

# Package ‘scRNAseqApp’

September 26, 2023

**Title** A single-cell RNAseq Shiny app-package

**Version** 1.0.1

**Description** scRNAseqApp is a Shiny app package that allows users to visualize single cell data interactively. It was modified from ShinyCell and repackaged to a tool to show multiple data.

It can visualize the data with multiple information side by side.

**License** GPL-3

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.3

**VignetteBuilder** knitr

**biocViews** Visualization, SingleCell, RNASeq

**Depends** R (>= 4.3.0)

**Imports** bibtext, bslib, circlize, ComplexHeatmap, data.table, DT, GenomicRanges, GenomeInfoDb, gg dendro, ggforce, ggplot2, ggrepel, ggridges, grDevices, grid, gridExtra, hdf5r, htmltools, IRanges, jsonlite, magrittr, methods, patchwork, plotly, RColorBrewer, RefManageR, Rsamtools, rtracklayer, S4Vectors, scales, sscript, Seurat, SeuratObject, shiny, shinyhelper, shinymanager, slingshot, SingleCellExperiment, 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\_17

**git\_last\_commit** f30cd71

**git\_last\_commit\_date** 2023-04-25

**Date/Publication** 2023-09-25

**Author** Jianhong Ou [aut, cre] (<<https://orcid.org/0000-0002-8652-2488>>)

**Maintainer** Jianhong Ou <jianhong.ou@duke.edu>

## R topics documented:

APPconf-class . . . . .	2
APPconf-methods . . . . .	3
createAppConfig . . . . .	4
createDataSet . . . . .	6
createSeuFromCellRanger . . . . .	7
createSeuFromMatrix . . . . .	7
scInit . . . . .	8
scRNAseqApp . . . . .	9

<b>Index</b>	<b>11</b>
--------------	-----------

---

APPconf-class	Class "APPconf"
---------------	-----------------

---

### Description

Ano 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). Tolder 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

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

---

APPconf-methods

*The methods for [APPconf-class](#)*

---

## Description

The assessment and replacement methods for [APPconf-class](#)

## Usage

```
## 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,ANY'
x[[i, j, ...]] <- value

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

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

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

## S4 method for signature 'APPconf'
markers(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 <a href="#">Extract</a>
drop	see <a href="#">drop</a>
X	an APPconf object.
FUN	function used by <code>lapply</code>
recursive, use.names	function used by <a href="#">unlist</a>

**Value**

A named character vector.

**Examples**

```
appconf <- readRDS(system.file("extdata", "data",
  "pbmc_small", "appconf.rds", package="scRNAseqApp"))
appconf
appconf$title
appconf[["title"]]
as.list(appconf)
as.character(appconf)
markers(appconf)
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  
)
```

**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

**Value**

An object of [APPconf](#) object

**Examples**

```
config <- createAppConfig(  
  title="pbmc_small",  
  destinationFolder = "pbmc_small",  
  species = "Homo sapiens",  
  doi="10.1038/nbt.3192",  
  datatype = "scRNAseq")
```

---

createDataSet	<i>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.</i>
---------------	--

---

## 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

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

## 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: <ul style="list-style-type: none"> <li>Seurat objects: "RNA" or "integrated" assay, default is "RNA"</li> </ul>
gexSlot	slot in single-cell assay to plot. Default is to use the "data" slot
atacAssayName	assay in single-cell data object to use for plotting open chromatin.
atacSlot	slot in single-cell atac assay to plot. Default is to use the "data" slot
LOCKER	Set locker if the file is required login
datafolder	app data folder

## Value

The updated Seurat object.

## Examples

```
library(Seurat)
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
```

---

## Description

load data from cellRanger

## Usage

```
createSeuFromCellRanger(outsFolder)
```

## Arguments

outsFolder      the outs folder of 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	<i>Create a scRNAseqApp project</i>
--------	-------------------------------------

---

**Description**

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

**Usage**

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

**Arguments**

app_path	path, a directory where do 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.

**Value**

no returns. This function will copy files to app\_path



**Examples**

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

---

scRNAseqApp

*scRNAseqApp main function*


---

**Description**

create a scRNAseqApp once the initialization is done.

**Usage**

```
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("&copy;"), "2020 -", format(Sys.Date(), "%Y"), "jianhong@duke"),
  maxRequestSize = 1073741824,
  timeout = 30,
  theme = bs_theme(bootswatch = "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**

```
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, 2
  
- createAppConfig, 4
- createDataSet, 6
- createSeuFromCellRanger, 7
- createSeuFromMatrix, 7
  
- 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