Package ‘scRNAseqApp’

May 30, 2024

Title A single-cell RNAseq Shiny app-package

Version 1.4.0

Description The scRNAseqApp is a Shiny app package designed for interactive visualization of single-cell data. It is an enhanced version derived from the ShinyCell, repackaged to accommodate multiple datasets. The app enables users to visualize data containing various types of information simultaneously, facilitating comprehensive analysis. Additionally, it includes a user management system to regulate database accessibility for different users.

License GPL-3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

VignetteBuilder knitr

biocViews Visualization, SingleCell, RNASeq

Depends R (>= 4.3.0)

Imports bibtex, bslib, circlize, ComplexHeatmap, data.table, DBI, DT, GenomicRanges, GenomeInfoDb, ggdendro, ggforce, ggplot2, ggrepel, ggridges, grDevices, grid, gridExtra, htmltools, IRRanges, jsonlite, magrittr, methods, patchwork, plotly, RColorBrewer, RefManageR, rhdf5, Rsamtools, RSQLite, rtracklayer, S4Vectors, scales, scrypt, Seurat, SeuratObject, shiny, shinyhelper, shinymanager, slingshot, SingleCellExperiment, sortable, stats, tools, xfun, xml2, utils

Suggests rmarkdown, knitr, testthat, BiocStyle

Enhances celldex, future, SingleR, SummarizedExperiment, tricycle

URL https://github.com/jianhong/scRNAseqApp

BugReports https://github.com/jianhong/scRNAseqApp/issues

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

git_branch RELEASE_3_19
APPconf-class

Description

An object of class "APPconf" represents the metadata for a dataset.

Usage

APPconf(...)

Arguments

... Each argument in ... becomes an slot in the new "APPconf"-class.

Value

A APPconf object.
**Slots**

- **title** character(1). Title of the data
- **id** character(1). Folder name of the data
- **species** character(1). species
- **ref** Reference information in a list with element bib, doi, pmid and entry. Entry must be an object of `bibentry`
- **type** character(1). Type of the data, scRNAseq or scATACseq.
- **markers** list. A list of data.frame represents cell markers.
- **keywords** character. A vector of characters represents the keywords of the study.
- **groupCol** character. The key group column name to separate the cells.

**Examples**

```r
appconf <- readRDS(system.file("extdata", "data", "pbmc_small", "appconf.rds", package="scRNAseqApp"))
appconf
```

---

**Description**

The assessment and replacement methods for `APPconf-class`

**Usage**

```r
## S4 method for signature 'APPconf'
show(object)

## S4 method for signature 'APPconf'
x$name

## S4 replacement method for signature 'APPconf'
x$name <- value

## S4 method for signature 'APPconf,ANY,ANY'
x[[i, j, ..., exact = TRUE]]

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

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

## S4 method for signature 'APPconf'
```

---

**APPconf-methods**

*The methods for APPconf-class*
as.list(x, ...)

## S4 method for signature 'APPconf'
as.character(x, ...)

## S4 method for signature 'APPconf'
makers(x)

## S4 method for signature 'APPconf'
lapply(X, FUN, ...)

## S4 method for signature 'APPconf'
unlist(x, recursive = TRUE, use.names = TRUE)

Arguments

- **object**: an object of APPconf
- **x**: APPconf object.
- **name**: A literal character string or a name (possibly backtick quoted).
- **value**: value to replace.
- **i, j**: indices specifying elements to extract or replace.
- **...**: Named or unnamed arguments to form a signature.
- **exact**: see Extract
- **drop**: see drop
- **X**: an APPconf object.
- **FUN**: function used by lapply
- **recursive, use.names**: function used by unlist

Value

A named character vector.

Examples

```r
appconf <- readRDS(system.file("extdata", "data", 
    "pbmc_small", "appconf.rds", package="scRNAseqApp"))
appconf
appconf$title
appconf["title"]
as.list(appconf)
as.character(appconf)
makers(appconf)
lapply(appconf, print)
lapply(appconf, print)
unlist(appconf)
```
createAppConfig

Create a metadata to describe the dataset

Description

The function will return a APPconf object which contain the reference, keywords for the dataset.

Usage

createAppConfig(
  title, destinationFolder, species, doi, pmid, bibentry,
  datatype = c("scRNAseq", "scATACseq", "scMultiome"), markers,
  keywords, abstract
)

Arguments

title The title of the dataset
destinationFolder The destination folder name of the dataset without the root folder of the datasets. The data will be saved as appdataFolder/destinationFolder
species The species of the dataset
doi, pmid The DOI or PMID of the reference
bibentry An object of bibentry
datatype character(1). Type of the data, scRNAseq, scATACseq or scMultiome.
markers A list of data.frame with gene symbols as rownames or a character vector.
keywords The keywords for the dataset. For example the condition, cell type, tissue information The keywords will be used for whole database search
abstract The abstract of the reference.

Value

An object of APPconf object
createDataSet

Create a dataset Create a dataset from a Seurat object. The function will try to find the markers in the Misc data named as `markers`. The misc data should be output of function `FindAllMarkers`.

**Description**

Create a dataset Create a dataset from a Seurat object. The function will try to find the markers in the Misc data named as `markers`. The misc data should be output of function `FindAllMarkers`.

**Usage**

```r
createDataSet(
  appconf,
  seu,
  config,
  contrast,
  assayName,
  gexSlot = c("data", "scale.data", "counts"),
  atacAssayName,
  atacSlot = c("data", "scale.data", "counts"),
  LOCKER = FALSE,
  datafolder = "data",
  default.symbol = "rownames"
)
```

**Arguments**

- `appconf` a list object represent the information about the dataset
- `seu` a Seurat object
- `config` config file for `makeShinyFiles`
- `contrast` The contrast group
- `assayName` assay in single-cell data object to use for plotting gene expression, which must match one of the following:
  - Seurat objects: "RNA" or "integrated" assay, default is "RNA"
createSeuFromCellRanger

Description
load data from cellRanger

Usage
createSeuFromCellRanger(outsFolder)

Arguments
outsFolder the outs folder of cellRanger

Value
An SeuratObject

.examples
library(Seurat)
if(interactive()){
  appconf <- createAppConfig(
    title="pbmc_small",
    destinationFolder = "pbmc_small",
    species = "Homo sapiens",
    doi="10.1038/nbt.3192",
    datatype = "scRNAseq"
  )
  createDataSet(appconf, pbmc_small, datafolder=tempdir())
}

createSeuFromCellRanger
    load data from cellRanger

Value
The updated Seurat object.

Examples
library(Seurat)
if(interactive()){
  appconf <- createAppConfig(
    title="pbmc_small",
    destinationFolder = "pbmc_small",
    species = "Homo sapiens",
    doi="10.1038/nbt.3192",
    datatype = "scRNAseq"
  )
  createDataSet(appconf, pbmc_small, datafolder=tempdir())
}

createSeuFromCellRanger
    load data from cellRanger

Value
The updated Seurat object.

Examples
library(Seurat)
if(interactive()){
  appconf <- createAppConfig(
    title="pbmc_small",
    destinationFolder = "pbmc_small",
    species = "Homo sapiens",
    doi="10.1038/nbt.3192",
    datatype = "scRNAseq"
  )
  createDataSet(appconf, pbmc_small, datafolder=tempdir())
}

createSeuFromCellRanger
    load data from cellRanger

Value
An SeuratObject
createSeuFromMatrix  
**load data from a count matrix**

**Description**

load data from a count matrix

**Usage**

createSeuFromMatrix(matrix, meta, genes, cluster, ...)

**Arguments**

- **matrix**: count matrix
- **meta**: cell-level meta data
- **genes**: character. gene names, will be the rownames of the matrix
- **cluster**: the cluster coordinates
- **...**: The parameter passed to read.delim when read cluster file.

**Value**

An SeuratObject

---

**scInit**  
*Create a scRNAseqApp project*

**Description**

To run scRNAseqApp, you need to first create a directory which contains the required files.

**Usage**

```r
scInit(
    app_path = getwd(),
    root = "admin",
    password = "scRNAseqApp",
    datafolder = "data",
    overwrite = FALSE,
    app_title = "scRNAseq Database",
    app_description = 
        "This database is a collection of single cell RNA-seq data.",
    passphrase = NULL
)
```
**Arguments**

- `app_path` path, a directory where you want to create the app
- `root` character(1), the user name for administrator
- `password` character(1), the password for administrator
- `datafolder` the folder where saved the dataset for the app
- `overwrite` logical(1), overwrite the `app_path` if there is a project.
- `app_title`, `app_description` character(1). The title and description of the home page.
- `passphrase` A password to protect the data inside the database.

**Value**

no returns. This function will copy files to `app_path`

**Examples**

```r
if(interactive()){
  scInit()
  scRNAseqApp()
}
```

---

**scRNAseqApp** *scRNAseqApp main function*

**Description**

create a scRNAseqApp once the initialization is done.

**Usage**

```r
scRNAseqApp(
  app_path = getwd(),
  datafolder = "data",
  defaultDataset = "pbmc_small",
  windowTitle = "scRNAseq/scATACseq database",
  banner = system.file("assets", "img", "banner.png", package = "scRNAseqApp"),
  footer = tagList(HTML("©"), "2020 - ", format(Sys.Date(), "%Y"), "jianhong@duke"),
  maxRequestSize = 1073741824,
  timeout = 30,
  theme = bs_theme(bootwatch = "lumen"),
  use_bs_themer = FALSE,
  ...
)
```
Arguments

- **app_path**: path, a directory where do you want to create the app
- **datafolder**: the folder where saved the dataset for the app
- **defaultDataset**: default dataset for the app.
- **windowTitle**: The title that should be displayed by the browser window.
- **banner**: The banner image.
- **footer**: The footer html contents.
- **maxRequestSize**: Maximal upload file size. Default is 1G.
- **timeout**: Timeout session (minutes) before logout if sleeping. Default to 30. 0 to disable.
- **theme**: A theme.
- **use_bs_themer**: logical(1). Used to determine the theme.
- **...**: parameters can be passed to shinyApp except ui and server.

Value

An object that represents the app.

Examples

```r
if(interactive()){
  app_path=tempdir()
  scInit(app_path=app_path)
  setwd(app_path)
  scRNAseqApp()
}
```
Index

* APPconf
  APPconf-methods, 3
  [,APPconf, ANY, ANY, ANY-method
     (APPconf-methods), 3
  [[,APPconf, ANY, ANY-method
     (APPconf-methods), 3
  [[<-,APPconf, ANY, ANY, ANY-method
     (APPconf-methods), 3
  $,APPconf-method (APPconf-methods), 3
  $<-,APPconf-method (APPconf-methods), 3

  APPconf, 5
  APPconf (APPconf-class), 2
  APPconf-class, 2, 3
  APPconf-methods, 3
  as.character,APPconf-method
     (APPconf-methods), 3
  as.list,APPconf-method
     (APPconf-methods), 3

  bibentry, 3

  createAppConfig, 5
  createDataSt, 6
  createSeuFromCellRanger, 7
  createSeuFromMatrix, 8

  drop, 4

  Extract, 4

  lapply,APPconf-method
     (APPconf-methods), 3

  markers (APPconf-methods), 3
  markers,APPconf-method
     (APPconf-methods), 3

  scInit, 8
  scRNAseqApp, 9
  show,APPconf-method (APPconf-methods), 3

unlist, 4
unlist,APPconf-method
     (APPconf-methods), 3

unlist,APPconf-method
     (APPconf-methods), 3

11