Package 'systemPipeTools'

November 12, 2026

Title Tools for data visualization

Version 1.20.0

Date 2021-05-11

Description systemPipeTools package extends the widely used systemPipeR (SPR) workflow environment with an enhanced toolkit for data visualization, including utilities to automate the data visualizaton for analysis of differentially expressed genes (DEGs). systemPipeTools provides data transformation and data exploration functions via scatterplots, hierarchical clustering heatMaps, principal component analysis, multidimensional scaling, generalized principal components, t-Distributed Stochastic Neighbor embedding (t-SNE), and MA and volcano plots. All these utilities can be integrated with the modular design of the systemPipeR environment that allows users to easily substitute any of these features and/or custom with alternatives.

Imports DESeq2, GGally, Rtsne, SummarizedExperiment, ape, dplyr, ggplot2, ggrepel, ggtree, glmpca, pheatmap, plotly, tibble, magrittr, DT, stats

Suggests systemPipeR, knitr, BiocStyle, rmarkdown, testthat (>= 3.0.0), BiocGenerics, Biostrings, methods

biocViews Infrastructure, DataImport, Sequencing, QualityControl, ReportWriting, ExperimentalDesign, Clustering, DifferentialExpression, MultidimensionalScaling, PrincipalComponent

License Artistic-2.0

Encoding UTF-8

LazyData false

VignetteBuilder knitr

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Config/testthat/edition 3

git_url https://git.bioconductor.org/packages/systemPipeTools

git_branch RELEASE_3_23

git_last_commit 0c99180

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Daniela Cassol [aut, cre],
Ponmathi Ramasamy [aut],
Le Zhang [aut],
Thomas Girke [aut]

Maintainer Daniela Cassol <danicassol@gmail.com>

Contents

exploreDDS 2
exploreDDSplot 4
GLMplot 5
hclustplot 6
heatMaplot 8
MAplot 9
MDSplot 11
PCAplot 12
showDT 13
tSNEplot 13
volcanoplot 14

Index **17**

`exploreDDS` *exploreDDS*

Description

Convenience wrapper function to transform raw read counts using the `DESeq2::DESeq2-package()` package transformations methods. The input file has to contain all the genes, not just differentially expressed ones.

Usage

```
exploreDDS(
  countMatrix,
  targets,
  cmp = cmp[[1]],
  preFilter = NULL,
  transformationMethod = "raw",
  blind = TRUE
)
```

Arguments

countMatrix	date.frame or matrix containing raw read counts.
targets	targets data.frame.
cmp	character matrix where comparisons are defined in two columns. This matrix should be generated with the systemPipeR::readComp() function from the targets file. Values used for comparisons need to match those in the Factor column of the targets file.
preFilter	allows removing rows in which there are very few reads. Accepts a numeric value with the minimum of total reads to keep. Default is NULL.
transformationMethod	a character string indicating which transformation method it will be used on the raw read counts. Supported methods include rlog and vst using the DESeq2 package or default raw for no data transformation.
blind	logical, whether to blind the transformation to the experimental design (see varianceStabilizingTransformation), from DESeq2::vst() or DESeq2::rlog().

Details

Note that the recommendation is to use the resulting transformed values in the transformationMethod argument only for visualization and clustering, not for differential expression analysis which needs raw counts. Users are strongly encouraged to consult the DESeq2::DESeq2-package() vignette for more detailed information on this topic and how to properly run DESeq2 on data sets with more complex experimental designs.

Value

returns an object of class DESeq2::DESeqTransform().

Author(s)

Daniela Cassol

References

For more details on DESeq2, please consult the following page: DESeq2. For more details on targets file definition, please consult the following page: systemPipeR.

Examples

```
## Targets file
targetspath <- system.file("extdata", "targets.txt", package = "systemPipeR")
targets <- read.delim(targetspath, comment = "#")
cmp <- systemPipeR::readComp(file = targetspath, format = "matrix",
delim = "-")
## Count table file
countMatrixPath <- system.file("extdata", "countDFeByg.xls",
package = "systemPipeR")
countMatrix <- read.delim(countMatrixPath, row.names = 1)
## Run
```

```
exploredds <- exploreDDS(countMatrix, targets,
    cmp = cmp[[1]],
    preFilter = NULL, transformationMethod = "raw"
)
exploredds
```

---

exploreDDSplot *exploreDDSplot*

Description

Scatterplot of transformed counts from two samples or grid of all samples

Usage

```
exploreDDSplot(
  countMatrix,
  targets,
  cmp = cmp[[1]],
  preFilter = NULL,
  samples,
  blind = TRUE,
  scattermatrix = FALSE,
  plotly = FALSE,
  savePlot = FALSE,
  filePlot = NULL
)
```

Arguments

countMatrix	date.frame or matrix containing raw read counts
targets	targets data.frame
cmp	character matrix where comparisons are defined in two columns. This matrix should be generated with the systemPipeR::readComp() function from the targets file. Values used for comparisons need to match those in the Factor column of the targets file.
preFilter	allows removing rows in which there are very few reads. Accepts a numeric value with the minimum of total reads to keep. Default is NULL.
samples	a character vector of two samples or ALL samples in the dataset. Could be specified the SampleName column name of the targets file or the respective numeric values. Also, if set as ALL, a correlation matrix it will be plot.
blind	logical, whether to blind the transformation to the experimental design (see varianceStabilizingTransformation), from DESeq2::vst() or DESeq2::rlog().
scattermatrix	if samples set as ALL, requires to assign TRUE to build a correlation matrix and plot the correlogram of all the samples.

plotly	logical: when FALSE (default), the ggplot2 plot will be returned. TRUE returns the plotly version of the plot.
savePlot	logical: when FALSE (default), the plot will not be saved. If TRUE the plot will be saved, and requires the filePlot argument.
filePlot	file name where the plot will be saved. For more information, please consult the ggplot2::ggsave() function.

Value

returns an object of ggplot2 plot.

Examples

```
## Targets file
targetspath <- system.file("extdata", "targets.txt", package = "systemPipeR")
targets <- read.delim(targetspath, comment = "#")
cmp <- systemPipeR::readComp(
    file = targetspath,
    format = "matrix", delim = "-"
)
## Count table file
countMatrixPath <- system.file("extdata", "countDFeByg.xls",
    package = "systemPipeR"
)
countMatrix <- read.delim(countMatrixPath, row.names = 1)
## Plot
exploreDDSplot(countMatrix, targets,
    cmp = cmp[[1]], preFilter = NULL,
    samples = c(3, 4)
)
```

GLMplot *Dimension Reduction with GLMplot*

Description

This function computes and plots generalized principal components analysis for dimension reduction of count expression matrix.

Usage

```
GLMplot(
  exploredds,
  L = 2,
  plotly = FALSE,
  savePlot = FALSE,
  filePlot = NULL,
  ...
)
```

Arguments

exploredds	object of class DESeq2::DESeqDataSet(), generated from exploreDDS function.
L	desired number of latent dimensions (positive integer).
plotly	logical: when FALSE (default), the ggplot2 plot will be returned. TRUE option returns the plotly version of the plot.
savePlot	logical: when FALSE (default), the plot will not be saved. If TRUE the plot will be saved, and requires the filePlot argument.
filePlot	file name where the plot will be saved. For more information, please consult the ggplot2::ggsave() function.
...	additional parameters for the glmpca::glmpca() function.

Value

returns an object of ggplot or plotly class.

References

F. William Townes and Kelly Street (2020). glmpca: Dimension Reduction of Non-Normally Distributed Data. R package version 0.2.0. https://CRAN.R-project.org/package=glmpca

Examples

```
## Targets file
targetspath <- system.file("extdata", "targets.txt", package = "systemPipeR")
targets <- read.delim(targetspath, comment = "#")
cmp <- systemPipeR::readComp(file = targetspath, format = "matrix",
delim = "-")
## Count table file
countMatrixPath <- system.file("extdata", "countDFeByg.xls",
package = "systemPipeR")
countMatrix <- read.delim(countMatrixPath, row.names = 1)
## Plot
exploredds <- exploreDDS(countMatrix, targets, cmp = cmp[[1]],
preFilter = NULL, transformationMethod = "raw")
GLMplot(exploredds, plotly = FALSE)
```

---

hclustplot *Hierarchical Clustering Dendrogram (hclustplot)*

Description

This function computes the sample-wise correlation coefficients using the stats::cor() function from the transformed expression values. After transformation to a distance matrix, hierarchical clustering is performed with the stats::hclust() function, and the result is plotted as a dendrogram.

Usage

```
hclustplot(
  exploredds,
  method = "spearman",
  plotly = FALSE,
  savePlot = FALSE,
  filePlot = NULL
)
```

Arguments

exploredds	object of class DESeq2::DESeqTransform().
method	a character string indicating which correlation coefficient is to be computed, based on the stats::cor() function. Options are: c("pearson" "kendall", "spearman").
plotly	logical: when FALSE (default), the ggplot2 plot will be returned. TRUE option returns the plotly version of the plot.
savePlot	logical: when FALSE (default), the plot will not be saved. If TRUE the plot will be saved, and requires the filePlot argument.
filePlot	file name where the plot will be saved. For more information, please consult the ggplot2::ggsave() function.

Value

returns an object of ggplot or plotly class.

Examples

```
## Targets file
targetspath <- system.file("extdata", "targets.txt",
package = "systemPipeR")
targets <- read.delim(targetspath, comment = "#")
cmp <- systemPipeR::readComp(file = targetspath,
format = "matrix", delim = "-")
## Count table file
countMatrixPath <- system.file("extdata", "countDFeByg.xls",
 package = "systemPipeR")
countMatrix <- read.delim(countMatrixPath, row.names = 1)
## Plot
exploredds <- exploreDDS(countMatrix, targets,
    cmp = cmp[[1]],
    preFilter = NULL, transformationMethod = "rlog"
)
hclustplot(exploredds, method = "spearman")
```

heatMaplot *Hierarchical Clustering HeatMap (heatMaplot)*

Description

This function performs hierarchical clustering on the transformed expression matrix generated with the DESeq2 package. It uses, by default, a Pearson correlation-based distance measure and complete linkage for cluster join.

Usage

```
heatMaplot(
  exploredds,
  clust,
  DEGlist = NULL,
  plotly = FALSE,
  savePlot = FALSE,
  filePlot = NULL,
  ...
)
```

Arguments

exploredds	object of class DESeq2::DESeqTransform().
clust	select the data to apply the distance matrix computation. If samples selected, it will be applied the stats::dist() function to the transformed count matrix to get sample-to-sample distances. If ind, it is necessary to provide the list of differentially expressed genes, for the exploredds subset.
DEGlist	List of up or down regulated gene/transcript identifiers meeting the chosen filter settings for all comparisons defined in data frames pval and log2FC.
plotly	logical: when FALSE (default), the ggplot2 plot will be returned. TRUE option returns the plotly version of the plot.
savePlot	logical: when FALSE (default), the plot will not be saved. If TRUE the plot will be saved, and requires the filePlot argument.
filePlot	file name where the plot will be saved. For more information, please consult the ggplot2::ggsave() function.
...	additional parameters for the pheatmap::pheatmap() function.

Value

returns an object of pheatmap or plotly class.

References

Raivo Kolde (2019). pheatmap: Pretty Heatmaps. R package version 1.0.12. https://CRAN.R-project.org/package=pheatmap

Examples

```
### Load data
targetspath <- system.file("extdata", "targets.txt",
package = "systemPipeR")
targets <- read.delim(targetspath, comment = "#")
cmp <- systemPipeR::readComp(file = targetspath,
format = "matrix", delim = "-")
countMatrixPath <- system.file("extdata", "countDFeByg.xls",
package = "systemPipeR")
countMatrix <- read.delim(countMatrixPath, row.names = 1)
## Samples plot
exploredds <- exploreDDS(countMatrix, targets,
    cmp = cmp[[1]],
    preFilter = NULL, transformationMethod = "rlog"
)
heatMaplot(exploredds, clust = "samples", plotly = TRUE)
## Individuals genes identified in DEG analysis
### DEG analysis with `systemPipeR`
degseqDF <- systemPipeR::run_DESeq2(
    countDF = countMatrix,
    targets = targets, cmp = cmp[[1]], independent = FALSE
)
DEG_list <- systemPipeR::filterDEGs(
    degDF = degseqDF,
    filter = c(Fold = 2, FDR = 10)
)
### Plot
heatMaplot(exploredds,
    clust = "ind",
    DEGlist = unique(as.character(unlist(DEG_list[[1]])))
)
```

`MAplot` *MAplot*

Description

This function plots log2 fold changes (y-axis) versus the mean of normalized counts (on the x-axis). Statistically significant features are colored.

Usage

```
MAplot(
  degseqDF,
  FDR.cutoff = 0.05,
  comparison,
  filter = c(Fold = 2, FDR = 10),
  genes = "NULL",
  plotly = FALSE,
```

```
  savePlot = FALSE,
  filePlot = NULL
)
```

Arguments

degseqDF	object of class `data.frame` generated by `systemPipeR::run_edgeR()` or `systemPipeR::run_DESeq2()`
FDR.cutoff	filter cutoffs for the p-value adjusted.
comparison	`character vector` specifying the factor names for comparison.
filter	Named vector with filter cutoffs of format c(Fold=2, FDR=1) where Fold refers to the fold change cutoff (unlogged) and FDR to the p-value cutoff.
genes	`character vector` of genes names to show on the plot.
plotly	logical: when `FALSE` (default), the `ggplot2` plot will be returned. `TRUE` option returns the `plotly` version of the plot.
savePlot	logical: when `FALSE` (default), the plot will not be saved. If `TRUE` the plot will be saved, and requires the `filePlot` argument.
filePlot	file name where the plot will be saved. For more information, please consult the `ggplot2::ggsave()` function.

Value

returns an object of `ggplot` or `plotly` class.

Examples

```
## Load targets file and count reads dataframe
targetspath <- system.file("extdata", "targets.txt", package = "systemPipeR")
targets <- read.delim(targetspath, comment = "#")
cmp <- systemPipeR::readComp(
    file = targetspath, format = "matrix",
    delim = "-"
)
countMatrixPath <- system.file("extdata", "countDFeByg.xls",
    package = "systemPipeR"
)
countMatrix <- read.delim(countMatrixPath, row.names = 1)
### DEG analysis with `systemPipeR`
degseqDF <- systemPipeR::run_DESeq2(
    countDF = countMatrix, targets = targets,
    cmp = cmp[[1]], independent = FALSE
)
DEG_list <- systemPipeR::filterDEGs(
    degDF = degseqDF,
    filter = c(Fold = 2, FDR = 10)
)
## Plot
MAplot(degseqDF,
    comparison = "M12-A12", filter = c(Fold = 1, FDR = 20),
    genes = "ATCG00280"
)
```

MDSplot *Multidimensional scaling with MDSplot*

Description

This function computes and plots multidimensional scaling analysis for dimension reduction of count expression matrix. Internally, it is applied the `stats::dist()` function to the transformed count matrix to get sample-to-sample distances.

Usage

```
MDSplot(
  exploredds,
  method = "spearman",
  plotly = FALSE,
  savePlot = FALSE,
  filePlot = NULL
)
```

Arguments

`exploredds`	object of class `DESeq2::DESeqDataSet()`, generated from `exploreDDS` function.
`method`	a `character string` indicating which correlation coefficient is to be computed, based on the `stats::cor()` function. Options are: c("pearson" "kendall", "spearman").
`plotly`	logical: when `FALSE` (default), the `ggplot2` plot will be returned. `TRUE` option returns the `plotly` version of the plot.
`savePlot`	logical: when `FALSE` (default), the plot will not be saved. If `TRUE` the plot will be saved, and requires the `filePlot` argument.
`filePlot`	file name where the plot will be saved. For more information, please consult the `ggplot2::ggsave()` function.

Value

returns an object of `ggplot` or `plotly` class.

Examples

```
## Targets file
targetspath <- system.file("extdata", "targets.txt", package = "systemPipeR")
targets <- read.delim(targetspath, comment = "#")
cmp <- systemPipeR::readComp(file = targetspath, format = "matrix",
delim = "-")
## Count table file
countMatrixPath <- system.file("extdata", "countDFeByg.xls",
package = "systemPipeR")
```

```
countMatrix <- read.delim(countMatrixPath, row.names = 1)
## Plot
exploredds <- exploreDDS(countMatrix, targets, cmp = cmp[[1]],
preFilter = NULL, transformationMethod = "rlog")
MDSplot(exploredds, plotly = FALSE)
```

PCAplot *PCAplot*

Description

This function plots a Principal Component Analysis (PCA) from transformed expression matrix. This plot shows samples variation based on the expression values and identifies batch effects.

Usage

```
PCAplot(exploredds, plotly = FALSE, savePlot = FALSE, filePlot = NULL)
```

Arguments

exploredds	object of class DESeq2::DESeqTransform().
plotly	logical: when FALSE (default), the ggplot2 plot will be returned. TRUE option returns the plotly version of the plot.
savePlot	logical: when FALSE (default), the plot will not be saved. If TRUE the plot will be saved, and requires the filePlot argument.
filePlot	file name where the plot will be saved. For more information, please consult the ggplot2::ggsave() function.

Value

returns an object of ggplot or plotly class.

Examples

```
## Targets file
targetspath <- system.file("extdata", "targets.txt", package = "systemPipeR")
targets <- read.delim(targetspath, comment = "#")
cmp <- systemPipeR::readComp(file = targetspath, format = "matrix",
delim = "-")
## Count table file
countMatrixPath <- system.file("extdata", "countDFeByg.xls",
package = "systemPipeR")
countMatrix <- read.delim(countMatrixPath, row.names = 1)
## Plot
exploredds <- exploreDDS(countMatrix, targets, cmp = cmp[[1]],
preFilter = NULL, transformationMethod = "rlog")
PCAplot(exploredds, plotly = TRUE)
```

showDT *Create an HTML table using DT package with fixed columns*

Description

Create an HTML table using DT package with fixed columns

Usage

```
showDT(data, ...)
```

Arguments

data	data object (either a matrix or a data frame).
...	Additional arguments used by dDT::atatable() function.

Value

returns an object of `datatables` and `htmlwidget`.

Examples

```
showDT(iris)
```

tSNEplot *t-Distributed Stochastic Neighbor embedding with tSNEplot*

Description

This function computes and plots t-Distributed Stochastic Neighbor embedding (t-SNE) analysis for unsupervised nonlinear dimensionality reduction of count expression matrix. Internally, it is applied the `Rtsne::Rtsne()` function, using the exact t-SNE computing with `theta=0.0`.

Usage

```
tSNEplot(
  countMatrix,
  targets,
  plotly = FALSE,
  savePlot = FALSE,
  filePlot = NULL,
  ...
)
```

Arguments

countMatrix	date.frame or matrix containing raw read counts.
targets	targets data.frame.
plotly	logical: when FALSE (default), the ggplot2 plot will be returned. TRUE option returns the plotly version of the plot.
savePlot	logical: when FALSE (default), the plot will not be saved. If TRUE the plot will be saved, and requires the filePlot argument.
filePlot	file name where the plot will be saved. For more information, please consult the ggplot2::ggsave() function.
...	additional parameters for the Rtsne::Rtsne() function.

Value

returns an object of ggplot or plotly class.

References

Jesse H. Krijthe (2015). Rtsne: T-Distributed Stochastic Neighbor Embedding using a Barnes-Hut Implementation, URL: https://github.com/jkrijthe/Rtsne

Examples

```
targetspath <- system.file("extdata", "targets.txt",
package = "systemPipeR")
targets <- read.delim(targetspath, comment = "#")
cmp <- systemPipeR::readComp(file = targetspath, format = "matrix",
delim = "-")
countMatrixPath <- system.file("extdata", "countDFeByg.xls",
package = "systemPipeR")
countMatrix <- read.delim(countMatrixPath, row.names = 1)
set.seed(42)
tSNEplot(countMatrix, targets, perplexity = 5)
```

---

volcanoplot *Volcano plot with* volcanoplot

Description

A simple function that shows statistical significance (p-value) versus magnitude of change (log2 fold change).

Usage

```
volcanoplot(
  degseqDF,
  comparison,
  filter = c(Fold = 2, FDR = 10),
  genes = "NULL",
  plotly = FALSE,
  savePlot = FALSE,
  filePlot = NULL
)
```

Arguments

degseqDF	object of class `data.frame` generated by `systemPipeR::run_edgeR()` or `systemPipeR::run_DESeq2()`
comparison	`character vector` specifying the factor names for comparison.
filter	Named vector with filter cutoffs of format c(Fold=2, FDR=1) where Fold refers to the fold change cutoff (unlogged) and FDR to the p-value cutoff.
genes	`character vector` of genes names to show on the plot.
plotly	logical: when `FALSE` (default), the `ggplot2` plot will be returned. `TRUE` option returns the `plotly` version of the plot.
savePlot	logical: when `FALSE` (default), the plot will not be saved. If `TRUE` the plot will be saved, and requires the `filePlot` argument.
filePlot	file name where the plot will be saved. For more information, please consult the `ggplot2::ggsave()` function.

Value

returns an object of `ggplot` or `plotly` class.

Examples

```
## Load targets file and count reads dataframe
targetspath <- system.file("extdata", "targets.txt", package = "systemPipeR")
targets <- read.delim(targetspath, comment = "#")
cmp <- systemPipeR::readComp(
    file = targetspath, format = "matrix",
    delim = "-"
)
countMatrixPath <- system.file("extdata", "countDFeByg.xls",
    package = "systemPipeR"
)
countMatrix <- read.delim(countMatrixPath, row.names = 1)
### DEG analysis with `systemPipeR`
degseqDF <- systemPipeR::run_DESeq2(
    countDF = countMatrix,
    targets = targets, cmp = cmp[[1]], independent = FALSE)
DEG_list <- systemPipeR::filterDEGs(
    degDF = degseqDF,
    filter = c(Fold = 2, FDR = 10))
```

```
## Plot
volcanoplot(degseqDF,
    comparison = "M12-A12", filter = c(Fold = 1, FDR = 20),
    genes = "ATCG00280")
```

Index

∗ **visualization**
GLMplot, 5
hclustplot, 6
heatMaplot, 8
MAplot, 9
MDSplot, 11
tSNEplot, 13

DESeq2::DESeqDataSet(), *6*, *11*
DESeq2::DESeqTransform(), *3*, *7, 8*, *12*
DESeq2::rlog(), *3, 4*
DESeq2::vst(), *3, 4*

exploreDDS, 2
exploreDDSplot, 4

ggplot2::ggsave(), *5–8*, *10–12*, *14, 15*
glmpca::glmpca(), *6*
GLMplot, 5

hclustplot, 6
heatMaplot, 8

MAplot, 9
MDSplot, 11

PCAplot, 12
pheatmap::pheatmap(), *8*

Rtsne::Rtsne(), *13, 14*

showDT, 13
stats::cor(), *6, 7*, *11*
stats::dist(), *8*, *11*
stats::hclust(), *6*
systemPipeR::readComp(), *3, 4*
systemPipeR::run_DESeq2(), *10*, *15*
systemPipeR::run_edgeR(), *10*, *15*

tSNEplot, 13

volcanoplot, 14