Package ‘ExperimentSubset’

March 30, 2021

Type Package

Title Manages subsets of data with Bioconductor Experiment objects

Version 1.0.0

Description
Experiment objects such as the SummarizedExperiment or SingleCellExperiment are data containers for one or more matrix-like assays along with the associated row and column data. Often only a subset of the original data is needed for down-stream analysis. For example, filtering out poor quality samples will require excluding some columns before analysis. The ExperimentSubset object is a container to efficiently manage different subsets of the same data without having to make separate objects for each new subset.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Depends R (>= 4.0.0), SummarizedExperiment, SingleCellExperiment

biocViews Infrastructure, Software, DataImport, DataRepresentation

Imports methods, Matrix

Suggests BiocStyle, knitr, rmarkdown, testthat, covr, stats, scran,
scater, scds, TENxPBMCData

VignetteBuilder knitr

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

git_branch RELEASE_3_12

git_last_commit ae54a23

git_last_commit_date 2020-10-27

Date/Publication 2021-03-29

Author Irzam Sarfraz [aut, cre] (<https://orcid.org/0000-0001-8121-792X>),
Muhammad Asif [aut, ths] (<https://orcid.org/0000-0003-1839-2527>),
Joshua D. Campbell [aut] (<https://orcid.org/0000-0003-0780-8662>)

Maintainer Irzam Sarfraz <irzam9095@gmail.com>

R topics documented:

altExp — *Alternative Experiment methods for ExperimentSubset objects*

Description

A wrapper to the altExp from altExps method with additional support for subsets.

Usage

```
altExp(x, e, withColData = FALSE, subsetName)

## S4 method for signature 'ANY'
altExp(x, e, withColData = FALSE, subsetName)
```

Arguments

x	ExperimentSubset Input ExperimentSubset object or any object supported by altExp from altExps method.
e	character(1) Same as altExp from altExps.
withColData	logical(1) Same as altExp from altExps. Default FALSE.
subsetName	character(1) Specify the name of the subset from which the altExp should be fetched from. If missing, altExp from altExps method is called on the main object.

Value

The altExp from the specified subset or same as altExp from altExps when subsetName is missing.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
es <- createSubset(es,
"subset1",
rows = c(10,11,50,56,98,99,102,105,109, 200),
cols = c(20,21,40,45,90,99,100,123,166,299),
parentAssay = "counts")
altExp(es, e = "altExample",
subsetName = "subset1") <- SingleCellExperiment(
assays = list(counts = assay(es, "subset1")))
altExp(es, subsetName = "subset1")
```

altExp<- *Alternative Experiment methods for ExperimentSubset objects*

Description

A wrapper to the `altExp<-` from altExps method with additional support for subsets.

Usage

```
altExp(x, e, withColData = FALSE, subsetName) <- value

## S4 replacement method for signature 'ANY'
altExp(x, e, withColData = FALSE, subsetName) <- value
```

Arguments

`x` `ExperimentSubset` Input `ExperimentSubset` object or any object supported by `altExp<-` from altExps method.

`e` `character(1)` Same as `altExp<-` from altExps method.

`withColData` `logical(1)` Same as `altExp<-` from altExps method. Default `FALSE`.

`subsetName` `character(1)` Specify the name of the subset to which the `altExp<-` should be set to. If `missing`, `altExp<-` from altExps method is called on the main object.

`value` `SingleCellExperiment` Input value same as `altExp<-` from altExps method.

Value

Input object with `altExp<-` set.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
es <- createSubset(es,
"subset1",
rows = c(10,11,50,56,98,99,102,105,109, 200),
cols = c(20,21,40,45,90,99,100,123,166,299),
parentAssay = "counts")
altExp(es, e = "altExample",
subsetName = "subset1") <- SingleCellExperiment(
assays = list(counts = assay(es, "subset1")))
```

altExpNames *Alternative Experiment methods for ExperimentSubset objects*

Description

A wrapper to the altExpNames from altExps method with additional support for subsets.

Usage

```
altExpNames(x, subsetName)

## S4 method for signature 'ANY'
altExpNames(x, subsetName)
```

Arguments

x	ExperimentSubset Input ExperimentSubset object or any object supported by altExpNames from altExps method.
subsetName	character(1) Specify the name of the subset from which the altExpNames should be fetched from. If missing, altExpNames from altExps method is called on the main object.

Value

The altExpNames from the specified subset or same as altExpNames from altExps when subsetName is missing.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
es <- createSubset(es,
"subset1",
rows = c(10,11,50,56,98,99,102,105,109, 200),
cols = c(20,21,40,45,90,99,100,123,166,299),
parentAssay = "counts")
altExp(es, e = "altExample",
subsetName = "subset1") <- SingleCellExperiment(
assays = list(counts = assay(es, "subset1")))
altExpNames(es, subsetName = "subset1")
```

altExpNames<- *Alternative Experiment methods for ExperimentSubset objects*

Description

A wrapper to the altExpNames<- from altExps method with additional support for subsets.

Usage

```
altExpNames(x, subsetName) <- value

## S4 replacement method for signature 'ANY'
altExpNames(x, subsetName) <- value
```

Arguments

x	`ExperimentSubset` Input `ExperimentSubset` object or any object supported by `altExpNames<-` from altExps method.
subsetName	`character(1)` Specify the name of the subset to which the `altExpNames<-` should be set to. If `missing`, `altExpNames<-` from altExps method is called on the main object.
value	`vector("character")` Input value same as `altExpNames<-` from altExps method.

Value

Input object with `altExpNames` set.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
es <- createSubset(es,
"subset1",
rows = c(10,11,50,56,98,99,102,105,109, 200),
cols = c(20,21,40,45,90,99,100,123,166,299),
parentAssay = "counts")
altExp(es, e = "altExample",
subsetName = "subset1") <- SingleCellExperiment(
assays = list(counts = assay(es, "subset1")))
altExpNames(es, subsetName = "subset1") <- c("altExpSubset1")
```

---

altExps *Alternative Experiment methods for ExperimentSubset objects*

Description

A wrapper to the altExps method with additional support for subsets.

Usage

```
altExps(x, withColData = FALSE, subsetName)

## S4 method for signature 'ANY'
altExps(x, withColData = FALSE, subsetName)
```

Arguments

`x` — `ExperimentSubset` Input `ExperimentSubset` object or any object supported by altExps method.

`withColData` — `logical(1)` Same as altExps. Default `FALSE`.

`subsetName` — `character(1)` Specify the name of the subset from which the `altExps` should be fetched from. If `missing`, altExps method is called on the main object.

Value

`altExps` from the specified subset or same as altExps when `subsetName` is `missing`.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
es <- createSubset(es, "subset1",
rows = c(10,11,50,56,98,99,102,105,109, 200),
cols = c(20,21,40,45,90,99,100,123,166,299),
parentAssay = "counts")
altExps(es, subsetName = "subset1") <- list(
alt1 = SingleCellExperiment(
assays = list(
counts = assay(es, "subset1"))),
alt2 = SingleCellExperiment(
assays = list(counts = assay(es, "subset1"))))
altExps(es, subsetName = "subset1")
```

`altExps<-` *Alternative Experiment methods for ExperimentSubset objects*

Description

A wrapper to the `altExps<-` from altExps method with additional support for subsets.

Usage

```
altExps(x, subsetName) <- value

## S4 replacement method for signature 'ANY'
altExps(x, subsetName) <- value
```

Arguments

`x` — `ExperimentSubset` Input `ExperimentSubset` object or any object supported by `altExps<-` from altExps method.

`subsetName` — `character(1)` Specify the name of the subset to which the `altExps<-` should be set to. If `missing`, `altExps<-` from altExps method is called on the main object.

`value` — `list()` Input value same as `altExps<-` from altExps method.

Value

Input object with `altExps<-` set.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
es <- createSubset(es,
"subset1",
rows = c(10,11,50,56,98,99,102,105,109, 200),
cols = c(20,21,40,45,90,99,100,123,166,299),
parentAssay = "counts")
altExps(es, subsetName = "subset1") <- list(
alt1 = SingleCellExperiment(
assays = list(counts = assay(es, "subset1"))),
alt2 = SingleCellExperiment(
assays = list(counts = assay(es, "subset1"))))
altExpNames(es, subsetName = "subset1")
```

`assay,ExperimentSubset,character-method` *Accessor method for assays in ExperimentSubset objects*

Description

Method to get an `assay` from an `ExperimentSubset` object or a subset from an `ExperimentSubset` object or any object supported by `assay` from `SummarizedExperiment`.

Usage

```
## S4 method for signature 'ExperimentSubset,character'
assay(x, i, withDimnames = FALSE, ...)
```

Arguments

`x`	`ExperimentSubset` Specify the input object which can be either `ExperimentSubset` or any object supported by `assay` from `SummarizedExperiment`.
`i`	`character(1)` Name of an assay or name of a subset or name of a subset `assay`.
`withDimnames`	`logical(1)` Set whether dimnames should be applied to `assay`. Default `FALSE`.
`...`	Additional parameters.

Value

The `assay` from the input object.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
es <- createSubset(es, "subset1",
rows = c(10,11,50,56,98,99,102,105,109, 200),
cols = c(20,21,40,45,90,99,100,123,166,299),
parentAssay = "counts")
assay(es, "subset1",
subsetAssayName = "subset1pAssay") <- assay(es, "subset1")[,] + 1
es
```

---

assay<-,ExperimentSubset,character-method

Setter method for assays in ExperimentSubset objects

Description

Method to set an assay to an ExperimentSubset object or a subset from an ExperimentSubset object or any object supported by assay<- from SummarizedExperiment.

Usage

```
## S4 replacement method for signature 'ExperimentSubset,character'
assay(x, i, withDimnames = FALSE, subsetAssayName = NULL, ...) <- value
```

Arguments

x	ExperimentSubset Input ExperimentSubset object or any object supported by assay from SummarizedExperiment.
i	character(1) Name of an assay or name of the subset if storing to an ExperimentSubset object.
withDimnames	logical(1) Set whether dimnames should be applied to assay. Default FALSE.
subsetAssayName	character(1) Name of the assay to store if storing to an ExperimentSubset object.
...	Additional parameters.
value	dgCMatrix The assay to store.

Value

Input object with assay stored.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
es <- createSubset(es, "subset1",
rows = c(10,11,50,56,98,99,102,105,109, 200),
cols = c(20,21,40,45,90,99,100,123,166,299),
parentAssay = "counts")
assay(es, "subset1",
subsetAssayName = "subset1pAssay") <- assay(es, "subset1")[,] + 1
es
```

AssaySubset *AssaySubset constructor*

Description

Constructor for creating a experiment object internally by the `ExperimentSubset` object.

Usage

```
AssaySubset(
  subsetName = "subset",
  rowIndices = NULL,
  colIndices = NULL,
  parentAssay = "counts",
  internalAssay = NULL
)
```

Arguments

`subsetName`	`character(1)` Name of the subset.
`rowIndices`	`vector("numeric")` Indices of the rows to include in the subset.
`colIndices`	`vector("numeric")` Indices of the columns to include in the subset.
`parentAssay`	`character(1)` Name of the parent of this subset.
`internalAssay`	`SummarizedExperiment` An internal object to store additional subset data.

Value

A `AssaySubset` object.

AssaySubset-class *An S4 class to create subset objects to store inside an `ExperimentSubset` object.*

Description

An S4 class to create subset objects to store inside an `ExperimentSubset` object.

Slots

`subsetName` `character(1)` Name of the subset.

`rowIndices` `vector("numeric")` Indices of the rows to include in the subset.

`colIndices` `vector("numeric")` Indices of the columns to include in the subset.

`parentAssay` `character(1)` Name of the parent of this subset.

`internalAssay` `SummarizedExperiment` An internal experiment object to store additional subset data.

`colData,ExperimentSubset-method` *Accessor method for colData in ExperimentSubset objects*

Description

Get `colData` from a subset of an input object or the object itself.

Usage

```
## S4 method for signature 'ExperimentSubset'
colData(x, subsetName = NULL, ...)
```

Arguments

`x`	`ExperimentSubset` Input `ExperimentSubset` object or any object supported by `colData` from `SummarizedExperiment`.
`subsetName`	`character(1)` Name of the subset to get `colData` from. If `NULL` or `missing`, `colData` from main input object is fetched.
`...`	Additional parameters.

Value

The `colData` from input object or subset of an input object.

`colData<-,ExperimentSubset,DataFrame-method` *Setter method for colData in ExperimentSubset objects*

Description

Set `colData` to a subset of an input object or the object itself.

Usage

```
## S4 replacement method for signature 'ExperimentSubset,DataFrame'
colData(x, ..., subsetName) <- value
```

Arguments

`x`	`ExperimentSubset` Input `ExperimentSubset` object or any object supported by `colData<-` from `SummarizedExperiment`.
`...`	Additional parameters.
`subsetName`	`character(1)` Name of the subset to set `colData` to. If `NULL` or `missing`, `colData` to main input object is set.
`value`	`DFrame` The `colData` to store in an object or subset of an object.

Value

Object with `colData` set.

colnames *Accessor method for colnames in ExperimentSubset objects*

Description

Get colnames from an ExperimentSubset object or a subset in the ExperimentSubset object or any object supported by colnames in base package.

Usage

```
colnames(object, ...)

## S4 method for signature 'ANY'
colnames(object, subsetName, ...)
```

Arguments

object	ExperimentSubset Input ExperimentSubset object or any object supported by colnames in base package.
...	Additional parameters and subsetName parameter to pass the name of the subset to get colnames from.
subsetName	character(1) Name of the subset to get colnames from. If missing, colnames from main object are returned.

Value

A vector of colnames.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
es <- createSubset(es, "subset1",
rows = c(10,11,50,56,98,99,102,105,109, 200),
cols = c(20,21,40,45,90,99,100,123,166,299),
parentAssay = "counts")
colnames(es, subsetName = "subset1")
```

colnames<- *Setter method for colnames in ExperimentSubset objects*

Description

Set colnames to an ExperimentSubset object or a subset in the ExperimentSubset object or any object supported by colnames in base package.

Usage

```
colnames(object, ...) <- value

## S4 replacement method for signature 'ANY'
colnames(object, subsetName, ...) <- value
```

Arguments

`object` `ExperimentSubset` Input `ExperimentSubset` object or any object supported by `colnames` in base package.

`...` Additional parameters and `subsetName` parameter to pass the name of the subset to get `colnames` from.

`value` `list()` A `list` of `colnames` to set to the input object.

`subsetName` `character(1)` Name of the subset to get `colnames` from. If `missing`, `colnames` from main object are returned.

Value

Input object with `colnames` set.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
es <- createSubset(es, "subset1",
rows = c(10,11,50,56,98,99,102,105,109, 200),
cols = c(20,21,40,45,90,99,100,123,166,299),
parentAssay = "counts")
colnames(es, subsetName = "subset1") <-
paste0("col", seq(subsetDim(es, subsetName = "subset1")[2]))
```

`createSubset` *Subset creation method for ExperimentSubset objects*

Description

Create a subset from an already available `assay` in the input `ExperimentSubset` object by specifying the rows and columns to include in the subset.

Usage

```
createSubset(object, subsetName, rows = NULL, cols = NULL, parentAssay = NULL)

## S4 method for signature 'ExperimentSubset'
createSubset(object, subsetName, rows = NULL, cols = NULL, parentAssay = NULL)

## S4 method for signature 'SingleCellExperiment'
createSubset(object, subsetName, rows = NULL, cols = NULL, parentAssay = NULL)

## S4 method for signature 'SummarizedExperiment'
createSubset(object, subsetName, rows = NULL, cols = NULL, parentAssay = NULL)
```

Arguments

object	`ExperimentSubset` Specify the object from which a subset should be created. Input can also be any object inherited from `SummarizedExperiment` for immediate conversion and subset formation.
subsetName	`character(1)` Specify the name of the subset to create.
rows	`vector("numeric")` Specify the rows to include in this subset. If `missing` or `NULL`, all rows are included in the subset. Values can be `numeric` or `character`. Default `NULL`.
cols	`vector("numeric")` Specify the columns to include in this subset. If `missing` or `NULL`, all columns are included in the subset. Values can be `numeric` or `character`. Default `NULL`.
parentAssay	`character(1)` Specify the parent `assay` of the subset. This parent `assay` must already be available in the `ExperimentSubset` object. If `NULL`, the first available main `assay` will be marked as parent. Default `NULL`.

Value

An `ExperimentSubset` object that now contains the newly created subset.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
es <- createSubset(es,
"subset1",
rows = c(10,11,50,56,98,99,102,105,109, 200),
cols = c(20,21,40,45,90,99,100,123,166,299),
parentAssay = "counts")
es
```

---

dim,ExperimentSubset-method

Dimensions retrieval method for ExperimentSubset objects

Description

Get dimensions of the `ExperimentSubset` object.

Usage

```
## S4 method for signature 'ExperimentSubset'
dim(x)
```

Arguments

x	codeExperimentSubset Input `ExperimentSubset` object.

Value

A `list` containing number of rows and number of columns of the object.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
dim(es)
```

`ExperimentSubset` *ExperimentSubset constructor*

Description

This constructor function is used to setup the `ExperimentSubset` object by passing either a `SingleCellExperiment` or `SummarizedExperiment` objects or objects inherited by these classes. A subset can also be directly created by passing a named `list` to the `subset` parameter. This named `list` should have parameter values named as `subsetName`, `rows`, `cols` and `parentAssay`.

Usage

```
ExperimentSubset(
  object,
  subset = list(subsetName = NA, rows = NA, cols = NA, parentAssay = NA)
)
```

Arguments

`object`	`SingleCellExperiment` or `SummarizedExperiment` Specify the root object.
`subset`	`list` Specify if a subset should be created from within the constructor. Named parameters in this list should be `subsetName`, `rows`, `cols` and `parentAssay`.

Value

A `ExperimentSubset` object.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
es
```

`ExperimentSubset-class` *An S4 class to create an* `ExperimentSubset` *object with support for subsets.*

Description

An S4 class to create an `ExperimentSubset` object with support for subsets.

Slots

`root` The root object from which all consequent subsets will be created. This can be any object that is inherited from `SummarizedExperiment`.

`subsets` A `list` of `AssaySubset` objects.

metadata — *Metadata accessor method for ExperimentSubset objects*

Description

Get `metadata` from an `ExperimentSubset` object.

Usage

```
metadata(object, subsetName)

## S4 method for signature 'ANY'
metadata(object, subsetName)
```

Arguments

`object`	`ExperimentSubset` Input `ExperimentSubset` object.
`subsetName`	`character(1)` Name of the subset to get the `metadata` from. If `missing`, `metadata` is fetched from the main input object.

Value

A `list` of `metadata` elements.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
es <- createSubset(es,
"subset1",
rows = c(10,11,50,56,98,99,102,105,109, 200),
cols = c(20,21,40,45,90,99,100,123,166,299),
parentAssay = "counts")
metadata(es, subsetName = "subset1") <- list(
meta1 = "This is an example for metadata in subset1")
metadata(es, subsetName = "subset1")
```

metadata<- — *Metadata setter method for ExperimentSubset objects*

Description

Set `metadata` to an `ExperimentSubset` object.

Usage

```
metadata(object, subsetName) <- value
```

Arguments

object	ExperimentSubset Input ExperimentSubset object.
subsetName	character(1) Name of the subset to set the metadata to. If missing, metadata is set to the main input object.
value	A list to set to the metadata slot.

Value

Input object with metadata set.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
es <- createSubset(es,
"subset1",
rows = c(10,11,50,56,98,99,102,105,109, 200),
cols = c(20,21,40,45,90,99,100,123,166,299),
parentAssay = "counts")
metadata(es, subsetName = "subset1") <- list(
meta1 = "This is an example for metadata in subset1")
```

metadata<-,ANY-method *Metadata setter method for ExperimentSubset objects*

Description

Set metadata to an ExperimentSubset object.

Usage

```
## S4 replacement method for signature 'ANY'
metadata(object, subsetName) <- value
```

Arguments

object	ExperimentSubset Input ExperimentSubset object.
subsetName	character(1) Name of the subset to set the metadata to. If missing, metadata is set to the main input object.
value	list() A list to set to the metadata slot.

Value

Input object with metadata set.

reducedDim *Methods for Reduced Dimensions in ExperimentSubset objects*

Description

A wrapper to the `reducedDim` from reducedDims method with additional support for subsets.

Usage

```
reducedDim(object, type, withDimnames, subsetName)

## S4 method for signature 'ANY'
reducedDim(object, type, withDimnames, subsetName)
```

Arguments

`object`	`ExperimentSubset` Input `ExperimentSubset` object or any object supported by `reducedDim` from reducedDims method.
`type`	`character(1)` Same as `type` in `reducedDim` from reducedDims method.
`withDimnames`	`logical(1)` Same as `withDimnames` in `reducedDim` from reducedDims method.
`subsetName`	`character(1)` Specify the name of the subset from which the `reducedDim` should be fetched from. If `missing`, `reducedDim` from reducedDims method is called on the main object.

Value

The `reducedDim` from the specified subset or same as `reducedDim` from reducedDims when `subsetName` is `missing`.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
es <- createSubset(es, "subset1",
rows = c(1:1500), cols = c(1:1500),
parentAssay = "counts")
reducedDim(es, type = "PCA",
subsetName = "subset1") <- scater::calculatePCA(
assay(es, "subset1"))
reducedDim(es, type = "PCA", subsetName = "subset1")
```

reducedDim<- *Methods for Reduced Dimensions in ExperimentSubset objects*

Description

A wrapper to the `reducedDim<-` from reducedDims method with additional support for subsets.

Usage

```
reducedDim(object, type, subsetName) <- value
```

Arguments

`object`	`ExperimentSubset` Input `ExperimentSubset` object or any object supported by `reducedDim<-` from reducedDims method.
`type`	`character(1)` Same as type in `reducedDim<-` from reducedDims method.
`subsetName`	`character(1)` Specify the name of the subset to which the `reducedDim` should be set to. If `missing`, `reducedDim<-` from reducedDims method is called on the main object.
`value`	`matrix` Value to set to `reducedDim`.

Value

Updated input object with `reducedDim` set.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
es <- createSubset(es, "subset1",
rows = c(1:1500), cols = c(1:1500),
parentAssay = "counts")
reducedDim(es, type = "PCA",
subsetName = "subset1") <- scater::calculatePCA(
assay(es, "subset1"))
```

`reducedDim<-,ANY-method` *Methods for Reduced Dimensions in ExperimentSubset objects*

Description

A wrapper to the `reducedDim<-` from reducedDims method with additional support for subsets.

Usage

```
## S4 replacement method for signature 'ANY'
reducedDim(object, type, subsetName) <- value
```

Arguments

`object`	`ExperimentSubset` Input `ExperimentSubset` object or any object supported by `reducedDim<-` from reducedDims method.
`type`	`character(1)` Same as type in `reducedDim<-` from reducedDims method.
`subsetName`	`character(1)` Specify the name of the subset to which the `reducedDim` should be set to. If `missing`, `reducedDim<-` from reducedDims method is called on the main object.
`value`	`matrix` Value to set to `reducedDim`.

Value

Updated input object with `reducedDim` set.

`reducedDimNames` *Methods for Reduced Dimensions in ExperimentSubset objects*

Description

A wrapper to the `reducedDimNames` from reducedDims method with additional support for subsets.

Usage

```
reducedDimNames(x, subsetName)

## S4 method for signature 'ANY'
reducedDimNames(x, subsetName)
```

Arguments

`x`	`ExperimentSubset` Input `ExperimentSubset` object or any object supported by `reducedDimNames` from reducedDims method.
`subsetName`	`character(1)` Specify the name of the subset from which the `reducedDimNames` should be fetched from. If `missing`, `reducedDimNames` from reducedDims method is called on the main object.

Value

The `reducedDimNames` from the specified subset or same as `reducedDimNames` from reducedDims when `subsetName` is `missing`.

`reducedDimNames<-` *Methods for Reduced Dimensions in ExperimentSubset objects*

Description

A wrapper to the `reducedDimNames<-` from reducedDims method with additional support for subsets.

Usage

```
reducedDimNames(x, subsetName) <- value

## S4 replacement method for signature 'ANY'
reducedDimNames(x, subsetName) <- value
```

Arguments

`x`	`ExperimentSubset` Input `ExperimentSubset` object or any object supported by `reducedDimNames<-` from reducedDims method.
`subsetName`	`character(1)` Specify the name of the subset to which the `reducedDimNames<-` should be set to. If `missing`, `reducedDimNames<-` from reducedDims method is called on the main object.
`value`	`vector("character")` Input value same as `reducedDimNames<-` from reducedDims method.

Value

Input object with `reducedDimNames<-` set.

`reducedDims` *Methods for Reduced Dimensions in ExperimentSubset objects*

Description

A wrapper to the reducedDims method with additional support for subsets.

Usage

```
reducedDims(object, withDimnames, subsetName)

## S4 method for signature 'ANY'
reducedDims(object, withDimnames, subsetName)
```

Arguments

`object`	`ExperimentSubset` Input `ExperimentSubset` object or any object supported by reducedDims method.
`withDimnames`	`logical(1)` Same as `withDimnames` in reducedDims method.
`subsetName`	`character(1)` Specify the name of the subset from which the `reducedDims` should be fetched from. If `missing`, reducedDims method is called on the main object.

Value

The `reducedDims` from the specified subset or same as link[SingleCellExperiment]reducedDims when `subsetName` is `missing`.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
es <- createSubset(es, "subset1",
rows = c(1:1500), cols = c(1:1500),
parentAssay = "counts")
reducedDim(es, type = "PCA_1",
subsetName = "subset1") <- scater::calculatePCA(
assay(es, "subset1"))
```

```
reducedDim(es, type = "PCA_2",
subsetName = "subset1") <- scater::calculatePCA(
assay(es, "subset1"))
reducedDims(es, subsetName = "subset1")
```

---

reducedDims<- *Methods for Reduced Dimensions in ExperimentSubset objects*

Description

A wrapper to the reducedDims<- from reducedDims method with additional support for subsets.

Usage

```
reducedDims(object, subsetName) <- value
```

Arguments

object	ExperimentSubset Input ExperimentSubset object or any object supported by reducedDims<- from reducedDims method.
subsetName	character(1) Specify the name of the subset to which the reducedDims should be set to. If missing, reducedDims<- from reducedDims method is called on the main object.
value	list() A list of values to set to reducedDims.

Value

Updated input object with reducedDims set.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
es <- createSubset(es, "subset1",
rows = c(1:1500), cols = c(1:1500),
parentAssay = "counts")
reducedDims(es, subsetName = "subset1") <- list(
PCA_1 = scater::calculatePCA(assay(es, "subset1")),
PCA_2 = scater::calculatePCA(assay(es, "subset1")))
reducedDims(es, subsetName = "subset1")
```

reducedDims<-,ANY-method *Methods for Reduced Dimensions in ExperimentSubset objects*

Description

A wrapper to the reducedDims<- from reducedDims method with additional support for subsets.

Usage

```
## S4 replacement method for signature 'ANY'
reducedDims(object, subsetName) <- value
```

Arguments

object	ExperimentSubset Input ExperimentSubset object or any object supported by reducedDims<- from reducedDims method.
subsetName	character(1) Specify the name of the subset to which the reducedDims should be set to. If missing, reducedDims<- from reducedDims method is called on the main object.
value	list() A list of values to set to reducedDims.

Value

Updated input object with reducedDims set.

rowData,ExperimentSubset-method *Accessor method for rowData in ExperimentSubset objects*

Description

Get rowData from a subset of an input object or the object itself.

Usage

```
## S4 method for signature 'ExperimentSubset'
rowData(x, subsetName = NULL, ...)
```

Arguments

x	ExperimentSubset Input ExperimentSubset object or any object supported by rowData from SummarizedExperiment.
subsetName	character(1) Name of the subset to get rowData from. If NULL or missing, rowData from main input object is fetched.
...	Additional parameters.

Value

The rowData from input object or subset of an input object.

rowData<-,ExperimentSubset-method *Setter method for rowData in ExperimentSubset objects*

Description

Set rowData to a subset of an input object or the object itself.

Usage

```
## S4 replacement method for signature 'ExperimentSubset'
rowData(x, ..., subsetName) <- value
```

Arguments

x	ExperimentSubset Input ExperimentSubset object or any object supported by rowData<- from SummarizedExperiment.
...	Additional parameters.
subsetName	character(1) Name of the subset to set rowData to. If NULL or missing, rowData to main input object is set.
value	DFrame The rowData to store in an object or subset of an object.

Value

Object with rowData set.

rownames *Accessor method for rownames in ExperimentSubset objects*

Description

Get rownames from an ExperimentSubset object or a subset in the ExperimentSubset object or any object supported by rownames in base package.

Usage

```
rownames(object, ...)

## S4 method for signature 'ANY'
rownames(object, subsetName, ...)
```

Arguments

object	ExperimentSubset Input ExperimentSubset object or any object supported by rownames in base package.
...	Additional parameters and subsetName parameter to pass the name of the subset to get rownames from.
subsetName	character(1) Name of the subset to get rownames from. If missing, rownames from main object are returned.

Value

A vector of rownames.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
es <- createSubset(es, "subset1",
rows = c(10,11,50,56,98,99,102,105,109, 200),
cols = c(20,21,40,45,90,99,100,123,166,299),
parentAssay = "counts")
rownames(es, subsetName = "subset1")
```

---

rownames<- *Setter method for rownames in ExperimentSubset objects*

Description

Set rownames to an ExperimentSubset object or a subset in the ExperimentSubset object or any object supported by rownames in base package.

Usage

```
rownames(object, ...) <- value

## S4 replacement method for signature 'ANY'
rownames(object, subsetName, ...) <- value
```

Arguments

object	ExperimentSubset Input ExperimentSubset object or any object supported by rownames in base package.
...	Additional parameters and subsetName parameter to pass the name of the subset to get rownames from.
value	codelist() A list of rownames to set to the input object.
subsetName	character(1) Name of the subset to get rownames from. If missing, rownames from main object are returned.

Value

Input object with rownames set.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
es <- createSubset(es, "subset1",
rows = c(10,11,50,56,98,99,102,105,109, 200),
cols = c(20,21,40,45,90,99,100,123,166,299),
parentAssay = "counts")
rownames(es, subsetName = "subset1") <-
paste0("row", seq(subsetDim(es, subsetName = "subset1")[1]))
```

`show,ExperimentSubset-method` *Display method for ExperimentSubset objects*

Description

Show the `ExperimentSubset` object

Usage

```
## S4 method for signature 'ExperimentSubset'
show(object)
```

Arguments

`object` `ExperimentSubset` Input `ExperimentSubset` object.

Value

Displays the overall contents of the `ExperimentSubset` object.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
es
```

`storeSubset` *Method for storing new assays in ExperimentSubset objects*

Description

Store a new subset `assay` inside a specified subset in the input `ExperimentSubset` object.

Usage

```
storeSubset(object, subsetName, inputMatrix, subsetAssayName)

## S4 method for signature 'ExperimentSubset'
storeSubset(object, subsetName, inputMatrix, subsetAssayName = NULL)
```

Arguments

`object` `ExperimentSubset` Specify the input object.

`subsetName` `character(1)` Specify the name of the existing subset inside which the new subset `assay` should be stored.

`inputMatrix` `dgCMatrix` The input subset `assay`.

`subsetAssayName`

`character(1)` Specify the name of the new `assay` against the `inputMatrix` parameter. If `NULL`, a new subset is created internally using the `createSubset` function. Default `NULL`.

Value

Updated `ExperimentSubset` object with the new `assay` stored inside the specified subset.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
es <- createSubset(es, "subset1",
rows = c(10,11,50,56,98,99,102,105,109, 200),
cols = c(20,21,40,45,90,99,100,123,166,299),
parentAssay = "counts")
counts1p <- assay(es, "subset1")
counts1p[,] <- counts1p[,] + 1
es <- storeSubset(es, "subset1", counts1p, "scaledSubset1")
es
```

`subsetAssayCount` *Count method for subset assays in ExperimentSubset objects*

Description

Get the count of the total available subsets and the subset assays inside these subsets in an `ExperimentSubset` object.

Usage

```
subsetAssayCount(object)

## S4 method for signature 'ExperimentSubset'
subsetAssayCount(object)
```

Arguments

`object`	`ExperimentSubset` Input `ExperimentSubset` object.

Value

A `numeric` value representing the sum of the subset count and subset assay count.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
es <- createSubset(es,
"subset1",
rows = c(10,11,50,56,98,99,102,105,109, 200),
cols = c(20,21,40,45,90,99,100,123,166,299),
parentAssay = "counts")
assay(es, "subset1",
subsetAssayName = "subset1pAssay") <- assay(es, "subset1")[,] + 1
subsetAssayCount(es)
```

subsetAssayNames *Name retrieval method for all subset assays in ExperimentSubset objects*

Description

Retrieves the names of all the subsets as well as the subset assays.

Usage

```
subsetAssayNames(object)

## S4 method for signature 'ExperimentSubset'
subsetAssayNames(object)
```

Arguments

object	ExperimentSubset Input ExperimentSubset object.

Value

A vector containing the names of the subsets and the subset assays available in the input ExperimentSubset object.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
es <- createSubset(es, "subset1",
rows = c(10,11,50,56,98,99,102,105,109, 200),
cols = c(20,21,40,45,90,99,100,123,166,299),
parentAssay = "counts")
assay(es, "subset1",
subsetAssayName = "subset1pAssay") <- assay(es, "subset1")[,] + 1
subsetAssayNames(es)
```

subsetColData *Accessor method for colData from subsets in ExperimentSubset objects*

Description

Get colData from a subset.

Usage

```
subsetColData(object, subsetName)

## S4 method for signature 'ExperimentSubset,character'
subsetColData(object, subsetName)
```

Arguments

object	ExperimentSubset Input ExperimentSubset object.
subsetName	character(1) Name of the subset to get colData from.

Value

The colData from input object.

---

subsetCount *Subset count method for ExperimentSubset objects*

Description

Get the total count of the available subsets in an ExperimentSubset object.

Usage

```
subsetCount(object)

## S4 method for signature 'ExperimentSubset'
subsetCount(object)
```

Arguments

object	ExperimentSubset Input ExperimentSubset object.

Value

A numeric value representing the total count of the subsets.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
es <- createSubset(es,
"subset1",
rows = c(10,11,50,56,98,99,102,105,109, 200),
cols = c(20,21,40,45,90,99,100,123,166,299),
parentAssay = "counts")
subsetCount(es)
```

subsetDim *Get dimensions of subsets in ExperimentSubset objects*

Description

Retrieves the dimensions of the specified subset in an `ExperimentSubset` object.

Usage

```
subsetDim(object, subsetName)

## S4 method for signature 'ExperimentSubset,character'
subsetDim(object, subsetName)
```

Arguments

object	`ExperimentSubset` Input `ExperimentSubset` object.
subsetName	`character(1)` Name of the subset to retrieve the dimensions from.

Value

A `vector` containing the dimensions of the specified subset i.e. the number of rows and the number of columns in the subset.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
es <- createSubset(es,
"subset1",
rows = c(10,11,50,56,98,99,102,105,109, 200),
cols = c(20,21,40,45,90,99,100,123,166,299),
parentAssay = "counts")
subsetDim(es, "subset1")
```

subsetNames *Get names of subsets in ExperimentSubset objects*

Description

Retrieves the names of the available subsets in an `ExperimentSubset` object.

Usage

```
subsetNames(object)

## S4 method for signature 'ExperimentSubset'
subsetNames(object)
```

Arguments

object	`ExperimentSubset` Specify the input object.

Value

A `vector` of subset names.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
es <- createSubset(es,
"subset1",
rows = c(10,11,50,56,98,99,102,105,109, 200),
cols = c(20,21,40,45,90,99,100,123,166,299),
parentAssay = "counts")
subsetNames(es)
```

`subsetParent` *Subset parent hierarchy retrieval method for ExperimentSubset objects*

Description

Retrieves a complete subset to parent link from a specified subset.

Usage

```
subsetParent(object, subsetName)

## S4 method for signature 'ANY'
subsetParent(object, subsetName)
```

Arguments

object	`ExperimentSubset` Input `ExperimentSubset` object.
subsetName	`character(1)` Specify the name of the subset against which the subset to parent link should be retrieved.

Value

A `list` containing the parent link of the subset.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
es <- createSubset(es, "subset1",
rows = c(10,11,50,56,98,99,102,105,109, 200),
cols = c(20,21,40,45,90,99,100,123,166,299),
parentAssay = "counts")
assay(es, "subset1",
subsetAssayName = "subset1pAssay") <- assay(es, "subset1")[,] + 1
subsetParent(es, "subset1pAssay")
```

subsetRowData *Accessor method for rowData from subsets in ExperimentSubset objects*

Description

Get `rowData` from a subset.

Usage

```
subsetRowData(object, subsetName)

## S4 method for signature 'ExperimentSubset,character'
subsetRowData(object, subsetName)
```

Arguments

`object`	`ExperimentSubset` Input `ExperimentSubset` object.
`subsetName`	`character(1)` Name of the subset to get `rowData` from.

Value

The `rowData` from input object.

subsetSummary *Method for displaying child-parent link structure of subsets in ExperimentSubset objects*

Description

The function displays the content of an `ExperimentSubset` object including all available main assays, all subsets and the subset assays inside these subsets. This function also depicts how and in what order the subsets in the object are linked with their parents. Moreover, all supplementary data inside the subsets such as `reducedDims` and `altExps` are also displayed against each subset entry.

Usage

```
subsetSummary(object)

## S4 method for signature 'ExperimentSubset'
subsetSummary(object)
```

Arguments

`object`	`ExperimentSubset` Input `ExperimentSubset` object.

Value

Prints all the available subset information against the input `ExperimentSubset` object.

Examples

```
data(sce_chcl, package = "scds")
es <- ExperimentSubset(sce_chcl)
es <- createSubset(es,
"subset1",
rows = c(10,11,50,56,98,99,102,105,109, 200),
cols = c(20,21,40,45,90,99,100,123,166,299),
parentAssay = "counts")
assay(es, "subset1",
subsetAssayName = "subset1pAssay") <- assay(es, "subset1")[,] + 1
subsetSummary(es)
```

Index