Package 'SingleCellExperiment'

April 12, 2018

Version 1.0.0

Date 2017-09-26

Title S4 Classes for Single Cell Data

Depends R (>= 3.4), SummarizedExperiment

Imports S4Vectors, methods, BiocGenerics, utils

Suggests testthat, BiocStyle, knitr, scRNAseq, magrittr, Rtsne

biocViews DataRepresentation, DataImport, Infrastructure, SingleCell

Description Defines a S4 class for storing data from single-cell experiments. This includes specialized methods to store and retrieve spike-in information, dimensionality reduction coordinates and size factors for each cell, along with the usual metadata for genes and libraries.

License GPL-3

VignetteBuilder knitr

NeedsCompilation no

Author Aaron Lun [aut, cph],
Davide Risso [aut, cre, cph]

Maintainer Davide Risso <risso.davide@gmail.com>

R topics documented:

Combining methods *Combining methods*

Description

Methods to combine SingleCellExperiment objects.

Usage

```
## S4 method for signature 'SingleCellExperiment'
rbind(..., deparse.level=1)

## S4 method for signature 'SingleCellExperiment'
cbind(..., deparse.level=1)
```

Arguments

`...`	One or more SingleCellExperiment objects.
`deparse.level`	An integer scalar; see `?base::cbind` for a description of this argument.

Details

For `rbind`, SingleCellExperiment objects are combined row-wise, i.e., rows in successive objects are appended to the first object. Column metadata, experimental metadata and `reducedDims` coordinates will only be taken from the first element in the list.

For `cbind`, SingleCellExperiment objects are combined columns-wise, i.e., columns in successive objects are appended to the first object. `reducedDims` coordinates will be combined row-wise to reflect the addition or more cells. Row and experimental metadata will only be taken from the first element in the list.

Value

A SingleCellExperiment object containing all rows/columns of the supplied objects.

Author(s)

Aaron Lun

Examples

```
example(SingleCellExperiment, echo=FALSE) # using the class example
rbind(sce, sce)
cbind(sce, sce)
dim(reducedDims(sce)[[1]])
```

`Miscellaneous methods` *Miscellaneous methods*

Description

Various methods for the SingleCellExperiment class.

Usage

```
## S4 method for signature 'SingleCellExperiment'
show(x)

## S4 method for signature 'SingleCellExperiment'
objectVersion(x)
```

Arguments

`x`	A SingleCellExperiment class.

Details

The `show` method will print out information about the data contained in `x`. This describes the stored assays and row/column metadata, as done in `show,SummarizedExperiment-method`. The names of the `reducedDims` slot and the names of the spike-ins (see `spikeNames`) are also reported.

The `objectVersion` method will return the version of the package with which `x` was constructed. This is useful for checking if the object is up to date relative to the latest versions of the package.

Value

For `show`, a message is printed to screen describing the data stored in `x`.

For `objectVersion`, an object of class `package_version` is returned.

See Also

`spikeNames`

Examples

```
example(SingleCellExperiment, echo=FALSE) # Using the class example
show(sce)
objectVersion(sce)
```

namedAssays *Named assay fields*

Description

Convenience methods to get or set named assay fields.

Usage

```
## S4 method for signature 'SingleCellExperiment'
counts(object)
## S4 replacement method for signature 'SingleCellExperiment'
counts(object) <- value

## S4 method for signature 'SingleCellExperiment'
normcounts(object)
## S4 replacement method for signature 'SingleCellExperiment'
normcounts(object) <- value

## S4 method for signature 'SingleCellExperiment'
logcounts(object)
## S4 replacement method for signature 'SingleCellExperiment'
logcounts(object) <- value

## S4 method for signature 'SingleCellExperiment'
cpm(object)
## S4 replacement method for signature 'SingleCellExperiment'
cpm(object) <- value

## S4 method for signature 'SingleCellExperiment'
tpm(object)
## S4 replacement method for signature 'SingleCellExperiment'
tpm(object) <- value
```

Arguments

object	A SingleCellExperiment object.
value	A numeric matrix of the same dimensions as object.

Details

These are wrapper methods for getting or setting assay(object, i=X) where X is the name of the method. For example, counts will get or set X="counts". This provide some convenience for users as well as encouraging standardization of naming across packages.

Our suggested interpretation of the fields are as follows:

counts: Raw count data, e.g., number of reads or transcripts.

normcounts: Normalized values on the same scale as the original counts. For example, counts divided by cell-specific size factors that are centred at unity.

logcounts: Log-transformed counts or count-like values. In most cases, this will be defined as log-transformed normcounts, e.g., using log base 2 and a pseudo-count of 1.

`cpm`: Counts-per-million. This is the read count for each gene in each cell, divided by the library size of each cell in millions.

`tpm`: Transcripts-per-million. This is the number of transcripts for each gene in each cell, divided by the total number of transcripts in that cell (in millions).

Value

Each method returns a matrix from the correspondingly named field in the `assays` slot.

Author(s)

Aaron Lun

See Also

`SingleCellExperiment`

Examples

```
example(SingleCellExperiment, echo=FALSE) # Using the class example
counts(sce) <- matrix(rnorm(nrow(sce)*ncol(sce)), ncol=ncol(sce))
dim(counts(sce))

# One possible way of computing normalized "counts"
sf <- 2^rnorm(ncol(sce))
sf <- sf/mean(sf)
normcounts(sce) <- t(t(counts(sce))/sf)
dim(normcounts(sce))

# One possible way of computing log-counts
logcounts(sce) <- log2(normcounts(sce)+1)
dim(normcounts(sce))
```

`Reduced dimensions` *Reduced dimensions methods*

Description

Methods to get or set the dimensionality reduction results.

Usage

```
## S4 method for signature 'SingleCellExperiment'
reducedDim(x, type)

## S4 replacement method for signature 'SingleCellExperiment'
reducedDim(x, type) <- value

## S4 method for signature 'SingleCellExperiment'
reducedDims(x)

## S4 replacement method for signature 'SingleCellExperiment'
reducedDims(x) <- value
```

Arguments

x	A SingleCellExperiment object.
type	A string containing the name for the dimensionality reduction results or a numeric index containing the position of the desired dimenionality reduction result.
value	For reducedDim<-, a matrix (usually double-precision) of coordinates, for each cell (row) and dimension (column). For reducedDims<-, a named SimpleList object containing such matrices.

Details

Dimensionality reduction is often used to interpreting the results of single-cell data analysis. These methods allow the results of dimensionality reduction methods to be stored in a SingleCellExperiment object. Multiple results can be stored in a single object by assigning to different type in reducedDim<-.

If type is NULL or missing for reducedDim, the first set of dimensionality reduction results is returned (or NULL, if no results are present). If value is NULL for reducedDim<-, the set of results corresponding to type is removed from the object.

Value

For reducedDim, a numeric matrix is returned containing coordinates for cells (rows) and dimensions (columns).

For reducedDims, a named SimpleList of matrices is returned, with one matrix for each type of dimensionality reduction method.

For reducedDim<- and reducedDims<-, a SingleCellExperiment object is returned with updated results in the reducedDims slot.

For reducedDimNames, a character vector containing the names of the elements in reducedDims.

Author(s)

Aaron Lun

See Also

SingleCellExperiment-class

Examples

```
example(SingleCellExperiment, echo=FALSE)
reducedDim(sce, "PCA")
reducedDim(sce, "tSNE")
reducedDims(sce)

reducedDim(sce, "PCA") <- NULL
reducedDims(sce)

reducedDims(sce) <- SimpleList()
reducedDims(sce)
```

SingleCellExperiment *SingleCellExperiment class*

Description

A description of the SingleCellExperiment class for storing single-cell sequencing data.

Usage

```
SingleCellExperiment(..., reducedDims=SimpleList())

## S4 method for signature 'SingleCellExperiment'
colData(x, internal=FALSE)
## S4 method for signature 'SingleCellExperiment'
rowData(x, internal=FALSE)
```

Arguments

...	Arguments to pass to the SummarizedExperiment constructor.
reducedDims	A SimpleList object containing matrices of cell coordinates in reduced space.
x	A SingleCellExperiment object.
internal	Whether the information contained in the internal slots should be returned. See details.

Details

The SingleCellExperiment class inherits from the SummarizedExperiment class, with several additional slots:

reducedDims: A SimpleList containing matrices of cell coordinates.

int_elementMetadata: A DataFrame containing internal row metadata (for each genomic feature).

int_colData: A DataFrame containing internal column metadata (for each cell).

int_metadata: A list containing internal experiment metadata.

The intended use of this class is the same as that for SummarizedExperiment instances. Rows should represent genomic features such as genes, while columns represent samples - in this case, single cells. Different quantifications (e.g., counts, CPMs, log-expression) can be stored simultaneously in the assays slot. Row and column metadata can be attached using rowData and colData, respectively.

The additional reducedDims slot allows storage of results from multiple dimensionality reduction methods, e.g., PCA or t-SNE. Each element of the SimpleList should be a matrix of coordinates for all cells from one reduction method. The number of rows of each matrix should be equal to the number of cells in the SingleCellExperiment object.

The internal metadata slots are not intended for external use. Please use the appropriate getter/setter functions instead, such as isSpike or sizeFactors.

It may sometimes be useful to return both the visible and the internal colData in a single DataFrame. This can be achieved by using colData(x, internal=TRUE), which will return the stored colData along with the int_colData (currently the sizeFactors). Similarly, rowData(x, internal=TRUE) will return the stored rowData along with the int_rowData (currently the columns corresponding to isSpike). Warnings will be raised in the unlikely event of any name clashes.

Value

A SingleCellExperiment object is returned from the constructor.

`colData` and `rowData` return a `DataFrame`.

Author(s)

Aaron Lun and Davide Risso

See Also

`isSpike`, `sizeFactors`, `reducedDims`

Examples

```
ncells <- 100
u <- matrix(rpois(20000, 5), ncol=ncells)
v <- log2(u + 1)

pca <- matrix(runif(ncells*5), ncells)
tsne <- matrix(rnorm(ncells*2), ncells)

sce <- SingleCellExperiment(assays=list(counts=u, logcounts=v),
    reducedDims=SimpleList(PCA=pca, tSNE=tsne))
sce

## coercion from SummarizedExperiment
se <- SummarizedExperiment(assays=list(counts=u, logcounts=v))
as(se, "SingleCellExperiment")
```

`Size factor methods` *Size factors methods*

Description

Gets or sets the size factors for all cells.

Usage

```
## S4 method for signature 'SingleCellExperiment'
sizeFactors(object, type=NULL)

## S4 replacement method for signature 'SingleCellExperiment'
sizeFactors(object, type=NULL) <- value
```

Arguments

`object`	A SingleCellExperiment object.
`type`	A string specifying the *type* of size factor to get or set.
`value`	A numeric vector of size factors for all cells.

Details

A size factor is a scaling factor used to divide the raw counts of a particular cell to obtain normalized expression values. The `sizeFactors` methods can be used to get or set size factors for all cells.

The `type` argument allows storage of multiple vectors of size factors (e.g., different values for spike-ins versus endogenous genes). If `type` is `NULL`, a "default" set of size factors is stored or returned.

If `value` is `NULL` for `isSpike<-`, size factors of `type` will be removed from `object`.

Value

For `sizeFactors`, a numeric vector is returned containing size factors of the set `type` for all cells. If `type` is not available, `NULL` is returned instead.

For `sizeFactors<-`, a SingleCellExperiment is returned with size factors stored in the internal metadata fields.

Author(s)

Aaron Lun

See Also

`SingleCellExperiment-class`

Examples

```
example(SingleCellExperiment, echo=FALSE) # Using the class example
sizeFactors(sce) <- runif(ncol(sce))
sizeFactors(sce)

sizeFactors(sce, "ERCC") <- runif(ncol(sce))
sizeFactors(sce, "ERCC")
sizeFactors(sce) # unchanged.

sizeFactors(sce, "ERCC") <- NULL
sizeFactors(sce, "ERCC")
```

`Spike-in methods` *Spike-in methods*

Description

Gets or sets the rows corresponding to spike-in transcripts.

Usage

```
## S4 method for signature 'SingleCellExperiment,character'
isSpike(x, type)

## S4 method for signature 'SingleCellExperiment,missing'
isSpike(x, type)
```

```
## S4 method for signature 'SingleCellExperiment,NULL'
isSpike(x, type)

## S4 replacement method for signature 'SingleCellExperiment,character'
isSpike(x, type) <- value

## S4 replacement method for signature 'SingleCellExperiment,missing'
isSpike(x, type) <- value

## S4 replacement method for signature 'SingleCellExperiment,NULL'
isSpike(x, type) <- value

## S4 method for signature 'SingleCellExperiment'
spikeNames(x)
```

Arguments

x	A SingleCellExperiment object.
type	A string containing the name of the spike-in set.
value	A vector indicating which rows correspond to spike-in transcripts.

Details

Spike-in transcripts may be added during library preparation in single-cell RNA sequencing experiments. These usually need to be handled differently during data analysis, compared to the endogenous genes. Thus, it is important to indicate which rows correspond to spike-in transcripts.

The `isSpike<-` method accepts any `value` that indicates which rows correspond to spike-ins. This can be a logical or integer subsetting vector, or a vector of row names. The `type` should be set to the name of the spike-in set, e.g., `"ERCC"` or `"SIRV"`.

In this manner, multiple types of spike-in sets are supported for a single experiment. The names of all available spike-in sets can be obtained using `spikeNames`. To remove spike-ins for a particular set, `value` should be set to `NULL` when using `isSpike<-`.

If `type` is missing or `NULL` for `isSpike<-`, all existing spike-in sets are removed. If `value` is non-`NULL` in such cases, it will be stored in `x` with an empty name. Otherwise, this will have the effect of removing all spike-in sets from `x`.

The `isSpike` getter methods will return a logical vector indicatng which rows represent spike-ins of the set specified by `type`. If `type` is missing or `NULL`, the vector will instead indicate whether each row is in *any* spike-in set. If `type` is specified but not available, an error will be raised.

Value

For `isSpike`, a logical vector is returned indicating whether each row is in the specified set `type` or any set.

For `isSpike<-`, a SingleCellExperiment is returned with spike-in information stored in the internal metadata fields.

For `spikeNames`, a character vector is returned containing the names of available spike-in sets.

Author(s)

Aaron Lun

See Also

SingleCellExperiment-class

Examples

```
example(SingleCellExperiment, echo=FALSE) # Using the class example
isSpike(sce, "ERCC") <- 1:10
isSpike(sce)

isSpike(sce, "SIRV") <- 11:20
spikeNames(sce)
which(isSpike(sce))
which(isSpike(sce, "SIRV"))

isSpike(sce, "ERCC") <- NULL
spikeNames(sce)
```

Subsetting methods *Subsetting methods*

Description

Methods to subset SingleCellExperiment objects.

Usage

```
## S4 method for signature 'SingleCellExperiment,ANY,ANY'
x[i, j, ..., drop=TRUE]

## S4 method for signature 'SingleCellExperiment'
subset(x, i, j)

## S4 replacement method for signature 'SingleCellExperiment,ANY,ANY,SingleCellExperiment'
x[i, j] <- value
```

Arguments

x	A SingleCellExperiment object.
i, j	A vector of logical or integer subscripts, indicating the rows and columns to be subsetted for i and j, respectively.
...	Extra arguments to be passed to [,SummarizedExperiment-method.
drop	A logical scalar that is ignored.
value	A SingleCellExperiment object with number of rows equal to length of i (or that of x, if i is not specified). The number of columns must be equal to the length of j (or number of columns in x, if j is not specified).

Details

Subsetting yields a SingleCellExperiment object containing the specified rows (features) and columns (cells). Internal row and column metadata fields will also be subsetted so that methods such as `isSpike` are still valid. If column subsetting is performed, values of the `reducedDims` will be modified to retain only the selected cells.

Subset assignment will replace the assay values and metadata of the specified rows or columns in `x` with those in `value`. If both `i` and `j` are set, the relevant block of assay values will be replaced, along with the metadata for the affected rows and columns. If neither `i` or `j` are set, `x` will be turned into `value`.

Value

For `[` and `subset`, a subsetted SingleCellExperiment object is returned.

For `[<-`, a modified SingleCellExperiment object is returned.

Author(s)

Aaron Lun

See Also

`SingleCellExperiment-class`

Examples

```
example(SingleCellExperiment, echo=FALSE) # using the class example

sce[1:10,]
sce[,1:5]

sce2 <- sce
sce2[1:10,] <- sce[11:20,]

# Can also use subset()
subset(sce, 1, 1)

# Can also use split()
split(sce, sample(LETTERS, nrow(sce), replace=TRUE))
```

Index