

# Package ‘RepeatedHighDim’

June 29, 2023

**Type** Package

**Title** Methods for High-Dimensional Repeated Measures Data

**Version** 2.2.0

**Author** Klaus Jung [aut, cre],  
Jochen Kruppa [aut],  
Sergej Ruff [aut]

**Maintainer** Klaus Jung <klaus.jung@tiho-hannover.de>

## Description

A toolkit for the analysis of high-dimensional repeated measurements, providing functions for outlier detection, differential expression analysis, gene-set tests, and binary random data generation.

**License** GPL (>= 3)

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Imports** ddalp, geometry, graphics, grDevices, MASS, mvtnorm, nlme,  
rgl, stats, utils

**Suggests** BiocManager, limma

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2023-06-29 11:20:02 UTC

## R topics documented:

bag . . . . .	2
depmed . . . . .	4
fc_ci . . . . .	5
fc_plot . . . . .	7
GA_diagplot . . . . .	8
gem . . . . .	10
GlobTestMissing . . . . .	12
gridfun . . . . .	13

hldepth . . . . .	14
iter_matrix . . . . .	15
loop . . . . .	16
RepeatedHighDim . . . . .	19
RHighDim . . . . .	21
rho_bounds . . . . .	22
rmvbinary_EP . . . . .	23
rmvbinary_QA . . . . .	24
sequence_probs . . . . .	25
start_matrix . . . . .	26
summary_RHD . . . . .	27
TestStatSimple . . . . .	28
TestStatSP . . . . .	28

## Index 29

---

bag	<i>Calculates the bag</i>
-----	---------------------------

---

### Description

Calculates the bag of a gemplot (i.e. the inner gemstone).

### Usage

```
bag(D, G)
```

### Arguments

D	Data set with rows representing the individuals and columns representing the features. In the case of three dimensions, the colnames of D must be c("x", "y", "z").
G	List containing the grid information produced by <a href="#">gridfun</a> and the halfspace location depths calculated by <a href="#">hldepth</a> .

### Details

Determines those grid points that belong to the bag, i.e. a convex hull that contains 50 percent of the data. In the case of a 3-dimensional data set, the bag can be visualized by an inner gemstone that can be accompanied by an outer gemstone ([loop](#)).

### Value

A list containing the following elements:

**coords** Coordinates of the grid points that belong to the bag. Each row represents a grid point and each column represents one dimension.

**hull** A data matrix that contains the indices of the margin grid points of the bag that cover the convex hull by triangles. Each row represents one triangle. The indices correspond to the rows of coords.

**Author(s)**

Jochen Kruppa, Klaus Jung

**References**

Rousseeuw, P. J., Ruts, I., & Tukey, J. W. (1999). The bagplot: a bivariate boxplot. *The American Statistician*, **53**(4), 382-387. doi: [10.1080/00031305.1999.10474494](https://doi.org/10.1080/00031305.1999.10474494)

Kruppa, J., & Jung, K. (2017). Automated multigroup outlier identification in molecular high-throughput data using bagplots and gemplots. *BMC bioinformatics*, **18**(1), 1-10. <https://link.springer.com/article/10.1186/s12859-017-1645-5>

**Examples**

```
## Attention: calculation is currently time-consuming.
## Remove #-Symbols to run examples

## Two 3-dimensional example data sets D1 and D2
# n <- 200
# x1 <- rnorm(n, 0, 1)
# y1 <- rnorm(n, 0, 1)
# z1 <- rnorm(n, 0, 1)
# D1 <- data.frame(cbind(x1, y1, z1))
# x2 <- rnorm(n, 1, 1)
# y2 <- rnorm(n, 1, 1)
# z2 <- rnorm(n, 1, 1)
# D2 <- data.frame(cbind(x2, y2, z2))
# colnames(D1) <- c("x", "y", "z")
# colnames(D2) <- c("x", "y", "z")

## Placing outliers in D1 and D2
# D1[17,] = c(4, 5, 6)
# D2[99,] = -c(3, 4, 5)

## Grid size and graphic parameters
# grid.size <- 20
# red <- rgb(200, 100, 100, alpha = 100, maxColorValue = 255)
# blue <- rgb(100, 100, 200, alpha = 100, maxColorValue = 255)
# yel <- rgb(255, 255, 102, alpha = 100, maxColorValue = 255)
# white <- rgb(255, 255, 255, alpha = 100, maxColorValue = 255)
# require(rgl)
# material3d(color=c(red, blue, yel, white),
# alpha=c(0.5, 0.5, 0.5, 0.5), smooth=FALSE, specular="black")

## Calculation and visualization of gemplot for D1
# G <- gridfun(D1, grid.size=20)
# G$H <- hldepth(D1, G, verbose=TRUE)
# dm <- depmed(G)
# B <- bag(D1, G)
# L <- loop(D1, B, dm=dm)
# bg3d(color = "gray39" )
# points3d(D1[L$outliers==0,1], D1[L$outliers==0,2], D1[L$outliers==0,3], col="green")
```

```

# text3d(D1[L$outliers==1,1], D1[L$outliers==1,2],D1[L$outliers==1,3],
# as.character(which(L$outliers==1)), col=yel)
# spheres3d(dm[1], dm[2], dm[3], col=yel, radius=0.1)
# material3d(1,alpha=0.4)
# gem(B$coords, B$hull, red)
# gem(L$coords.loop, L$hull.loop, red)
# axes3d(col="white")

## Calculation and visualization of gemplot for D2
# G <- gridfun(D2, grid.size=20)
# G$H <- hldepth(D2, G, verbose=TRUE)
# dm <- depmed(G)
# B <- bag(D2, G)
# L <- loop(D2, B, dm=dm)
# points3d(D2[L$outliers==0,1], D2[L$outliers==0,2], D2[L$outliers==0,3], col="green")
# text3d(D2[L$outliers==1,1], D2[L$outliers==1,2],D2[L$outliers==1,3],
# as.character(which(L$outliers==1)), col=yel)
# spheres3d(dm[1], dm[2], dm[3], col=yel, radius=0.1)
# gem(B$coords, B$hull, blue)
# gem(L$coords.loop, L$hull.loop, blue)

```

---

depmed

*Calculates the depth median.*


---

## Description

Calculates the depth median.

## Usage

```
depmed(G)
```

## Arguments

G List containing the grid information produced by [gridfun](#) and the halfspace location depths produced by [hldepth](#).

## Details

Calculates the depth median in a specified grid array with given halfspace location depth at each grid location.

## Value

An vector with a length equal to the number of dimension of the array in G, containing the coordinates of the depth median.

## Author(s)

Jochen Kruppa, Klaus Jung

## References

Rousseeuw, P. J., Ruts, I., & Tukey, J. W. (1999). The bagplot: a bivariate boxplot. *The American Statistician*, 53(4), 382-387.

## Examples

```
## Attention: calculation is currently time-consuming.
## Remove #-Symbols to run examples

## A 3-dimensional example data set D1
# n <- 200
# x1 <- rnorm(n, 0, 1)
# y1 <- rnorm(n, 0, 1)
# z1 <- rnorm(n, 0, 1)
# D1 <- data.frame(cbind(x1, y1, z1))
# colnames(D1) <- c("x", "y", "z")

## Specification of the grid and calculation of the halfspace location depth at each grid location.
# G <- gridfun(D1, grid.size=20)
# G$H <- hldepth(D1, G, verbose=TRUE)
# dm <- depmed(G) ## Calculation of the depth median
```

---

 fc\_ci

*Calculation of adjusted confidence intervals*


---

## Description

Calculation of adjusted confidence intervals

## Usage

```
fc_ci(fit, alpha = 0.05, method = "raw")
```

## Arguments

fit	Object as returned from the function eBayes of the limma package
alpha	1 - confidence level (e.g., if confidence level is 0.95, alpha is 0.05)
method	Either 'raw' for unadjusted confidence intervals, or 'BH' for Bejamini Hochberg-adjusted confidence intervals, or 'BY' for Benjamini Yekutieli-adjusted confidence intervals

## Details

Calculation of unadjusted and adjusted confidence intervals for the log fold change

## Value

A results matrix with one row per gene, and one column for the p-value, the log fold change, the lower limit of the CI, and the upper limit of the CI

**Author(s)**

Klaus Jung

**References**

Dudoit, S., Shaffer, J. P., & Boldrick, J. C. (2003). Multiple hypothesis testing in microarray experiments. *Statistical Science*, **18**(1), 71-103. <https://projecteuclid.org/journals/statistical-science/volume-18/issue-1/Multiple-Hypothesis-Testing-in-Microarray-Experiments/10.1214/ss/1056397487.full>

Jung, K., Friede, T., & Beißbarth, T. (2011). Reporting FDR analogous confidence intervals for the log fold change of differentially expressed genes. *BMC bioinformatics*, **12**, 1-9. <https://link.springer.com/article/10.1186/1471-2105-12-288>

**Examples**

```
### Artificial microarray data
d = 1000 ### Number of genes
n = 10 ### Sample per group
fc = rlnorm(d, 0, 0.1)
mu1 = rlnorm(d, 0, 1) ### Mean vector group 1
mu2 = mu1 * fc ### Mean vector group 2
sd1 = rnorm(d, 1, 0.2)
sd2 = rnorm(d, 1, 0.2)
X1 = matrix(NA, d, n) ### Expression levels group 1
X2 = matrix(NA, d, n) ### Expression levels group 2
for (i in 1:n) {
  X1[,i] = rnorm(d, mu1, sd=sd1)
  X2[,i] = rnorm(d, mu2, sd=sd2)
}
X = cbind(X1, X2)
heatmap(X)

### Differential expression analysis with limma
if(check_limma()){
  group = gl(2, n)
  design = model.matrix(~ group)
  fit1 = limma::lmFit(X, design)
  fit = limma::eBayes(fit1)

### Calculation of confidence intervals
CI = fc_ci(fit=fit, alpha=0.05, method="raw")
head(CI)
CI = fc_ci(fit=fit, alpha=0.05, method="BH")
head(CI)
CI = fc_ci(fit=fit, alpha=0.05, method="BY")
head(CI)

fc_plot(CI, xlim=c(-0.5, 3), ylim=-log10(c(1, 0.0001)), updown="up")
fc_plot(CI, xlim=c(-3, 0.5), ylim=-log10(c(1, 0.0001)), updown="down")
fc_plot(CI, xlim=c(-3, 3), ylim=-log10(c(1, 0.0001)), updown="all")
}
```

---

`fc_plot`*Volcano plot of adjusted confidence intervals*

---

**Description**

Volcano plot of adjusted confidence intervals

**Usage**

```
fc_plot(  
  CI,  
  alpha = 0.05,  
  updown = "all",  
  xlim = c(-3, 3),  
  ylim = -log10(c(1, 0.001))  
)
```

**Arguments**

CI	Object as returned from the function <code>fc_ci</code>
alpha	1 - confidence level (e.g., if confidence level is 0.95, alpha is 0.05)
updown	Character, 'all' if CIs for all genes, 'down' if CIs for down-regulated genes, or 'up' if CIs for up-regulated genes to be plotted
xlim	Vector of length 2 with the lower and upper limits for the X-axis
ylim	Vector of length 2 with the lower and upper limits for the Y-axis. Please note, that p-values are usually displayed on the -log <sub>10</sub> -scale in a volcano plot

**Details**

Volcano plot of adjusted confidence intervals

**Author(s)**

Klaus Jung

**References**

Dudoit, S., Shaffer, J. P., & Boldrick, J. C. (2003). Multiple hypothesis testing in microarray experiments. *Statistical Science*, **18**(1), 71-103. <https://projecteuclid.org/journals/statistical-science/volume-18/issue-1/Multiple-Hypothesis-Testing-in-Microarray-Experiments/10.1214/ss/1056397487.full>

Jung, K., Friede, T., & Beißbarth, T. (2011). Reporting FDR analogous confidence intervals for the log fold change of differentially expressed genes. *BMC bioinformatics*, **12**, 1-9. <https://link.springer.com/article/10.1186/1471-2105-12-288>

**Examples**

```

### Artificial microarray data
d = 1000 ### Number of genes
n = 10 ### Sample per group
fc = rlnorm(d, 0, 0.1)
mu1 = rlnorm(d, 0, 1) ### Mean vector group 1
mu2 = mu1 * fc ### Mean vector group 2
sd1 = rnorm(d, 1, 0.2)
sd2 = rnorm(d, 1, 0.2)
X1 = matrix(NA, d, n) ### Expression levels group 1
X2 = matrix(NA, d, n) ### Expression levels group 2
for (i in 1:n) {
  X1[,i] = rnorm(d, mu1, sd=sd1)
  X2[,i] = rnorm(d, mu2, sd=sd2)
}
X = cbind(X1, X2)
heatmap(X)

### Differential expression analysis with limma
if(check_limma()){
  group = gl(2, n)
  design = model.matrix(~ group)
  fit1 = limma::lmFit(X, design)
  fit = limma::eBayes(fit1)

### Calculation of confidence intervals
CI = fc_ci(fit=fit, alpha=0.05, method="raw")
head(CI)
CI = fc_ci(fit=fit, alpha=0.05, method="BH")
head(CI)
CI = fc_ci(fit=fit, alpha=0.05, method="BY")
head(CI)

fc_plot(CI, xlim=c(-0.5, 3), ylim=-log10(c(1, 0.0001)), updown="up")
fc_plot(CI, xlim=c(-3, 0.5), ylim=-log10(c(1, 0.0001)), updown="down")
fc_plot(CI, xlim=c(-3, 3), ylim=-log10(c(1, 0.0001)), updown="all")
}

```

---

GA\_diagplot

*Diagnostic plot for comparison of two correlation matrices.*


---

**Description**

A diagnostic plot that compares the entries of two correlation matrices using a color scale.

**Usage**

```

GA_diagplot(
  R,

```



```

    Rt,
    eps = 0.05,
    col.method = "trafficlight",
    color = c(0, 8),
    top = ""
  )

```

### Arguments

R	Specified correlation matrix.
Rt	Correlation matrix of the data generated by the genetic algorithm.
eps	Permitted difference between the entries of two matrices. Must only be specified if col.method="trafficlight".
col.method	Method to use for color scaling the difference between the matrices. If method="trafficlight" only two colors are used, indicating whether the entries deviated at least by a difference of eps. If method="updown" a discrete gray scale is used.
color	Value of two color that are used if method="trafficlight"
top	Specifies the main title of the plot

### Details

A diagnostic plot that compares the entries of two correlation matrices using a color scale.

### Author(s)

Jochen Kruppa, Klaus Jung

### References

Kruppa, J., Lepenies, B., & Jung, K. (2018). A genetic algorithm for simulating correlated binary data from biomedical research. *Computers in biology and medicine*, **92**, 1-8. doi: [10.1016/j.combiomed.2017.10.023](https://doi.org/10.1016/j.combiomed.2017.10.023)

### Examples

```

R1 = diag(10)
X0 <- start_matrix(p=c(0.4, 0.2, 0.5, 0.15, 0.4, 0.35, 0.2, 0.25, 0.3, 0.4), k = 5000)
Xt <- iter_matrix(X0, R = diag(10), T = 10000, e.min = 0.00001)
GA_diagplot(R1, Rt = Xt$Rt, col.method = "trafficlight")
GA_diagplot(R1, Rt = Xt$Rt, col.method = "updown")

```

---

`gem`*Plots a gemstone to an interactive graphics device*

---

**Description**

Plots a gemstone to an interactive graphics device.

**Usage**

```
gem(coords, hull, clr)
```

**Arguments**

<code>coords</code>	Matrix with coordinates of the grid or of data points that belong to the gemstone, calculated by either <code>bag</code> or <code>loop</code> . Each row represents a grid point and each column represents one dimension.
<code>hull</code>	Matrix with indices of triangles that cover a convex hull around the gemstone. Each row represents one triangle and the indices refer to the rows of <code>coords</code> .
<code>clr</code>	Specifies the color of the gemstone.

**Details**

Only applicable to 3-dimensional data sets. Transparent colors are recommended for outer gemstone of the gemplot. Further graphical parameters can be set using `material3d()` of the `rgl`-package.

**Author(s)**

Jochen Kruppa, Klaus Jung

**References**

Rousseeuw, P. J., Ruts, I., & Tukey, J. W. (1999). The bagplot: a bivariate boxplot. *The American Statistician*, **53**(4), 382-387. doi: [10.1080/00031305.1999.10474494](https://doi.org/10.1080/00031305.1999.10474494)

Kruppa, J., & Jung, K. (2017). Automated multigroup outlier identification in molecular high-throughput data using bagplots and gemplots. *BMC bioinformatics*, **18**(1), 1-10. <https://link.springer.com/article/10.1186/s12859-017-1645-5>

**Examples**

```
## Attention: calculation is currently time-consuming.
## Remove #-Symbols to run examples

## Two 3-dimensional example data sets D1 and D2
# n <- 200
# x1 <- rnorm(n, 0, 1)
# y1 <- rnorm(n, 0, 1)
# z1 <- rnorm(n, 0, 1)
# D1 <- data.frame(cbind(x1, y1, z1))
```

```

# x2 <- rnorm(n, 1, 1)
# y2 <- rnorm(n, 1, 1)
# z2 <- rnorm(n, 1, 1)
# D2 <- data.frame(cbind(x2, y2, z2))
# colnames(D1) <- c("x", "y", "z")
# colnames(D2) <- c("x", "y", "z")

## Placing outliers in D1 and D2
# D1[17,] = c(4, 5, 6)
# D2[99,] = -c(3, 4, 5)

## Grid size and graphic parameters
# grid.size <- 20
# red <- rgb(200, 100, 100, alpha = 100, maxColorValue = 255)
# blue <- rgb(100, 100, 200, alpha = 100, maxColorValue = 255)
# yel <- rgb(255, 255, 102, alpha = 100, maxColorValue = 255)
# white <- rgb(255, 255, 255, alpha = 100, maxColorValue = 255)
# require(rgl)
# material3d(color=c(red, blue, yel, white),
# alpha=c(0.5, 0.5, 0.5, 0.5), smooth=FALSE, specular="black")

## Calculation and visualization of gemplot for D1
# G <- gridfun(D1, grid.size=20)
# G$H <- hldepth(D1, G, verbose=TRUE)
# dm <- depmed(G)
# B <- bag(D1, G)
# L <- loop(D1, B, dm=dm)
# bg3d(color = "gray39" )
# points3d(D1[L$outliers==0,1], D1[L$outliers==0,2], D1[L$outliers==0,3], col="green")
# text3d(D1[L$outliers==1,1], D1[L$outliers==1,2], D1[L$outliers==1,3],
# as.character(which(L$outliers==1)), col=yel)
# spheres3d(dm[1], dm[2], dm[3], col=yel, radius=0.1)
# material3d(1,alpha=0.4)
# gem(B$coords, B$hull, red)
# gem(L$coords.loop, L$hull.loop, red)
# axes3d(col="white")

## Calculation and visualization of gemplot for D2
# G <- gridfun(D2, grid.size=20)
# G$H <- hldepth(D2, G, verbose=TRUE)
# dm <- depmed(G)
# B <- bag(D2, G)
# L <- loop(D2, B, dm=dm)
# points3d(D2[L$outliers==0,1], D2[L$outliers==0,2], D2[L$outliers==0,3], col="green")
# text3d(D2[L$outliers==1,1], D2[L$outliers==1,2], D2[L$outliers==1,3],
# as.character(which(L$outliers==1)), col=yel)
# spheres3d(dm[1], dm[2], dm[3], col=yel, radius=0.1)
# gem(B$coords, B$hull, blue)
# gem(L$coords.loop, L$hull.loop, blue)

## Example of outlier detection with four principal components.
## Attention: calculation is currently time-consuming.

```

```
# set.seed(123)
# n <- 200
# x1 <- rnorm(n, 0, 1)
# x2 <- rnorm(n, 0, 1)
# x3 <- rnorm(n, 0, 1)
# x4 <- rnorm(n, 0, 1)
# D <- data.frame(cbind(x1, x2, x3, x4))
# D[67,] = c(7, 0, 0, 0)

# date()
# G = gridfun(D, 20, 4)
# G$H = h1depth(D, G, verbose=TRUE)
# dm = depmed(G)
# B = bag(D, G)
# L = loop(D, B, dm=dm)
# which(L$outliers==1)
# date()
```

---

GlobTestMissing

*Detection of global group effect*

---

### Description

Detection of global group effect

### Usage

```
GlobTestMissing(X1, X2, nperm = 100)
```

### Arguments

X1	Matrix of expression levels in first group. Rows represent features, columns represent samples.
X2	Matrix of expression levels in second group. Rows represent features, columns represent samples.
nperm	Number of permutations.

### Details

Tests a global effect for a set of molecular features (e.g. genes, proteins,...) between the two groups of samples. Missing values are allowed in the expression data. Samples of the two groups are supposed to be unpaired.

### Value

The p-value of a permutation test.

**Author(s)**

Klaus Jung

**References**

Jung K, Dihazi H, Bibi A, Dihazi GH and Beissbarth T (2014): Adaption of the Global Test Idea to Proteomics Data with Missing Values. *Bioinformatics*, **30**, 1424-30. doi: [10.1093/bioinformatics/btu062](https://doi.org/10.1093/bioinformatics/btu062)

**Examples**

```
### Global comparison of a set of 100 proteins between two experimental groups,
### where (tau * 100) percent of expression levels are missing.
n1 = 10
n2 = 10
d = 100
tau = 0.1
X1 = t(matrix(rnorm(n1*d, 0, 1), n1, d))
X2 = t(matrix(rnorm(n2*d, 0.1, 1), n2, d))
X1[sample(1:(n1*d), tau * (n1*d))] = NA
X2[sample(1:(n2*d), tau * (n2*d))] = NA
GlobTestMissing(X1, X2, nperm=100)
```

gridfun

*Specifies grid for the calculation of the halfspace location depths***Description**

Specifies a k-dimensional array as grid for the calculation of the halfspace location depths.

**Usage**

```
gridfun(D, grid.size, k = 4)
```

**Arguments**

D	Data set with rows representing the individuals and columns representing the features. In the case of three dimensions, the colnames of D must be c("x", "y", "z").
grid.size	Number of grid points in each dimension.
k	Number of dimensions of the grid. Needs only be specified if D has more than columns.

**Details**

D must have at least three columns. If D has three columns, automatically a 3-dimensional grid is generated. If D has more than three columns, k must be specified.

**Value**

A list containing the following elements:

***H*** The k-dimensional array.

In the case of a 3-dimensional array, additional elements are:

***grid.x***, ***grid.y***, ***grid.z*** The coordinates of the grid points at each dimension.

In the case that the array has more than three dimensions, additional elements are:

***grid.k*** A matrix with the coordinates of the grid. Row represents dimensions and columns represent grid points.

**Author(s)**

Jochen Kruppa, Klaus Jung

---

 hldepth

*Calculates the halfspace location depth*

---

**Description**

Calculates the halfspace location depth for each point in a given grid.

**Usage**

```
hldepth(D, G, verbose = TRUE)
```

**Arguments**

D	Data set with rows representing the individuals and columns representing the features. In the case of three dimensions, the colnames of D must be c("x", "y", "z").
G	List containing the grid information produced by <a href="#">gridfun</a> .
verbose	Logical. Indicates whether progress information is printed during calculation.

**Details**

Calculation of the halfspace location depth at each grid point is mandatory before calculating the depth median ([depmed](#)), the bag ([bag](#)) and the loop ([loop](#)). Ideally, the output is assigned to the array H produced by [gridfun](#).

**Value**

***H*** An array of the same dimension as the array in argument G. The elements contain the halfspace location depth at the related grid location.

**Author(s)**

Jochen Kruppa, Klaus Jung

**References**

Rousseeuw, P. J., Ruts, I., & Tukey, J. W. (1999). The bagplot: a bivariate boxplot. *The American Statistician*, 53(4), 382-387.

**Examples**

```
## Attention: calculation is currently time-consuming.
## Remove #-Symbols to run examples

## A 3-dimensional example data set D1
# n <- 200
# x1 <- rnorm(n, 0, 1)
# y1 <- rnorm(n, 0, 1)
# z1 <- rnorm(n, 0, 1)
# D1 <- data.frame(cbind(x1, y1, z1))
# colnames(D1) <- c("x", "y", "z")

## Specification of the grid and calculation of the halfspace location depth at each grid location.
# G <- gridfun(D1, grid.size=20)
# G$H <- hldepth(D1, G, verbose=TRUE)
```

---

iter\_matrix

*Genetic algorithm for generating correlated binary data*


---

**Description**

Starts the genetic algorithm based on a start matrix with specified marginal probabilities.

**Usage**

```
iter_matrix(X0, R, T = 1000, e.min = 1e-04, plt = TRUE, perc = TRUE)
```

**Arguments**

X0	Start matrix with specified marginal probabilities. Can be generated by <a href="#">start_matrix</a> .
R	Desired correlation matrix the data should have after running the genetic algorithm.
T	Maximum number of iterations after which the genetic algorithm stops.
e.min	Minimum error (RMSE) between the correlation of the iterated data matrix and R.
plt	Boolean parameter that indicates whether to plot e.min versus the iteration step.
perc	Boolean parameter that indicates whether to print the percentage of iteration steps relative to T.

**Details**

In each step, the genetic algorithm swaps two randomly selected entries in each column of  $X_0$ . Thus it can be guaranteed that the marginal probabilities do not change. If the correlation matrix is closer to  $R$  than that of  $x_0(t-1)$ ,  $X_0(t)$  replaces  $X_0(t-1)$ .

**Value**

A list with four entries:

**$X_t$**  Final representative data matrix with specified marginal probabilities and a correlation as close as possible to  $R$

**$t$**  Number of performed iteration steps ( $t \leq T$ )

**$R_t$**  Empirical correlation matrix of  $X_t$

**$RMSE$**  Final RSME error between desired and achieved correlation matrix

**Author(s)**

Jochen Kruppa, Klaus Jung

**References**

Kruppa, J., Lepenies, B., & Jung, K. (2018). A genetic algorithm for simulating correlated binary data from biomedical research. *Computers in biology and medicine*, **92**, 1-8. doi: [10.1016/j.combiomed.2017.10.023](https://doi.org/10.1016/j.combiomed.2017.10.023)

**Examples**

```
### Generation of the representative matrix Xt
X0 <- start_matrix(p = c(0.5, 0.6), k = 1000)
Xt <- iter_matrix(X0, R = diag(2), T = 10000, e.min = 0.00001)$Xt

### Drawing of a random sample S of size n = 10
S <- Xt[sample(1:1000, 10, replace = TRUE),]
```

---

loop

*Calculates the fence and the loop*

---

**Description**

Calculates the fence and the loop of a gemplot (i.e. the outer gemstone).

**Usage**

```
loop(D, B, inflation = 3, dm)
```



**Arguments**

D	Data set with rows representing the individuals and columns representing the features. In the case of three dimensions, the colnames of D must be c("x", "y", "z").
B	List containing the information about the coordinates of the bag and the convex hull that forms the bag (determined by <a href="#">bag</a> ).
inflation	A numeric value > 0 that specifies the inflation factor of the bag relative to the median (default = 3).
dm	The coordinates of the depth median as produced by <a href="#">depmed</a> .

**Details**

The fence inflates the the bag relative to the depth median by the factor inflation. Data points outside the bag and inside the fence the loop or outer gemstone are flagged as outliers. Data points outside the fence are marked as outliers. In the case of a 3-dimensional data set, the loop can be visualized by an outer gemstone around the inner gemstone or bag.

**Value**

A list containing the following elements:

***coords.loop*** Coordinates of the data points that are inside the convex hull around the loop.

***hull.loop*** A data matrix that contains the indices of the margin data points of the loop that cover the convex hull by triangles. Each row represents one triangle. The indices correspond to the rows of *coords.loop*.

***coords.fence*** Coordinates of the grid points that are inside the fence but outside the bag.

***hull.fence*** A data matrix that contains the indices of the margin grid points of the fence that cover the convex hull around the fence by triangles. Each row represents one triangle. The indices correspond to the rows of *coords.fence*.

***outliers*** A vector of length equal to the sample size. Data points that are inside the fence are labelled by 0 and values outside the fence (i.e. outliers) are labelled by 1.

**Author(s)**

Jochen Kruppa, Klaus Jung

**References**

Rousseeuw, P. J., Ruts, I., & Tukey, J. W. (1999). The bagplot: a bivariate boxplot. *The American Statistician*, **53**(4), 382-387. doi: [10.1080/00031305.1999.10474494](https://doi.org/10.1080/00031305.1999.10474494)

Kruppa, J., & Jung, K. (2017). Automated multigroup outlier identification in molecular high-throughput data using bagplots and gemplots. *BMC bioinformatics*, **18**(1), 1-10. <https://link.springer.com/article/10.1186/s12859-017-1645-5>

**Examples**

```

## Attention: calculation is currently time-consuming.
## Remove #-Symbols to run examples

## Two 3-dimensional example data sets D1 and D2
# n <- 200
# x1 <- rnorm(n, 0, 1)
# y1 <- rnorm(n, 0, 1)
# z1 <- rnorm(n, 0, 1)
# D1 <- data.frame(cbind(x1, y1, z1))
# x2 <- rnorm(n, 1, 1)
# y2 <- rnorm(n, 1, 1)
# z2 <- rnorm(n, 1, 1)
# D2 <- data.frame(cbind(x2, y2, z2))
# colnames(D1) <- c("x", "y", "z")
# colnames(D2) <- c("x", "y", "z")

## Placing outliers in D1 and D2
# D1[17,] = c(4, 5, 6)
# D2[99,] = -c(3, 4, 5)

## Grid size and graphic parameters
# grid.size <- 20
# red <- rgb(200, 100, 100, alpha = 100, maxColorValue = 255)
# blue <- rgb(100, 100, 200, alpha = 100, maxColorValue = 255)
# yel <- rgb(255, 255, 102, alpha = 100, maxColorValue = 255)
# white <- rgb(255, 255, 255, alpha = 100, maxColorValue = 255)
# require(rgl)
# material3d(color=c(red, blue, yel, white),
# alpha=c(0.5, 0.5, 0.5, 0.5), smooth=FALSE, specular="black")

## Calculation and visualization of gemplot for D1
# G <- gridfun(D1, grid.size=20)
# G$H <- hldepth(D1, G, verbose=TRUE)
# dm <- depmed(G)
# B <- bag(D1, G)
# L <- loop(D1, B, dm=dm)
# bg3d(color = "gray39" )
# points3d(D1[L$outliers==0,1], D1[L$outliers==0,2], D1[L$outliers==0,3], col="green")
# text3d(D1[L$outliers==1,1], D1[L$outliers==1,2], D1[L$outliers==1,3],
# as.character(which(L$outliers==1)), col=yel)
# spheres3d(dm[1], dm[2], dm[3], col=yel, radius=0.1)
# material3d(1,alpha=0.4)
# gem(B$coords, B$hull, red)
# gem(L$coords.loop, L$hull.loop, red)
# axes3d(col="white")

## Calculation and visualization of gemplot for D2
# G <- gridfun(D2, grid.size=20)
# G$H <- hldepth(D2, G, verbose=TRUE)
# dm <- depmed(G)
# B <- bag(D2, G)

```

```
# L <- loop(D2, B, dm=dm)
# points3d(D2[L$outliers==0,1], D2[L$outliers==0,2], D2[L$outliers==0,3], col="green")
# text3d(D2[L$outliers==1,1], D2[L$outliers==1,2], D2[L$outliers==1,3],
# as.character(which(L$outliers==1)), col=yel)
# spheres3d(dm[1], dm[2], dm[3], col=yel, radius=0.1)
# gem(B$coords, B$hull, blue)
# gem(L$coords.loop, L$hull.loop, blue)
```

---

RepeatedHighDim

*RepeatedHighDim Package*

---

## Description

A comprehensive toolkit for repeated high-dimensional analysis.

## Details

The RepeatedHighDim-package is a collection of functions for the analysis of high-dimensional repeated measures data, e.g. from Omics experiments. It provides function for outlier detection, differential expression analysis, self-contained gene-set testing, and generation of correlated binary data.

For more information and examples, please refer to the package documentation and the tutorial available at <https://software.klausjung-lab.de/>.

## Functions

This package includes the following functions:

### B:

- `bag`: Calculates the bag.

### D:

- `depmed`: Calculates the depth median.

### F:

- `fc_ci`: Calculates adjusted confidence intervals.
- `fc_plot`: Creates a volcano plot of adjusted confidence intervals.

### G:

- `GA_diagplot`: Generates a diagnostic plot for comparing two correlation matrices.
- `gem`: Plots a gemstone to an interactive graphics device.
- `GlobTestMissing`: Detects global group effects.
- `gridfun`: Specifies a grid for calculating halfspace location depths.

### H:

- `hldepth`: Calculates the halfspace location depth.

**I:**

- `iter_matrix`: Implements a genetic algorithm for generating correlated binary data.

**L:**

- `loop`: Calculates the fence and the loop.

**R:**

- `RHighDim`: Detects global group effects.
- `rho_bounds`: Calculates lower and upper bounds for pairwise correlations.
- `rmvbinary_EP`: Simulates correlated binary variables using the algorithm by Emrich and Piedmonte (1991).
- `rmvbinary_QA`: Simulates correlated binary variables using the algorithm by Qaqish (2003).

**S:**

- `sequence_probs`: Calculates probabilities for binary sequences.
- `start_matrix`: Sets up the start matrix.
- `summary_RHD`: Provides a summary of the RHighDim function.

**T:**

- `TestStatSimple`: Calculates the test statistic for RHighDim.
- `TestStatSP`: Calculates the test statistic for RHighDim.

**Author(s)**

**Maintainer:** Klaus Jung (<klaus.jung@tiho-hannover.de>)

**Other contributors:**

- Jochen Kruppa (<j.kruppa@hs-osnabrueck.de>)
- Sergej Ruff (<Sergej.Ruff@tiho-hannover.de>)

If you have any questions, suggestions, or issues, please feel free to contact the maintainer, Klaus Jung (<klaus.jung@tiho-hannover.de>).

**See Also**

For more information, please refer to the package's documentation and the tutorial: <https://software.klausjung-lab.de/>.

---

**RHighDim***Detection of global group effect*

---

**Description**

Detection of global group effect

**Usage**

```
RHighDim(X1, X2, paired = TRUE)
```

**Arguments**

X1	Matrix of expression levels in first group. Rows represent features, columns represent samples.
X2	Matrix of expression levels in second group. Rows represent features, columns represent samples.
paired	FALSE if samples are unpaired, TRUE if samples are paired.

**Details**

Global test for a set of molecular features (e.g. genes, proteins,...) between two experimental groups. Paired or unpaired design is allowed.

**Value**

An object that contains the test results. Contents can be displayed by the summary function.

**Author(s)**

Klaus Jung

**References**

Brunner, E (2009) Repeated measures under non-sphericity. *Proceedings of the 6th St. Petersburg Workshop on Simulation*, 605-609.

Jung K, Becker B, Brunner B and Beissbarth T (2011) Comparison of Global Tests for Functional Gene Sets in Two-Group Designs and Selection of Potentially Effect-causing Genes. *Bioinformatics*, **27**, 1377-1383. doi: [10.1093/bioinformatics/btr152](https://doi.org/10.1093/bioinformatics/btr152)

**Examples**

```
### Global comparison of a set of 100 genes between two experimental groups.  
X1 = matrix(rnorm(1000, 0, 1), 10, 100)  
X2 = matrix(rnorm(1000, 0.1, 1), 10, 100)  
RHD = RHighDim(X1, X2, paired=FALSE)  
summary_RHD(RHD)
```

---

`rho_bounds`*Calculate lower and upper the bounds for pairwise correlations*

---

**Description**

Calculate lower and upper the bounds for pairwise correlations

**Usage**

```
rho_bounds(R, p)
```

**Arguments**

<code>R</code>	Correlation matrix
<code>p</code>	Vector of marginal frequencies

**Details**

The function calculates upper and lower bounds for pairwise correlations given a vector of marginal probabilities as detailed in Emrich and Piedmonte (1991).

**Value**

A list with three entries:

*L* Matrix of lower bounds

*U* Matrix of upper bounds

*Z* Matrix that indicates whether specified correlations in *R* are bigger or smaller than the calculated bounds

**Author(s)**

Jochen Kruppa, Klaus Jung

**References**

Emrich, L.J., Piedmonte, M.R.: A method for generating highdimensional multivariate binary variates. *The American Statistician*, **45(4)**, 302 (1991). doi: [10.1080/00031305.1991.10475828](https://doi.org/10.1080/00031305.1991.10475828)

**Examples**

```
### A simple example
R <- diag(4)
p <- c(0.1, 0.2, 0.4, 0.5)

rho_bounds(R, p)
```

---

rmvbinary_EP	<i>Simulating correlated binary variables using the algorithm by Emrich and Piedmonte (1991)</i>
--------------	--

---

### Description

Generation of random sample of binary correlated variables

### Usage

```
rmvbinary_EP(n, R, p)
```

### Arguments

n	Sample size
R	Correlation matrix
p	Vector of marginal probabilities

### Details

The function implements the algorithm proposed by Emrich and Piedmonte (1991) to generate a random sample of  $d$  ( $=\text{length}(p)$ ) correlated binary variables. The sample is generated based on given marginal probabilities  $p$  of the  $d$  variables and their correlation matrix  $R$ . The algorithm generates first determines an appropriate correlation matrix  $R'$  for the multivariate normal distribution. Next, a sample is drawn from  $N_d(0, R')$  and each variable is finally dichotomized with respect to  $p$ .

### Value

Sample  $(n \times p)$ -matrix with representing a random sample of size  $n$  from the specified multivariate binary distribution.

### Author(s)

Jochen Kruppa, Klaus Jung

### References

Emrich, L.J., Piedmonte, M.R. (1991) A method for generating highdimensional multivariate binary variates. *The American Statistician*, **45(4)**, 302. doi: [10.1080/00031305.1991.10475828](https://doi.org/10.1080/00031305.1991.10475828)

### Examples

```
## Generation of a random sample  
rmvbinary_EP(n = 10, R = diag(2), p = c(0.5, 0.6))
```

---

`rmvbinary_QA`*Simulating correlated binary variables using the algorithm by Qaqish (2003)*

---

## Description

Generation of random sample of binary correlated variables

## Usage

```
rmvbinary_QA(n, R, p)
```

## Arguments

<code>n</code>	Sample size
<code>R</code>	Correlation matrix
<code>p</code>	Vector of marginal probabilities

## Details

The function implements the algorithm proposed by Qaqish (2003) to generate a random sample of  $d$  ( $=\text{length}(p)$ ) correlated binary variables. The sample is generated based on given marginal probabilities  $p$  of the  $d$  variables and their correlation matrix  $R$ . The algorithm starts by generating a data for the first variable  $X_1$  and generates successively the data for  $X_2, \dots$  based on their conditional probabilities  $P(X_j|X_{[i-1]}, \dots, X_1), j=1, \dots, d$ .

## Value

Sample  $(n \times p)$ -matrix representing a random sample of size  $n$  from the specified multivariate binary distribution.

## Author(s)

Jochen Kruppa, Klaus Jung

## References

Qaqish, B. F. (2003) A family of multivariate binary distributions for simulating correlated binary variables with specified marginal means and correlations. *Biometrika*, **90**(2), 455-463. doi: [10.1093/biomet/90.2.455](https://doi.org/10.1093/biomet/90.2.455)

## Examples

```
## Generation of a random sample  
rmvbinary_QA(n = 10, R = diag(2), p = c(0.5, 0.6))
```



---

sequence_probs	<i>Calculation of probabilities for binary sequences</i>
----------------	--

---

**Description**

Calculation of probabilities for binary sequences based on the final matrix generated by the genetic algorithm

**Usage**

```
sequence_probs(Xt)
```

**Arguments**

Xt                      Representative matrix generated by the genetic algorithm with [iter\\_matrix](#)

**Details**

Observation of binary correlated binary data can be expressed as binary sequences. In the case of two binary variables possible observations are (0,0), (0,1), (1,0) and (1,1). In general,  $2^m$  binary sequences are possible, where  $m$  is the number of binary variables. Based on the representative matrix generated by the genetic algorithm the probability for each binary sequence is determined.

**Value**

A vector of probabilities for the binary sequences

**Author(s)**

Jochen Kruppa, Klaus Jung

**References**

Kruppa, J., Lepenies, B., & Jung, K. (2018). A genetic algorithm for simulating correlated binary data from biomedical research. *Computers in biology and medicine*, **92**, 1-8. doi: [10.1016/j.combiomed.2017.10.023](https://doi.org/10.1016/j.combiomed.2017.10.023)

**Examples**

```
### Generation of the representative matrix Xt
X0 <- start_matrix(p = c(0.5, 0.6), k = 1000)
Xt <- iter_matrix(X0, R = diag(2), T = 10000, e.min = 0.00001)$Xt

### Calculation of probabilities for binary sequences
sequence_probs(Xt = Xt)
```

---

`start_matrix`*Setup of the start matrix*

---

### Description

Generation of the start matrix with n rows and specified marginal probabilities p.

### Usage

```
start_matrix(p, k)
```

### Arguments

p	Marginal probabilities of the start matrix.
k	Number of rows to be generated.

### Details

The start matrix needs to be setup for further use in the genetic algorithm implemented in the function `iter_matrix`. For high-dimensional cases or if the marginal probabilities have multiple decimal places, the number k of rows should be large (up to multiple thousand).

### Value

A (k x p)-Matrix with with entries 0 and 1 according to the specified marginal probabilities p.

### Author(s)

Jochen Kruppa, Klaus Jung

### References

Kruppa, J., Lepenies, B., & Jung, K. (2018). A genetic algorithm for simulating correlated binary data from biomedical research. *Computers in biology and medicine*, **92**, 1-8. doi: [10.1016/j.combiomed.2017.10.023](https://doi.org/10.1016/j.combiomed.2017.10.023)

### Examples

```
X0 <- start_matrix(p = c(0.5, 0.6), k = 10000)

## check if p can be restored
apply(X0, 2, mean)
```

---

summary\_RHD

*Summary of RHighDim function*

---

## Description

Summary of RHighDim function

## Usage

```
summary_RHD(object, ...)
```

## Arguments

object	An object provided by the RHighDim function.
...	additional arguments affecting the summary produced.

## Details

Summarizes the test results obtained by the RHighDim function.

## Value

No value

## Author(s)

Klaus Jung

## References

Brunner, E (2009) Repeated measures under non-sphericity. *Proceedings of the 6th St. Petersburg Workshop on Simulation*, 605-609.

Jung K, Becker B, Brunner B and Beissbarth T (2011) Comparison of Global Tests for Functional Gene Sets in Two-Group Designs and Selection of Potentially Effect-causing Genes. *Bioinformatics*, **27**, 1377-1383. doi: [10.1093/bioinformatics/btr152](https://doi.org/10.1093/bioinformatics/btr152)

## Examples

```
### Global comparison of a set of 100 genes between two experimental groups.  
X1 = matrix(rnorm(1000, 0, 1), 10, 100)  
X2 = matrix(rnorm(1000, 0.1, 1), 10, 100)  
RHD = RHighDim (X1, X2, paired=FALSE)  
summary_RHD(RHD)
```

---

TestStatSimple	<i>Calculates test statistic for RHighDim</i>
----------------	---

---

**Description**

Calculates test statistic for RHighDim

**Usage**

```
TestStatSimple(Y, H)
```

**Arguments**

Y	A matrix.
H	A matrix.

**Value**

A list.

---

TestStatSP	<i>Calculates test statistic for RHighDim</i>
------------	---

---

**Description**

Calculates test statistic for RHighDim

**Usage**

```
TestStatSP(Y1, Y2)
```

**Arguments**

Y1	A matrix.
Y2	A matrix.

**Value**

A list.

# Index

- \* **bag**
  - RepeatedHighDim, 19
- \* **gem**
  - RepeatedHighDim, 19
- \* **outlier**
  - RepeatedHighDim, 19
  
- bag, 2, 10, 14, 17, 19
  
- depmed, 4, 14, 17, 19
  
- fc\_ci, 5, 19
- fc\_plot, 7, 19
  
- GA\_diagplot, 8, 19
- gem, 10, 19
- GlobTestMissing, 12, 19
- gridfun, 2, 4, 13, 14, 19
  
- hldepth, 2, 4, 14, 20
  
- iter\_matrix, 15, 20, 25, 26
  
- loop, 2, 10, 14, 16, 20
  
- RepeatedHighDim, 19
- RepeatedHighDim-package
  - (RepeatedHighDim), 19
- RHighDim, 20, 21
- rho\_bounds, 20, 22
- rmvbinary\_EP, 20, 23
- rmvbinary\_QA, 20, 24
  
- sequence\_probs, 20, 25
- start\_matrix, 15, 20, 26
- summary\_RHD, 20, 27
  
- TestStatSimple, 20, 28
- TestStatSP, 20, 28