

Package ‘BRIL’

June 11, 2021

Title Bootstrap and Refine Iterative Locator

Date 2021-05-19

Version 1.0.0

Author Adrien Brilhault [aut, cre] (<<https://orcid.org/0000-0002-8135-2715>>),

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

Description The BRIL algorithm estimate the main mode of bivariate distributions through three principal steps:
BOOTSTRAP: a recursive depth trimming (or recursive convex body minimizers) to locate a first estimate.
REFINE: a two-pass outliers filtering, the first relying on euclidian distance to the first estimate and unimodality tests, the second on robust distances and multinormality tests.
ITERATE: removing the samples selected in the REFINE step from the global distribution, the same process is reapplied, the final mode corresponding to the iteration with the largest number of samples.

Depends R (>= 3.5.0)

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

Suggests mvtnorm

License GPL-3 | file LICENSE

URL <https://github.com/adrienbrilhault/BRIL>

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

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

NeedsCompilation no

R topics documented:

bril	2
depth_values	4
filter_outliers	5
median_mv	7
median_rec	8

plot.BRIL	9
plot.BRIL.Filtering	11
plot.BRIL.MedianRec	12
print.BRIL	13
print.BRIL.Filtering	14
print.BRIL.MedianRec	14
test_multinormality	15
test_unimodality	15

Index	17
--------------	-----------

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 unasigned 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 median_rec()

testUnimodal	Statistical test used for unimodality. Valid options are only "DIP", see filter_outliers()
threshUnimodal	Threshold of significance for the unimodality test (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 filter_outliers()
threshNormal	Threshold of significance for the normality test (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 filter_outliers()
exitWhenUnimodal	Logical value. TRUE will terminate the execution of the algorithm as soon as an unimodal subset is encountered on the start of an iteration. FALSE (default) will let that last iteration proceed before exiting
debug	Logical value. TRUE will compute all p.values in the filtering steps (even after the p.values exceed the threshold, see plot.BRIL.Filtering())
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 to the function (i.e. data, maxIterations, minUnassigned, method, alpha, testUnimodal, threshUnimodal, distUnimodal, testNormal, threshNormal, distNormal, trimmedPerFilteringIteration, and exitWhenUnimodal)
iterations	A list of each iteration, containing the two filtering procedures performed: filteringUnimodal and filteringNormal (both being S3 object of class BRIL.Filtering, see filter_outliers())
nbClusters	Number of groups encountered (one per iteration)
labels	Labels of the groups encountered (coresponding to the number of the iteration they were identified from)
clustersCenters	Matrix containing the coordinates of the centers of each group (row-wise) the filtering
clustersSizes	Array with the number of sample in each group
mainCluster	Index of the group identified as main mode

See Also

[plot.BRIL\(\)](#), [print.BRIL\(\)](#), [median_rec\(\)](#), [median_mv\(\)](#), [filter_outliers\(\)](#), [depth_values\(\)](#)

Examples

```
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, "iterations")

## End(Not run)
```

depth_values

Depth function 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` isn't 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 NULL, depth values will be computed for all observations from data.
<code>method</code>	Depth function to use. 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 called.

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

`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)
)

# compute depths
D <- depth_values(XY, method = "L2")

# plot results
graphics::par(mfrow = c(1, 2))
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(1:nrow(XY), D, pch = 20, col = plotColors[as.numeric(cut(D, breaks = 20))], xlab = "Index", ylab = "Depth")
graphics::par(mfrow = c(1, 1))

# 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 robust distances and multinormality tests

Description

Recursive outlier filtering based on robust distances 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 order samples based on their euclidian distance (array of numerical values)
test	Statistical test used. 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 (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 (default: 1)
debug	Logical value. TRUE will compute all p.values (even after exceeding the threshold, for plotting purpose, see plot.BRIL.Filtering())
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 contains a large amount of outliers). For normality tests robust distances are preferable, based on 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 to the function (i.e. <code>data</code> , <code>test</code> , <code>testType</code> , <code>center</code> , <code>threshold</code> , <code>trimmedPerIteration</code> and <code>distType</code>)
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 at the end of the filtering

See Also

[plot.BRIL.Filtering\(\)](#), [print.BRIL.Filtering\(\)](#), [median_rec\(\)](#), [median_mv\(\)](#), [bril\(\)](#)

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 "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

```
## Example 1

# 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 median
m <- median_mv(XY, method = "Liu")

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

## Example 2

median_mv(XY, method = "L2")
median_mv(XY, method = "Tukey", sampleMedian = TRUE)
median_mv(XY, method = "Tukey", sampleMedian = FALSE)
```

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 iterations

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

<code>x</code>	An object of class BRIL (see bril())
<code>contents</code>	Contents to be displayed, options are "scatterplot", or "iterations" (only one option possible)
<code>showClusters</code>	Logical value. Used with contents="scatterplot", to show or not the different clusters
<code>showMode</code>	Logical value. Used for contents="clusters", to show or not the main mode to show or not the different clusters
<code>col</code>	default color of samples
<code>colMode</code>	Color of the mode when contents="scatterplot" (if showMode is TRUE)
<code>colClusters</code>	List (or array) of colors (length must be at least equal to the number of clusters identified by the function bril() , i.e. 'x\$nbClusters')
<code>iterationsIndices</code>	Numerical value or array of numerical values, used with contents="iterations", providing 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 bril() will be ignored
<code>iterationsOptions</code>	List of additional parameters to be passed to the plot.BRIL.Filtering function when contents="iterations" is selected (see plot.BRIL.Filtering for details). <i>Example: iterationsOptions = list(Xlab=NA, Ylab=NA, contents=c("p.values", "scatterplot"))</i>
<code>...</code>	Other arguments passed to or from other methods (such as pch for the symbols, main and sub for title and subtitle, xlab, xmin, ...)

See Also

[bril\(\)](#), [print.BRIL](#), [filter_outliers\(\)](#), [print.BRIL.Filtering\(\)](#)

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

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

<code>x</code>	An object of class <code>BRIL.Filtering</code> (see filter_outliers())
<code>contents</code>	Contents to be displayed, options are "p.values", "scatterplot", "dist", "hist" and "all"
<code>showCenter</code>	Logical value, to show the center used in the filtering
<code>showSelection</code>	Logical value, to highlight the samples selected by the filtering process
<code>col</code>	Default color for non-selected samples (default: "black")
<code>colSelection</code>	Color of the selected samples (default: "red")
<code>colCenter</code>	Color for center in "scatterplot" (default: "orange")
<code>mtextTitles</code>	Logical value, TRUE to set smaller titles/subtitles on top, FALSE to use the default plot options.
<code>mfrow</code>	Number of rows and columns of the figure (example: <code>c(4,1)</code>)
<code>...</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> , ...)

Details

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

Notes:

*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`*

See Also

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

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", showCe
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 median_rec())
nbIterations	Number of iterations to display, or 0 to show all of them (default: 5)
showMedian	Logical, to show the final recursive median (indicated by a "+")
showMax	Logical, 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	<i>Print method for BRIL objects</i>
------------	--------------------------------------

Description

Print method for BRIL objects

Usage

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

Arguments

x	An object of class BRIL (see bril())
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

[plot.BRIL\(\)](#), [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 filter_outliers())
maxDisplayed	Number of elements to display in the output (default: 500). 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 median_rec())
...	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 (for now the only option "DIP")
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)

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)

Index

`bril`, 2
`bril()`, 5, 6, 8–10, 12–14

`depth_values`, 4
`depth_values()`, 3, 8, 9

`filter_outliers`, 5
`filter_outliers()`, 3, 10–12, 14

`median_mv`, 7
`median_mv()`, 3, 5, 6, 9, 12, 13
`median_rec`, 8
`median_rec()`, 2, 3, 5, 6, 8, 12–14

`plot.BRIL`, 9
`plot.BRIL()`, 3, 14
`plot.BRIL.Filtering`, 10, 11
`plot.BRIL.Filtering()`, 3, 6, 14
`plot.BRIL.MedianRec`, 12
`plot.BRIL.MedianRec()`, 9, 14
`print.BRIL`, 10, 13
`print.BRIL()`, 3
`print.BRIL.Filtering`, 14
`print.BRIL.Filtering()`, 6, 10, 12
`print.BRIL.MedianRec`, 14
`print.BRIL.MedianRec()`, 9

`test_multinormality`, 15
`test_unimodality`, 15