Package 'RepeatedHighDim'

May 7, 2026

Type Package

Title Methods for High-Dimensional Repeated Measures Data

Version 2.5.0

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)

URL https://software.klausjung-lab.de

Encoding UTF-8

RoxygenNote 7.3.3

Imports alphahull, circlize, ComplexHeatmap, ddalpha, dplyr, FNN, geometry, ggplot2, graphics, grDevices, MASS, mvtnorm, netmeta, nlme, patchwork, progress, RColorBrewer, rgl, rlang, scales, Seurat, SeuratObject, stats, utils

Suggests BiocManager, invgamma, limma

Depends R (>= 3.5)

NeedsCompilation no

Repository CRAN

Date/Publication 2026-01-13 09:00:02 UTC

Contents

`bag` *Calculates the bag*

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 `gridfun` and the halfspace location depths calculated by `hldepth`.

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 containg 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

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

See Also

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

Examples

```
## Attention: calculation is currently time-consuming.

## Not run:
## 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")

# Calucation 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")

# Calucation 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)

## End(Not run)
```

---

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.

See Also

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

Examples

```
## Attention: calculation is currently time-consuming.
## Not run:

# 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

## End(Not run)
```

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

See Also

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

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 fc_ci
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 -log10-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

See Also

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

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.compbiomed.2017.10.023

See Also

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

Examples

```
## Not run:

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")


## End(Not run)
```

gem *Plots a gemstone to an interactive graphics device*

Description

Plots a gemstone to an interactive graphics device.

Usage

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

Arguments

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

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

See Also

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

Examples

```
## Attention: calculation is currently time-consuming.
## Not run:

# 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")

# Calucation 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")

# Calucation 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$H = hldepth(D, G, verbose=TRUE)
dm = depmed(G)
B = bag(D, G)
L = loop(D, B, dm=dm)
which(L$outliers==1)
date()

## End(Not run)
```

`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 allowd 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

See Also

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

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

See Also

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

---

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 gridfun.
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.

See Also

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

Examples

```
## Attention: calculation is currently time-consuming.
## Not run:

# 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)

## End(Not run)
```

`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 `start_matrix`.
`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 relativ to T.

Details

In each step, the genetic algorithm swaps two randomly selected entries in each column of X0. Thus it can be guaranteed that the marginal probabilities do not change. If the correlation matrix is closer to R than that of x0(t-1), X0(t) replaces X0(t-1).

Value

A list with four entries:

Xt Final representativ data matrix with specified marginal probabilities and a correlation as close as possible to R

t Number of performed iteration steps (t <= T)

Rt Empirical correlation matrix of Xt

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.compbiomed.2017.10.023

See Also

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

Examples

```
### Generation of the representive 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 bag).
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 depmed.

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 represnts 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 represnts 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

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

See Also

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

Examples

```
## Attention: calculation is currently time-consuming.
## Not run:

# 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")

# Calucation 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")

# Calucation 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)

## End(Not run)
```

---

netRNA *netRNA:Network meta-analysis for gene expression data*

Description

This function conducts network meta-analysis using gene expression data to make indirect comparisons between different groups. It computes the p values for each gene and the fold changes, and provides a dataframe containing these results.

Usage

```
netRNA(TE, seTE, treat1, treat2, studlab)
```

Arguments

TE	A list containing log fold changes from two individual studies. Index names of the list should be the gene names; otherwise, each value of the 'name' column in the output dataframe will correspond to the position in the list, rather than gene identifiers.
seTE	A list containing standard errors of log fold changes from two individual studies.
treat1	A vector with Label/Number for first treatment.
treat2	A vector with Label/Number for second treatment.
studlab	A vector containing study labels

Details

The function supports a simple network with three nodes, where one node represents a control group and the two other nodes represent treatment (or diseased) groups. While the user provides fold changes and their standard errors of each treatment versus control as input, the function calculates the fold changes for the indirect comparison between the two treatments.

Value

A list containing the p values for each gene, the fold changes, the upper and lower bounds for the 95% CI of the log fold changes, and a summary dataframe with results for each gene.

Author(s)

Klaus Jung, Sergej Ruff

References

Winter, C., Kosch, R., Ludlow, M. et al. Network meta-analysis correlates with analysis of merged independent transcriptome expression data. *BMC Bioinformatics* **20**, 144 (2019). doi:10.1186/s1285901927059

Rücker G. (2012). Network meta-analysis, electrical networks and graph theory. *Research synthesis methods*, **3(4)**, 312–324. doi:10.1002/jrsm.1058

See Also

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

Examples

```
## Not run:
#'########################
### Data generation ###
########################
n = 100 ### Sample size per group
G = 100 ### Number of genes

### Basic expression, fold change, batch effects and error
alpha.1 = rnorm(G, 0, 1)
alpha.2 = rnorm(G, 0, 1)
beta.1 = rnorm(G, 0, 1)
beta.2 = rnorm(G, 0, 1)
gamma.1 = rnorm(G, 0, 1)
gamma.2 = rnorm(G, 2, 1)
delta.1 = sqrt(invgamma::rinvgamma(G, 1, 1))
delta.2 = sqrt(invgamma::rinvgamma(G, 1, 2))
sigma.g = rep(1, G)

# Generate gene names
gene_names <- paste("Gene", 1:G, sep = "")

### Data matrices of control and treatment (disease) groups
C.1 = matrix(NA, G, n)
C.2 = matrix(NA, G, n)
T.1 = matrix(NA, G, n)
T.2 = matrix(NA, G, n)

for (j in 1:n) {
 C.1[,j] = alpha.1 + (0 * beta.1) + gamma.1 + (delta.1 * rnorm(1, 0, sigma.g))
 C.2[,j] = alpha.1 + (0 * beta.2) + gamma.2 + (delta.2 * rnorm(1, 0, sigma.g))
 T.1[,j] = alpha.2 + (1 * beta.1) + gamma.1 + (delta.1 * rnorm(1, 0, sigma.g))
 T.2[,j] = alpha.2 + (1 * beta.2) + gamma.2 + (delta.2 * rnorm(1, 0, sigma.g))
}

study1 = cbind(C.1, T.1)
study2 = cbind(C.2, T.2)

# Assign gene names to row names
#rownames(study1) <- gene_names
#rownames(study2) <- gene_names
############################
### Differential Analysis ###
############################

if(check_limma()){
### study1: treatment A versus control
group = gl(2, n)
M = model.matrix(~ group)
fit = limma::lmFit(study1, M)
fit = limma::eBayes(fit)
p.S1 = fit$p.value[,2]
```

```
fc.S1 = fit$coefficients[,2]
fce.S1 = sqrt(fit$s2.post) * sqrt(fit$cov.coefficients[2,2])

### study2: treatment B versus control
group = gl(2, n)
M = model.matrix(~ group)
fit = limma::lmFit(study2, M)
fit = limma::eBayes(fit)
p.S2 = fit$p.value[,2]
fc.S2 = fit$coefficients[,2]
fce.S2 = sqrt(fit$s2.post) * sqrt(fit$cov.coefficients[2,2])



############################
### Network meta-analysis ###
############################
p.net = rep(NA, G)
fc.net = rep(NA, G)
treat1 = c("uninfected", "uninfected")
treat2 = c("ZIKA", "HSV1")
studlab = c("experiment1", "experiment2")
fc.true = beta.2 - beta.1

TEs <- list(fc.S1, fc.S2)
seTEs <- list(fce.S1, fce.S2)
}

# Example usage:
test <- netRNA(TE = TEs, seTE = seTEs, treat1 = treat1, treat2 = treat2, studlab = studlab)

## End(Not run)
```

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.

N:

- netRNA: network meta-analysis using gene expression data.

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:

- scTC_bpplot: Post-trim breakpoint heatmap for scTrimClust results.
- scTC_trim_effect: Compare scTrimClust trimming against default Seurat analysis.
- scTrimClust: Clustering with alpha hull-based outlier detection.
- sequence_probs: Calculates probabilities for binary sequences.
- start_matrix: Sets up the start matrix.
- summary_RHD: Provides a summary of the RHighDim function.

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> ,second maintainer)

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

See Also

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

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

R	Correlation matrix
p	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

See Also

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

Examples

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

rho_bounds(R, p)
```

`rmvbinary_EP` *Simulating correlated binary variables using the algorithm by Emrich and Piedmonte (1991)*

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 (=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 finnaly dichotomized with respect to p.

Value

Sample (n x 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

See Also

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

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

n	Sample size
R	Correlation matrix
p	Vector of marginal probabilities

Details

The function implements the algorithm proposed by Qaqish (2003) to generate a random sample of d (=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 succesively the data for X_2, ... based on their conditional probabilities P(X_j|X_[i-1],...,X_1), j=1,...,d.

Value

Sample (n x 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

See Also

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

Examples

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

---

scTC_bpplot *scTC_bpplot: Post-trim breakpoint heatmap for scTrimClust results*

Description

Generates a heatmap showing the percentage overlap of marker genes between the original (untrimmed) cluster markers and markers identified after trimming at various percentages.

Usage

```
scTC_bpplot(
  ...,
  trim_percent_vector,
  plot_title = "scTrimClust: Post-trim Breakpoint Heatmap",
  legend_title = "Percent markes\nof non-trimmed",
  color = brewer.pal(n = 11, name = "RdYlBu")
)
```

Arguments

...	Two or more data.frames/tibbles containing marker genes from 'FindAllMarkers'.
trim_percent_vector	Numeric vector of trim percentages.
plot_title	Character string for the heatmap title.
legend_title	Character string for the legend title.
color	Color palette for the heatmap.

Details

scTC_bpplot compares marker genes between the original (untrimmed) clustering and trimmed versions. For each cluster, it calculates what percentage of the original markers are retained at each trim level. Clusters are ordered by the number of markers in the original (untrimmed) results.

At least two data.frames/tibbles containing marker genes from 'FindAllMarkers' (from the 'Seurat' package) should be provided to ... as input. The first data frame should be the original (untrimmed) results, followed by trimmed results.

trim_percent_vector must be a numeric vector of trim percentages corresponding to each input data frame (e.g., c(0,10,20,30,40) for untrimmed, 10 input data.frames/tibbles.

Value

A ComplexHeatmap object

Examples

```
## Not run:
scTC_bpplot(
  covid_markers = RepeatedHighDim:::covid_markers,
  robust_covid_markers = RepeatedHighDim:::robust_covid_markers,
  robust_covid_markers_02trim = RepeatedHighDim:::robust_covid_markers_02trim,
  robust_covid_markers_03trim = RepeatedHighDim:::robust_covid_markers_03trim,
  robust_covid_data = RepeatedHighDim:::robust_covid_data,
  trim_percent_vector = c(0, 10, 20, 30, 40),
  plot_title = "CLR, nPCs:5, nFeatures:1000",
  legend_title = "Percent markers of non-trimmed"
)

## End(Not run)
```

scTC_trim_effect *scTC_trim_effect: Compare scTrimClust trimming against default Seurat analysis*

Description

Visualizes the impact of scTrimClust's trimming by comparing gene sets between: 1) Default Seurat analysis (no trimming) 2) scTrimClust post-trimming results

Usage

```
scTC_trim_effect(
  method_pairs,
  method_colors,
  set_colors = setNames(c("#4D4D4D", "#AEAEAE", "#E6E6E6"), c("S1:standard",
    "S2:intersect", "S3:trimmed")),
 heatmap_color_palette = colorRamp2(seq(0, 100, 1), heat.colors(101, rev = TRUE)),
  column_title = "",
  row_names_side = "right",
  legend_name = "No. of\nmarkers",
  row_names_gp = 10,
  column_title_gp = 12
)
```

Arguments

method_pairs	A named list of method comparisons. Each element should be a list with two components (*data1 (untrimmed)* and *data2 (trimmed)*) containing data frames with: • *cluster*: Cluster identifiers • *gene*: Gene identifiers
method_colors	Named vector of colors for method annotations. Names should match the names in *method_pairs*.
set_colors	Named vector of colors for set annotations (S1-S3). Default: *c("S1:standard", "S2:intersect", "S3:trimmed")* with grey colors.
heatmap_color_palette	Color mapping function for heatmap. Default: *colorRamp2(seq(0, 100, 1), heat.colors(101, rev = TRUE))*.
column_title	Main title for the heatmap columns.
row_names_side	Side for row names ("left" or "right"). Default: "right".
legend_name	Title for the heatmap legend. Default: "No. of markers".
row_names_gp	Graphics parameters for row names. Default: 10.
column_title_gp	Graphics parameters for column title. Default: 12.

Details

scTC_trim_effect creates a heatmap showing the percentage differences in gene sets between method pairs across clusters. The heatmap shows three components for each method comparison:

- Column 1-3: Unique to method1 (untrimmed), Intersection, Unique to method2 (trimmed)
- Rows represent (cell) clusters with counts from first method in parentheses
- Columns are split by method pairs

Value

A *Heatmap* object from the ComplexHeatmap package.

Examples

```
## Not run:

method_pairs <- list(
  CLR = list(
    data1 = RepeatedHighDim:::scTC_eff_clr,
    data2 = RepeatedHighDim:::scTC_eff_clr_robust
  ),
  LogNorm = list(
    data1 = RepeatedHighDim:::scTC_eff_log,
    data2 = RepeatedHighDim:::scTC_eff_log_robust
  )
)

method_colors <- setNames(grey.colors(2), c("CLR", "LogNorm"))

scTC_trim_effect(
  method_pairs = method_pairs,
  method_colors = method_colors,
  column_title = "nPCs:5, nFeatures:1000"
)

set_colors <- grey.colors(3)
names(set_colors) <- c("S1:standard", "S2:intersect", "S3:trimmed")

scTC_trim_effect(
  method_pairs = method_pairs,
  method_colors = method_colors,
  set_colors = setNames(c("blue", "green", "red"), names(set_colors)),
  heatmap_color_palette = colorRamp2(c(0, 50, 100), c("white", "pink", "purple")),
  column_title = "Custom Color Example"
)

## End(Not run)
```

scTrimClust *scTrimClust: Cluster visualization with alpha hull-based outlier detection*

Description

Visualizes cell clusters in low-dimensional space (t-SNE, UMAP, etc.) and identifies/removes potential outliers based on their distance from cluster alpha hulls.

Usage

```
scTrimClust(
  object,
  dims = c(1, 2),
  cells = NULL,
  cols = NULL,
  pt.size = NULL,
  reduction = NULL,
  group.by = NULL,
  split.by = NULL,
  shape.by = NULL,
  order = NULL,
  shuffle = FALSE,
  seed = 1,
  label = FALSE,
  label.size = 4,
  label.color = "black",
  label.box = FALSE,
  repel = FALSE,
  alpha = 1,
  stroke.size = NULL,
  cells.highlight = NULL,
  cols.highlight = "#DE2D26",
  sizes.highlight = 1,
  na.value = "grey50",
  ncol = NULL,
  combine = TRUE,
  raster = NULL,
  raster.dpi = c(512, 512),
  add.alpha.hull = TRUE,
  hull.alpha = 2,
  hull.color = NULL,
  hull.size = 0.5,
  outlier.quantile = 0.4,
  remove.outliers = FALSE,
  outlier.alpha = 0.1,
  outlier.color = NULL,
```

```
  outlier.colors = NULL,
  outline.color = NULL,
  outline.size = 0.5,
  outline.alpha = 1,
  outline.outliers = FALSE
)
```

Arguments

object	A Seurat object containing dimensionality reduction results.
dims	Integer vector of length 2 specifying which dimensions to plot (e.g., c(1, 2)).
cells	Vector of cells to include (NULL uses all cells).
cols	Vector of colors for clusters.
pt.size	Point size for cells.
reduction	Name of dimensionality reduction to use (e.g., "umap", "tsne").
group.by	Metadata column to group cells by (default: 'ident' uses cluster IDs).
split.by	Metadata column to split plots by (creates multiple facets).
shape.by	Metadata column to determine point shapes.
order	Vector specifying order to plot cells (affects z-ordering).
shuffle	Logical to randomly shuffle plotting order.
seed	Random seed for reproducibility when shuffle=TRUE.
label	Logical to add cluster labels.
label.size	Size of cluster labels.
label.color	Color of cluster labels.
label.box	Logical to add background box to labels.
repel	Logical to use ggrepel for label placement.
alpha	Transparency level for points (0-1).
stroke.size	Size of point borders.
cells.highlight	Specific cells to highlight.
cols.highlight	Color(s) for highlighted cells.
sizes.highlight	Size(s) for highlighted cells.
na.value	Color for NA values.
ncol	Number of columns for faceted plots.
combine	Logical to combine multiple plots into one.
raster	Logical to rasterize points (for large datasets).
raster.dpi	Resolution for rasterized points.
add.alpha.hull	Logical to compute and plot alpha hulls.
hull.alpha	Alpha parameter for hull calculation. Higher values produce smoother hulls that encompass more cells (default: 2).

`hull.color` Color of the alpha hull lines (default: Null = same color as cluster points).

`hull.size` Thickness of the alpha hull lines (default: 0.5).

`outlier.quantile`
Quantile threshold (0-1) for outlier detection based on hull distance. Cells with distances below this quantile are considered outliers (default: 0.4).

`remove.outliers`
Logical - whether to remove outliers from the returned Seurat object (default: FALSE).

`outlier.alpha` Transparency level for outlier points (0-1; default: 0.1).

`outlier.color` Single color to use for all outlier points. If NULL, uses cluster colors.

`outlier.colors` A named vector of colors to be assigned to outliers.If NULL, uses cluster colors.

`outline.color` Color for the outline of points. If NULL, no outline is added.

`outline.size` Thickness of the outline around points (default: 0.5).

`outline.alpha` Transparency of the outline around points (default: 1).

`outline.outliers`
Logical whether to add outlines to outlier points (default: FALSE).

Value

A list containing:

- *plot*: ggplot object of the visualization with hulls and highlighted outliers
- *object*: Modified Seurat object with outliers removed (if remove.outliers=TRUE)
- *outlier_coords*: Dataframe containing coordinates of outlier cells, their IDs and cluster assignments
- *hull_info*: List containing alpha hull geometries (if add.alpha.hull=TRUE)

Examples

```
## Not run:

scTrimClust(RepeatedHighDim:::seurat_obj,reduction = 'tsne',
group.by = 'CellType',
hull.alpha = 2,
remove.outliers = FALSE,
outlier.quantile = 0.2,
outlier.alpha = 0.3,
outlier.color = "red",
pt.size = 5,
outline.color = "black",
outline.outliers = TRUE)

# second example with custom outlier col per cluster

scTrimClust(RepeatedHighDim:::seurat_obj,reduction = 'tsne',
group.by = 'CellType',
hull.alpha = 2,
```

```
remove.outliers = FALSE,
outlier.quantile = 0.2,
outlier.alpha = 0.3,
outlier.colors = c('TypeA'="black",
'TypeB'='violet','TypeC' ='pink'),
pt.size = 5,
outline.color = "black",
outline.outliers = TRUE)$plot


## End(Not run)
```

`scTrimDist` *ScTrimDist: Trim extreme cells based on kNN distance within cell types*

Description

Identifies and removes extreme (outlier) cells within each cell type or cluster based on k-nearest neighbour (kNN) distances computed in the normalized high-dimensional gene expression space. Cells located in sparsely populated regions at the periphery of clusters are excluded prior to downstream analyses.

Usage

```
scTrimDist(
  seurat_obj,
  celltype_col,
  knn_k = 30,
  keep_frac = 0.05,
  normalization_method = "LogNormalize",
  nfeatures = 2000,
  assay = "RNA",
  npcs = 20,
  resolution = 0.5,
  log2FC_filter = 1,
  pred,
  verbose = TRUE
)
```

Arguments

`seurat_obj`	A `Seurat` object containing single-cell expression data.
`celltype_col`	Character scalar specifying the column in `seurat_obj@meta.data` defining cell types or clusters.
`knn_k`	Integer specifying the number of nearest neighbours.

keep_frac	Numeric in (0,1) specifying the fraction of most extreme cells to remove per cell type.
normalization_method	Normalization method passed to Seurat::NormalizeData.
nfeatures	Number of variable features selected.
assay	Assay used for expression data extraction.
npcs	Number of principal components used downstream.
resolution	Clustering resolution for FindClusters.
log2FC_filter	Minimum log2 fold-change threshold for marker filtering. If NULL, no filtering is applied.
pred	A SingleR result object. Row names must correspond to cell barcodes; pred$labels is used for annotation.
verbose	Logical indicating whether progress messages are printed.

Details

For each cell type (or cluster), a kNN search is performed using the normalized gene expression matrix obtained from a standard Seurat preprocessing workflow. For a given cell i in cluster k, the Euclidean distances $D^k_{(j,i)}$ to its $j = 1, \ldots, K$ nearest neighbours are computed.

The minimum distance

$$\min D_i^k = \min_{j=1,\ldots,K} D^k_{(j,i)}$$

is used as a measure of local neighbourhood density. Cells with large minimum distances are interpreted as extreme or non-representative cells.

A fraction α (specified via keep_frac) of the most extreme cells is removed per cluster, defined as cells with

$$\min D_i^k > Q_{1-\alpha}$$

where $Q_{1-\alpha}$ is the $(1-\alpha)$ quantile of the minimum kNN distance distribution within the cluster.

After trimming, the remaining cells are re-normalized and reprocessed using standard Seurat workflows. Cell type annotations are assigned using a **precomputed SingleR result** supplied by the user, and cluster-specific marker genes are identified.

Value

A named list containing:

- plot_outliers: ggplot showing t-SNE with outliers highlighted.
- trimmed_object: Seurat object after trimming and reprocessing.
- all_markers: Data frame of marker genes.
- knn_res: List of kNN results per cell type.

sequence_probs *Calculation of probabilities for binary sequences*

Description

Calculation of proabilities 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.compbiomed.2017.10.023

See Also

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

Examples

```
### Generation of the representive 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.compbiomed.2017.10.023

See Also

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

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

See Also

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

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)
```

Index