# Package 'DIscBIO'

November 13, 2020

**Date** 2020-11-13

**Title** A User-Friendly Pipeline for Biomarker Discovery in Single-Cell Transcriptomics

**Version** 1.1.0

**Description** An open, multi-algorithmic pipeline for easy, fast and efficient analysis of cellular sub-populations and the molecular signatures that characterize them. The pipeline consists of four successive steps: data pre-processing, cellular clustering with pseudo-temporal ordering, defining differential expressed genes and biomarker identification. This package implements extensions of the work published by Ghannoum et. al. (2019) <doi:10.1101/700989>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Imports** methods, TSCAN, boot, httr, mclust, statmod, igraph, RWeka, philentropy, NetIndices, png, grDevices, readr, RColorBrewer, ggplot2, rpart, fpc, cluster, rpart.plot, tsne, AnnotationDbi, org.Hs.eg.db, graphics, stats, utils, impute

**Depends** R (>= 4.0), SingleCellExperiment

**Suggests** testthat, Seurat

**LazyData** true

**RoxygenNote** 7.1.1

**URL** https://github.com/ocbe-uio/DIscBIO

**BugReports** https://github.com/ocbe-uio/DIscBIO/issues

**Collate** 'DIscBIO-classes.R' 'DIscBIO-generic-ClassVectoringDT.R' 'DIscBIO-generic-ClustDiffGenes.R' 'DIscBIO-generic-Clustexp.R' 'DIscBIO-generic-DEGanalysis.R' 'DIscBIO-generic-DEGanalysis2clust.R' 'DIscBIO-generic-Exprmclust.R' 'DIscBIO-generic-FinalPreprocessing.R' 'DIscBIO-generic-FindOutliers.R' 'DIscBIO-generic-NoiseFiltering.R'

'DIscBIO-generic-Normalizedata.R'
'DIscBIO-generic-PCAplotSymbols.R'
'DIscBIO-generic-PlotmclustMB.R'
'DIscBIO-generic-clusteringOrder.R'
'DIscBIO-generic-clustheatmap.R' 'DIscBIO-generic-comptSNE.R'
'DIscBIO-generic-plotExptSNE.R' 'DIscBIO-generic-plotGap.R'
'DIscBIO-generic-plotLabelstSNE.R'
'DIscBIO-generic-plotOrderTsne.R'
'DIscBIO-generic-plotSilhouette.R'
'DIscBIO-generic-plotSymbolstSNE.R'
'DIscBIO-generic-plottSNE.R'
'DIscBIO-generic-pseudoTimeOrdering.R' 'J48DT.R' 'J48DTeval.R'
'Jaccard.R' 'NetAnalysis.R' 'Networking.R' 'PPI.R'
'PlotMBpca.R' 'RpartDT.R' 'RpartEVAL.R' 'VolcanoPlot.R'
'customConverters.R' 'datasets.R'
'internal-functions-samr-adapted.R' 'internal-functions.R'

**NeedsCompilation** no

**Author** Salim Ghannoum [aut, cph],
Alvaro Köhn-Luque [aut, ths],
Waldir Leoncio [cre, aut],
Damiano Fantini [ctb]

**Maintainer** Waldir Leoncio <w.l.netto@medisin.uio.no>

**Repository** CRAN

**Date/Publication** 2020-11-13 10:20:08 UTC

# R topics documented:

---

`as.DISCBIO` *Convert Single Cell Data Objects to DISCBIO.*

---

## Description

Initialize a DISCBIO-class object starting from a SingleCellExperiment object or a Seurat object

## Usage

```
as.DISCBIO(x, ...)
```

## Arguments

| | |
|---|---|
| `x` | an object of class Seurat or SingleCellExperiment |
| `...` | additional parameters to pass to the function |

### Details

Additional parameters to pass to ‘list‘ include, if x is a Seurat object, "assay", which is a string indicating the assay slot used to obtain data from (defaults to ’RNA’)

### Value

a DISCBIO-class object

---

## check.format *Check format*

---

### Description

Check format

### Usage

```
check.format(y, resp.type, censoring.status = NULL)
```

### Arguments

| | |
|---|---|
| y | y |
| resp.type | resp type |
| censoring.status | censoring status |

---

## ClassVectoringDT *Generating a class vector to be used for the decision tree analysis.*

---

### Description

This function generates a class vector for the input dataset so the decision tree analysis can be implemented afterwards.

### Usage

```
ClassVectoringDT(
  object,
  Clustering = "K-means",
  K,
  First = "CL1",
  Second = "CL2",
  sigDEG,
  quiet = FALSE
)
```

```
## S4 method for signature 'DISCBIO'
ClassVectoringDT(
  object,
  Clustering = "K-means",
  K,
  First = "CL1",
  Second = "CL2",
  sigDEG,
  quiet = FALSE
)
```

## Arguments

| | |
|---|---|
| object | DISCBIO class object. |
| Clustering | Clustering has to be one of the following: ["K-means", "MB"]. Default is "K-means" |
| K | A numeric value of the number of clusters. |
| First | A string vector showing the first target cluster. Default is "CL1" |
| Second | A string vector showing the second target cluster. Default is "CL2" |
| sigDEG | A data frame of the differentially expressed genes (DEGs) generated by running "DEGanalysis()" or "DEGanalysisM()". |
| quiet | If 'TRUE', suppresses intermediary output |

## Value

A data frame.

---

ClustDiffGenes *ClustDiffGenes*

---

## Description

Creates a table of cluster differences

## Usage

```
ClustDiffGenes(
  object,
  K,
  pValue = 0.05,
  fdr = 0.01,
  export = FALSE,
  quiet = FALSE,
  filename_up = "Up-DEG-cluster",
```

```
  filename_down = "Down-DEG-cluster",
  filename_binom = "binomial-DEGsTable",
  filename_sigdeg = "binomial-sigDEG"
)

## S4 method for signature 'DISCBIO'
ClustDiffGenes(
  object,
  K,
  pValue = 0.05,
  fdr = 0.01,
  export = FALSE,
  quiet = FALSE,
  filename_up = "Up-DEG-cluster",
  filename_down = "Down-DEG-cluster",
  filename_binom = "binomial-DEGsTable",
  filename_sigdeg = "binomial-sigDEG"
)
```

## Arguments

| | |
|---|---|
| object | DISCBIO class object. |
| K | A numeric value of the number of clusters. |
| pValue | A numeric value of the p-value. Default is 0.05. |
| fdr | A numeric value of the false discovery rate. Default is 0.01. |
| export | A logical vector that allows writing the final gene list in excel file. Default is TRUE. |
| quiet | if 'TRUE', suppresses intermediate text output |
| filename_up | Name of the exported "up" file (if 'export=TRUE') |
| filename_down | Name of the exported "down" file (if 'export=TRUE') |
| filename_binom | Name of the exported binomial file |
| filename_sigdeg | Name of the exported sigDEG file |

## Value

A list containing two tables.

## Examples

```
sc <- DISCBIO(valuesG1msTest)
sc <- Clustexp(sc, cln=3, quiet=TRUE)
cdiff <- ClustDiffGenes(sc, K=3, fdr=.3, export=FALSE)
str(cdiff)
cdiff[[2]]
```

`Clustexp` *Clustering of single-cell transcriptome data*

## Description

This functions performs the initial clustering of the RaceID algorithm.

## Usage

```
Clustexp(
  object,
  clustnr = 3,
  bootnr = 50,
  metric = "pearson",
  do.gap = TRUE,
  SE.method = "Tibs2001SEmax",
  SE.factor = 0.25,
  B.gap = 50,
  cln = 0,
  rseed = NULL,
  quiet = FALSE
)

## S4 method for signature 'DISCBIO'
Clustexp(
  object,
  clustnr = 3,
  bootnr = 50,
  metric = "pearson",
  do.gap = TRUE,
  SE.method = "Tibs2001SEmax",
  SE.factor = 0.25,
  B.gap = 50,
  cln = 0,
  rseed = NULL,
  quiet = FALSE
)
```

## Arguments

| | |
|---|---|
| `object` | `DISCBIO` class object. |
| `clustnr` | Maximum number of clusters for the derivation of the cluster number by the saturation of mean within-cluster-dispersion. Default is 20. |
| `bootnr` | A numeric value of booststrapping runs for `clusterboot`. Default is 50. |
| `metric` | Is the method to transform the input data to a distance object. Metric has to be one of the following: ["spearman", "pearson", "kendall", "euclidean", "maximum", "manhattan", "canberra", "binary", "minkowski"]. |

| | |
|---|---|
| do.gap | A logical vector that allows generating the number of clusters based on the gap statistics. Default is TRUE. |
| SE.method | The SE.method determines the first local maximum of the gap statistics. The SE.method has to be one of the following:["firstSEmax", "Tibs2001SEmax", "globalSEmax", "firstmax", "globalmax"]. Default is "Tibs2001SEmax" |
| SE.factor | A numeric value of the fraction of the standard deviation by which the local maximum is required to differ from the neighboring points it is compared to. Default is 0.25. |
| B.gap | Number of bootstrap runs for the calculation of the gap statistics. Default is 50 |
| cln | Number of clusters to be used. Default is NULL and the cluster number is inferred by the saturation criterion. |
| rseed | Random integer to enforce reproducible clustering results. |
| quiet | if 'TRUE', intermediate output is suppressed |

## Value

The DISCBIO-class object input with the cpart slot filled.

## Examples

```
sc <- DISCBIO(valuesG1msTest) # changes signature of data
sc <- Clustexp(sc, cln=2)
```

---

clustheatmap *Plotting clusters in a heatmap representation of the cell distances*

---

## Description

This functions plots a heatmap of the distance matrix grouped by clusters. Individual clusters are highlighted with rainbow colors along the x and y-axes.

## Usage

```
clustheatmap(
  object,
  clustering_method = "k-means",
  hmethod = "single",
  rseed = NULL,
  quiet = FALSE,
  plot = TRUE
)

## S4 method for signature 'DISCBIO'
clustheatmap(
  object,
  clustering_method = "k-means",
```

```
  hmethod = "single",
  rseed = NULL,
  quiet = FALSE,
  plot = TRUE
)
```

## Arguments

| | |
|---|---|
| object | DISCBIO class object. |
| clustering_method | either "k-means" or "model-based" ("k" and "mb" are also accepted) |
| hmethod | Agglomeration method used for determining the cluster order from hierarchical clustering of the cluster medoids. This should be one of "ward.D", "ward.D2", "single", "complete", "average". Default is "single". |
| rseed | Random integer to fix random results. |
| quiet | if 'TRUE', intermediary output is suppressed |
| plot | if 'TRUE', plots the heatmap; otherwise, just prints cclmo |

## Value

Unless otherwise specified, a heatmap and a vector of the underlying cluster order.

---

# comptSNE *Computing tSNE*

---

## Description

This function is used to compute the t-Distributed Stochastic Neighbor Embedding (t-SNE).

## Usage

```
comptSNE(
  object,
  rseed = NULL,
  max_iter = 5000,
  epoch = 500,
  quiet = FALSE,
  ...
)

## S4 method for signature 'DISCBIO'
comptSNE(
  object,
  rseed = NULL,
  max_iter = 5000,
  epoch = 500,
```

```
  quiet = FALSE,
  ...
)
```

### Arguments

| | |
|---|---|
| object | DISCBIO class object. |
| rseed | Random integer to to yield reproducible maps across different runs |
| max_iter | maximum number of iterations to perform. |
| epoch | The number of iterations in between update messages. |
| quiet | if 'TRUE', suppresses intermediate output |
| ... | other parameters to be passed to 'tsne::tsne' |

### Value

The DISCBIO-class object input with the tsne slot filled.

### Examples

```
sc <- DISCBIO(valuesG1msTest) # changes signature of data
sc <- Clustexp(sc, cln=2) # data must be clustered before plottin
sc <- comptSNE(sc, max_iter=30)
head(sc@tsne)
```

---

## customConvertFeats *Automatic Feature Id Conversion.*

---

### Description

Attempt to automatically convert non-ENSEMBL feature identifiers to ENSEMBL identifiers. Features are included as rownames of the input data.frame (or matrix). It is assumed that feature identifiers (i.e., rownames of x) come from human or mouse genomes, and are either OFFICIAL SYMBOLS or ENTREZIDS. If less than 20 is identified, an error will be thrown.

### Usage

```
customConvertFeats(x, verbose = TRUE)
```

### Arguments

| | |
|---|---|
| x | data.frame or matrix including raw counts (typically, UMIs), where rows are features (genes) and rownames are feature identifiers (SYMBOLs or ENTREZIDs). |
| verbose | logical, shall messages be printed to inform about conversion advances. |

### Value

a data.frame where rownames are ENSEMBL IDs. The new feature IDs are automatically imputed based on existing feature IDs (SYMBOLs or ENTREZIDs).

DEGanalysis *Determining differentially expressed genes (DEGs) between all individual clusters.*

## Description

This function defines DEGs between all individual clusters generated by either K-means or model based clustering.

## Usage

```
DEGanalysis(
  object,
  K,
  Clustering = "K-means",
  fdr = 0.05,
  name = "Name",
  export = FALSE,
  quiet = FALSE,
  plot = TRUE,
  filename_deg = "DEGsTable",
  filename_sigdeg = "sigDEG",
  ...
)

## S4 method for signature 'DISCBIO'
DEGanalysis(
  object,
  K,
  Clustering = "K-means",
  fdr = 0.05,
  name = "Name",
  export = FALSE,
  quiet = FALSE,
  plot = TRUE,
  filename_deg = "DEGsTable",
  filename_sigdeg = "sigDEG",
  ...
)
```

## Arguments

| | |
|---|---|
| object | DISCBIO class object. |
| K | A numeric value of the number of clusters. |
| Clustering | Clustering has to be one of the following: ["K-means","MB"]. Default is "K-means" |

| | |
|---|---|
| fdr | A numeric value of the false discovery rate. Default is 0.05. |
| name | A string vector showing the name to be used to save the resulted tables. |
| export | A logical vector that allows writing the final gene list in excel file. Default is TRUE. |
| quiet | if 'TRUE', suppresses intermediate text output |
| plot | if 'TRUE', plots are generated |
| filename_deg | Name of the exported DEG table |
| filename_sigdeg | Name of the exported sigDEG table |
| ... | additional parameters to be passed to samr() |

## Value

A list containing two tables.

---

DEGanalysis2clust *Determining differentially expressed genes (DEGs) between two particular clusters.*

---

## Description

This function defines DEGs between particular clusters generated by either K-means or model based clustering.

## Usage

```
DEGanalysis2clust(
  object,
  K,
  Clustering = "K-means",
  fdr = 0.05,
  name = "Name",
  First = "CL1",
  Second = "CL2",
  export = FALSE,
  quiet = FALSE,
  plot = TRUE,
  filename_deg = "DEGsTable",
  filename_sigdeg = "sigDEG",
  ...
)

## S4 method for signature 'DISCBIO'
DEGanalysis2clust(
  object,
```

```
  K,
  Clustering = "K-means",
  fdr = 0.05,
  name = "Name",
  First = "CL1",
  Second = "CL2",
  export = FALSE,
  quiet = FALSE,
  plot = TRUE,
  filename_deg = "DEGsTable",
  filename_sigdeg = "sigDEG",
  ...
)
```

## Arguments

| | |
|---|---|
| object | DISCBIO class object. |
| K | A numeric value of the number of clusters. |
| Clustering | Clustering has to be one of the following: ["K-means","MB"]. Default is "K-means" |
| fdr | A numeric value of the false discovery rate. Default is 0.05. |
| name | A string vector showing the name to be used to save the resulted tables. |
| First | A string vector showing the first target cluster. Default is "CL1" |
| Second | A string vector showing the second target cluster. Default is "CL2" |
| export | A logical vector that allows writing the final gene list in excel file. Default is TRUE. |
| quiet | if 'TRUE', suppresses intermediate text output |
| plot | if 'TRUE', plots are generated |
| filename_deg | Name of the exported DEG table |
| filename_sigdeg | Name of the exported sigDEG table |
| ... | additional parameters to be passed to samr() |

## Value

A list containing two tables.

DISCBIO *The DISCBIO Class*

## Description

The DISCBIO class is the central object storing all information generated throughout the pipeline.

## Arguments

| | |
|---|---|
| `object` | An DISCBIO object. |

## Details

DISCBIO

## Slots

`SingleCellExperiment` Representation of the single cell input data, including both cells from regular and ERCC spike-in samples. Data are stored in a SingleCellExperiment object.

`expdata` The raw expression data matrix with cells as columns and genes as rows in sparse matrix format. It does not contain ERCC spike-ins.

`expdataAll` The raw expression data matrix with cells as columns and genes as rows in sparse matrix format. It can contain ERCC spike-ins.

`ndata` Data with expression normalized to one for each cell.

`fdata` Filtered data with expression normalized to one for each cell.

`distances` A distance matrix.

`tsne` A data.frame with coordinates of two-dimensional tsne layout for the K-means clustering.

`background` A list storing the polynomial fit for the background model of gene expression variability. It is used for outlier identification.

`out` A list storing information on outlier cells used for the prediction of rare cell types.

`cpart` A vector containing the final clustering partition computed by K-means.

`fcol` A vector contaning the colour scheme for the clusters.

`filterpar` A list containing the parameters used for cell and gene filtering based on expression.

`clusterpar` A list containing the parameters used for the K-means clustering.

`outlierpar` A list containing the parameters used for outlier identification.

`kmeans` A list containing the results of running the Clustexp() function.

`MBclusters` A vector containing the final clustering partition computed by Model-based clustering.

`kordering` A vector containing the Pseudo-time ordering based on k-means clusters.

`MBordering` A vector containing the Pseudo-time ordering based on Model-based clusters.

`MBtsne` A data.frame with coordinates of two-dimensional tsne layout for the Model-based clustering.

`noiseF` A vector containing the gene list resulted from running the noise filtering.

`FinalGeneList` A vector containing the final gene list resulted from running the noise filtering or/and the expression filtering.

## Examples

```
class(valuesG1msTest)
G1_reclassified <- DISCBIO(valuesG1msTest)
class(G1_reclassified)
str(G1_reclassified, max.level=2)
identical(G1_reclassified@expdataAll, valuesG1msTest)
```

---

# DISCBIO2SingleCellExperiment

*Convert a DISCBIO object to a SingleCellExperiment.*

---

## Description

Extract the SingleCellExperiment input data from the corresponding input slot in a DISCBIO-class object

## Usage

```
DISCBIO2SingleCellExperiment(x)
```

## Arguments

| | |
|---|---|
| x | an object of class DISCBIO |

## Value

a SingleCellExperiment-class object

## Examples

```
g1_disc <- DISCBIO(valuesG1msTest)
class(g1_disc)
g1_sce <- DISCBIO2SingleCellExperiment(g1_disc)
class(g1_sce)
```

Exprmclust *Performing Model-based clustering on expression values*

## Description

this function first uses principal component analysis (PCA) to reduce dimensionality of original data. It then performs model-based clustering on the transformed expression values.

## Usage

```
Exprmclust(
  object,
  K = 3,
  modelNames = "VVV",
  reduce = TRUE,
  cluster = NULL,
  quiet = FALSE
)

## S4 method for signature 'DISCBIO'
Exprmclust(
  object,
  K = 3,
  modelNames = "VVV",
  reduce = TRUE,
  cluster = NULL,
  quiet = FALSE
)

## S4 method for signature 'data.frame'
Exprmclust(
  object,
  K = 3,
  modelNames = "VVV",
  reduce = TRUE,
  cluster = NULL,
  quiet = FALSE
)
```

## Arguments

| | |
|---|---|
| object | DISCBIO class object. |
| K | An integer vector specifying all possible cluster numbers. Default is 3. |
| modelNames | model to be used in model-based clustering. By default "ellipsoidal, varying volume, shape, and orientation" is used. |

| | |
|---|---|
| reduce | A logical vector that allows performing the PCA on the expression data. Default is TRUE. |
| cluster | A vector showing the ID of cells in the clusters. |
| quiet | if 'TRUE', suppresses intermediary output |

## Value

If 'object' is of class DISCBIO, the output is the same object with the MBclusters slot filled. If the 'object' is a data frame, the function returns a named list containing the four objects that together correspond to the contents of the MBclusters slot.

---

`FinalPreprocessing` *Final Preprocessing*

---

## Description

This function generates the final filtered normalized dataset.

## Usage

```
FinalPreprocessing(
  object,
  GeneFlitering = "NoiseF",
  export = FALSE,
  quiet = FALSE,
  fileName = "filteredDataset"
)

## S4 method for signature 'DISCBIO'
FinalPreprocessing(
  object,
  GeneFlitering = "NoiseF",
  export = FALSE,
  quiet = FALSE,
  fileName = "filteredDataset"
)
```

## Arguments

| | |
|---|---|
| object | DISCBIO class object. |
| GeneFlitering | GeneFlitering has to be one of the followings: ["NoiseF","ExpF"]. Default is "NoiseF" |
| export | A logical vector that allows writing the final gene list in excel file. Default is TRUE. |
| quiet | if 'TRUE', intermediary output is suppressed |
| fileName | File name for exporting (if 'export = TRUE') |

## Value

The DISCBIO-class object input with the FinalGeneList slot filled.

## Examples

```
sc <- DISCBIO(valuesG1msTest)
sc <- NoiseFiltering(sc, percentile=0.9, CV=0.2, export=FALSE)
sc <- FinalPreprocessing(sc, GeneFlitering="NoiseF", export=FALSE)
```

---

# FindOutliers *Inference of outlier cells*

---

## Description

This functions performs the outlier identification for k-means and model-based clustering

## Usage

```
FindOutliers(
  object,
  K,
  outminc = 5,
  outlg = 2,
  probthr = 0.001,
  thr = 2^-(1:40),
  outdistquant = 0.75,
  plot = TRUE,
  quiet = FALSE
)

## S4 method for signature 'DISCBIO'
FindOutliers(
  object,
  K,
  outminc = 5,
  outlg = 2,
  probthr = 0.001,
  thr = 2^-(1:40),
  outdistquant = 0.75,
  plot = TRUE,
  quiet = FALSE
)
```

### Arguments

| | |
|---|---|
| object | DISCBIO class object. |
| K | Number of clusters to be used. |
| outminc | minimal transcript count of a gene in a clusters to be tested for being an outlier gene. Default is 5. |
| outlg | Minimum number of outlier genes required for being an outlier cell. Default is 2. |
| probthr | outlier probability threshold for a minimum of outlg genes to be an outlier cell. This probability is computed from a negative binomial background model of expression in a cluster. Default is 0.001. |
| thr | probability values for which the number of outliers is computed in order to plot the dependence of the number of outliers on the probability threshold. Default is 2**-(1:40).set |
| outdistquant | Real number between zero and one. Outlier cells are merged to outlier clusters if their distance smaller than the outdistquant-quantile of the distance distribution of pairs of cells in the orginal clusters after outlier removal. Default is 0.75. |
| plot | if 'TRUE', produces a plot of -log10prob per K |
| quiet | if 'TRUE', intermediary output is suppressed |

### Value

A named vector of the genes containing outlying cells and the number of cells on each.

### Examples

```
sc <- DISCBIO(valuesG1msTest)
sc <- Clustexp(sc, cln=2) # K-means clustering
FindOutliers(sc, K=2)
```

---

## foldchange.seq.twoclass.unpaired

*Foldchange of twoclass unpaired sequencing data*

---

### Description

Foldchange of twoclass unpaired sequencing data

### Usage

```
foldchange.seq.twoclass.unpaired(x, y, depth)
```

### Arguments

| | |
|---|---|
| x | x |
| y | y |
| depth | depth |

---

## HumanMouseGeneIds *Human and Mouse Gene Identifiers.*

---

### Description

Data.frame including ENTREZID, SYMBOL, and ENSEMBL gene identifiers of human and mouse genes.

### Source

Data were imported, modified, and formatted from the Mus.musculus (ver 1.3.1) and the Homo.sapiens (ver 1.3.1) BioConductor libraries.

---

## J48DT *J48 Decision Tree*

---

### Description

The decision tree analysis is implemented over a training dataset, which consisted of the DEGs obtained by either SAMseq or the binomial differential expression.

### Usage

```
J48DT(data, quiet = FALSE, plot = TRUE)
```

### Arguments

| | |
|---|---|
| data | A data frame resulted from running the function ClassVectoringDT. |
| quiet | If 'TRUE', suppresses intermediary output |
| plot | If 'FALSE', suppresses plot output |

### Value

Information about the J48 model and, by default, a plot of the decision tree.

## J48DTeval *Evaluating the performance of the J48 decision tree.*

### Description

This function evaluates the performance of the generated trees for error estimation by ten-fold cross validation assessment.

### Usage

```
J48DTeval(data, num.folds = 10, First = "CL1", Second = "CL2", quiet = FALSE)
```

### Arguments

| | |
|---|---|
| data | The resulted data from running the function J48DT. |
| num.folds | A numeric value of the number of folds for the cross validation assessment. Default is 10. |
| First | A string vector showing the first target cluster. Default is "CL1" |
| Second | A string vector showing the second target cluster. Default is "CL2" |
| quiet | If 'TRUE', suppresses intermediary output |

### Value

Statistics about the J48 model

## Jaccard *Jaccard's similarity*

### Description

Robustness of the clusters can be assessed by Jaccard's similarity, which reflects the reproducibility of individual clusters across bootstrapping runs. Jaccard's similarity is the intersect of two clusters divided by the union.

### Usage

```
Jaccard(object, Clustering = "K-means", K, plot = TRUE, R = 100, ...)
```

### Arguments

| | |
|---|---|
| object | DISCBIO class object. |
| Clustering | Clustering has to be one of the following: ["K-means","MB"]. Default is "K-means" |
| K | A numeric value of the number of clusters |
| plot | if 'TRUE', plots the mean Jaccard similarities |
| R | number of bootstrap replicates |
| ... | Further arguments passed to boot::boot |

### Value

A plot of the mean Jaccard similarity coefficient per cluster.

---

## KmeanOrder *Pseudo-time ordering based on k-means clusters*

---

### Description

This function takes the exact output of exprmclust function and construct Pseudo-time ordering by mapping all cells onto the path that connects cluster centers.

### Usage

```
KmeanOrder(
  object,
  quiet = FALSE,
  export = FALSE,
  filename = "Cellular_pseudo-time_ordering_based_on_k-meansc-lusters"
)

## S4 method for signature 'DISCBIO'
KmeanOrder(
  object,
  quiet = FALSE,
  export = FALSE,
  filename = "Cellular_pseudo-time_ordering_based_on_k-meansc-lusters"
)
```

### Arguments

| | |
|---|---|
| object | DISCBIO class object. |
| quiet | if 'TRUE', suppresses intermediary output |
| export | if 'TRUE', exports order table to csv |
| filename | Name of the exported file (if 'export=TRUE') |

### Value

The DISCBIO-class object input with the kordering slot filled.

### Note

This function has been replaced by pseudoTimeOrdering(), but it is being kept for legacy purposes. It will, however, be removed from future versions of DIscBIO.

---

## NetAnalysis *Networking analysis.*

---

### Description

This function checks the connectivity degree and the betweenness centrality, which reflect the communication flow in the defined PPI networks

### Usage

```
NetAnalysis(data, export = FALSE, FileName = "NetworkAnalysisTable-1")
```

### Arguments

| | |
|---|---|
| data | Protein-protein interaction data frame resulted from running the PPI function. |
| export | if 'TRUE', exports the analysis table as a csv file |
| FileName | suffix for the file name (if export = TRUE) |

### Value

A network analysis table

---

## Networking *Plotting the network.*

---

### Description

This function uses STRING-api to plot the network.

### Usage

```
Networking(
  data,
  FileName = NULL,
  species = "9606",
  plot_width = 25,
  plot_height = 15
)
```

### Arguments

| | |
|---|---|
| data | A gene list. |
| FileName | A string vector showing the name to be used to save the resulted network. If 'NULL', the network will be saved to a temporary directory |
| species | The taxonomy name/id. Default is "9606" for Homo sapiens. |
| plot_width | Plot width |
| plot_height | Plot height |

### Value

A plot of the network

---

NoiseFiltering *Noise Filtering*

---

### Description

Given a matrix or data frame of count data, this function estimates the size factors as follows: Each column is divided by the geometric means of the rows. The median (or, if requested, another location estimator) of these ratios (skipping the genes with a # geometric mean of zero) is used as the size factor for this column. Source: DESeq package.

### Usage

```
NoiseFiltering(
  object,
  percentile = 0.8,
  CV = 0.3,
  geneCol = "yellow",
  FgeneCol = "black",
  erccCol = "blue",
  Val = TRUE,
  plot = TRUE,
  export = FALSE,
  quiet = FALSE,
  filename = "Noise_filtering_genes_test"
)

## S4 method for signature 'DISCBIO'
NoiseFiltering(
  object,
  percentile = 0.8,
  CV = 0.3,
  geneCol = "yellow",
  FgeneCol = "black",
```

```
  erccCol = "blue",
  Val = TRUE,
  plot = TRUE,
  export = FALSE,
  quiet = FALSE,
  filename = "Noise_filtering_genes_test"
)
```

## Arguments

| | |
|---|---|
| object | DISCBIO class object. |
| percentile | A numeric value of the percentile. It is used to validate the ERCC spik-ins. Default is 0.8. |
| CV | A numeric value of the coefficient of variation. It is used to validate the ERCC spik-ins. Default is 0.5. |
| geneCol | Color of the genes that did not pass the filtration. |
| FgeneCol | Color of the genes that passt the filtration. |
| erccCol | Color of the ERCC spik-ins. |
| Val | A logical vector that allows plotting only the validated ERCC spike-ins. Default is TRUE. If Val=FALSE will plot all the ERCC spike-ins. |
| plot | A logical vector that allows plotting the technical noise. Default is TRUE. |
| export | A logical vector that allows writing the final gene list in excel file. Default is TRUE. |
| quiet | if 'TRUE', suppresses printed output |
| filename | Name of the exported file (if 'export=TRUE') |

## Value

The DISCBIO-class object input with the noiseF slot filled.

## Note

This function should be used only if the dataset has ERCC.

## Examples

```
sc <- DISCBIO(valuesG1msTest) # changes signature of data
sd_filtered <- NoiseFiltering(sc, export=FALSE)
str(sd_filtered)
```

`Normalizedata` *Normalizing and filtering*

## Description

This function allows filtering of genes and cells to be used in the downstream analysis.

## Usage

```
Normalizedata(
  object,
  mintotal = 1000,
  minexpr = 0,
  minnumber = 0,
  maxexpr = Inf,
  downsample = FALSE,
  dsn = 1,
  rseed = NULL
)

## S4 method for signature 'DISCBIO'
Normalizedata(
  object,
  mintotal = 1000,
  minexpr = 0,
  minnumber = 0,
  maxexpr = Inf,
  downsample = FALSE,
  dsn = 1,
  rseed = NULL
)
```

## Arguments

| | |
|---|---|
| `object` | `DISCBIO` class object. |
| `mintotal` | minimum total transcript number required. Cells with less than `mintotal` transcripts are filtered out. Default is 1000. |
| `minexpr` | minimum required transcript count of a gene in at least `minnumber` cells. All other genes are filtered out. Default is 0. |
| `minnumber` | minimum number of cells that are expressing each gene at minexpr transcripts. Default is 0. |
| `maxexpr` | maximum allowed transcript count of a gene in at least a single cell after normalization or downsampling. All other genes are filtered out. Default is Inf. |
| `downsample` | A logical vector. Default is FALSE. If downsample is set to TRUE, then transcript counts are downsampled to mintotal transcripts per cell, instead of the |

| | |
|---|---|
| | normalization. Downsampled versions of the transcript count data are averaged across dsn samples |
| dsn | A numeric value of the number of samples to be used to average the downsampled versions of the transcript count data. Default is 1 which means that sampling noise should be comparable across cells. For high numbers of dsn the data will become similar to the median normalization. |
| rseed | Random integer to enforce reproducible clustering. results |

## Value

The DISCBIO-class object input with the ndata and fdata slots filled.

## Examples

```
sc <- DISCBIO(valuesG1msTest) # changes signature of data

# In this case this function is used to normalize the reads
sc_normal <- Normalizedata(
    sc, mintotal=1000, minexpr=0, minnumber=0, maxexpr=Inf, downsample=FALSE,
    dsn=1, rseed=17000
)
summary(sc_normal@fdata)
```

---

# PCAplotSymbols *Plot PCA symbols*

---

## Description

Generates a plot of grouped PCA components

## Usage

```
PCAplotSymbols(object, types = NULL)

## S4 method for signature 'DISCBIO'
PCAplotSymbols(object, types = NULL)
```

## Arguments

| | |
|---|---|
| object | DISCBIO class object. |
| types | If types=NULL then the names of the cells will be grouped automatically. Default is NULL |

## Value

Plot of the Principal Components

## plotExptSNE *Highlighting gene expression in the t-SNE map*

### Description

The t-SNE map representation can also be used to analyze expression of a gene or a group of genes, to investigate cluster specific gene expression patterns

### Usage

```
plotExptSNE(object, g, n = NULL)

## S4 method for signature 'DISCBIO'
plotExptSNE(object, g, n = NULL)
```

### Arguments

| | |
|---|---|
| object | DISCBIO class object. |
| g | Individual gene name or vector with a group of gene names corresponding to a subset of valid row names of the ndata slot of the DISCBIO object. |
| n | String of characters representing the title of the plot. Default is NULL and the first element of g is chosen. |

### Value

t-SNE plot for one particular gene

## plotGap *Plotting Gap Statistics*

### Description

Plotting Gap Statistics

### Usage

```
plotGap(object, y_limits = NULL)

## S4 method for signature 'DISCBIO'
plotGap(object, y_limits = NULL)
```

### Arguments

| | |
|---|---|
| object | DISCBIO class object. |
| y_limits | 2-length numeric vector with the limits for the gap plot |

### Value

A plot of the gap statistics

---

`plotLabelstSNE` *tSNE map with labels*

---

### Description

Visualizing k-means or model-based clusters using tSNE maps

### Usage

```
plotLabelstSNE(object)

## S4 method for signature 'DISCBIO'
plotLabelstSNE(object)
```

### Arguments

| | |
|---|---|
| `object` | `DISCBIO` class object. |

### Value

Plot containing the ID of the cells in each cluster

---

`PlotMBpca` *Plotting pseudo-time ordering or gene expression in Model-based clustering in PCA*

---

### Description

The PCA representation can either be used to show pseudo-time ordering or the gene expression of a particular gene.

### Usage

```
PlotMBpca(object, type = "order", g = NULL, n = NULL)
```

### Arguments

| | |
|---|---|
| `object` | `DISCBIO` class object. |
| `type` | either ‘order‘ to plot pseudo-time ordering or ‘exp‘ to plot gene expression |
| `g` | Individual gene name or vector with a group of gene names corresponding to a subset of valid row names of the `ndata` slot of the `DISCBIO` object. Ignored if ‘type="order"‘. |
| `n` | String of characters representing the title of the plot. Default is NULL and the first element of g is chosen. Ignored if ‘type="order"‘. |

## Value

A plot of the PCA.

---

`PlotmclustMB` *Plotting the Model-based clusters in PCA.*

---

## Description

Plot the model-based clustering results

## Usage

```
PlotmclustMB(object)

## S4 method for signature 'DISCBIO'
PlotmclustMB(object)
```

## Arguments

| | |
|---|---|
| `object` | `DISCBIO` class object. |

## Value

A plot of the PCA.

---

`plotOrderTsne` *Plotting the pseudo-time ordering in the t-SNE map*

---

## Description

The tSNE representation can also be used to show the pseudo-time ordering.

## Usage

```
plotOrderTsne(object)

## S4 method for signature 'DISCBIO'
plotOrderTsne(object)
```

## Arguments

| | |
|---|---|
| `object` | `DISCBIO` class object. |

## Value

A plot of the pseudo-time ordering.

## plotSilhouette *Silhouette Plot for K-means clustering*

### Description

The silhouette provides a representation of how well each point is represented by its cluster in comparison to the closest neighboring cluster. It computes for each point the difference between the average similarity to all points in the same cluster and to all points in the closest neighboring cluster. This difference it normalize such that it can take values between -1 and 1 with higher values reflecting better representation of a point by its cluster.

### Usage

```
plotSilhouette(object, K)

## S4 method for signature 'DISCBIO'
plotSilhouette(object, K)
```

### Arguments

| | |
|---|---|
| object | DISCBIO class object. |
| K | A numeric value of the number of clusters |

### Value

A silhouette plot

## plotSymbolstSNE *tSNE map for K-means clustering with symbols*

### Description

Visualizing the K-means clusters using tSNE maps

### Usage

```
plotSymbolstSNE(object, types = NULL, legloc = "bottomright")

## S4 method for signature 'DISCBIO'
plotSymbolstSNE(object, types = NULL, legloc = "bottomright")
```

### Arguments

| | |
|---|---|
| object | DISCBIO class object. |
| types | If types=NULL then the names of the cells will be grouped automatically. Default is NULL |
| legloc | A keyword from the list "bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright", "right" and "center". Default is "bottomright" |

### Value

Plot of tsne objet slot, grouped by gene.

---

## plottSNE *tSNE map*

---

### Description

Visualizing the k-means or model-based clusters using tSNE maps

### Usage

```
plottSNE(object)

## S4 method for signature 'DISCBIO'
plottSNE(object)
```

### Arguments

| | |
|---|---|
| object | DISCBIO class object. |

### Value

A plot of t-SNEs.

---

## PPI *Defining protein-protein interactions (PPI) over a list of genes,*

---

### Description

This function uses STRING-api. The outcome of STRING analysis will be stored in tab separated values (TSV) files.

### Usage

```
PPI(data, FileName = NULL, species = "9606")
```

### Arguments

| | |
|---|---|
| data | A gene list. |
| FileName | A string vector showing the name to be used to save the resulted table. If null, no file will be exported |
| species | The taxonomy name/id. Default is "9606" for Homo sapiens. |

### Value

Either a TSV file stored in the user's file system and its corresponding 'data.frame' object in R or and R object containing that information.

---

## prepExampleDataset *Prepare Example Dataset*

---

### Description

Internal function that prepares a pre-treated dataset for use in several examples

### Usage

```
prepExampleDataset(dataset, save = TRUE)
```

### Arguments

| | |
|---|---|
| dataset | Dataset used for transformation |
| save | save results? |

### Details

This function serves the purpose of treating datasets such as valuesG1msReduced to reduce examples of other functions by bypassing some analysis steps covered in the vignettes.

### Value

Two rda files, ones for K-means clustering and another for Model-based clustering.

### Author(s)

Waldir Leoncio

## pseudoTimeOrdering *Pseudo-time ordering*

### Description

This function takes the exact output of exprmclust function and construct Pseudo-time ordering by mapping all cells onto the path that connects cluster centers.

### Usage

```
pseudoTimeOrdering(
  object,
  quiet = FALSE,
  export = FALSE,
  filename = "Cellular_pseudo-time_ordering"
)

## S4 method for signature 'DISCBIO'
pseudoTimeOrdering(
  object,
  quiet = FALSE,
  export = FALSE,
  filename = "Cellular_pseudo-time_ordering"
)
```

### Arguments

| | |
|---|---|
| object | DISCBIO class object. |
| quiet | if 'TRUE', suppresses intermediary output |
| export | if 'TRUE', exports order table to csv |
| filename | Name of the exported file (if 'export=TRUE') |

### Value

The DISCBIO-class object input with the kordering slot filled.

## rankcols *Rank columns*

### Description

Ranks the elements within each col of the matrix x and returns these ranks in a matrix

### Usage

```
rankcols(x)
```

### Arguments

| | |
|---|---|
| x | x |

### Note

this function is equivalent to ‘samr::rankcol‘, but uses ‘apply‘ to rank the colums instead of a compiled Fortran function which was causing our DEGanalysis functions to freeze in large datasets.

---

## reformatSiggenes *Reformat Siggenes Table*

---

### Description

Reformats the Siggenes table output from the SAMR package

### Usage

```
reformatSiggenes(table)
```

### Arguments

| | |
|---|---|
| table | output from ‘samr::samr.compute.siggenes.table‘ |

### Author(s)

Waldir Leoncio

### See Also

replaceDecimals

---

## replaceDecimals *Replace Decimals*

---

### Description

Replaces decimals separators between comma and periods on a character vector

### Usage

```
replaceDecimals(x, from = ",", to = ".")
```

### Arguments

| | |
|---|---|
| x | vector of characters |
| from | decimal separator on input file |
| to | decimal separator for output file |

### Note

This function was especially designed to be used with retormatSiggenes

### See Also

reformatSiggenes

---

## resa *Resampling*

---

### Description

Corresponds to ‘samr::resample‘

### Usage

```
resa(x, d, nresamp = 20)
```

### Arguments

| | |
|---|---|
| x | data matrix. nrow=#gene, ncol=#sample |
| d | estimated sequencing depth |
| nresamp | number of resamplings |

### Value

xresamp: an rank array with dim #gene*#sample*nresamp

---

## RpartDT *RPART Decision Tree*

---

### Description

The decision tree analysis is implemented over a training dataset, which consisted of the DEGs obtained by either SAMseq or the binomial differential expression.

### Usage

```
RpartDT(data, quiet = FALSE, plot = TRUE)
```

### Arguments

| | |
|---|---|
| data | The exact output of the exprmclust function. |
| quiet | If ‘TRUE‘, suppresses intermediary output |
| plot | If ‘FALSE‘, suppresses plot output |

## Value

Information about the model and, by default, a plot of the decision tree.

---

`RpartEVAL` *Evaluating the performance of the RPART Decision Tree.*

---

## Description

This function evaluates the performance of the generated trees for error estimation by ten-fold cross validation assessment.

## Usage

```
RpartEVAL(data, num.folds = 10, First = "CL1", Second = "CL2", quiet = FALSE)
```

## Arguments

| | |
|---|---|
| `data` | The resulted data from running the function J48DT. |
| `num.folds` | A numeric value of the number of folds for the cross validation assessment. Default is 10. |
| `First` | A string vector showing the first target cluster. Default is "CL1" |
| `Second` | A string vector showing the second target cluster. Default is "CL2" |
| `quiet` | If ‘TRUE‘, suppresses intermediary output |

## Value

Performance statistics of the model

---

`sammy` *Significance analysis of microarrays*

---

## Description

This function is an adaptation of ‘samr::samr‘

## Usage

```
sammy(
  data,
  resp.type = c("Quantitative", "Two class unpaired", "Survival", "Multiclass",
    "One class", "Two class paired", "Two class unpaired timecourse",
    "One class timecourse", "Two class paired timecourse", "Pattern discovery"),
  assay.type = c("array", "seq"),
  s0 = NULL,
  s0.perc = NULL,
  nperms = 100,
  center.arrays = FALSE,
  testStatistic = c("standard", "wilcoxon"),
  time.summary.type = c("slope", "signed.area"),
  regression.method = c("standard", "ranks"),
  return.x = FALSE,
  knn.neighbors = 10,
  random.seed = NULL,
  nresamp = 20,
  nresamp.perm = NULL,
  xl.mode = c("regular", "firsttime", "next20", "lasttime"),
  xl.time = NULL,
  xl.prevfit = NULL
)
```

## Arguments

| | |
|---|---|
| data | Data object with components x- p by n matrix of features, one observation per column (missing values allowed); y- n-vector of outcome measurements; censoring.status- n-vector of censoring censoring.status (1= died or event occurred, 0=survived, or event was censored), needed for a censored survival outcome |
| resp.type | Problem type: "Quantitative" for a continuous parameter (Available for both array and sequencing data); "Two class unpaired" (for both array and sequencing data); "Survival" for censored survival outcome (for both array and sequencing-data); "Multiclass": more than 2 groups (for both array and sequencing data); "One class" for a single group (only for array data); "Two class paired" for two classes with paired observations (for both array and sequencing data); "Two class unpaired timecourse" (only for array data), "One class time course" (only for array data), "Two class.paired timecourse" (only for array data), or "Pattern discovery" (only for array data) |
| assay.type | Assay type: "array" for microarray data, "seq" for counts from sequencing |
| s0 | Exchangeability factor for denominator of test statistic; Default is automatic choice. Only used for array data. |
| s0.perc | Percentile of standard deviation values to use for s0; default is automatic choice; -1 means s0=0 (different from s0.perc=0, meaning s0=zeroeth percentile of standard deviation values= min of sd values. Only used for array data. |
| nperms | Number of permutations used to estimate false discovery rates |

| | |
|---|---|
| center.arrays | Should the data for each sample (array) be median centered at the outset? Default =FALSE. Only used for array data. |
| testStatistic | Test statistic to use in two class unpaired case.Either "standard" (t-statistic) or ,"wilcoxon" (Two-sample wilcoxon or Mann-Whitney test). Only used for array data. |
| time.summary.type | Summary measure for each time course: "slope", or"signed.area"). Only used for array data. |
| regression.method | Regression method for quantitative case: "standard",(linear least squares) or "ranks" (linear least squares on ranked data). Only used for array data. |
| return.x | Should the matrix of feature values be returned? Only useful for time course data, where x contains summaries of the features over time. Otherwise x is the same as the input data data\$x |
| knn.neighbors | Number of nearest neighbors to use for imputation of missing features values. Only used for array data. |
| random.seed | Optional initial seed for random number generator (integer) |
| nresamp | For assay.type="seq", number of resamples used to construct test statistic. Default 20. Only used for sequencing data. |
| nresamp.perm | For assay.type="seq", number of resamples used to construct test statistic for permutations. Default is equal to nresamp and it must be at most nresamp. Only used for sequencing data. |
| xl.mode | Used by Excel interface |
| xl.time | Used by Excel interface |
| xl.prevfit | Used by Excel interface |

---

## samr.estimate.depth *Estimate sequencing depths*

---

### Description

Estimate sequencing depths

### Usage

```
samr.estimate.depth(x)
```

### Arguments

| | |
|---|---|
| x | data matrix. nrow=#gene, ncol=#sample |

### Value

depth: estimated sequencing depth. a vector with len sample.

## valuesG1msTest *Single-cells data from a myxoid liposarcoma cell line*

### Description

A sample of single cells from a myxoid liposarcoma cell line. Columns refer to samples and rows refer to genes. The last rows refer to external RNA controls consortium (ERCC) spike-ins. This dataset is part of a larger dataset containing 94 single cells. The complete dataset is fully compatible with this package and an rda file can be obtained at https://github.com/ocbe-uio/DIscBIO/blob/dev/data/valuesG1ms.rda

## VolcanoPlot *Volcano Plot*

### Description

Plotting differentially expressed genes (DEGs) in a particular cluster. Volcano plots are used to readily show the DEGs by plotting significance versus fold-change on the y and x axes, respectively.

### Usage

```
VolcanoPlot(object, value = 0.05, name = NULL, fc = 0.5, FS = 0.4)
```

### Arguments

| | |
|---|---|
| object | A data frame showing the differentially expressed genes (DEGs) in a particular cluster |
| value | A numeric value of the false discovery rate. Default is 0.05.. Default is 0.05 |
| name | A string vector showing the name to be used on the plot title |
| fc | A numeric value of the fold change. Default is 0.5. |
| FS | A numeric value of the font size. Default is 0.4. |

### Value

A volcano plot

`wilcoxon.unpaired.seq.func` *Twoclass Wilcoxon statistics*

## Description

Twoclass Wilcoxon statistics

## Usage

```
wilcoxon.unpaired.seq.func(xresamp, y)
```

## Arguments

| | |
|---|---|
| `xresamp` | an rank array with dim #gene*#sample*nresamp |
| `y` | outcome vector of values 1 and 2 |

## Value

the statistic.

# Index