Package 'SCArray'

January 26, 2023

Type Package

Title Large-scale single-cell RNA-seq data manipulation with GDS files

Version 1.7.5

Date 2023-01-21

Depends R (>= 3.5.0), gdsfmt (>= 1.35.4), methods, DelayedArray (>= 0.16.0)

Imports S4Vectors, SummarizedExperiment, SingleCellExperiment, DelayedMatrixStats, BiocParallel, BiocSingular, utils

Suggests BiocGenerics, Matrix, scater, uwot, RUnit, knitr, markdown, rmarkdown, rhdf5, HDF5Array

Description Provides large-scale single-cell RNA-seq data manipulation using Genomic Data Structure (GDS) files. It combines dense and sparse matrices stored in GDS files and the Bioconductor infrastructure framework (SingleCellExperiment and DelayedArray) to provide out-of-memory data storage and large-scale manipulation using the R programming language.

License GPL-3

VignetteBuilder knitr

ByteCompile TRUE

URL https://github.com/AbbVie-ComputationalGenomics/SCArray

biocViews Infrastructure, DataRepresentation, DataImport, SingleCell, RNASeq

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

git_branch master

git_last_commit 5d8f930

git_last_commit_date 2023-01-25

Date/Publication 2023-01-26

Author Xiuwen Zheng [aut, cre] (<https://orcid.org/0000-0002-1390-0708>)

Maintainer Xiuwen Zheng <xiuwen.zheng@abbvie.com>

R topics documented:

`SCArray-package` *Large-scale single-cell RNA-seq data manipulation with GDS files*

Description

The package combines dense/sparse matrices stored in GDS files and the Bioconductor infrastructure framework to provide out-of-memory data storage and manipulation using the R programming language.

Details

Package:	SCArray
Type:	Package
License:	GPL version 3

Author(s)

Xiuwen Zheng <xiuwen.zheng@abbvie.com>

Examples

```
# a GDS file for SingleCellExperiment
fn <- system.file("extdata", "example.gds", package="SCArray")

sce <- scExperiment(fn)
sce

rm(sce)
```

`scArray` *Get an DelayedArray instance*

Description

Gets an DelayedArray instance from a single-cell GDS file.

Usage

```
scArray(gdsfile, varname)
```

Arguments

`gdsfile`	character for a file name, or a single-cell GDS object with class `SCArrayFileClass`
`varname`	character for the node name in the GDS file

Value

Return an object of class `DelayedArray`.

Author(s)

Xiuwen Zheng

See Also

`scOpen`, `scExperiment`

Examples

```
# a GDS file for SingleCellExperiment
fn <- system.file("extdata", "example.gds", package="SCArray")

cnt <- scArray(fn, "counts")
cnt

rm(cnt)
```

SCArray-classes *Class list defined in SCArray*

Description

The package combines dense/sparse matrices stored in GDS files and the Bioconductor infrastructure framework to provide out-of-memory data storage and manipulation using the R programming language.

Author(s)

Xiuwen Zheng <xiuwen.zheng@abbvie.com>

SCArray-stats *SC_GDSMatrix row/column summarization*

Description

The row/column summarization methods for the SC_GDSMatrix matrix, extending the S4 methods in the **DelayedArray** and **DelayedMatrixStats** packages.

Usage

```
## S4 method for signature 'SC_GDSMatrix'
rowSums(x, na.rm=FALSE, dims=1)
## S4 method for signature 'SC_GDSMatrix'
colSums(x, na.rm=FALSE, dims=1)
## S4 method for signature 'SC_GDSMatrix'
rowSums2(x, rows=NULL, cols=NULL, na.rm=FALSE, ..., useNames=NA)
## S4 method for signature 'SC_GDSMatrix'
colSums2(x, rows=NULL, cols=NULL, na.rm=FALSE, ..., useNames=NA)

## S4 method for signature 'SC_GDSMatrix'
rowProds(x, rows=NULL, cols=NULL, na.rm=FALSE, ..., useNames=NA)
## S4 method for signature 'SC_GDSMatrix'
colProds(x, rows=NULL, cols=NULL, na.rm=FALSE, ..., useNames=NA)

## S4 method for signature 'SC_GDSMatrix'
rowMeans(x, na.rm=FALSE, dims=1)
## S4 method for signature 'SC_GDSMatrix'
colMeans(x, na.rm=FALSE, dims=1)
## S4 method for signature 'SC_GDSMatrix'
rowMeans2(x, rows=NULL, cols=NULL, na.rm=FALSE, ..., useNames=NA)
## S4 method for signature 'SC_GDSMatrix'
colMeans2(x, rows=NULL, cols=NULL, na.rm=FALSE, ..., useNames=NA)
```

```
## S4 method for signature 'SC_GDSMatrix'
rowWeightedMeans(x, w=NULL, rows=NULL, cols=NULL, na.rm=FALSE, ..., useNames=NA)
## S4 method for signature 'SC_GDSMatrix'
colWeightedMeans(x, w=NULL, rows=NULL, cols=NULL, na.rm=FALSE, ..., useNames=NA)

## S4 method for signature 'SC_GDSMatrix'
rowVars(x, rows=NULL, cols=NULL, na.rm=FALSE, center=NULL, ..., useNames=NA)
## S4 method for signature 'SC_GDSMatrix'
colVars(x, rows=NULL, cols=NULL, na.rm=FALSE, center=NULL, ..., useNames=NA)
## S4 method for signature 'SC_GDSMatrix'
rowWeightedVars(x, w=NULL, rows=NULL, cols=NULL, na.rm=FALSE, ..., useNames=NA)
## S4 method for signature 'SC_GDSMatrix'
colWeightedVars(x, w=NULL, rows=NULL, cols=NULL, na.rm=FALSE, ..., useNames=NA)

## S4 method for signature 'SC_GDSMatrix'
rowSds(x, rows=NULL, cols=NULL, na.rm=FALSE, center=NULL, ..., useNames=NA)
## S4 method for signature 'SC_GDSMatrix'
colSds(x, rows=NULL, cols=NULL, na.rm=FALSE, center=NULL, ..., useNames=NA)
## S4 method for signature 'SC_GDSMatrix'
rowWeightedSds(x, w=NULL, rows=NULL, cols=NULL, na.rm=FALSE, ..., useNames=NA)
## S4 method for signature 'SC_GDSMatrix'
colWeightedSds(x, w=NULL, rows=NULL, cols=NULL, na.rm=FALSE, ..., useNames=NA)

## S4 method for signature 'SC_GDSMatrix'
rowMins(x, rows=NULL, cols=NULL, na.rm=FALSE)
## S4 method for signature 'SC_GDSMatrix'
colMins(x, rows=NULL, cols=NULL, na.rm=FALSE)
## S4 method for signature 'SC_GDSMatrix'
rowMaxs(x, rows=NULL, cols=NULL, na.rm=FALSE)
## S4 method for signature 'SC_GDSMatrix'
colMaxs(x, rows=NULL, cols=NULL, na.rm=FALSE)
## S4 method for signature 'SC_GDSMatrix'
rowRanges(x, rows=NULL, cols=NULL, na.rm=FALSE)
## S4 method for signature 'SC_GDSMatrix'
colRanges(x, rows=NULL, cols=NULL, na.rm=FALSE)

# Get means and variances together for each row or column,
#     return a matrix with two columns for mean and variance
scRowMeanVar(x, na.rm=FALSE, useNames=FALSE)
scColMeanVar(x, na.rm=FALSE, useNames=FALSE)

## S4 method for signature 'SC_GDSMatrix'
rowCollapse(x, idxs, rows=NULL, ..., useNames=NA)
## S4 method for signature 'SC_GDSMatrix'
colCollapse(x, idxs, cols=NULL, ..., useNames=NA)
```

Arguments

x	A SC_GDSMatrix object (inherited from DelayedMatrix)

dims	not used, it should be 1
rows, cols	specify the subset of rows (and/or columns) to operate over; if NULL, no subsetting
na.rm	if TRUE, missing values (NaN and NA) will be removed
w	NULL or a numeric vector for weights
center	NULL, or a vector of pre-calculated row (column) means
useNames	if TRUE, the name attributes of result are set
idxs	An index vector specifying the columns (rows) to be extracted; the vector will be reused if the length is less than the number of columns or rows
...	additional arguments passed to specific methods

Details

All these operations are block-processed according to the data stored in the GDS file.

Author(s)

Xiuwen Zheng

See Also

- The **DelayedMatrixStats** package for more row/column summarization methods for DelayedMatrix objects.
- DelayedMatrix-utils for other common operations on DelayedMatrix objects.
- DelayedMatrix objects.
- matrix objects in base R.

SCArray-utils *SC_GDSArray subsetting, Ops, Math*

Description

Subsetting, Arith, Compare, Logic and Math operations on the SC_GDSArray object.

Usage

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

## S4 method for signature 'SC_GDSArray'
```

```
Ops(e1, e2)
## S4 method for signature 'SC_GDSArray'
Math(x)

# names(x) <- value
# dimnames(x) <- value
```

Arguments

x	A SC_GDSArray or SC_GDSMatrix object
i, j, ...	indices specifying elements to extract
drop	if TRUE the result will be coerced to the lowest possible dimension
e1, e2	objects
value	NULL, a character vector for names<- or a list of character vectors for dimnames<-

Details

All these operations return a SC_GDSArray or SC_GDSMatrix object.

Arith: "+", "-", "*", "^", "%%", "%/%", "/"

Compare: "==", ">", "<", "!=", "<=", ">="

Logic: "&", "|".

Ops: "Arith", "Compare", "Logic"

Math: "abs", "sign", "sqrt", "ceiling", "floor", "trunc", "cummax", "cummin", "cumprod", "cumsum", "log", "log10", "log2", "log1p", "acos", "acosh", "asin", "asinh", "atan", "atanh", "exp", "expm1", "cos", "cosh", "cospi", "sin", "sinh", "sinpi", "tan", "tanh", "tanpi", "gamma", "lgamma", "digamma", "trigamma"

Value

All these operations return a SC_GDSArray or SC_GDSMatrix object.

Author(s)

Xiuwen Zheng

See Also

Ops, Math, SCArray-stats

Examples

```
1
```

scConvGDS *Create a GDS file*

Description

Creates a single-cell GDS file from an R object.

Usage

```
scConvGDS(obj, outfn, save.sp=TRUE, type=c("float32", "float64", "int32"),
    compress="LZMA_RA", clean=TRUE, verbose=TRUE)
```

Arguments

obj	a dense/sparse matrix, DelayedMatrix, SummarizedExperiment or SingleCellExperiment
outfn	the output file name in GDS format
save.sp	if TRUE, save it to a sparse matrix in GDS; otherwise, store dense matrix
type	numeric data type in the output file
compress	the compression method, see add.gdsn
clean	TRUE
verbose	if TRUE, show information

Value

Return the path of the output file.

Author(s)

Xiuwen Zheng

See Also

scOpen, scClose, scMEX2GDS, scHDF2GDS

Examples

```
# load a SingleCellExperiment object
fn <- system.file("extdata", "example.rds", package="SCArray")
sce <- readRDS(fn)
sce

scConvGDS(sce, "test.gds")

# remove the temporary file
unlink("test.gds")
```

scExperiment *Get a SummarizedExperiment*

Description

Gets an instance of SingleCellExperiment or SummarizedExperiment.

Usage

```
scExperiment(gdsfile, sce=TRUE, use.names=TRUE, load.row=TRUE, load.col=TRUE)
```

Arguments

gdsfile	character for a file name, or a single-cell GDS object with class SCArrayFileClass
sce	if TRUE, return an instance of SingleCellExperiment, otherwise an instance of SummarizedExperiment
use.names	if TRUE, load dimnames from 'feature.id' and 'sample.id'
load.row	TRUE for loading rowData from the gds node "feature.data" in gdsfile
load.col	TRUE for loading colData from the gds node "sample.data" in gdsfile

Value

Return an instance of SingleCellExperiment or SummarizedExperiment.

Author(s)

Xiuwen Zheng

See Also

scOpen, scClose

Examples

```
# a GDS file for SingleCellExperiment
fn <- system.file("extdata", "example.gds", package="SCArray")

sce <- scExperiment(fn)
sce

remove(sce)
```

scHDF2GDS *Convert HDF5 files to GDS*

Description

Creates a single-cell GDS file from Cell Ranger HDF5 files.

Usage

```
scHDF2GDS(h5_fn, outfn, group=c("matrix", "mm10"), feature_path=character(),
    type=c("float32", "float64", "int32"), compress="LZMA_RA", clean=TRUE,
    verbose=TRUE)
```

Arguments

h5_fn	the input HDF5 file name
outfn	the output file name in GDS format
group	the name of the group in the HDF5 file where the sparse matrix is stored; if there are more than one group names, the first existing group in the HDF5 file is used; "mm10" is usually used for 10x Genomics datasets
feature_path	a character vector for feature variables, otherwise detecting automatically
type	numeric data type in the output file
compress	the compression method, see add.gdsn
clean	TRUE
verbose	if TRUE, show information

Details

The packages **rhdf5** and **HDF5Array** should be installed.

Value

Return the path of the output file.

Author(s)

Xiuwen Zheng

See Also

scConvGDS, scMEX2GDS

scMEX2GDS *Convert MEX files to GDS*

Description

Creates a single-cell GDS file from Cell Ranger MEX files.

Usage

```
scMEX2GDS(feature_fn, barcode_fn, mtx_fn, outfn,
    feature_colnm=c("id", "gene", "feature_type"),
    type=c("float32", "float64", "int32"), compress="LZMA_RA", clean=TRUE,
    verbose=TRUE)
```

Arguments

feature_fn	the input file name for features
barcode_fn	the input file name for barcodes
mtx_fn	the input count matrix in MEX format
outfn	the output file name in GDS format
feature_colnm	the column names used in feature_fn
type	numeric data type in the output file
compress	the compression method, see add.gdsn
clean	TRUE
verbose	if TRUE, show information

Value

Return the path of the output file.

Author(s)

Xiuwen Zheng

See Also

scConvGDS, scHDF2GDS

scObj *DelayedArray Object in GDS*

Description

Convert to SC_GDSArray/SC_GDSMatrix for utilizing GDS specific functions.

Usage

```
scObj(obj, verbose=FALSE)
```

Arguments

obj	a SummarizedExperiment, SingleCellExperiment or DelayedArray object
verbose	if TRUE, show information

Value

Return the object obj with the object class DelayedArray replaced by the class SC_GDSMatrix or SC_GDSArray.

Author(s)

Xiuwen Zheng

See Also

scArray, scExperiment

scOpen *Open/Close a Single-cell GDS File*

Description

Opens or closes a single-cell GDS file.

Usage

```
scOpen(gdsfn, readonly=TRUE, allow.duplicate=TRUE)
scClose(gdsfile)
```

Arguments

gdsfn	the input file name
readonly	whether read-only or not
allow.duplicate	if TRUE, it is allowed to open a GDS file with read-only mode when it has been opened in the same R session
gdsfile	a single-cell GDS object with class SCArrayFileClass

Value

Return an object of class SCArrayFileClass inherited from gds.class.

Author(s)

Xiuwen Zheng

See Also

scArray

Examples

```
# a GDS file for SingleCellExperiment
fn <- system.file("extdata", "example.gds", package="SCArray")

# open the GDS file
(f <- scOpen(fn))

# read a GDS file
cell.id <- read.gdsn(index.gdsn(f, "feature.id"))
samp.id <- read.gdsn(index.gdsn(f, "sample.id"))

# get a DelayedArray object
(cnt <- scArray(f, "counts"))

scClose(f)
```

---

scReplaceNA *Replacement*

Description

Replace NA/NaN in a GDS-specific DelayedArray by a specified value.

Usage

```
scReplaceNA(x, v=0L)
```

Arguments

x	a SC_GDSArray object
v	a length-one double or integer value

Value

Return an object with the class SC_GDSMatrix or SC_GDSArray.

Author(s)

Xiuwen Zheng

See Also

scSetMin, scSetMax, scSetBounds

Examples

```
suppressPackageStartupMessages(library(DelayedArray))

m <- matrix(1:12, nrow=3)
m[2, c(1,3)] <- NA
(mat <- DelayedArray(m))

new_m <-  scObj(mat)  # wrap a in-memory DelayedMatrix
class(new_m)  # SC_GDSMatrix

scReplaceNA(new_m,  999)
```

---

scRunPCA *Perform PCA on expression data*

Description

Perform a Principal Components Analysis (PCA) on cells in the SingleCellExperiment object.

Usage

```
scRunPCA(x, ncomponents=50, ntop=500, subset_row=NULL, scale=FALSE,
    altexp=NULL, name="PCA", exprs_values="logcounts", dimred=NULL,
    n_dimred=NULL, BSPARAM=NULL, BPPARAM=SerialParam(), verbose=TRUE)
```

Arguments

x	a SingleCellExperiment or SummarizedExperiment object
ncomponents	# of calculated principal components
ntop	# of features with the highest variances to use for PCA
subset_row	specifying the subset of features to use
scale	if TRUE, expression values will be be standardized
altexp	String or integer scalar specifying an alternative experiment containing the input data
name	the name to be used to store the result in reducedDims
exprs_values	the assay name containing the expression values
dimred	String or integer scalar specifying the existing dimensionality reduction results to use
n_dimred	Integer scalar or vector specifying the dimensions to use if dimred is specified
BSPARAM	A BiocSingularParam object specifying which algorithm to be used in runPCA in the BiocSingular package
BPPARAM	A BiocParallelParam object for parallelized calculation
verbose	if TRUE, show information

Value

Return an object with the class SC_GDSMatrix or SC_GDSArray.

Author(s)

Xiuwen Zheng

See Also

scExperiment

Examples

```
1
```

---

scSetBounds *Set the bounds*

Description

Set the maximum and/or minimum on a GDS-specific DelayedArray.

Usage

```
scSetMax(x, vmax)
scSetMin(x, vmin)
scSetBounds(x, vmin=NaN, vmax=NaN)
```

Arguments

x	a SC_GDSArray object
vmax	maximum, length-one
vmin	minimum, length-one

Value

Return an object with the class SC_GDSMatrix or SC_GDSArray.

Author(s)

Xiuwen Zheng

See Also

scReplaceNA

Examples

```
suppressPackageStartupMessages(library(DelayedArray))

m <- matrix(1:12, nrow=3)
(mat <- DelayedArray(m))

new_m <-  scObj(mat)  # wrap a in-memory DelayedMatrix
class(new_m)  # SC_GDSMatrix

scSetMax(new_m, 5)
scSetMin(new_m, 5)
scSetBounds(new_m, 4, 9)
```

Index