Package 'scDDboost'

May 5, 2023

Type Package

Title A compositional model to assess expression changes from single-cell rna-seq data

Version 1.3.0

Date 2018-10-31

Description

scDDboost is an R package to analyze changes in the distribution of single-cell expression data between two experimental conditions. Compared to other methods that assess differential expression, scDDboost benefits uniquely from information conveyed by the clustering of cells into cellular subtypes. Through a novel empirical Bayesian formulation it calculates gene-specific posterior probabilities that the marginal expression distribution is the same (or different) between the two conditions. The implementation in scDDboost treats gene-level expression data within each condition as a mixture of negative binomial distributions.

License GPL (>= 2)

Imports Rcpp (>= 0.12.11), RcppEigen (>= 0.3.2.9.0),EBSeq, BiocParallel, mclust, SingleCellExperiment, cluster, Oscope, SummarizedExperiment, stats, methods

biocViews SingleCell, Software, Clustering, Sequencing, GeneExpression, DifferentialExpression, Bayesian

Depends R (>= 4.2), ggplot2

LinkingTo Rcpp, RcppEigen, BH

Suggests knitr, rmarkdown, BiocStyle, testthat

SystemRequirements c++11

Roxygen list(wrap=FALSE)

RoxygenNote 7.1.2

VignetteBuilder knitr

BugReports https://github.com/wiscstatman/scDDboost/issues

URL https://github.com/wiscstatman/scDDboost

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

git_branch devel

git_last_commit a1f7b3f

git_last_commit_date 2023-04-25

Date/Publication 2023-05-04

Author Xiuyu Ma [cre, aut],
Michael A. Newton [ctb]

Maintainer Xiuyu Ma <watsonforfun@gmail.com>

R topics documented:

`scDDboost-package` *A compositional model to assess expression changes from single-cell rna-seq data*

Description

scDDboost is an R package to analyze changes in the distribution of single-cell expression data between two experimental conditions. Compared to other methods that assess differential expression, scDDboost benefits uniquely from information conveyed by the clustering of cells into cellular subtypes. Through a novel empirical Bayesian formulation it calculates gene-specific posterior probabilities that the marginal expression distribution is the same (or different) between the two conditions. The implementation in scDDboost treats gene-level expression data within each condition as a mixture of negative binomial distributions.

Details

The DESCRIPTION file: This package was not yet installed at build time.

Index: This package was not yet installed at build time.
Package used to score evidence of differential distribution in single-cell RNA-seq data

Author(s)

NA

Maintainer: NA

References

https://projecteuclid.org/journals/annals-of-applied-statistics/volume-15/issue-2/A-compositional-model-to-assess-expression-changes-from-single-cell/10.1214/20-AOAS1423.short

See Also

https://github.com/wiscstatman/scDDboost/blob/master/DESCRIPTION

Examples

```
data(sim_dat)
dat = extractInfo(sim_dat)
data_counts = dat$count_matrix
cd = dat$condition
bp <- BiocParallel::MulticoreParam(4)
D_c = calD(data_counts,bp)
pDD = pdd(data_counts,cd,bp,D_c)
```

---

calD *calculate distance matrix*

Description

calculate distance matrix

Usage

```
calD(data, bp)
```

Arguments

data	transcripts
bp	bioc parallel parameter

Value

distance matrix

Examples

```
data(sim_dat)
dat <- extractInfo(sim_dat)
data_counts <- dat$count_matrix
bp <- BiocParallel::MulticoreParam(4)
D_c <- calD(data_counts,bp)
```

---

clusHelper *function to get intra and inter distance for clusters*

Description

function to get intra and inter distance for clusters

Usage

```
clusHelper(D, i)
```

Arguments

D	distance matrix
i	number of clusters

Value

vector of intra and inter distance

detK *determine the number of clusters*

Description

determine the number of clusters

Usage

```
detK(D, epi = 1)
```

Arguments

D	distance matrix
epi	threshold for cutting off

Value

number of clusters

Examples

```
data(sim_dat)
dat <- extractInfo(sim_dat)
data_counts <- dat$count_matrix
bp <- BiocParallel::MulticoreParam(4)
D_c <- calD(data_counts,bp)
detK(D_c)
```

EBS *accelerated empirical bayesian*

Description

accelerated empirical bayesian

Usage

```
EBS(data, conditions, gclus, sf, iter = 10, hyper, PP, stp1, stp2)
```

Arguments

data	single cell expression matrix, row as genes column as cells
conditions	partition of cells
gclus	partition of genes
sf	size factors
iter	maximum iteration step of EM
hyper	hyper parameters for beta distributions
PP	pattern of partitions
stp1	step size of hyperparameter alpha (shared by all units) in one step EM
stp2	step size of hyperparameter beta (unit specific) in one step EM

Value

posterior probability of mean expression pattern

---

extractInfo *extract count matrix from SingleCellExperiment object*

Description

extract count matrix from SingleCellExperiment object

Usage

```
extractInfo(data)
```

Arguments

data	SingleCellExperiment object

Value

list of count matrix and condition vector

Examples

```
data(sim_dat)
dat <- extractInfo(sim_dat)
```

gCl *gene_level cluster*

Description

gene_level cluster

Usage

```
gCl(data, bp)
```

Arguments

data	transcripts
bp	bioc parallel parameter

Value

return a matrix whose row represent gene specific cluster

genRClus *generate random clusterings*

Description

generate random clusterings

Usage

```
genRClus(D, a, K)
```

Arguments

D	distance matrix of cells
a	paramter for weights
K	number of subtypes

Value

random generated clustering of cells

getDD *index of DD genes under FDR control*

Description

index of DD genes under FDR control

Usage

```
getDD(pDD, FDR = 0.01)
```

Arguments

pDD	probability of genes being DD
FDR	fdr to be controlled

Value

index of positive genes

Examples

```
p_dd <- c(0.01,0.99,0.7,0.5)
getDD(p_dd)
```

getSizeofDD *number of DD genes under FDR control*

Description

number of DD genes under FDR control

Usage

```
getSizeofDD(pDD, FDR = 0.01)
```

Arguments

pDD	estimated probability of being DD
FDR	fdr to be controlled

Value

number of positive genes

Examples

```
p_dd <- c(0.1,0.99,1,0.05,0.05)
getSizeofDD(p_dd)
```

`getZ1Z2` *function to get counts of cluster sizes at two conditions*

Description

function to get counts of cluster sizes at two conditions

Usage

```
getZ1Z2(ccl, cd)
```

Arguments

`ccl`	clustering label
`cd`	condition label

Value

return list of counts

`gRef` *generate reference matrix*

Description

generate reference matrix

Usage

```
gRef(Posp)
```

Arguments

`Posp`	possible partition of data

Value

return a matrix indicate the refinement relation between different partitions.

isRef *check refinement relation between two clusters*

Description

check refinement relation between two clusters

Usage

```
isRef(x, y)
```

Arguments

x	a cluster
y	a cluster

Value

whether x refines y

LL *likelihood function for hyperparameters estimation*

Description

likelihood function for hyperparameters estimation

Usage

```
LL(param, x, d0)
```

Arguments

param	parameters to be determined by MLE
x	distance matrix of cells
d0	rate parameter of prior of 1 / true distance

Value

return hyperparameteres a.

lpt1t2 *log likelihood of z1,z2 given t1,t2*

Description

log likelihood of z1,z2 given t1,t2

Usage

```
lpt1t2(z1, z2, pp, alpha1, alpha2)
```

Arguments

z1	counts of each group in condition 1
z2	counts of each group in condition 2
pp	a partition
alpha1	parameter of double dirichlet prior
alpha2	parameter of double dirichlet prior

Value

log likelihood of z1,z2 given t1,t2

lpzgt *log likelihood of aggregated multinomial counts z given aggregated proportions t*

Description

log likelihood of aggregated multinomial counts z given aggregated proportions t

Usage

```
lpzgt(z, pp, alpha)
```

Arguments

z	counts of each group in one condition
pp	a partition
alpha	parameter of double dirichlet prior

Value

log likelihood of aggregated multinomial counts z given aggregated proportions t

mdd *posterior of proportion change given mixture double dirichlet prior*

Description

posterior of proportion change given mixture double dirichlet prior

Usage

```
mdd(z1, z2, pat, alpha1, alpha2)
```

Arguments

z1	counts of each group in condition 1
z2	counts of each group in condition 2
pat	partition patterns
alpha1	parameter of double dirichlet prior
alpha2	parameter of double dirichlet prior

Value

posterior of proportion change

pat *generating partition patterns*

Description

generating partition patterns

Usage

```
pat(K)
```

Arguments

K	number of elements

Value

all possible partition of K elements

Examples

```
pat(3)
```

pdd *calculate posterior probabilities of a gene to be differential distributed*

Description

calculate posterior probabilities of a gene to be differential distributed

Usage

```
pdd(
  data,
  cd,
  bp,
  D,
  random = TRUE,
  norm = TRUE,
  epi = 1,
  Upper = 1000,
  nrandom = 50,
  iter = 20,
  reltol = 0.001,
  stp1 = 1e-06,
  stp2 = 0.01,
  K = 0
)
```

Arguments

data	normalized preprocessed transcripts
cd	conditions label
bp	bioc parallel parameter
D	distance matrix of cells or cluster of cells or a given clustering
random	boolean indicator of whether randomzation has been been implemented on distance matrix
norm	boolean indicator of whether the input expression data is normalized
epi	tol for change of validity score in determining number of clusters
Upper	bound for hyper parameters optimization
nrandom	number of random generated distance matrix
iter	max number of iterations for EM
reltol	relative tolerance for optim on weighting paramters
stp1	step size of hyperparameter alpha (shared by all units) in one step EM
stp2	step size of hyperparameter beta (unit specific) in one step EM
K	number of subtypes, could be user specified or determined internally(set to 0)

Value

posterior probabilities of a gene to be differential distributed

Examples

```
data(sim_dat)
dat <- extractInfo(sim_dat)
data_counts <- dat$count_matrix
cd <- dat$condition
bp <- BiocParallel::MulticoreParam(4)
D_c <- calD(data_counts,bp)
pDD <- pdd(data_counts,cd,bp,D_c)
```

---

pddAggregate *function to aggregate intermediate results and get prob of DD*

Description

function to aggregate intermediate results and get prob of DD

Usage

```
pddAggregate(z1, z2, Posp, DE, K, REF)
```

Arguments

z1	counts of cluster sizes in condition 1
z2	counts of cluster sizes in condition 2
Posp	partition of cells
DE	posterior probabilities of DE patterns
K	number of clusters
REF	reference matrix indicating relation of nested partitions

Value

return vector of prob of DD

pddRandom *calculate PDD when add random noise in distance matrix*

Description

calculate PDD when add random noise in distance matrix

Usage

```
pddRandom(data, cd, K, D, a, sz, hp, Posp, iter, REF, stp1, stp2)
```

Arguments

data	normalized preprocessed transcripts
cd	condition label
K	number of subgroups
D	distance matrix of cells
a	shape param for weights
sz	size factors
hp	hyper parameters for EBSeq
Posp	parition patterns
iter	max number of iterations for EM in EBSeq
REF	refinement relation matrix
stp1	step size of hyperparameter alpha (shared by all units) in one step EM
stp2	step size of hyperparameter beta (unit specific) in one step EM

Value

posterior probabilities under random distance matrix

rwMle *MLE for random weighting parameter*

Description

MLE for random weighting parameter

Usage

```
rwMle(D, reltol)
```

Arguments

D,	distance matrix of cells
reltol,	tolerance of convergence

Value

MLE of random weighting parameter

---

sim_dat *scDDboost*

Description

simulated data for demonstration, data are mixture negative binomial distributed

Usage

```
data(sim_dat)
```

Format

An object of class "list".

Examples

```
data(sim_dat)
```

Index