

# Package ‘SparseICA’

May 7, 2026

**Type** Package

**Title** Sparse Independent Component Analysis

**Version** 0.1.4

**Date** 2025-01-15

**Author** Zihang Wang [aut, cre] (ORCID: <<https://orcid.org/0000-0001-9032-5412>>),  
Irina Gaynanova [aut] (ORCID: <<https://orcid.org/0000-0002-4116-0268>>),  
Aleksandr Aravkin [aut] (ORCID:  
<<https://orcid.org/0000-0002-1875-1801>>),  
Benjamin Risk [aut] (ORCID: <<https://orcid.org/0000-0003-1090-0777>>)

**Maintainer** Zihang Wang <[zhwang0378@gmail.com](mailto:zhwang0378@gmail.com)>

**BugReports** <https://github.com/thebrisklab/SparseICA/issues>

**URL** <https://github.com/thebrisklab/SparseICA>

**Description** Provides an implementation of the Sparse ICA method in Wang et al. (2024) <[doi:10.1080/01621459.2024.2370593](https://doi.org/10.1080/01621459.2024.2370593)> for estimating sparse independent source components of cortical surface functional MRI data, by addressing a non-smooth, non-convex optimization problem through the relax-and-split framework. This method effectively balances statistical independence and sparsity while maintaining computational efficiency.

**License** GPL-3

**LinkingTo** Rcpp, RcppArmadillo

**RoxygenNote** 7.3.1

**Encoding** UTF-8

**Imports** Rcpp (>= 1.0.13), MASS (>= 7.3-58), irlba (>= 2.3.5), clue (>= 0.3), ciftiTools (>= 0.16), parallel (>= 4.1)

**Depends** R (>= 4.1.0)

**LazyData** true

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2025-01-29 18:50:02 UTC

## Contents

BIC_sparseICA . . . . .	2
create_group_list . . . . .	4
est.M.ols . . . . .	5
example_sim123 . . . . .	5
gen.inits . . . . .	6
gen_groupPC . . . . .	6
givens.rotation . . . . .	7
group_sparseICA . . . . .	8
matchICA . . . . .	10
relax_and_split_ICA . . . . .	11
signchange . . . . .	13
soft_thresh_R . . . . .	13
sparseICA . . . . .	14
theta2W . . . . .	16
whitener . . . . .	17
<b>Index</b>	<b>18</b>

---

BIC\_sparseICA

*BIC-like Criterion for Tuning Parameter Selection in Sparse ICA*

---

### Description

This function uses a BIC-like criterion to select the optimal tuning parameter  $\nu$  for Sparse ICA.

### Usage

```
BIC_sparseICA(
  xData,
  n.comp,
  nu_list = seq(0.1, 4, 0.1),
  whiten = c("eigenvec", "sqrtprec", "none"),
  lngca = FALSE,
  orth.method = c("svd", "givens"),
  method = c("C", "R"),
  use_irlba = TRUE,
  eps = 1e-06,
  maxit = 500,
  verbose = FALSE,
  col.stand = TRUE,
  row.stand = FALSE,
  iter.stand = 0,
  BIC_plot = FALSE
)
```

**Arguments**

<code>xData</code>	A numeric matrix of input data with dimensions $P \times T$ , where $P$ is the number of features and $T$ is the number of samples.
<code>n.comp</code>	An integer specifying the number of components to estimate.
<code>nu_list</code>	A numeric vector specifying the list of candidate tuning parameters. Default is <code>seq(0.1, 4, 0.1)</code> .
<code>whiten</code>	A character string specifying the method for whitening the input <code>xData</code> . Options are "eigenvec", "sqrtprec", or "none". Default is "eigenvec".
<code>lngca</code>	A logical value indicating whether to perform Linear Non-Gaussian Component Analysis (LNGCA). Default is FALSE.
<code>orth.method</code>	A character string specifying the method for generating initial values of the $U$ matrix. Default is "svd".
<code>method</code>	A character string specifying the computation method. If "C" (default), C code is used for Sparse ICA to improve performance. If "R", computations are performed entirely in R.
<code>use_irlba</code>	A logical value indicating whether to use the <code>irlba</code> method for fast truncated Singular Value Decomposition (SVD) during whitening. This can improve memory efficiency for intermediate datasets. Default is TRUE.
<code>eps</code>	A numeric value specifying the convergence threshold. Default is $1e-6$ .
<code>maxit</code>	An integer specifying the maximum number of iterations for the Sparse ICA method using Laplace density. Default is 500.
<code>verbose</code>	A logical value indicating whether to print convergence information during execution. Default is FALSE.
<code>col.stand</code>	A logical value indicating whether to standardize columns. For each column, the mean of the entries in the column equals 0, and the variance of the entries in the column equals 1. Default is TRUE.
<code>row.stand</code>	A logical value indicating whether to standardize rows. For each row, the mean of the entries in the row equals 0, and the variance of the entries in the row equals 1. Default is FALSE.
<code>iter.stand</code>	An integer specifying the number of iterations for achieving both row and column standardization when <code>col.stand = TRUE</code> and <code>row.stand = TRUE</code> . Default is 5.
<code>BIC_plot</code>	A logical value indicating whether to generate a plot showing the trace of BIC values for different <code>nu</code> candidates. Default is FALSE.

**Value**

A list containing the following elements:

`BIC` A numeric vector of BIC values corresponding to each candidate `nu` in `nu_list`.

`nu_list` A numeric vector of candidate tuning parameter values.

`best_nu` The optimal `nu` selected based on the BIC-like criterion.

**Examples**

```
#get simulated data
data(example_sim123)

select_sparseICA = BIC_sparseICA(xData = example_sim123$xmat, n.comp = 3,
  method="C", BIC_plot = TRUE, verbose = TRUE, nu_list = seq(0.1,4,0.1))

(my_nu = select_sparseICA$best_nu)
```

---

create_group_list	<i>Create a List of fMRI Files for Group ICA Analysis</i>
-------------------	---

---

**Description**

This function scans a BIDS-formatted directory for subject-specific fMRI files that match a specified pattern and returns a list of these files for use in group ICA analysis.

**Usage**

```
create_group_list(bids_path, pattern = "task-rest.*\\.dtseries\\.nii$")
```

**Arguments**

bids_path	A character string specifying the path to the root directory of the BIDS-formatted dataset. This directory should contain subject folders (e.g., sub-*).
pattern	A character string specifying the pattern to match fMRI files. The default is "task-rest.*\\.dtseries\\.nii\$".

**Value**

A named list where each element corresponds to a subject directory and contains a vector of matched fMRI file paths. The names of the list are the subject IDs.

**Examples**

```
# Example usage:
# Assuming `bids_dir` is the path to a BIDS dataset:
# group_list <- create_group_list(bids_path = bids_dir, pattern = "task-rest.*\\.dtseries\\.nii$")
# Print the structure of the list:
# str(group_list)
```

---

est.M.ols	<i>Estimate mixing matrix from estimates of components</i>
-----------	--

---

**Description**

Estimate mixing matrix from estimates of components

**Usage**

```
est.M.ols(sData, xData, intercept = TRUE)
```

**Arguments**

sData	S Dimension: P x Q
xData	X Dimension: P x T
intercept	default = TRUE

**Value**

a mixing matrix M, dimension Q x T.

---

example_sim123	<i>Example sim123 Dataset</i>
----------------	-------------------------------

---

**Description**

A simple dataset for demonstration purposes.

**Usage**

```
example_sim123
```

**Format**

A list containing 3 data frames:

**smat** A 1089 x 3 numeric matrix of the true source signals. Each column is an 33 x 33 image.

**mmat** A 3 x 50 numeric mixing matrix of the true time series. Each row is a time series of corresponding column in smat.

**xmat** A 1089 x 50 numeric matrix of the simulated data. Each column is the simulated mixed signal at a time point.

**Examples**

```
data(example_sim123)
str(example_sim123)
```

---

gen.inits	<i>Function for generating random starting points</i>
-----------	---

---

**Description**

Function for generating random starting points

**Usage**

```
gen.inits(p, d, runs, orth.method = c("svd", "givens"))
```

**Arguments**

p	The number of rows.
d	The number of columns.
runs	The number of random starts.
orth.method	The method used for generating initial values of U matrix. The default is "svd".

**Value**

A list of random initialization of matrices.

---

gen_groupPC	<i>Generate Group-Level Principal Components (PCs) for fMRI Data</i>
-------------	--

---

**Description**

This function computes subject-level principal components (PCs) from fMRI data and performs a group-level PCA for dimension reduction, designed for cortical surface fMRI data in BIDS format.

**Usage**

```
gen_groupPC(
  bids_path,
  subj_list,
  n.comp = 30,
  ncore = 1,
  npc = 85,
  iter_std = 5,
  brainstructures = c("left", "right"),
  verbose = TRUE
)
```

**Arguments**

<code>bids_path</code>	A character string specifying the root directory of the BIDS-formatted dataset.
<code>subj_list</code>	A named list generated from <code>create_group_list</code> containing fMRI file paths for each subject.
<code>n.comp</code>	An integer specifying the number of components to retain during group-level PCA. Default is 30.
<code>ncore</code>	An integer specifying the number of cores to use for parallel processing. Default is 1.
<code>npc</code>	An integer specifying the number of components to retain during subject-level PCA. Default is 85.
<code>iter_std</code>	An integer specifying the number of iterative standardization steps to apply to fMRI data. Default is 5.
<code>brainstructures</code>	A character vector specifying the brain structures to include in the analysis. Options are "left" (left cortex), "right" (right cortex), and/or "subcortical" (subcortex and cerebellum). Can also be "all" (obtain all three brain structures). Default is <code>c("left", "right")</code> .
<code>verbose</code>	A logical value indicating whether to print convergence information during execution. Default is TRUE.

**Details**

NOTE: This function requires the `ciftiTools` package to be installed, and set up the path to the Connectome Workbench folder by `ciftiTools.setOption()`. See the package `ciftiTools` documentation for more information.

**Value**

A numeric matrix containing the group-level principal components, with dimensions determined by the number of retained components (`n.comp`) and the concatenated data across all subjects.

---

`givens.rotation`      *For a given angle theta, returns a d x d Givens rotation matrix*

---

**Description**

For a given angle theta, returns a d x d Givens rotation matrix

**Usage**

```
givens.rotation(theta = 0, d = 2, which = c(1, 2))
```

**Arguments**

<code>theta</code>	The value of theta.
<code>d</code>	The value of d.
<code>which</code>	The value of which.

---

group\_sparseICA      *Perform Group Sparse Independent Component Analysis (Sparse ICA)*

---

### Description

This function performs Sparse ICA on group-level fMRI data. It processes BIDS-formatted fMRI datasets, performs PCA to reduce dimensionality, selects a tuning parameter  $\nu$  (optionally using a BIC-like criterion), and executes Sparse ICA to estimate independent components.

### Usage

```
group_sparseICA(
  bids_path,
  subj_list = NULL,
  nu = "BIC",
  n.comp = 30,
  method = "C",
  ncore = 1,
  npc = 85,
  iter_std = 5,
  brainstructures = c("left", "right"),
  restarts = 40,
  positive_skewness = TRUE,
  use_irlba = TRUE,
  eps = 1e-06,
  maxit = 500,
  BIC_plot = TRUE,
  nu_list = seq(0.1, 4, 0.05),
  verbose = TRUE,
  BIC_verbose = FALSE,
  converge_plot = FALSE
)
```

### Arguments

bids_path	A character string specifying the root directory of the BIDS-formatted dataset.
subj_list	A named list where each element corresponds to a subject and contains vectors of fMRI file names. If NULL, the subject list is generated automatically using <a href="#">create_group_list</a> . Default is NULL.
nu	A numeric value for the tuning parameter, or "BIC" to select nu using a BIC-like criterion. Default is "BIC".
n.comp	An integer specifying the number of components to estimate. Default is 30.
method	A character string specifying the computation method for Sparse ICA. Options are "C" (default) for C-based computation or "R" for R-based computation.
ncore	An integer specifying the number of cores to use for parallel processing. Default is 1.

npc	An integer specifying the number of components to retain during subject-level PCA. Default is 85.
iter_std	An integer specifying the number of iterative standardization steps to apply to fMRI data. Default is 5.
brainstructures	A character vector specifying the brain structures to include in the analysis. Options are "left" (left cortex), "right" (right cortex), and/or "subcortical" (subcortex and cerebellum). Can also be "all" (obtain all three brain structures). Default is c("left", "right").
restarts	An integer specifying the number of random initializations for Sparse ICA. Default is 40.
positive_skewness	A logical value indicating whether to enforce positive skewness on the estimated components. Default is TRUE.
use_irlba	A logical value indicating whether to use the <code>irlba</code> method for fast truncated Singular Value Decomposition (SVD) during whitening. This can improve memory efficiency for intermediate datasets. Default is TRUE.
eps	A numeric value specifying the convergence threshold. Default is 1e-6.
maxit	An integer specifying the maximum number of iterations for Sparse ICA. Default is 500.
BIC_plot	A logical value indicating whether to generate a plot of BIC values for different <code>nu</code> candidates when selecting <code>nu</code> . Default is TRUE.
nu_list	A numeric vector specifying candidate values for <code>nu</code> when selecting it using a BIC-like criterion. Default is seq(0.1, 4, 0.05).
verbose	A logical value indicating whether to print progress messages. Default is TRUE.
BIC_verbose	A logical value indicating whether to print detailed messages during the BIC-based selection of <code>nu</code> . Default is FALSE.
converge_plot	A logical value indicating whether to generate a plot showing the convergence trace during Sparse ICA. Default is FALSE.

## Details

The function operates in four main steps:

1. If `subj_list` is not provided, it creates a list of subject-specific fMRI files using [create\\_group\\_list](#).
2. Performs subject-level PCA using [gen\\_groupPC](#) to reduce data dimensionality.
3. Selects the tuning parameter `nu` using a BIC-like criterion (if `nu = "BIC"`) or uses the provided `nu`.
4. Executes Sparse ICA on the group-level PCs to estimate independent components.

## Value

A list containing the results of the group Sparse ICA analysis, including:

`loglik` The minimal log-likelihood value among the random initializations.

estS A numeric matrix of estimated sparse independent components with dimensions  $P \times Q$ .

estU The estimated U matrix with dimensions  $Q \times Q$ .

whitener The whitener matrix used for data whitening.

converge The trace of convergence for the U matrix.

best\_nu The selected nu value (if nu = "BIC").

BIC A numeric vector of BIC values for each nu candidate (if nu = "BIC").

nu\_list The list of nu candidates used in the BIC-based selection (if nu = "BIC").

### See Also

[create\\_group\\_list](#), [gen\\_groupPC](#), [BIC\\_sparseICA](#), [sparseICA](#)

---

matchICA

*Match ICA results based on L2 distances and Hungarian*

---

### Description

Match ICA results based on L2 distances and Hungarian

### Usage

```
matchICA(S, template, M = NULL)
```

### Arguments

S	loading variable matrix
template	template for match
M	subject score matrix

### Value

the match result

---

relax\_and\_split\_ICA *Relax-and-split ICA Function for Sparse ICA wrapper*


---

### Description

This function performs Sparse Independent Component Analysis (Sparse ICA), implemented in both pure R and RCpp for efficiency.

### Usage

```
relax_and_split_ICA(
  xData,
  n.comp,
  nu = 1,
  U.list = NULL,
  whiten = c("eigenvec", "sqrtprec", "none"),
  lngca = FALSE,
  orth.method = c("svd", "givens"),
  method = c("C", "R"),
  restarts = 40,
  use_irlba = TRUE,
  eps = 1e-06,
  maxit = 500,
  verbose = FALSE,
  converge_plot = FALSE,
  col.stand = TRUE,
  row.stand = FALSE,
  iter.stand = 5,
  positive_skewness = TRUE
)
```

### Arguments

xData	A numeric matrix of input data with dimensions $P \times T$ , where $P$ is the number of features and $T$ is the number of samples.
n.comp	An integer specifying the number of components to estimate.
nu	A numeric tuning parameter controlling the balance between accuracy and sparsity of the results. It can be selected using a BIC-like criterion or based on expert knowledge. Default is 1.
U.list	An optional matrix specifying the initialization of the U matrix. Default is NULL.
whiten	A character string specifying the method for whitening the input xData. Options are "eigenvec", "sqrtprec", "lngca", or "none". Default is "eigenvec".
lngca	A logical value indicating whether to perform Linear Non-Gaussian Component Analysis (LNGCA). Default is FALSE.
orth.method	A character string specifying the method used for generating initial values for the U matrix. Default is "svd".

<code>method</code>	A character string specifying the computation method. If "C" (default), C code is used for most computations for better performance. If "R", computations are performed entirely in R.
<code>restarts</code>	An integer specifying the number of random initializations for optimization. Default is 40.
<code>use_irlba</code>	A logical value indicating whether to use the <code>irlba</code> method for fast truncated Singular Value Decomposition (SVD) during whitening. This can improve memory efficiency for intermediate datasets. Default is TRUE.
<code>eps</code>	A numeric value specifying the convergence threshold. Default is 1e-6.
<code>maxit</code>	An integer specifying the maximum number of iterations for the Sparse ICA method using Laplace density. Default is 500.
<code>verbose</code>	A logical value indicating whether to print convergence information during execution. Default is FALSE.
<code>converge_plot</code>	A logical value indicating whether to generate a line plot showing the convergence trace. Default is FALSE.
<code>col.stand</code>	A logical value indicating whether to standardize columns. For each column, the mean of the entries in the column equals 0, and the variance of the entries in the column equals 1. Default is TRUE.
<code>row.stand</code>	A logical value indicating whether to standardize rows. For each row, the mean of the entries in the row equals 0, and the variance of the entries in the row equals 1. Default is FALSE.
<code>iter.stand</code>	An integer specifying the number of iterations for achieving both row and column standardization when <code>col.stand = TRUE</code> and <code>row.stand = TRUE</code> . Default is 5.
<code>positive_skewness</code>	A logical value indicating whether to enforce positive skewness on the estimated components. Default is TRUE.

## Value

A list containing the following elements:

`loglik` The minimal log-likelihood value among the random initializations.

`estS` A numeric matrix of estimated sparse independent components with dimensions  $P \times Q$ .

`estU` The estimated U matrix with dimensions  $Q \times Q$ .

`estM` The estimated mixing matrix with dimensions  $Q \times T$ .

`whitener` The whitener matrix used for data whitening.

`converge` Convergence information for the U matrix.

---

signchange	<i>Change the sign of S and M matrices to positive skewness.</i>
------------	--

---

**Description**

Change the sign of S and M matrices to positive skewness.

**Usage**

```
signchange(S, M = NULL)
```

**Arguments**

S	The S matrix with dimension P x Q.
M	The M matrix with dimension Q x T.

**Value**

A list of S and M matrices with positive skewness.

---

soft_thresh_R	<i>Soft-threshold function</i>
---------------	--------------------------------

---

**Description**

Soft-threshold function

**Usage**

```
soft_thresh_R(x, nu = 1, lambda = sqrt(2)/2)
```

**Arguments**

x	The input scalar.
nu	The tuning parameter.
lambda	The lambda parameter of the Laplace density.

---

 sparseICA

*Sparse Independent Component Analysis (Sparse ICA) Function*


---

### Description

This function performs Sparse Independent Component Analysis (Sparse ICA), implemented in both pure R and RCpp for efficiency.

### Usage

```
sparseICA(
  xData,
  n.comp,
  nu = "BIC",
  nu_list = seq(0.1, 4, 0.1),
  U.list = NULL,
  whiten = c("eigenvec", "sqrtprec", "none"),
  lngca = FALSE,
  orth.method = c("svd", "givens"),
  method = c("C", "R"),
  restarts = 40,
  use_irlba = TRUE,
  eps = 1e-06,
  maxit = 500,
  verbose = TRUE,
  BIC_verbose = FALSE,
  converge_plot = FALSE,
  col.stand = TRUE,
  row.stand = FALSE,
  iter.stand = 5,
  positive_skewness = TRUE
)
```

### Arguments

xData	A numeric matrix of input data with dimensions $P \times T$ , where $P$ is the number of features and $T$ is the number of samples.
n.comp	An integer specifying the number of components to estimate.
nu	A positive numeric value or a character "BIC" specifying the tuning parameter controlling the balance between accuracy and sparsity of the results. It can be selected using a BIC-like criterion ("BIC") or based on expert knowledge (a positive number). Default is "BIC".
nu_list	A numeric vector specifying the list of candidate tuning parameters. Default is <code>seq(0.1, 4, 0.1)</code> .
U.list	An optional matrix specifying the initialization of the U matrix. Default is NULL.

<code>whiten</code>	A character string specifying the method for whitening the input <code>xData</code> . Options are "eigenvec", "sqrtprec", "lngca", or "none". Default is "eigenvec".
<code>lngca</code>	A logical value indicating whether to perform Linear Non-Gaussian Component Analysis (LNGCA). Default is FALSE.
<code>orth.method</code>	A character string specifying the method used for generating initial values for the U matrix. Default is "svd".
<code>method</code>	A character string specifying the computation method. If "C" (default), C code is used for most computations for better performance. If "R", computations are performed entirely in R.
<code>restarts</code>	An integer specifying the number of random initializations for optimization. Default is 40.
<code>use_irlba</code>	A logical value indicating whether to use the <code>irlba</code> method for fast truncated Singular Value Decomposition (SVD) during whitening. This can improve memory efficiency for intermediate datasets. Default is TRUE.
<code>eps</code>	A numeric value specifying the convergence threshold. Default is 1e-6.
<code>maxit</code>	An integer specifying the maximum number of iterations for the Sparse ICA method using Laplace density. Default is 500.
<code>verbose</code>	A logical value indicating whether to print convergence information during execution. Default is TRUE.
<code>BIC_verbose</code>	A logical value indicating whether to print BIC selection information. Default is FALSE.
<code>converge_plot</code>	A logical value indicating whether to generate a line plot showing the convergence trace. Default is FALSE.
<code>col.stand</code>	A logical value indicating whether to standardize columns. For each column, the mean of the entries in the column equals 0, and the variance of the entries in the column equals 1. Default is TRUE.
<code>row.stand</code>	A logical value indicating whether to standardize rows. For each row, the mean of the entries in the row equals 0, and the variance of the entries in the row equals 1. Default is FALSE.
<code>iter.stand</code>	An integer specifying the number of iterations for achieving both row and column standardization when <code>col.stand = TRUE</code> and <code>row.stand = TRUE</code> . Default is 5.
<code>positive_skewness</code>	A logical value indicating whether to enforce positive skewness on the estimated components. Default is TRUE.

## Value

A list containing the following elements:

`loglik` The minimal log-likelihood value among the random initializations.

`estS` A numeric matrix of estimated sparse independent components with dimensions  $P \times Q$ .

`estM` The estimated mixing matrix with dimensions  $Q \times T$ .

`estU` The estimated U matrix with dimensions  $Q \times Q$ .

whitener The whitener matrix used for data whitening.  
 converge The trace of convergence for the U matrix.  
 BIC A numeric vector of BIC values corresponding to each candidate nu in nu\_list.  
 nu\_list A numeric vector of candidate tuning parameter values.  
 best\_nu The optimal nu selected based on the BIC-like criterion.

### Examples

```
#get simulated data
data(example_sim123)

my_sparseICA <- sparseICA(xData = example_sim123$xmat, n.comp = 3, nu = "BIC", method="C",
  restarts = 40, eps = 1e-6, maxit = 500, verbose=TRUE)

res_matched <- matchICA(my_sparseICA$estS, example_sim123$smat)

# Visualize the estimated components
oldpar <- par()$mfrow
par(mfrow=c(1,3))
image(matrix(res_matched[,1],33,33))
image(matrix(res_matched[,2],33,33))
image(matrix(res_matched[,3],33,33))
par(mfrow=oldpar)
```

---

 theta2W

*Convert angle vector into orthodox matrix*


---

### Description

Convert angle vector into orthodox matrix

### Usage

```
theta2W(theta)
```

### Arguments

theta A vector of angles theta.

### Value

An orthodox matrix.

---

`whitener`*The function for perform whitening.*

---

**Description**

The function for perform whitening.

**Usage**

```
whitener(X, n.comp = ncol(X), center.row = FALSE, use_irlba = TRUE)
```

**Arguments**

<code>X</code>	The data matrix with dimension $P \times T$ .
<code>n.comp</code>	The number of components.
<code>center.row</code>	Whether to center the row of data. Default is FALSE.
<code>use_irlba</code>	Whether to use the irlba method to perform fast truncated singular value decomposition in whitening step, helpful for memorying intermediate dataset. Default is TRUE.

**Value**

A list including the whitener matrix, the whitened data matrix, and the mean of the input data.

# Index

## \* datasets

- example\_sim123, 5
- BIC\_sparseICA, 2, 10
- create\_group\_list, 4, 8–10
- est.M.ols, 5
- example\_sim123, 5
- gen.inits, 6
- gen\_groupPC, 6, 9, 10
- givens.rotation, 7
- group\_sparseICA, 8
- matchICA, 10
- relax\_and\_split\_ICA, 11
- signchange, 13
- soft\_thresh\_R, 13
- sparseICA, 10, 14
- theta2W, 16
- whitener, 17