  normA = FALSE
)
```

**Arguments**

BOLD	Subject-level fMRI data matrix ( $V \times T$ ). Rows will be centered.
GICA	Group-level independent components ( $V \times Q$ )
scale	"global" (default), "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.
scale_sm_xifti, scale_sm_FWHM	Only applies if scale=="local" 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.
detrend_DCT	Detrend the data? This is an integer number of DCT bases to use for detrending. If 0 (default), do not detrend.
center_Bcols	Center BOLD across columns (each image)? This is equivalent to performing global signal regression. Default: FALSE.
normA	Scale each IC timeseries (column of $A$ ) in the dual regression estimates? Default: FALSE (not recommended). Note that the product $A \times S$ remains the same with either option.

**Value**

A list containing the subject-level independent components  $\mathbf{S}$  ( $V \times Q$ ), and subject-level mixing matrix  $\mathbf{A}$  ( $T \times Q$ ).

**Examples**

```
nT <- 30
nV <- 400
nQ <- 7
mU <- matrix(rnorm(nV*nQ), nrow=nV)
mS <- mU %*% diag(seq(nQ, 1)) %*% matrix(rnorm(nQ*nT), nrow=nQ)
BOLD <- mS + rnorm(nV*nT, sd=.05)
GICA <- mU
dual_reg(BOLD=BOLD, GICA=mU, scale="local")
```

EM\_FCtemplateICA

*EM Algorithm for FC Template ICA Model***Description**

EM Algorithm for FC Template ICA Model

**Usage**

```
EM_FCtemplateICA(
  template_mean,
  template_var,
  template_FC,
  prior_params = c(0.001, 0.001),
  BOLD,
  AS_0,
  maxiter = 100,
  epsilon = 0.01,
  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
template_FC	(list) Parameters of functional connectivity template
prior_params	Alpha and beta parameters of IG prior on $\tau^2$ (error variance)
BOLD	( $V \times T$ matrix) preprocessed fMRI data

AS_0	(list) initial guess at latent variables: A ( $T \times Q$ mixing matrix), and S ( $Q \times V$ matrix of spatial ICs)
maxiter	Maximum number of EM iterations. Default: 100.
epsilon	Smallest proportion change in parameter estimates between iterations. Default: 0.01.
verbose	If TRUE, display progress of algorithm. Default: FALSE.

### Details

EM\_FCtemplateICA implements the expectation-maximization (EM) algorithm for the functional connectivity (FC) template ICA model

### Value

A list: theta (list of final parameter estimates), subICmean (estimates of subject-level ICs), subICvar (variance of subject-level ICs), mixing\_mean (estimates of subject-level mixing matrix), mixing\_var (variance of subject-level mixing matrix), success (flag indicating convergence (TRUE) or not (FALSE))

---

EM_templateICA	<i>EM Algorithms for Template ICA Models</i>
----------------	--

---

### Description

EM Algorithms for Template ICA Models

### Usage

```
EM_templateICA.spatial(
  template_mean,
  template_var,
  meshes,
  BOLD,
  theta0,
  C_diag,
  maxiter = 100,
  usePar = FALSE,
  epsilon = 0.01,
  verbose = FALSE
)

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

```

    C_diag,
    maxiter = 100,
    epsilon = 0.01,
    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. 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 over voxels? Default: FALSE. Can be the number of cores to use or TRUE, which will use the number on the PC minus two. Not implemented yet for spatial template ICA.
epsilon	Smallest proportion change between iterations. Default: 0.01.
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\_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,
  inds = NULL,
  scale = c("global", "local", "none"),
  scale_sm_surfL = NULL,
  scale_sm_surfR = NULL,
  scale_sm_FWHM = 2,
  detrend_DCT = 0,
  center_Bcols = FALSE,
  normA = FALSE,
  Q2 = 0,
  Q2_max = NULL,
  brainstructures = c("left", "right"),
  mask = NULL,
```



```
    keep_DR = FALSE,
    FC = FALSE,
    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("global", "local", "none"),
  scale_sm_surfl = NULL,
  scale_sm_surfr = NULL,
  scale_sm_FWHM = 2,
  detrend_DCT = 0,
  center_Bcols = FALSE,
  normA = FALSE,
  Q2 = 0,
  Q2_max = NULL,
  brainstructures = c("left", "right"),
  keep_DR = 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("global", "local", "none"),
  scale_sm_surfl = NULL,
  scale_sm_surfr = NULL,
  scale_sm_FWHM = 2,
  detrend_DCT = 0,
  center_Bcols = FALSE,
  normA = FALSE,
  Q2 = 0,
```

```

Q2_max = NULL,
brainstructures = c("left", "right"),
keep_DR = 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("global", "local", "none"),
  detrend_DCT = 0,
  center_Bcols = FALSE,
  normA = FALSE,
  Q2 = 0,
  Q2_max = NULL,
  mask = NULL,
  keep_DR = 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** Group ICA maps in a format compatible with BOLD. Can also be a (vectorized) numeric matrix ( $V \times Q$ ) no matter the format of BOLD. Its columns will be centered.

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>"global" (default), "local", or "none". Global scaling will divide the entire data matrix by the mean image standard deviation (<math>\text{mean}(\text{sqrt}(\text{rowVars}(\text{BOLD})))</math>). Local scaling will divide each data location's time series by its estimated standard deviation.</p>
scale_sm_surfl, scale_sm_surfr, scale_sm_FWHM	<p>Only applies if scale=="local" 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 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>
detrend_DCT	<p>Detrend the data? This is an integer number of DCT bases to use for detrending. If 0 (default), do not detrend.</p>
center_Bcols	<p>Center BOLD across columns (each image)? This is equivalent to performing global signal regression. Default: FALSE.</p>
normA	<p>Scale each IC timeseries (column of <math>A</math>) in the dual regression estimates? Default: FALSE (not recommended). Note that the product <math>A \times S</math> remains the same with either option.</p>
Q2, Q2_max	<p>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.</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 (default) to skip denoising.</p> <p>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 <math>T * .75 - Q</math> where <math>T</math> is the minimum number of timepoints in each fMRI scan and <math>Q</math> is the number of group ICs. If NULL (default), Q2_max will be set to <math>T * .50 - Q</math>, rounded.</p>
brainstructures	<p>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("left", "right")</code> (cortical surface only).</p>
mask	<p>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.</p>

keep_DR	Keep the DR estimates? If FALSE (default), do not save the DR estimates and only return the templates. If TRUE, the DR estimates 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: FALSE (not fully supported yet.)
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 if isTRUE(keep\_DR).

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 <- 30
nV <- 400
nQ <- 7
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)

## Not run:
estimate_template(
  run1_cifti_fnames, run2_cifti_fnames,
  gICA_cifti_fname, brainstructures="all",
  scale="local", detrend_DCT=7, Q2=NULL, varTol=10
)

## End(Not run)

```

---

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, "xifti", or "nifti" format, to match the format of the original BOLD data. Otherwise, the paths to the new files specified by out\_fname.

### 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)
```

---

Gibbs\_AS\_posteriorCPP *Use a Gibbs sampler for the A and S variables (E-step of the EM)*

---

### Description

Use a Gibbs sampler for the A and S variables (E-step of the EM)

### Usage

```
Gibbs_AS_posteriorCPP(
  nsamp,
  nburn,
  template_mean,
  template_var,
  S,
  G,
  tau_v,
  Y,
  alpha,
  final,
  return_samp
)
```

### Arguments

nsamp	the number of posterior samples to output after burn-in
nburn	the number of posterior samples to throw away before saving
template_mean	a matrix with dimensions $V \times Q$ giving the mean value of the independent components
template_var	a matrix with dimensions $V \times Q$ giving the variance of the independent components
S	a matrix with dimensions $V \times Q$ of subject independent components
G	a $Q \times Q$ matrix of the prior covariance of A
tau_v	a length $V$ vector with noise variance for each data location
Y	a matrix with dimensions $V \times T$ of observed BOLD data
alpha	a length $Q$ vector of the prior mean of all rows of A
final	a boolean. Should posterior samples be returned instead of summary measures?
return_samp	a boolean. Should posterior samples be returned?

### Value

List with estimates for A, S, and possibly other quantities

---

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 = c("left", "right"),
  num_PCs = 100,
  num_ICs,
  max_rows_GPCA = 10000,
  center_Bcols = FALSE,
  scale = c("global", "local", "none"),
  scale_sm_FWHM = 2,
  detrend_DCT = 0,
  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" (subcor-



	tical and cerebellar gray matter). Can also be "all" (obtain all three brain structures). Default: c("left", "right") (cortical surface only).
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 <code>length(cifti_fnames) * num_PCs</code> rows.
center_Bcols, scale, scale_sm_FWHM, detrend_DCT	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, but do not detrend.
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 <a href="#">smooth_cifti</a> . The defaults here are the same. Note that <code>smooth_zeroes_as_NA</code> will control both of the corresponding surface and volume arguments to <a href="#">smooth_cifti</a> . These arguments only apply if <code>smooth</code> .

### 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)

---

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 <a href="#">make_surf</a> and <a href="#">is.surf</a> .
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("global", "local", "none"),  
  scale_sm_xiffti = NULL,  
  scale_sm_FWHM = 2,  
  detrend_DCT = 0  
)
```

**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	"global" (default), "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.
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.
detrend_DCT	Detrend the data? This is an integer number of DCT bases to use for detrending. If 0 (default), do not detrend.

**Value**

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

---

PCA

PCA

---

**Description**

Efficient PCA for a tall matrix (many more rows than columns). Uses the SVD of the covariance matrix.

**Usage**

`PCA(X, center = TRUE, Q = NULL, Q_max = 100, nV = 0)`

**Arguments**

X	$V \times T$ fMRI timeseries data matrix, centered by columns.
center	Center the columns of X? Default: TRUE. Set to FALSE if already centered.
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.
nV	Number of principal directions to obtain. Default: 0. Can also be "Q" to set equal to the value of Q. Note that setting this value less than Q does not speed up computation time, but does save on memory. Note that the directions will be with respect to X, not its covariance matrix.

**Value**

The SVD decomposition

---

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

---

<code>resample_template</code>	<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.

---

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

---

**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**

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

**Value**

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

Nothing, invisibly.

Nothing, invisibly.

---

`summary.tICA.cifti`*Summarize a "tICA.cifti" object*

---

**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.

---

 templateICA

*Template ICA*


---

**Description**

Perform template independent component analysis (ICA) using expectation-maximization (EM).

**Usage**

```
templateICA(
  BOLD,
  template,
  tvar_method = c("non-negative", "unbiased"),
  scale = c("global", "local", "none"),
  scale_sm_surfl = NULL,
  scale_sm_surfr = NULL,
  scale_sm_FWHM = 2,
  detrend_DCT = 0,
  center_Bcols = FALSE,
  normA = FALSE,
  Q2 = NULL,
  Q2_max = NULL,
  brainstructures = c("left", "right"),
  mask = NULL,
```

```

time_inds = NULL,
varTol = 1e-06,
spatial_model = NULL,
resamp_res = NULL,
rm_mwall = TRUE,
reduce_dim = TRUE,
maxiter = 100,
epsilon = 0.01,
kappa_init = 0.2,
usePar = FALSE,
verbose = TRUE
)

```

### Arguments

BOLD	<p>Vector of subject-level fMRI data in one of the following formats: CIFTI file paths, "xiffti" objects, NIFTI file paths, "nifti" objects, or <math>V \times T</math> numeric matrices, where <math>V</math> is the number of data locations and <math>T</math> is the number of time-points.</p> <p>If multiple BOLD data are provided, they will be independently centered, scaled, and detrended (if applicable), and then they will be concatenated together followed by denoising (if applicable) and computing the initial dual regression estimate.</p>
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" (default), "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.
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: 2).</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>
detrend_DCT	Detrend the data? This is an integer number of DCT bases to use for detrending. If 0 (default), do not detrend.

center_Bcols	Center BOLD across columns (each image)? This is equivalent to performing global signal regression. Default: FALSE.
normA	Scale each IC timeseries (column of $A$ ) in the dual regression estimates? Default: FALSE (not recommended). Note that the product $A \times S$ remains the same with either option.
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 (default), or zero to skip denoising.</p> <p>If is.null(Q2), use Q2_max to specify the maximum number of nuisance ICs that should be estimated by PESEL. Q2_max 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 (default), Q2_max will be set to <math>T * .50 - Q</math>, rounded.</p>
brainstructures	<p>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: c("left", "right") (cortical surface only).</p>
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. If NULL (default), use all volumes. Subsetting is performed before any centering, scaling, detrending, and denoising.
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. 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, spatial_model 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 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.</p> <p>If BOLD represents NIFTI-format data, spatial modeling is not yet available.</p> <p>If BOLD is a numeric matrix, spatial_model should be a list of meshes (see <a href="#">make_mesh</a>).</p>

resamp_res	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.
rm_mwall	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.
reduce_dim	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.
maxiter	Maximum number of EM iterations. Default: 100.
epsilon	Smallest proportion change between iterations. Default: .01.
kappa_init	Starting value for kappa. Default: 0.2.
usePar	Parallelize the computation over data locations? Default: FALSE. Can be the number of cores to use or TRUE, which will use the number on the PC minus two.
verbose	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  $\mathbf{S}$ ; subjICse, the standard errors of  $\mathbf{S}$ ; the mask of locations without template values due to too many low variance or missing values; 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\_FCtemplateICA

*Update FC Template ICA parameters (tau\_sq, alpha, G)*

---

### Description

Update FC Template ICA parameters (tau\_sq, alpha, G)

**Usage**

```
UpdateTheta_FCtemplateICA(
  template_mean,
  template_var,
  template_FC,
  prior_params,
  BOLD,
  post_sums,
  verbose
)
```

**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
template_FC	(list) Parameters of functional connectivity template
prior_params	Alpha and beta parameters of IG prior on $\tau^2$ (error variance)
BOLD	( $V \times Q$ matrix) dimension-reduced fMRI data
post_sums	TBD
verbose	If TRUE, display progress of algorithm. Default: FALSE.

---

UpdateTheta\_FCtemplateICAcpp

*Update parameters (M-step of the EM)*

---

**Description**

Update parameters (M-step of the EM)

**Usage**

```
UpdateTheta_FCtemplateICAcpp(
  template_mean,
  template_var,
  template_FC,
  G,
  prior_params,
  BOLD,
  Y_sq_sum,
  post_sums,
  sigma2_alpha,
  verbose
)
```

**Arguments**

template_mean	a matrix with dimensions $V \times Q$ giving the mean value of the independent components
template_var	a matrix with dimensions $V \times Q$ giving the variance of the independent components
template_FC	a list with two elements: psi, a $Q \times Q$ matrix, and nu, a scalar. These two values are the parameters of the Wishart prior on G
G	a $Q \times Q$ matrix of the prior covariance of A
prior_params	a length 2 vector with the prior parameters for tau_v
BOLD	a $V \times T$ matrix of BOLD values
Y_sq_sum	a length V vector with the sum of squared BOLD values at each data location
post_sums	a list of posterior summary statistics including the named summaries AS_sq_sum, yAS_sum, A_sum, and AtA_sum
sigma2_alpha	a scalar multiplier for the prior variance of alpha
verbose	a boolean. Should messages be generated and output?

**Value**

A list with quantities tau\_sq, alpha, and G

---

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,
  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,
    verbose
)

```

### 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
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".

### Value

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

---

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)



# Index

activations, [3](#)

dual\_reg, [4](#)

EM\_FCtemplateICA, [5](#)

EM\_templateICA, [6](#)

estimate\_nu\_matrix, [8](#)

estimate\_template, [8](#), [34](#)

estimate\_template\_from\_DR, [13](#)

export\_template, [12](#), [14](#)

Gibbs\_AS\_posteriorCPP, [15](#)

groupICA.cifti, [16](#)

is\_surf, [18](#)

IW\_var, [17](#)

make\_mesh, [7](#), [18](#), [35](#), [39](#)

make\_mesh\_2D, [19](#)

make\_surf, [11](#), [18](#), [34](#)

norm\_BOLD, [17](#), [19](#)

PCA, [20](#)

plot.template.cifti, [21](#)

plot.template.gifti, [21](#)

plot.template.matrix, [22](#)

plot.template.nifti, [23](#)

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

plot.tICA.matrix, [24](#)

plot.tICA.nifti, [25](#)

plot.tICA\_act.cifti, [25](#)

print.summary.template.cifti  
(summary.template.cifti), [26](#)

print.summary.template.gifti  
(summary.template.gifti), [27](#)

print.summary.template.matrix  
(summary.template.matrix), [28](#)

print.summary.template.nifti  
(summary.template.nifti), [29](#)

print.summary.tICA.cifti  
(summary.tICA.cifti), [29](#)

print.summary.tICA.matrix  
(summary.tICA.matrix), [30](#)

print.summary.tICA.nifti  
(summary.tICA.nifti), [31](#)

print.summary.tICA\_act.cifti  
(summary.tICA\_act.cifti), [32](#)

print.summary.tICA\_act.matrix  
(summary.tICA\_act.matrix), [32](#)

print.template.cifti  
(summary.template.cifti), [26](#)

print.template.gifti  
(summary.template.gifti), [27](#)

print.template.matrix  
(summary.template.matrix), [28](#)

print.template.nifti  
(summary.template.nifti), [29](#)

print.tICA.cifti (summary.tICA.cifti),  
[29](#)

print.tICA.matrix  
(summary.tICA.matrix), [30](#)

print.tICA.nifti (summary.tICA.nifti),  
[31](#)

print.tICA\_act.cifti  
(summary.tICA\_act.cifti), [32](#)

print.tICA\_act.matrix  
(summary.tICA\_act.matrix), [32](#)

resample\_template, [26](#)

smooth\_cifti, [17](#)

summary.template.cifti, [26](#)

summary.template.gifti, [27](#)

summary.template.matrix, [28](#)

summary.template.nifti, [29](#)

summary.tICA.cifti, [29](#)

summary.tICA.matrix, [30](#)

summary.tICA.nifti, [31](#)

summary.tICA\_act.cifti, [32](#)

`summary.tICA_act.matrix`, [32](#)

`templateICA`, [3](#), [33](#)

`UpdateTheta_FCtemplateICA`, [36](#)

`UpdateTheta_FCtemplateICApp`, [37](#)

`UpdateTheta_templateICA`, [38](#)

`var_sq_err`, [39](#)