Package ‘SEtools’

May 25, 2024

<table>
<thead>
<tr>
<th>Type</th>
<th>Package</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>SEtools: tools for working with SummarizedExperiment</td>
</tr>
<tr>
<td>Version</td>
<td>1.18.0</td>
</tr>
<tr>
<td>Depends</td>
<td>R (&gt;= 4.0), SummarizedExperiment, sechm</td>
</tr>
<tr>
<td>Description</td>
<td>This includes a set of convenience functions for working with the SummarizedExperiment class. Note that plotting functions historically in this package have been moved to the sechm package (see vignette for details).</td>
</tr>
<tr>
<td>Imports</td>
<td>BiocParallel, Matrix, DESeq2, S4Vectors, data.table, edgeR, openxlsx, pheatmap, stats, circlize, methods, sva</td>
</tr>
<tr>
<td>Suggests</td>
<td>BiocStyle, knitr, rmarkdown, ggplot2</td>
</tr>
<tr>
<td>biocViews</td>
<td>GeneExpression</td>
</tr>
<tr>
<td>VignetteBuilder</td>
<td>knitr</td>
</tr>
<tr>
<td>License</td>
<td>GPL</td>
</tr>
<tr>
<td>Encoding</td>
<td>UTF-8</td>
</tr>
<tr>
<td>RoxygenNote</td>
<td>7.2.1</td>
</tr>
</tbody>
</table>

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

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

git_branch RELEASE_3_19

git_last_commit d3e34af

git_last_commit_date 2024-04-30

Repository Bioconductor 3.19

Date/Publication 2024-05-24

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

Maintainer Pierre-Luc Germain <pierre-luc.germain@hest.ethz.ch>
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 dis-
card rowData.

Value

An object of class ‘SummarizedExperiment’
Examples

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

Description

Casts a data.frame as a `SummarizedExperiment-class`.

Usage

```r
castSE(
  x,
  rowNames = NULL,
  colNames = NULL,
  assayNames = NULL,
  colData = NULL,
  rowData = NULL,
  sparse = FALSE
)
```

Arguments

- `x`: A data.frame
- `rowNames`: Column of `x` containing the row.names (if omitted, will build from 'rowData')
- `colNames`: Column of `x` containing the column names (if omitted, will build from 'colData')
- `assayNames`: Columns of `x` to turn into assays
- `colData`: Columns of `x` to use as colData
- `rowData`: Columns of `x` to use as rowData
- `sparse`: Local, whether to keep the assays sparse.

Value

A `SummarizedExperiment-class`

Examples

```r
d <- data.frame(transcript=rep(LETTERS[1:10],each=2), gene=rep(LETTERS[1:5],each=4),
  count=rpois(20, 10), sample=letters[1:2])
head(d)
castSE(d, rowData=c("transcript","gene"), colNames="sample")
```
### data

**Example dataset**

**Description**

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

**Value**

a `SummarizedExperiment-class`.

**References**


---

### 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 fold-change assay will be computed.

**Value**

A SummarizedExperiment
Description

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

Usage

```r
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

```r
log2FC( matrix(rnorm(40), ncol=4), controls=1:2 )
```
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.
**resetAllSEtoolsOptions**

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] ) )

---

**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")

Description

Deprecated pheatmap wrapper for SummarizedExperiment-class. **This function has been replaced by the sechm function from the ‘sechm’ package and is retained here solely for backward compatibility.**

Usage

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 = NULL,
  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 vector 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 the heatmap function; if missing, will be set automatically according to toporder.
anno_rows  Columns of 'rowData' to use for left annotation.
anno_columns  Columns of 'colData' to use for top annotation.
anno_colors  List of colors to use for annotation.
show_rownames  Whether to show row names (default TRUE if less than 50 rows to plot).
show_colnames  Whether to show column names (default FALSE).
...  Further arguments passed to 'pheatmap'

Value

A heatmap.

svacor

Description

A wrapper around SVA-based correction, providing a corrected assay. If this is RNAseq data or similar, use a count assay assay with 'useVST=TRUE'; otherwise (e.g. proteomics) a log-normalized assay is recommended.
svacor

Usage

svacor(
  SE,
  form,
  form0 = ~1,
  assayName = NULL,
  regressOutNull = TRUE,
  useVST = TRUE,
  n.sv = NULL,
  ...
)

Arguments

  SE          An object of class ‘SummarizedExperiment’.
  form        The formula of the differential expression model
  form0       An optional formula for the null model
  assayName   The name (or index) of the assay to use.
  regressOutNull Logical; whether to regress out the variables of ‘form0’.
  useVST      Logical; whether to use DESeq2’s variance-stabilizing transformation; (for count data!)
  n.sv        The number of surrogate variables (if omitted, sva will attempt to estimate it)
  ...         Any other argument passed to the sva command.

Value

Returns the ‘SummarizedExperiment’ with a ‘corrected’ assay and the surrogate variables in ‘col-Data’.

Examples

data("SE", package="SEtools")
SE <- svacor(SE, ~Condition)
Index

aggSE, 2, 6

castSE, 3

data, 4

flattenPB, 4

getBreaks, 9

log2FC, 5

mergeSEs, 6

resetAllSEtoolsOptions, 7

SE (data), 4

se2xls, 7

sechm, 8

sehm, 8

sva, 10

svacor, 9