Package 'CTSV'

October 16, 2024

Type Package

Title Identification of cell-type-specific spatially variable genes accounting for excess zeros

Version 1.6.0

Description The R package CTSV implements the CTSV approach developed by Jinge Yu and Xiangyu Luo that detects cell-type-specific spatially variable genes accounting for excess zeros. CTSV directly models sparse raw count data through a zero-inflated negative binomial regression model, incorporates cell-type proportions, and performs hypothesis testing based on R package pscl. The package outputs p-values and q-values for genes in each cell type, and CTSV is scalable to datasets with tens of thousands of genes measured on hundreds of spots. CTSV can be installed in Windows, Linux, and Mac OS.

License GPL-3

Encoding UTF-8

RoxygenNote 7.2.0

Depends R (>= 4.2),

URL https://github.com/jingeyu/CTSV

BugReports https://github.com/jingeyu/CTSV/issues

Imports stats, pscl, qvalue, BiocParallel, methods, knitr, SpatialExperiment, SummarizedExperiment

Suggests testthat, BiocStyle

biocViews GeneExpression, StatisticalMethod, Regression, Spatial, Genetics

NeedsCompilation yes

VignetteBuilder knitr

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

git_branch RELEASE_3_19

git_last_commit f423f78

git_last_commit_date 2024-04-30

Repository Bioconductor 3.19

Date/Publication 2024-10-16

Author Jinge Yu Developer [aut, cre],
Xiangyu Luo Developer [aut]

Maintainer Jinge Yu Developer <yjgruc@ruc.edu.cn>

Contents

CTSV *Detection of cell-type-specific spatially variable genes*

Description

Detection of cell-type-specific spatially variable genes

Usage

```
CTSV(spe, W, num_core = 1, BPPARAM = NULL)
```

Arguments

spe	A SpatialExperiment class.
W	A n by K cell-type proportion matrix, where K is the number of cell types. The column names of W are cell type names.
num_core	Number of cores if using paralleling. The default is one.
BPPARAM	Optional additional argument for parallelization. The default is NULL, in which case num_core will be used. If provided, this should be an instance of BiocParallelParam. For most users, the recommended option is to use the num_core argument.

Value

A list with a G by 2K matrix of p-values and a G by 2K matrix of q-values.

pval	A G by 2K matrix of p-values. The first K columns correspond to the first coordinate, and the last K columns to the second coordinate.
qval	A G by 2K matrix of q-values. The first K columns correspond to the first coordinate, and the last K columns to the second coordinate.

Examples

```
library(CTSV)
#read example data
data(CTSVexample_data)
spe <- CTSVexample_data[[1]]
W <- CTSVexample_data[[2]]
gamma_true <- CTSVexample_data[[3]]
# gene number
G <- nrow(spe)
# spot number
n <- ncol(spe)
# cell type number
K <- ncol(W)
G
n
K
# SV genes in each cell type:
rownames(W)[which(gamma_true[,1] == 1)]
rownames(W)[which(gamma_true[,2] == 1)]
# Number of SV genes at the aggregated level:
sum(rowSums(gamma_true)>0)
#--- Run CTSV ----
result <- CTSV(spe,W,num_core = 8)
# View on q-value matrix
head(result$qval)
# detect SV genes
re <- svGene(result$qval,0.05)
#SV genes in each cell type:
re$SVGene
```

CTSVexample_data *A simulated data set*

Description

A simulated data set for demonstrating how to use the ctsv function.

gamma_true A 20 by 2 0-1 matrix, indicator of SV genes.

spe A SpatialExperiment class.

W A 100 by 2 cell-type proportion matrix.

Format

A list containing 3 elements.

Examples

```
library(SpatialExperiment)
rDirichlet <- function(n,alpha){
    l <- length(alpha)
    x <- matrix(rgamma(l * n, alpha), ncol = l, byrow = TRUE)
    sm <- x
    return(x/as.vector(sm))

}
seed <- 20210509
set.seed(seed)
#  gene numbers
G <- 20
# cell type numbers
K <- 2
# spot numbers
n <- 100
# number of DE genes
DE_num <- 10
# drop out probability
pai <- 0.5
# parameter of NB distribution
size = 100

# coordinates of spots
loc <- NULL
for(i in 1:10){
    for(j in 1:10){
        loc <- rbind(loc,c(i,j))
    }
}
rownames(loc) <- paste0("spot",1:n)
colnames(loc) <- c("x","y")
NDE_scrna <- rnorm(G, mean=2, sd=0.2)
scrna_1 <- NDE_scrna
scrna_2 <- NDE_scrna
scrna_2[sample(1:G,DE_num,replace = FALSE)] <- rnorm(DE_num, mean=3, sd=0.2)
eta <- cbind(scrna_1,scrna_2)

gamma_true <- matrix(0, G, K)
gamma_true[11:13,1] <- 1
gamma_true[14:16,2] <- 1
beta1 <- matrix(0, G, K)
beta2 <- matrix(0, G, K)

# cell type proportion
W <- rDirichlet(n, c(1,2))
W <- t(W)

S <- t(loc) - colMeans(loc)
S <- t(S / apply(S, 1, sd))
```

```
h1 <- S[,1]
h2 <- S[,2]
beta1[gamma_true == 1] <- 1
beta2[gamma_true == 1] <- 0.5

log_lambda <- eta
W <- t(W)
Y <- matrix(rnbinom(G*n,size = size, mu = exp(c(log_lambda))), G, n)
set.seed(5)
r_unif <- matrix(runif(G*n),G,n)
Y[r_unif <= pai] <- 0
colnames(Y) = rownames(loc)
rownames(W) = rownames(loc)
rownames(Y) <- paste0("gene",1:G)
spe <- SpatialExperiment(
    assay = list(counts = Y),
    colData = loc,
    spatialCoordsNames = c("x","y")
)
CTSVexample_data <- list(spe,W,gamma_true)
```

---

svGene *Report spatially variable genes*

Description

Report spatially variable genes

Usage

```
svGene(Q_val, thre.alpha = 0.05)
```

Arguments

Q_val	A G by 2K q-value matrix, where G is the number of genes and K is the number of cell types.
thre.alpha	numeric, a q-value threshold to control FDR less than thre.alpha.

Value

A list with a G by 2K 0-1 matrix and a list with SV gene names in each cell type. The first K columns of the 0-1 matrix correspond to the coordinate of S_1, and the last K columns to the coordinate of S_2.

SV	A G by 2K 0-1 matrix. The first K columns correspond to the coordinate of S_1, the last K columns to the coordinate of S_2.
SVGene	A list with SV gene names in each cell type.

Examples

```
library(CTSV)
# Simulate a Q value matrix
K <- 2 # cell-type number
G <- 10 # gene number
set.seed(1)
Q_val <-matrix(runif(G*K,0,0.1),G,K)
rownames(Q_val) <- paste0("gene",seq_len(G))
# detect SV genes
re <- svGene(Q_val,0.05)
#SV genes in each cell type:
re$SVGene
```

Index