Package ‘SEtools’

October 17, 2020

Type Package

Title SEtools: tools for working with SummarizedExperiment

Version 1.2.0

Depends R (>= 4.0)

Description This includes a set of tools for working with the SummarizedExperiment class, including merging, melting, aggregation and plotting functions.

Imports S4Vectors, SummarizedExperiment, data.table, pheatmap,
seriation, ComplexHeatmap, circlize, methods, BiocParallel,
randomcoloR, edgeR, openxlsx

Suggests BiocStyle, knitr, rmarkdown, ggplot2

biocViews GeneExpression, Visualization

VignetteBuilder knitr

License GPL

Encoding UTF-8

RoxygenNote 7.1.0

BugReports `https://github.com/plger/SEtools`

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

git_branch RELEASE_3_11

git_last_commit 5ab00d8

git_last_commit_date 2020-04-27

Date/Publication 2020-10-16

Author Pierre-Luc Germain [cre, aut] (<https://orcid.org/0000-0003-3418-4218>)

Maintainer Pierre-Luc Germain <`pierre-luc.germain@hest.ethz.ch`>

R topics documented:

aggSE *aggSE*

Description

Aggregates the rows of a ‘SummarizedExperiment‘.

Usage

```
aggSE(x, by, assayFun = NULL, rowDatFuns = list())
```

Arguments

`x`	An object of class ‘SummarizedExperiment‘
`by`	Vector by which to aggregate, or column of ‘rowData(x)‘
`assayFun`	Function by which to aggregate, or a list of such functions (or vector of function names) of the same length as there are assays. If NULL will attempt to use an appropriate function (and notify the functions used), typically the mean.
`rowDatFuns`	A named list providing functions by which to aggregate each rowData columns. If a given column has no specified function, the default will be used, i.e. logical are transformed into a proportion, numerics are aggregated by median, and unique factors/characters are pasted together. Use ‘rowDataFuns=NULL‘ to discard rowData.

Value

An object of class ‘SummarizedExperiment‘

Examples

```
library(SummarizedExperiment)
data("SE", package="SEtools")
# arbitrary IDs for example aggregation:
rowData(SE)$otherID <- rep(LETTERS[1:10],each=10)
SE <- aggSE(SE, "otherID")
```

crossHm *crossHm*

Description

Plot a multi-panel heatmap from a list of `SummarizedExperiment-class`.

Usage

```
crossHm(
  ses,
  genes,
  do.scale = TRUE,
  uniqueScale = FALSE,
  assayName = .getDef("assayName"),
  sortBy = seq_along(ses),
  only.common = TRUE,
  cluster_cols = FALSE,
  cluster_rows = is.null(sortBy),
  toporder = NULL,
  hmcols = NULL,
  breaks = .getDef("breaks"),
  gaps_at = .getDef("gaps_at"),
  gaps_row = NULL,
  anno_rows = .getDef("anno_rows"),
  anno_columns = .getDef("anno_columns"),
  name = NULL,
  anno_colors = list(),
  show_rownames = NULL,
  show_colnames = FALSE,
  rel.width = NULL,
  ...
)
```

Arguments

`ses` A (named) list of `SummarizedExperiment-class`.

`genes` A vector of genes/row.names to plot.

`do.scale` Logical; whether to scale rows in each SE (default TRUE).

`uniqueScale` Logical; whether to force the same colorscale for each heatmap.

`assayName` The name of the assay to use; if multiple names are given, the first available will be used. Defaults to "logcpm", "lognorm".

`sortBy` Names or indexes of ‘ses‘ to use for sorting rows (default all)

`only.common` Logical; whether to plot only rows common to all SEs (default TRUE).

`cluster_cols` Logical; whether to cluster columns (default FALSE).

`cluster_rows` Logical; whether to cluster rows (default TRUE if ‘do.sortRows=FALSE‘, FALSE otherwise).

`toporder` Optional verctor of categories on which to supra-order when sorting rows, or name of a ‘rowData‘ column to use for this purpose.

`hmcols` Colors for the heatmap.

`breaks` Breaks for the heatmap colors. Alternatively, symmetrical breaks can be generated automatically by setting ‘breaks‘ to a numerical value between 0 and 1. The value is passed as the ‘split.prop‘ argument to the `getBreaks` function, and indicates the proportion of the points to map to a linear scale, while the more extreme values will be plotted on a quantile scale. ‘breaks=FALSE‘ will disable symmetrical scale and quantile capping, while retaining automatic breaks. ‘breaks=1‘ will produce a symmetrical scale without quantile capping.

`gaps_at` Columns of ‘colData‘ to use to establish gaps between columns.

`gaps_row` A named vector according to which rows will be split.

`anno_rows` Columns of ‘rowData‘ to use for annotation.

`anno_columns` Columns of ‘colData‘ to use for annotation.

`name` The title of the heatmap key.

`anno_colors` List of colors to use for annotation.

`show_rownames` Whether to show row names (default TRUE if 50 rows or less).

`show_colnames` Whether to show column names (default FALSE).

`rel.width` Relative width of the heatmaps

`...` Any other parameter passed to each call of `Heatmap`.

Value

A Heatmap list.

Examples

```
data("SE", package="SEtools")
se1 <- SE[,1:10]
se2 <- SE[,11:20]
se3 <- mergeSEs( list(se1=se1, se2=se2) )
```

`data` *Example dataset*

Description

A `SummarizedExperiment-class` containing (a subset of) whole-hippocampus RNAseq of mice after different stressors.

Value

a `SummarizedExperiment-class`.

References

Floriou-Servou et al. (2018). Distinct Proteomic, Transcriptomic, and Epigenetic Stress Responses in Dorsal and Ventral Hippocampus. *Biological Psychiatry*, **84**(7): 531-541. DOI: 10.1016/j.biopsych.2018.02.003.

flattenPB *flattenPB*

Description

Flattens a pseudo-bulk SummarizedExperiment as produced by 'muscat::aggregateData' so that all cell types are represented in a single assay. Optionally normalizes the data and calculates per-sample logFCs.

Usage

```
flattenPB(pb, norm = TRUE, lfc_group = NULL)
```

Arguments

`pb`	a pseudo-bulk SummarizedExperiment as produced by 'muscat::aggregateData', with different celltypes/clusters are assays.
`norm`	Logical; whether to calculate logcpm (TMM normalization).
`lfc_group`	the colData column to use to calculate foldchange. If NULL (default), no foldchange assay will be computed.

Value

A SummarizedExperiment

getBreaks *getBreaks*

Description

Produces symmetrical breaks for a color scale, with the scale steps increasing for large values, which is useful to avoid outliers influencing too much the color scale.

Usage

```
getBreaks(x, n, split.prop = 0.98, symmetric = TRUE)
```

Arguments

`x`	A matrix of log2FC (or any numerical values centered around 0)
`n`	The desired number of breaks.
`split.prop`	The proportion of the data points to plot on a linear scale; the remaining will be plotted on a scale with regular frequency per step (quantile).
`symmetric`	Logical; whether breaks should be symmetric around 0 (default TRUE)

Value

A vector of breaks of length = 'n'

Examples

```
dat <- rnorm(100,sd = 10)
getBreaks(dat, 10)
```

`log2FC` *log2FC*

Description

Generates log2(foldchange) matrix/assay, eventually on a per-batch fashion.

Usage

```
log2FC(
  x,
  fromAssay = NULL,
  controls,
  by = NULL,
  isLog = NULL,
  agFun = rowMeans,
  toAssay = "log2FC"
)
```

Arguments

`x`	A numeric matrix, or a ‘SummarizedExperiment‘ object
`fromAssay`	The assay to use if ‘x‘ is a ‘SummarizedExperiment‘
`controls`	A vector of which samples should be used as controls for foldchange calculations.
`by`	An optional vector indicating groups/batches by which the controls will be averaged to calculate per-group foldchanges.
`isLog`	Logical; whether the data is log-transformed. If NULL, will attempt to figure it out from the data and/or assay name
`agFun`	Aggregation function for the baseline (default rowMeans)
`toAssay`	The name of the assay in which to save the output.

Value

An object of same class as ‘x‘; if a ‘SummarizedExperiment‘, will have the additional assay named from ‘toAssay‘.

Examples

```
log2FC( matrix(rnorm(40), ncol=4), controls=1:2 )
```

meltSE *meltSE*

Description

Melts a SE object into a ggplot-ready long data.frame.

Usage

```
meltSE(x, genes, assayName = NULL, colDat.columns = NULL, rowDat.columns = NA)
```

Arguments

x	An object of class SummarizedExperiment-class
genes	A vector of genes to include. Use 'genes=NULL' to include all.
assayName	The name(s) of the assay(s) to use. If NULL and the assays are named, all of them will be included (if they are not named, the first one will be used).
colDat.columns	The colData columns to include (defaults includes all). Use 'colDat.columns=NA' in order not to include any.
rowDat.columns	The rowData columns to include (none included by default). Use 'rowData=NULL' to include all.

Value

A data.frame.

Examples

```
data("SE", package="SEtools")
head(meltSE(SE,"Fos"))
```

mergeSEs *mergeSEs*

Description

Merges a list of SummarizedExperiment-class, either by row.names or through specified rowData fields. In cases of many-to-many (or one-to-many) mappings, 'aggFun' determines whether the records are aggregated by linking ID (if an aggregation method is given) or all combinations are returned (if 'aggFun=NULL' - default).

Usage

```
mergeSEs(
  ll,
  use.assays = NULL,
  do.scale = TRUE,
  commonOnly = TRUE,
  colColumns = NULL,
  mergeBy = NULL,
  aggFun = NULL,
  addDatasetPrefix = TRUE,
  defValues = list(),
  keepRowData = TRUE,
  BPPARAM = SerialParam()
)
```

Arguments

`ll` A (named) list of `SummarizedExperiment-class`

`use.assays` Names (or indexes) of the assays to use. By default, all common assays are used.

`do.scale` A logical vector indicating (globally or for each assay) whether to perform row unit-variance scaling on each dataset before merging (default TRUE).

`commonOnly` Logical; whether to restrict to rows present in all datasets (default TRUE).

`colColumns` A character vector specifying ‘colData‘ columns to include (if available in at least one of the datasets). If NULL, everything is kept.

`mergeBy` The ‘rowData‘ column to merge with. If NULL, row.names are used.

`aggFun` The aggregation function to use when multiple rows have the same ‘mergeBy‘ value. If merging multiple assays, a different function per assay can be passed as a named list (see `aggSE`). If NULL (default), entries will be reused to have each combination.

`addDatasetPrefix` Logical; whether the name of the dataset should be appended to the sample names (default TRUE).

`defValues` An optional named list of default ‘colData‘ values when some columns are missing from some SEs.

`keepRowData` Logical, whether to keep the rowData (default TRUE).

`BPPARAM` For multithreading the aggregation step.

Value

An object of class `SummarizedExperiment-class`

Examples

```
data("SE", package="SEtools")
mergeSEs( list( se1=SE[,1:10], se2=SE[,11:20] ) )
```

`qualitativeColors` *qualitativeColors*

Description

qualitativeColors

Usage

```
qualitativeColors(names, ...)
```

Arguments

`names`	The names to which the colors are to be assigned, or an integer indicating the desired number of colors
`...`	passed to ‘randomcoloR::distinctColorPalette‘

Value

A vector (eventually named) of colors

`resetAllSEtoolsOptions` *resetAllSEtoolsOptions*

Description

Resents all global options relative to SEtools.

Usage

```
resetAllSEtoolsOptions()
```

Value

None

Examples

```
resetAllSEtoolsOptions()
```

SE-heatmap *Heatmap wrappers for* SummarizedExperiment-class.

Description

Heatmap wrappers for SummarizedExperiment-class.

Usage

```
sechm(
  se,
  genes,
  do.scale = FALSE,
  assayName = .getDef("assayName"),
  sortRowsOn = seq_len(ncol(se)),
  cluster_cols = FALSE,
  cluster_rows = is.null(sortRowsOn),
  toporder = NULL,
  hmcols = NULL,
  breaks = .getDef("breaks"),
  gaps_at = .getDef("gaps_at"),
  gaps_row = NULL,
  anno_rows = .getDef("anno_rows"),
  anno_columns = .getDef("anno_columns"),
  name = NULL,
  anno_colors = list(),
  show_rownames = NULL,
  show_colnames = FALSE,
  isMult = FALSE,
  show_heatmap_legend = !isMult,
  includeMissing = FALSE,
  ...
)

sehm(
  se,
  genes,
  do.scale = FALSE,
  assayName = .getDef("assayName"),
  sortRowsOn = seq_len(ncol(se)),
  cluster_cols = FALSE,
  cluster_rows = is.null(sortRowsOn),
  toporder = NULL,
  hmcols = NULL,
  breaks = .getDef("breaks"),
  gaps_at = .getDef("gaps_at"),
  gaps_row = NULL,
  anno_rows = .getDef("anno_rows"),
  anno_columns = .getDef("anno_columns"),
  anno_colors = .getAnnoCols(se),
  show_rownames = NULL,
  show_colnames = FALSE,
  ...
)
```

Arguments

`se` A `SummarizedExperiment-class`.

`genes` An optional vector of genes (i.e. row names of ‘se‘)

`do.scale` Logical; whether to scale rows (default FALSE).

`assayName` An optional vector of assayNames to use. The first available will be used, or the first assay if NULL.

`sortRowsOn` Sort rows by MDS polar order using the specified columns (default all)

`cluster_cols` Whether to cluster columns (default F)

`cluster_rows` Whether to cluster rows; default FALSE if ‘do.sortRows=TRUE‘.

`toporder` Optional verctor of categories on which to supra-order when sorting rows, or name of a ‘rowData‘ column to use for this purpose.

`hmcols` Colors for the heatmap.

`breaks` Breaks for the heatmap colors. Alternatively, symmetrical breaks can be generated automatically by setting ‘breaks‘ to a numerical value between 0 and 1. The value is passed as the ‘split.prop‘ argument to the `getBreaks` function, and indicates the proportion of the points to map to a linear scale, while the more extreme values will be plotted on a quantile scale. ‘breaks=FALSE‘ will disable symmetrical scale and quantile capping, while retaining automatic breaks. ‘breaks=1‘ will produce a symmetrical scale without quantile capping.

`gaps_at` Columns of ‘colData‘ to use to establish gaps between columns.

`gaps_row` Passed to `pheatmap`; if missing, will be set automatically according to toporder.

`anno_rows` Columns of ‘rowData‘ to use for annotation.

`anno_columns` Columns of ‘colData‘ to use for annotation.

`name` The name of the heatmap, eventually appearing as title of the color scale.

`anno_colors` List of colors to use for annotation.

`show_rownames` Whether to show row names (default TRUE if 50 rows or less).

`show_colnames` Whether to show column names (default FALSE).

`isMult` Logical; used to silence labels when plotting mulitple heatmaps

`show_heatmap_legend`
Logical; whether to show heatmap legend

`includeMissing` Logical; whether to include missing genes (default FALSE)

`...` Further arguments passed to ‘pheatmap‘ (‘sehm‘) or ‘Heatmap‘ (‘sechm‘).

Value

A heatmap (see `pheatmap`), or, for ‘sechm‘, a `Heatmap-class`.

Examples

```
data("SE", package="SEtools")
sehm(SE, row.names(SE)[1:10], do.scale=TRUE)
```

se2xls *se2xlsx*

Description

Writes a SummarizedExperiment to an excel/xlsx file. Requires the ‘openxlsx‘ package.

Usage

```
se2xls(se, filename, addSheets = NULL)
```

Arguments

se	The ‘SummarizedExperiment‘
filename	xlsx file name
addSheets	An optional list of additional tables to save as sheets.

Value

Saves to file.

Examples

```
data("SE", package="SEtools")
# not run
# se2xls(SE, filename="SE.xlsx")
```

sortRows *sortRows*

Description

sortRows

Usage

```
sortRows(
  x,
  z = FALSE,
  toporder = NULL,
  na.rm = FALSE,
  method = "MDS_angle",
  toporder.meth = "before"
)
```

Arguments

`x` A numeric matrix or data.frame.

`z` Whether to scale rows for the purpose of calculating order (default FALSE).

`toporder` Optional verctor of categories (length=nrow(x)) on which to supra-order when sorting rows.

`na.rm` Wheter to remove missing values and invariant rows (default FALSE).

`method` Seriation method; 'MDS_angle' (default) or 'R2E' recommended.

`toporder.meth` Whether to perform higher-order sorting 'before' (default) or 'after' the lower-order sorting.

Value

A reordered matrix or data.frame.

Examples

```
# random data
m <- matrix( round(rnorm(100,mean=10, sd=2)), nrow=10,
             dimnames=list(LETTERS[1:10], letters[11:20]) )
m
sortRows(m)
```

Index