

# Package ‘BRIL’

July 2, 2021

**Title** Bootstrap and Refine Iterative Locator

**Version** 1.0.2

**Maintainer** Adrien Brilhault <adrien.brilhault@gmail.com>

**Description** The BRIL algorithm estimates the main mode of bivariate distributions through three principal steps:

- BOOTSTRAP: a recursive depth (or convex body minimizers) trimming to locate a first estimate.
- REFINE: a two-pass outliers filtering, the first relying on euclidean distances to the first estimate and unimodality tests, the second on robust distances and multinormality tests.
- ITERATE: after removing the samples selected in the REFINE step from the global distribution, the same process is reapplied, the first mode corresponding to the iteration with the largest number of samples.

**Depends** R (>= 3.5.0)

**Imports** ddalpna,

depth,  
OjaNP,  
MASS,  
diptest,  
robustbase,  
stats,  
utils,  
nortest,  
ICS,  
graphics,  
grDevices

**Suggests** mvtnorm

**License** GPL-3 | file LICENSE

**URL** <https://adrienbrilhault.github.io/BRIL/>, <https://github.com/adrienbrilhault/BRIL/>

**BugReports** <https://github.com/adrienbrilhault/BRIL/issues>

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.1.1

R topics documented:

bril . . . . .	2
depth_values . . . . .	4
filter_outliers . . . . .	5
median_mv . . . . .	7
median_rec . . . . .	8
plot.BRIL . . . . .	10
plot.BRIL.Filtering . . . . .	11
plot.BRIL.MedianRec . . . . .	13
print.BRIL . . . . .	14
print.BRIL.Filtering . . . . .	15
print.BRIL.MedianRec . . . . .	15
test_multinormality . . . . .	16
test_unimodality . . . . .	17
<b>Index</b>	<b>18</b>

---

bril	<i>Bootstrap and Refine Iterative Locator</i>
------	---

---

Description

Robust Estimate of Mode in Multivariate Distribution

Usage

```
bril(  
  data,  
  maxIterations = NULL,  
  minUnassigned = 0.1,  
  method = "Projection",  
  alpha = 0.5,  
  testUnimodal = "DIP",  
  threshUnimodal = 0.05,  
  distUnimodal = "Euclidean",  
  testNormal = "Mardia",  
  threshNormal = 0.05,  
  distNormal = "MCD",  
  trimmedPerFilteringIteration = 1,  
  exitWhenUnimodal = FALSE,  
  debug = FALSE,  
  warnings = FALSE  
)
```

Arguments

data	Matrix of numerical values containing the observations (one per row, with two columns for X and Y coordinates)
maxIterations	Maximum number of iterations performed by the algorithm (set to NULL or 0 for unlimited number)

minUnassigned	Numerical value between 0 and 1 (default: 0.1), providing the proportion of unassigned samples from data below which the algorithm will terminate
method	Method to use. Valid options are "MCD" and "MVE" for convex body minimizers, or "L2", "Lui", "Mahalanobis", "Oja", "Projection" (default), "Spatial" and "Tukey" for depth functions
alpha	Proportion of samples trimmed at each iteration of the recursive median estimate (numerical value between 0 and 1, default: 0.5), see <a href="#">median_rec()</a>
testUnimodal	Statistical test used for unimodality. Valid options are "DIP" only, see <a href="#">filter_outliers()</a>
threshUnimodal	Threshold of significance for the unimodality test (numerical value between 0 and 1, default: 0.05)
distUnimodal	Distance metric used for ordering the samples in the unimodal filtering. Valid options are "Euclidean" (default), or "MCD", "MVE", and "OGK" for robust distances. "Euclidean" is strongly advised for unimodality tests.
testNormal	Statistical test used for normality. Valid options are "Mardia" (default), "Kurtosis", "Skewness", "KS", "KS-adj", "Shapiro", "Lillie" and "Chisq", see <a href="#">filter_outliers()</a>
threshNormal	Threshold of significance for the normality test (numerical value between 0 and 1, default: 0.05)
distNormal	Distance metric used for ordering the samples in the normal filtering. Valid options are "Euclidean", or "MCD" (default), "MVE", and "OGK" for robust distances. Robust distances are strongly advised for normality tests.
trimmedPerFilteringIteration	Number of samples trimmed at each iteration of the unimodality and normality filtering (default: 1), see <a href="#">filter_outliers()</a>
exitWhenUnimodal	Logical value. TRUE will terminate the execution of the algorithm as soon as an unimodal subset is encountered on the start of a global iteration. FALSE (default) will let that last iteration proceed before terminating
debug	Logical value. TRUE will compute all p.values in the filtering steps (even after they exceed the selection threshold, see <a href="#">plot.BRIL.Filtering()</a> )
warnings	Logical value, to display the warnings and errors caught

## Value

The function returns an S3 object of type BRIL containing the following values:

call	Parameters of the call (contains data, maxIterations, minUnassigned, method, alpha, testUnimodal, threshUnimodal, distUnimodal, testNormal, threshNormal, distNormal, trimmedPerFilteringIteration, and exitWhenUnimodal)
iterations	A list with every global iteration of the algorithm, each containing the two filtering procedures performed: filteringUnimodal and filteringNormal (both being S3 object of class BRIL.Filtering, see <a href="#">filter_outliers()</a> )
nbClusters	Number of groups encountered
labels	Labels of the groups encountered (corresponding to the number of the iteration they were identified in)
clustersCenters	Matrix containing the coordinates of the centers of each group (row-wise)
clustersSizes	Array with the number of samples in each group
mainCluster	Index of the group identified as main mode
mode	Coordinates of the main mode

**See Also**

`plot.BRIL()`, `print.BRIL()`, `filter_outliers()`, `median_rec()`, `median_mv()`, `depth_values()`

**Examples**

```
# Create a sample distribution and run bril() function
XY <- rbind(
  mvtnorm::rmvnorm(300, c(0, 0), diag(2) * 3 - 1),
  mvtnorm::rmvnorm(100, c(15, 20), diag(2)),
  mvtnorm::rmvnorm(150, c(-10, 15), diag(2) * 2 - 0.5),
  mvtnorm::rmvnorm(200, c(5, 5), diag(2) * 200)
)

res <- bril(XY, debug = TRUE)
print(res)

# Plot the mode and groups encountered
plot(res)

# Plot the different iterations (interactive)
## Not run:
plot(res, contents = "iterations", asp = 1)

## End(Not run)

# See ?plot.BRIL() for other plotting examples
```

---

depth\_values

*Depth functions wrapper*

---

**Description**

Computes the depth values with respect to the distribution provided in data of either all the coordinates given in `u`, or all observations from data if `u` is not provided. Depth computations rely on the packages `depth` and `ddalpha`.

**Usage**

```
depth_values(data, u = NULL, method = "Projection", warnings = FALSE)
```

**Arguments**

<code>data</code>	Matrix of numerical values containing the observations (one per row, with two columns for X and Y coordinates)
<code>u</code>	Matrix of numerical values containing the coordinates for which depth values are to be computed (in rows, with two columns for X and Y coordinates). When missing or NULL, depth values will be computed for all observations from data
<code>method</code>	Depth function used. Valid options are "L2", "Lui", "Mahalanobis", "Oja", "Projection" (default), "Spatial", "Tukey"
<code>warnings</code>	Logical value, to display the warnings and error raised by the underlying depth functions

**Value**

Array of numerical values containing the depth of each observation from data, or from u if provided (these values are all set to 0 in the occurrence of errors)

**See Also**

[bril\(\)](#), [filter\\_outliers\(\)](#), [median\\_rec\(\)](#), [median\\_mv\(\)](#)

**Examples**

```
# Illustrative data
XY <- rbind(
  mvtnorm::rmvnorm(300, c(0, 0), diag(2)),
  mvtnorm::rmvnorm(100, c(15, 20), diag(2) * 3 - 1),
  mvtnorm::rmvnorm(150, c(-10, 15), diag(2) * 2 - 0.5),
  mvtnorm::rmvnorm(200, c(5, 5), diag(2) * 200)
)

# Compute depths
D <- depth_values(XY, method = "L2", warnings = TRUE)

# Plot distribution with depth color scale
plotColors <- colorRampPalette(c("maroon4", "steelblue4", "green4", "gold"))(20)
plot(XY, pch = 20, asp = 1, col = plotColors[as.numeric(cut(D, breaks = 20))],
     xlab = "X", ylab = "Y")

# Plot depth values
plot(1:nrow(XY), D, pch = 20, col = plotColors[as.numeric(cut(D, breaks = 20))],
     xlab = "Index", ylab = "Depth")

# Compute depth for a single point
depth_values(XY, c(10, 3), method = "L2")

# Compute depth for three sets of coordinates
depth_values(XY, rbind(c(10, 3), c(65, 8), c(0, 1)), method = "L2")
```

---

filter\_outliers

---

Recursive outlier filtering based on unimodality and multinormality tests

---

**Description**

Recursive outlier filtering based on unimodality and multinormality tests

**Usage**

```
filter_outliers(
  data,
  center,
  test = "Mardia",
  threshold = 0.05,
  distType,
```

```

    trimmedPerIteration = 1,
    debug = FALSE,
    warnings = FALSE
  )

```

### Arguments

data	Matrix of numerical values containing the observations (one per row, with two columns for X and Y coordinates)
center	Coordinates used to compute the distances of the samples and order them (array of numerical two values, for X and Y)
test	Statistical test to use. Valid options are "DIP" for unimodality test, or "Mardia", "Kurtosis", "Skewness", "KS", "KS-adj", "Shapiro", "Lillie", and "Chisq" for multivariate normality test
threshold	Threshold of significance for the statistical test (between 0 and 1, default: 0.05)
distType	Distance metric used to order the samples. Valid options are "Euclidean", "MCD", "MVE", and "OGK". If empty or NULL, "Euclidean" will be automatically selected for unimodality tests, and "MCD" for normality tests
trimmedPerIteration	Number of samples trimmed at each iteration (positive integer, default: 1)
debug	Logical value. TRUE will compute all p.values, even after exceeding the threshold, for plotting purpose (see <a href="#">plot.BRIL.Filtering()</a> )
warnings	Logical value, to display the warnings and errors caught

### Details

For unimodality tests parameter `distType` should be set to "Euclidean" (as the distribution might contain a large amount of outliers). For normality tests robust distances are preferable, using a robust estimate estimates of location and scatter ("MCD", "MVE", or "OGK").

### Value

The function returns an S3 object of type `BRIL.Filtering` containing the following values:

call	Parameters of the call (contains data, test, testType, center, threshold, trimmedPerIteration and distType)
distances	Distances of each sample from data to the center provided
p.values	P.Values of the test at each iteration
index.p.values	Subset size corresponding to each P.Value, for plotting purpose
selected	Indices of the samples from data selected at the end of the filtering
cutoffDistance	Distance of the furthest inlier selected

### See Also

[plot.BRIL.Filtering\(\)](#), [print.BRIL.Filtering\(\)](#), [bril\(\)](#), [median\\_rec\(\)](#), [median\\_mv\(\)](#), [depth\\_values\(\)](#)

## Examples

```
## Example 1

# Illustrative data
XY <- rbind(
  mvtnorm::rmvnorm(300, c(0, 0), diag(2) * 3 - 1),
  mvtnorm::rmvnorm(100, c(15, 20), diag(2)),
  mvtnorm::rmvnorm(150, c(-10, 15), diag(2) * 2 - 0.5),
  mvtnorm::rmvnorm(200, c(5, 5), diag(2) * 200)
)

# Compute an estimate for the center
center <- median_rec(XY)$median

# Remove non unimodal outliers from this location
filtering <- filter_outliers(XY, center, test = "DIP", debug = TRUE)
print(filtering, maxDisplayed = 200)
plot(filtering)

## Example 2

# Illustrative data
XY <- rbind(
  mvtnorm::rmvnorm(300, c(0, 0), diag(2) * 4 - 1.5),
  mvtnorm::rmvnorm(150, c(5, 5), diag(2) * 400)
)

# Compute an estimate for the center
center <- median_rec(XY)$median

# Remove non normal outliers from this location
filtering <- filter_outliers(XY, center, test = "Chisq", distType = "MVE", debug = TRUE)
print(filtering)
plot(filtering, asp = 1)
```

---

median\_mv

*Multivariate Median*


---

## Description

Computes the Multivariate Median of the distribution provided in data (depth computations rely on the packages **depth**, **OjaNP** and **ddalpha**, while convex body minimizers "MCD" and "MVE" use the package **MASS**).

## Usage

```
median_mv(data, method = "Projection", sampleMedian = TRUE, warnings = FALSE)
```

## Arguments

data	Matrix of numerical values containing the observations (one per row, with two columns for X and Y coordinates)
------	--

method	Method to use. Valid options are "CW", "MCD", "MVE", "L2", "Lui", "Mahalanobis", "Oja", "Projection" (default), "Spatial" and "Tukey"
sampleMedian	Logical value. If TRUE (default), the function will return the Sample Median (observation from data with the highest depth). If FALSE, it will return the classic Multivariate Median (point in space with the highest depth), when applicable (methods "Oja", "Turkey" and "Spatial")
warnings	Logical value, to display the warnings and error caught by the underlying functions

### Value

Coordinates of the Multivariate Median

### See Also

[median\\_rec\(\)](#), [depth\\_values\(\)](#), [bril\(\)](#)

### Examples

```
# Illustrative data
XY <- rbind(
  mvtnorm::rmvnorm(200, c(0, 0), diag(2)),
  mvtnorm::rmvnorm(100, c(15, 20), diag(2) * 3 - 1),
  mvtnorm::rmvnorm(150, c(-10, 15), diag(2) * 2 - 0.5),
  mvtnorm::rmvnorm(100, c(5, 5), diag(2) * 200)
)

# Compute median
m <- median_mv(XY, method = "L2", warnings = TRUE)

# Plot results
plot(XY, asp = 1, xlab = "X", ylab = "Y")
points(m[1], m[2], col = "red", pch = 3, cex = 1.5, lwd = 3)

## Others examples of medians
## Not run:
median_mv(XY, method = "Oja")
median_mv(XY, method = "Tukey", sampleMedian = TRUE)
median_mv(XY, method = "Tukey", sampleMedian = FALSE)

## End(Not run)
```

---

median\_rec

*Recursive estimate of central location*

---

### Description

Recursive estimate of central location based on depth measures (from the packages [depth](#) and [ddalpha](#)) or convex body minimizers (package [MASS](#)).



**Usage**

```
median_rec(
  data,
  method = "Projection",
  alpha = 0.5,
  maxIterations = NULL,
  warnings = FALSE
)
```

**Arguments**

data	Matrix of numerical values containing the observations (one per row, with two columns for X and Y coordinates)
method	Method to use. Valid options are "MCD" and "MVE" for convex body minimizers, or "L2", "Lui", "Mahalanobis", "Oja", "Projection" (default), "Spatial" and "Tukey" for depth functions
alpha	Proportion of samples trimmed at each iteration (numerical value between 0 and 1, default: 0.5)
maxIterations	Set to a positive integer to limit the number of iterations, to NULL or 0 (default) for no limits
warnings	Logical value, to display the warnings and error raised by the underlying functions

**Value**

The function returns an S3 object of type `BRIL.MedianRec`, containing the following values:

median	Coordinate of the recursive median
max	Coordinate of the sample with the highest depth (or the center of the first iteration in the case of convex body minimizers)
iterations	List containing the indices from the samples of data selected at each iteration

**See Also**

[plot.BRIL.MedianRec\(\)](#), [print.BRIL.MedianRec\(\)](#), [median\\_mv\(\)](#), [depth\\_values\(\)](#), [bril\(\)](#)

**Examples**

```
# Illustrative data
XY <- rbind(
  mvtnorm::rmvnorm(300, c(0, 0), diag(2)),
  mvtnorm::rmvnorm(100, c(15, 20), diag(2) * 3 - 1),
  mvtnorm::rmvnorm(150, c(-10, 15), diag(2) * 2 - 0.5),
  mvtnorm::rmvnorm(200, c(5, 5), diag(2) * 200)
)

# Compute the recursive median
res <- median_rec(XY)

print(res)
plot(res)
```

plot.BRIL

*Plot method for BRIL objects***Description**

Plot method for BRIL objects

**Usage**

```
## S3 method for class 'BRIL'
plot(
  x,
  contents = "plot",
  showClusters = TRUE,
  showMode = TRUE,
  col,
  colMode,
  colClusters = NULL,
  iterationsIndices = NULL,
  iterationsOptions = NULL,
  ...
)
```

**Arguments**

x	An object of class BRIL (see <a href="#">bril()</a> )
contents	Contents to be displayed, options are "scatterplot", or "iterations" (only one option possible)
showClusters	Logical value used when contents = "scatterplot", to show or not the different clusters
showMode	Logical value used when contents = "scatterplot", to show or not the main mode
col	Default color of samples
colMode	Color of the mode when contents = "scatterplot"
colClusters	List (or array) of colors for each of the clusters/iterations (length must be at least equal to the number of groups identified by the function <a href="#">bril()</a> , i.e. <code>x\$nbClusters</code> )
iterationsIndices	Numerical value or array of numerical values, used when contents = "iterations", which provides the indices of the iterations to be plotted. If more than one iteration is requested, an interactive menu in the console will be used for the selection. 0 or NULL (default) will include all the iterations. Values that are negative or superior to the number of iterations performed by the execution <a href="#">bril()</a> will be ignored
iterationsOptions	List of additional parameters to be passed to the <a href="#">plot.BRIL.Filtering()</a> function when contents = "iterations" is selected (see <a href="#">plot.BRIL.Filtering()</a> for details). Example: <code>iterationsOptions = list(xlab = NA, ylab = NA, contents = c("p.values", "scatterplot"), asp = 1)</code>
...	Other arguments passed to or from other methods (such as <code>pch</code> for the symbols, <code>main</code> and <code>sub</code> for title and subtitle, <code>xlab</code> , <code>xmin</code> , ...)

**See Also**

`bril()`, `print.BRIL()`, `filter_outliers()`, `print.BRIL.Filtering()`

**Examples**

```
# Create a sample distribution and run bril() function
XY <- rbind(
  mvtnorm::rmvnorm(300, c(0, 0), diag(2) * 3 - 1),
  mvtnorm::rmvnorm(100, c(15, 20), diag(2)),
  mvtnorm::rmvnorm(150, c(-10, 15), diag(2) * 2 - 0.5),
  mvtnorm::rmvnorm(200, c(5, 5), diag(2) * 200)
)
res <- bril(XY, debug = TRUE)

# Plot the mode and groups encountered (default)
plot(res)

# Plot the mode only
plot(res, showClusters = FALSE)

# Plot the mode only (with extra graphic options)
plot(res, showClusters = FALSE, main = "Multivariate Mode Estimate",
      col = "blue", colMode = "black", asp = 1, pch = 3 )

# Plot the clusters without the mode
plot(res, showMode = FALSE, col = "gray",
      colClusters = c("yellow", "cyan", "purple", "red"))

# Plot the second iteration
plot(res, contents = "iteration", iterationsIndices = 2)

# Plot the second iteration (with arguments to plot.BRIL.filtering())
plot(res, contents = "iteration", iterationsIndices = 2,
      iterationsOptions = list(
        contents = c("scatterplot", "p.values"),
        colSelection = "blue", mfrow = c(2,1), asp = 1))

## Not run:
# Plot all iterations (interactive mode)
plot(res, contents = "iterations")

# Plot the 3 first iterations with options (interactive mode)
plot(res, contents = "iterations", iterationsIndices = c(1:3),
      iterationsOptions = list(
        contents = c("scatterplot"),
        xlim = c(-50,50), ylim = c(-30,30), asp = 1))

## End(Not run)
```

## Description

Plot method for BRIL.Filtering objects

## Usage

```
## S3 method for class 'BRIL.Filtering'
plot(
  x,
  contents = c("p.values", "scatterplot", "dist", "hist"),
  showCenter = TRUE,
  showSelection = TRUE,
  col = "black",
  colSelection = "red",
  colCenter = "orange",
  mtextTitles = TRUE,
  mfrow,
  ...
)
```

## Arguments

x	An object of class BRIL.Filtering (see <a href="#">filter_outliers()</a> )
contents	Contents to be displayed, options are "p.values", "scatterplot", "dist", "hist" and "all"
showCenter	Logical value, to show the center used in the filtering
showSelection	Logical value, to highlight the samples selected by the filtering process
col	Default color for non-selected samples (default: "black")
colSelection	Color of the selected samples (default: "red")
colCenter	Color for the center in "scatterplot" (default: "orange")
mtextTitles	Logical value, TRUE to set smaller titles/subtitles on top, FALSE to use the default plot title options.
mfrow	Number of rows and columns of the figure (example: c(4,1))
...	Other arguments passed to or from other methods (such as pch for the symbols, main and sub for title and subtitle, xlab, xmin, ...)

## Details

*Red intercept lines correspond to the selection based on the p.values exceeding the given threshold. To display all the p.values, rerun the function [filter\\_outliers\(\)](#) with the parameter debug = TRUE*

contents options:

- **"p.values"** provides a plot of the test p.values (in function of the subset size)
- **"scatterplot"** displays the data in cartesian coordinates. Selected samples are displayed in red, and the center used to compute distances as an orange cross
- **"dist"** shows the distances of each sample to the center provided in [filter\\_outliers\(\)](#) (in function of sample index)
- **"hist"** draws an histogram of the distances of the samples to the center provided in [filter\\_outliers\(\)](#)
- **"all"** displays a figure with all of the options above

**See Also**

[filter\\_outliers\(\)](#), [print.BRIL.Filtering\(\)](#), [bril\(\)](#), [median\\_rec\(\)](#), [median\\_mv\(\)](#), [depth\\_values\(\)](#)

**Examples**

```
# Illustrative data
XY <- rbind(
  mvtnorm::rmvnorm(300, c(0, 0), diag(2) * 3 - 1),
  mvtnorm::rmvnorm(100, c(15, 20), diag(2)),
  mvtnorm::rmvnorm(150, c(-10, 15), diag(2) * 2 - 0.5),
  mvtnorm::rmvnorm(200, c(5, 5), diag(2) * 200)
)

# Process the data
filtering <- filter_outliers(XY, median_rec(XY)$median, test = "DIP", debug = TRUE)

# Plot all default figures
plot(filtering)

# Plot P.Values and Scatterplot only
plot(filtering, contents = c("pvalues", "scatterplot"))

# Change the layout to vertical
plot(filtering, contents = c("pvalues", "scatterplot"), mfrow = c(2, 1))

# Remove title, subtitle, and axis labels
plot(filtering, contents = "scatterplot", main = "", sub = "",
      ylab = "", xlab = "")

# Other graphical options
plot(filtering, contents = "scatterplot", asp = 1,
      xlim = c(-30, 30), ylim = c(-30, 30))

plot(filtering,
      contents = "scatterplot", asp = 1, pch = 4, lwd = 2, col = "blue",
      colSelection = "green", showCenter = FALSE
)

plot(filtering, contents = "hist", main = "My Histogram",
      showSelection = FALSE, breaks = 50)
```

---

plot.BRIL.MedianRec     *Plot method for BRIL.MedianRec objects*

---

**Description**

Plot method for BRIL.MedianRec objects

**Usage**

```
## S3 method for class 'BRIL.MedianRec'
plot(x, nbIterations = 5, showMedian = FALSE, showMax = FALSE, ...)
```

**Arguments**

x	An object of class BRIL.MedianRec (see <a href="#">median_rec()</a> )
nbIterations	Number of iterations to display, or 0 to show all of them (default: 5)
showMedian	Logical value, to show the final recursive median (indicated by a "+")
showMax	Logical value, to show the overall deepest point, or the center of the first MCD/MVE iteration (indicated by a "x")
...	Other arguments passed to or from other methods

**See Also**

[median\\_rec\(\)](#), [median\\_mv\(\)](#), [bril\(\)](#)

**Examples**

```
# Illustrative data
XY <- rbind(
  mvtnorm::rmvnorm(300, c(0, 0), diag(2)),
  mvtnorm::rmvnorm(100, c(15, 20), diag(2) * 3 - 1),
  mvtnorm::rmvnorm(150, c(-10, 15), diag(2) * 2 - 0.5),
  mvtnorm::rmvnorm(200, c(5, 5), diag(2) * 200)
)

# Process the data
res <- median_rec(XY)

# Default plot
plot(res)

# Adjust axis
plot(res, asp = 1, xlim = c(-20, 20), ylim = c(-20, 20))

# Change other graphical options
plot(res, showMedian = TRUE, pch = 16, main = "Recursive Median")
```

---

print.BRIL

---

*Print method for BRIL objects*


---

**Description**

Print method for BRIL objects

**Usage**

```
## S3 method for class 'BRIL'
print(x, maxDisplayed = NULL, ...)
```

**Arguments**

x	An object of class BRIL (see <a href="#">bril()</a> )
maxDisplayed	Number of elements to display in the output. Set to NULL (or 0) to show all values.
...	Other arguments passed to or from other methods

**See Also**

[bril\(\)](#), [plot.BRIL\(\)](#)

---

print.BRIL.Filtering    *Print method for BRIL.Filtering objects*

---

**Description**

Print method for BRIL.Filtering objects

**Usage**

```
## S3 method for class 'BRIL.Filtering'
print(x, maxDisplayed = 200, ...)
```

**Arguments**

x	An object of class BRIL.Filtering (see <a href="#">filter_outliers()</a> )
maxDisplayed	Number of elements to display in the output (default: 200). Set to NULL (or 0) to show all values.
...	Other arguments passed to or from other methods

**See Also**

[filter\\_outliers\(\)](#), [plot.BRIL.Filtering\(\)](#), [bril\(\)](#)

---

print.BRIL.MedianRec    *Print method for BRIL.MedianRec objects*

---

**Description**

Print method for BRIL.MedianRec objects

**Usage**

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

**Arguments**

x	An object of class BRIL.MedianRec (see <a href="#">median_rec()</a> )
...	Other arguments passed to or from other methods

**See Also**

[median\\_rec\(\)](#), [plot.BRIL.MedianRec\(\)](#)

---

test_multinormality	<i>Test the multivariate normality of a distribution</i>
---------------------	--

---

## Description

Test the multivariate normality of a distribution

## Usage

```
test_multinormality(  
  values,  
  test = "Mardia",  
  threshold = 0.05,  
  data,  
  center,  
  warnings = FALSE  
)
```

## Arguments

values	Unidimensional array of numerical values (distances)
test	Statistical test used, valid options are "Mardia", "Kurtosis", "Skewness", "KS", "KS-adj", "Shapiro", "Lillie", and "Chisq"
threshold	Threshold of significance for the statistical test (default: 0.05)
data	Matrix of numerical values containing the observations (one per row, with two columns for X and Y coordinates)
center	Center of the observations from data
warnings	Logical value, to display the warnings and errors caught

## Details

Parameter data is only required for the tests "Mardia", "Skewness", "Kurtosis" and "Chisq", while parameter center is only required for "Chisq"

## Value

p-value of the test (lower values suggest non normality)

## See Also

[filter\\_outliers\(\)](#)



---

test_unimodality	<i>Test the unimodality of a distribution</i>
------------------	---

---

**Description**

Test the unimodality of a distribution

**Usage**

```
test_unimodality(values, test = "DIP", warnings = FALSE)
```

**Arguments**

values	Unidimensional array of numerical values (distances)
test	Statistical test used (for now the only option is "DIP")
warnings	Logical value, to display the warnings and errors caught

**Value**

p-value of the test (lower values suggest multimodality)

**See Also**

[filter\\_outliers\(\)](#)

# Index

`bril`, 2  
`bril()`, 5, 6, 8–11, 13–15

`depth_values`, 4  
`depth_values()`, 4, 6, 8, 9, 13

`filter_outliers`, 5  
`filter_outliers()`, 3–5, 11–13, 15–17

`median_mv`, 7  
`median_mv()`, 4–6, 9, 13, 14  
`median_rec`, 8  
`median_rec()`, 3–6, 8, 13–15

`plot.BRIL`, 10  
`plot.BRIL()`, 4, 15  
`plot.BRIL.Filtering`, 11  
`plot.BRIL.Filtering()`, 3, 6, 10, 15  
`plot.BRIL.MedianRec`, 13  
`plot.BRIL.MedianRec()`, 9, 15  
`print.BRIL`, 14  
`print.BRIL()`, 4, 11  
`print.BRIL.Filtering`, 15  
`print.BRIL.Filtering()`, 6, 11, 13  
`print.BRIL.MedianRec`, 15  
`print.BRIL.MedianRec()`, 9

`test_multinormality`, 16  
`test_unimodality`, 17