

Package ‘TooManyCellsR’

March 5, 2019

Title An R Wrapper for 'TooManyCells'

Version 0.1.0.0

Author Gregory W. Schwartz

Maintainer Gregory W. Schwartz <gsch@pennmedicine.upenn.edu>

Description An R wrapper for using 'TooManyCells', a command line program for clustering, visualizing, and quantifying cell clade relationships. See <<https://gregoryschwartz.github.io/too-many-cells/>> for more details.

Imports Matrix, imager, ggplot2, cowplot, jsonlite

SystemRequirements 'TooManyCells'
(<https://github.com/GregorySchwartz/too-many-cells>)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2019-03-05 11:20:03 UTC

R topics documented:

importResults	2
tooManyCells	3
tryFunc	4
writeMatrixFiles	5

Index	6
--------------	----------

importResults

Import some 'too-many-cells make-tree' outputs into a data frame.

Description

This function will import some of the files resulting from a 'too-many-cells make-tree' run into R as data frames. Does not include cluster list. Look at the main tooManyCells function for the cluster list.

Usage

```
importResults(dir = "out")
```

Arguments

dir The output directory of a 'too-many-cells' run.

Value

A list of each output. Reads the following files, see <https://gregoryschwartz.github.io/too-many-cells/> for more details: "dendrogram.svg", "clumpiness.pdf", "projection.pdf", "label_projection.pdf", "clumpiness.csv", "cluster_info.csv", "node_info.csv", and "cluster_diversity.csv".

Examples

```
input <- system.file("extdata", "mat.csv", package="TooManyCellsR")
inputLabels <- system.file("extdata", "labels.csv", package="TooManyCellsR")
df = read.csv(input, row.names = 1, header = TRUE)
mat = Matrix::Matrix(as.matrix(df), sparse = TRUE)
labelsDf = read.csv(inputLabels, header = TRUE)
# Here we draw this small toy example with no filter or normalization, and
# decrease the size of the branches and increase the size of the leaf nodes.
# With non-toy real world single cell data, these options should not be
# necessary.
## Not run:
tooManyCells( mat, labels = labelsDf
              , args = c( "make-tree"
                          , "--no-filter"
                          , "--normalization", "NoneNorm"
                          , "--draw-max-node-size", "40"
                          , "--draw-max-leaf-node-size", "70"
                          )
              )
res = importResults("out")
plot(res$treePlot, axes = FALSE)

## End(Not run)
```

tooManyCells	<i>Execute 'too-many-cells'.</i>
--------------	----------------------------------

Description

This function will run 'too-many-cells' on a Matrix. Requires 'TooManyCells' to be installed (follow instructions at <https://gregoryschwartz.github.io/too-many-cells/>).

Usage

```
tooManyCells(mat, args = c("make-tree"), labels = NULL,
             output = "out", prior = NULL)
```

Arguments

mat	The input Matrix with gene row names and cell barcode column names.
args	The arguments to give to the command line program. See https://gregoryschwartz.github.io/too-many-cells/ for more information. Defaults to "make-tree".
labels	The input labels data frame with item (cell barcodes) and label (whatever labels you want to give them, such as tissue of origin, celltype, etc.) columns. Optional.
output	The output folder for the 'too-many-cells' process. Defaults to "out".
prior	The location of the tree that was already made (previous 'too-many-cells' output) so quick visual or pruning changes can be made without remaking the tree (can potentially save hours).

Value

A list of each output, including the stdout. Reads the following files, see <https://gregoryschwartz.github.io/too-many-cells/> for more details: "dendrogram.svg", "clumpiness.pdf", "projection.pdf", "label_projection.pdf", "clumpiness.csv", "cluster_info.csv", "node_info.csv", and "cluster_diversity.csv".

Examples

```
input <- system.file("extdata", "mat.csv", package="TooManyCellsR")
inputLabels <- system.file("extdata", "labels.csv", package="TooManyCellsR")
df = read.csv(input, row.names = 1, header = TRUE)
mat = Matrix::Matrix(as.matrix(df), sparse = TRUE)
labelsDf = read.csv(inputLabels, header = TRUE)
# Here we draw this small toy example with no filter or normalization, and
# decrease the size of the branches and increase the size of the leaf nodes.
# With non-toy real world single cell data, these options should not be
# necessary.
## Not run:
res = tooManyCells( mat, labels = labelsDf
                   , args = c( "make-tree"
```

```

        , "--no-filter"
        , "--normalization", "NoneNorm"
        , "--draw-max-node-size", "40"
        , "--draw-max-leaf-node-size", "70"
    )
)
plot(res$treePlot, axes = FALSE)
res$stdout
res$nodeInfo

## End(Not run)

```

tryFunc

Different error for importing data.

Description

This function will fail gracefully instead of stopping the program with an error.

Usage

```
tryFunc(f, file)
```

Arguments

f	The function to use.
file	The input file to be read.

Value

The imported data frame or NULL if an error occurred.

Examples

```

input <- system.file("extdata", "mat.csv", package="TooManyCellsR")
fail = tryFunc(read.csv, "fail.csv")
fail
success = tryFunc(read.csv, input)
success

```

writeMatrixFiles	<i>Write a Matrix to a folder.</i>
------------------	------------------------------------

Description

This function will write a Matrix from the Matrix library to a temporary directory containing matrix.mtx, genes.tsv, barcodes.tsv, and optionally a labels.csv file.

Usage

```
writeMatrixFiles(mat, labels = NULL)
```

Arguments

mat	The input Matrix with gene row names and cell barcode column names.
labels	The input labels data frame with item (cell barcodes) and label (whatever labels you want to give them, such as tissue of origin, celltype, etc.) columns. Optional.

Value

None

Examples

```
input <- system.file("extdata", "mat.csv", package="TooManyCellsR")
df = read.csv(input, row.names = 1, header = TRUE)
mat = Matrix::Matrix(as.matrix(df), sparse = TRUE)
writeMatrixFiles(mat)
```

Index

`importResults`, 2

`tooManyCells`, 3

`tryFunc`, 4

`writeMatrixFiles`, 5