

Package: eds (via r-universe)

June 13, 2024

Title eds: Low-level reader for Alevin EDS format

Version 1.7.0

Description This packages provides a single function, readEDS. This is a low-level utility for reading in Alevin EDS format into R. This function is not designed for end-users but instead the package is predominantly for simplifying package dependency graph for other Bioconductor packages.

Depends Matrix

Imports Rcpp

Suggests knitr, tximportData, testthat (>= 3.0.0)

LinkingTo Rcpp

SystemRequirements C++11

License GPL-2

Encoding UTF-8

URL <https://github.com/mikelove/eds>

biocViews Sequencing, RNASeq, GeneExpression, SingleCell

VignetteBuilder knitr

LazyData true

RoxygenNote 7.1.2

Config/testthat/edition 3

Repository <https://bioc.r-universe.dev>

RemoteUrl <https://github.com/bioc/eds>

RemoteRef HEAD

RemoteSha 4b935eb5ebc5294ffbd8850b98dc09772472a7e5

Contents

readEDS	2
Index	3

readEDS *A low-level utility function for quickly reading in Alevin EDS format*

Description

This provides a simple utility for reading in EDS format. Note that most users will prefer to use `tximport` or `tximeta`. This function and package exist in order to simplify the dependency graph for other packages.

Usage

```
readEDS(numOfGenes, numOfOriginalCells, countMatFilename, tierImport = FALSE)
```

Arguments

<code>numOfGenes</code>	number of genes
<code>numOfOriginalCells</code>	number of cells
<code>countMatFilename</code>	pointer to the EDS file, <code>quants_mat.gz</code>
<code>tierImport</code>	whether the <code>countMatFilename</code> refers to a <code>quants</code> tier file

Value

a genes x cells sparse matrix, of the class `dgCMatrx`

Examples

```
# point to files
dir0 <- system.file("extdata", package="tximportData")
samps <- list.files(file.path(dir0, "alevin"))
dir <- file.path(dir0, "alevin", samps[3], "alevin")
quant.mat.file <- file.path(dir, "quants_mat.gz")
barcode.file <- file.path(dir, "quants_mat_rows.txt")
gene.file <- file.path(dir, "quants_mat_cols.txt")

# readEDS() requires knowing the number of cells and genes
cell.names <- readLines(barcode.file)
gene.names <- readLines(gene.file)
num.cells <- length(cell.names)
num.genes <- length(gene.names)

# reading in the sparse matrix
mat <- readEDS(numOfGenes=num.genes,
               numOfOriginalCells=num.cells,
               countMatFilename=quant.mat.file)
```

Index

readEDS, [2](#)