Package 'bayNorm'

November 25, 2024

Type Package

Title Single-cell RNA sequencing data normalization

Version 1.24.0

Description bayNorm is used for normalizing single-cell RNA-seq data.

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.0.2

Depends R (>= 3.5),

Imports Rcpp (>= 0.12.12), BB, foreach, iterators, doSNOW, Matrix, parallel, MASS, locfit, fitdistrplus, stats, methods, graphics, grDevices, SingleCellExperiment, SummarizedExperiment, BiocParallel, utils

LinkingTo Rcpp, RcppArmadillo,RcppProgress

Suggests knitr, rmarkdown, BiocStyle, devtools, testthat

VignetteBuilder knitr

biocViews ImmunoOncology, Normalization, RNASeq, SingleCell,Sequencing

URL https://github.com/WT215/bayNorm

BugReports https://github.com/WT215/bayNorm/issues

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

git_branch RELEASE_3_20

git_last_commit 59c02c9

git_last_commit_date 2024-10-29

Repository Bioconductor 3.20

Date/Publication 2024-11-25

Author Wenhao Tang [aut, cre],
Fran<U+00E7>ois Bertaux [aut],
Philipp Thomas [aut],
Claire Stefanelli [aut],
Malika Saint [aut],
Samuel Marguerat [aut],
Vahid Shahrezaei [aut]

Maintainer Wenhao Tang <wt215@ic.ac.uk>

Contents

`AdjustSIZE_fun` *Adjust MME size estimate*

Description

This function adjusts MME estimated size parameter of prior, which is a negative binomial distribution, using estimates from maximizing marginal distirbution (`BB_SIZE`). Simulation studies has shown this hybrid method of using adjusted MME size estimates is the most robust (see bayNorm paper). Hence, this is the default option for estimating size in bayNorm.

Usage

```
AdjustSIZE_fun(BB_SIZE, MME_MU, MME_SIZE)
```

Arguments

`BB_SIZE`	size estimated from `BB_Fun`.
`MME_MU`	mu estimated from EstPrior.
`MME_SIZE`	size estimated from EstPrior.

Value

MME_SIZE_adjust: A vector of estimated size. Adjusted MME_SIZE based on BB_SIZE (size estimated by maximizing marginal distribution)

Examples

```
data('EXAMPLE_DATA_list')
MME_MU<-rlnorm(100,meanlog=5,sdlog=1)
MME_SIZE<-rlnorm(100,meanlog=1,sdlog=1)
BB_SIZE<-rlnorm(100,meanlog=0.5,sdlog=1)
adjustt<-AdjustSIZE_fun(BB_SIZE, MME_MU, MME_SIZE)
```

---

asMatrix *Rcpp version: as.matrix*

Description

Rcpp version: as.matrix

Usage

```
asMatrix(rp, cp, z, nrows, ncols)
```

Arguments

rp	vector
cp	vector
z	vector
nrows	nrows
ncols	ncols

Details

Rcpp version: as.matrix

Value

Matrix object in R.

Examples

```
data("EXAMPLE_DATA_list")
#Should not run by the users, it is used in prior estimation.
## Not run:
```

as_matrix *Rcpp version's as.matrix*

Description

Transform sparse matrix to matrix.

Usage

```
as_matrix(mat)
```

Arguments

mat	Sparse matrix.

Value

Matrix.

Examples

```
aa<-matrix(seq(1,6),nrow=2,ncol=3)
qq<-as(as.matrix(aa), "dgCMatrix")
all.equal(unname(as_matrix(qq)),unname(as.matrix(qq)))
```

bayNorm *A wrapper function of prior estimation and bayNorm function*

Description

This is the main wrapper function for bayNorm. The input is a matrix of raw scRNA-seq data and a vector of capture efficiencies of cells. You can also specify the condition of cells for normalizing multiple groups of cells separately.

Usage

```
bayNorm(
  Data,
  BETA_vec = NULL,
  Conditions = NULL,
  UMI_sffl = NULL,
  Prior_type = NULL,
  mode_version = FALSE,
  mean_version = FALSE,
  S = 20,
  parallel = TRUE,
  NCores = 5,
  FIX_MU = TRUE,
  GR = FALSE,
  BB_SIZE = TRUE,
```

```
  verbose = TRUE,
  out.sparse = FALSE
)
```

Arguments

Data	A matrix of single-cell expression where rows are genes and columns are samples (cells). Data can be of class SummarizedExperiment (the assays slot contains the expression matrix, is named "Counts"), just matrix or sparse matrix.
BETA_vec	A vector of capture efficiencies (probabilities) of cells. If it is null, library size (total count) normalized to 0.06 will be used as the input BETA_vec. BETA_vec less and equal to 0 or greater and equal to 1 will be replaced by the minimum and maximum of the BETA_vec which range between (0,1) respectively.
Conditions	vector of condition labels, this should correspond to the columns of the Data. D efault is NULL, which assumes that all cells belong to the same group.
UMI_sffl	Scaling factors are required only for non-UMI based data for which Data is devided by UMI_sffl. If non-null and Conditions is non-null, then UMI_sffl should be a vector of length equal to the number of groups. Default is NULL.
Prior_type	Determines what groups of cells is used in estimating prior using Conditions. Default is NULL. If Conditions is NULL, priors are estimated based on all cells. If Conditions is not NULL and if Prior_type is LL, priors are estimated within each group respectively. If Prior_type is GG, priors are estimated based on cells from all groups. LL is suitable for DE detection. GG is preferred if reduction of batch effect between samples are desired for example for technical replicates (see bayNorm paper).
mode_version	If TRUE, bayNorm return modes of posterior estimates as normalized data which is a 2D matrix rather than samples from posterior which is a 3D array. Default is FALSE.
mean_version	If TRUE, bayNorm return means of posterior estimates as normalized data, which is a 2D matrix rather than samples from posterior which is a 3D array. Default is FALSE.
S	The number of samples you would like to generate from estimated posterior distribution (The third dimension of 3D array). Default is 20. S needs to be specified if mode_version=FALSE.
parallel	If TRUE, NCores cores will be used for parallelization. Default is TRUE.
NCores	number of cores to use, default is 5. This will be used to set up a parallel environment using either MulticoreParam (Linux, Mac) or SnowParam (Windows) with NCores using the package BiocParallel.
FIX_MU	Whether fix mu (the mean parameter of prior distribution) to its MME estimate, when estimating prior parameters by maximizing marginal distribution. If TRUE, then 1D optimization is used, otherwise 2D optimization for both mu and size is used (slow). Default is TRUE.
GR	If TRUE, the gradient function will be used in optimization. However since the gradient function itself is very complicated, it does not help too much in speeding up. Default is FALSE.
BB_SIZE	If TRUE, estimate size parameter of prior using maximization of marginal likelihood, and then use it for adjusting MME estimate of SIZE Default is TRUE.
verbose	print out status messages. Default is TRUE.
out.sparse	Only valid for mean version: Whether the output is of type dgCMatrix or not. Default is FALSE.

Details

A wrapper function of prior estimation and bayNorm function.

Value

List containing 3D arrays of normalized expression (if `mode_version=FALSE`) or 2D matrix of normalized expression (if `mode_version=TRUE` or `mean_version=TRUE`), a list contains estimated priors and a list contains input parameters used: `BETA_vec`, `Conditions` (if specified), `UMI_sffl` (if specified), `Prior_type`, `FIX_MU`, `BB_SIZE` and `GR`.

References

Wenhao Tang, Francois Bertaux, Philipp Thomas, Claire Stefanelli, Malika Saint, Samuel Blaise Marguerat, Vahid Shahrezaei bayNorm: Bayesian gene expression recovery, imputation and normalisation for single cell RNA-sequencing data Bioinformatics, btz726; doi: 10.1093/bioinformatics/btz726

Examples

```
data('EXAMPLE_DATA_list')
#Return 3D array normalzied data:
bayNorm_3D<-bayNorm(
Data=EXAMPLE_DATA_list$inputdata[,seq(1,30)],
BETA_vec = EXAMPLE_DATA_list$inputbeta[seq(1,30)],
mode_version=FALSE,parallel =FALSE)
```

---

bayNorm_sup *bayNorm with estimated parameters as input*

Description

This is a supplementary wrapper function for bayNorm. It is useful if one has already estimated prior parameters and wants to simulate 2D or 3D normalized output using the same prior estimates.

Usage

```
bayNorm_sup(
  Data,
  PRIORS = NULL,
  input_params = NULL,
  mode_version = FALSE,
  mean_version = FALSE,
  S = 20,
  parallel = TRUE,
  NCores = 5,
  BB_SIZE = TRUE,
  verbose = TRUE,
  out.sparse = FALSE
)
```

Arguments

Data	A matrix of single-cell expression where rows are genes and columns are samples (cells). Data can be of class SummarizedExperiment (the assays slot contains the expression matrix, is named "Counts"), just matrix or sparse matrix.
PRIORS	A list of estimated prior parameters obtained from bayNorm.
input_params	A list of input parameters which have been used: BETA_vec, Conditions, UMI_sffl, Prior_type, FIX_MU, BB_SIZE and GR.
mode_version	If TRUE, bayNorm return mode version normalized data which is of 2D matrix instead of 3D array. Default is FALSE.
mean_version	If TRUE, bayNorm return mean version normalized data which is of 2D matrix instead of 3D array. Default is FALSE.
S	The number of samples you would like to generate from estimated posterior distribution (The third dimension of 3D array). Default is 20. S needs to be specified if mode_version=FALSE.
parallel	If TRUE, NCores cores will be used for parallelization. Default is TRUE.
NCores	number of cores to use, default is 5. This will be used to set up a parallel environment using either MulticoreParam (Linux, Mac) or SnowParam (Windows) with NCores using the package BiocParallel.
BB_SIZE	If TRUE (default), use adjusted size for normalization. The adjusted size is obtained by adjusting MME estimated size by a factor. The factor is calculated based on both MME estimated size and BB estimated size. If FALSE, use MME estimated SIZE.
verbose	print out status messages. Default is TRUE.
out.sparse	Only valid for mean version: Whether the output is of type dgCMatrix or not. Default is FALSE.

Details

If you have run bayNorm before and obtained a list of estimated prior parameters, then you may not want to run parameter estimation again. You can just use previous estimated parameters for obtaining 3D or 2D normalized data.

Value

List containing 3D arrays of normalized expression (if mode_version=FALSE) or 2D matrix of normalized expression (if mode_version=TRUE or mean_version=TRUE), a list contains estimated priors and a list contains input parameters used: BETA_vec, Conditions (if specified), UMI_sffl (if specified), Prior_type, FIX_MU, BB_SIZE and GR.

References

Wenhao Tang, Francois Bertaux, Philipp Thomas, Claire Stefanelli, Malika Saint, Samuel Blaise Marguerat, Vahid Shahrezaei bayNorm: Bayesian gene expression recovery, imputation and normalisation for single cell RNA-sequencing data Bioinformatics, btz726; doi: 10.1093/bioinformatics/btz726

Examples

```
data('EXAMPLE_DATA_list')
#Return 3D array normalzied data:
bayNorm_3D<-bayNorm(
Data=EXAMPLE_DATA_list$inputdata[,seq(1,30)],
BETA_vec = EXAMPLE_DATA_list$inputbeta[seq(1,30)]
,mode_version=FALSE,parallel =FALSE)

#Now if you want to generate 2D matrix using the same prior
#estimates as generated before:
bayNorm_2D<-bayNorm_sup(
Data=EXAMPLE_DATA_list$inputdata[,seq(1,30)]
,PRIORS=bayNorm_3D$PRIORS,
input_params = bayNorm_3D$input_params
,mode_version=TRUE)
```

BB_Fun *Estimating size for each gene by either 1D or 2D maximization of marginal distribution*

Description

Estimating parameters of the prior distribution for each gene by maximizing marginal distribution: 1D (optimize with respect to size using MME estimate of mu, 2D (optimize with respect to both mu and size)

Usage

```
BB_Fun(
  Data,
  BETA_vec,
  INITIAL_MU_vec,
  INITIAL_SIZE_vec,
  MU_lower = 0.01,
  MU_upper = 500,
  SIZE_lower = 0.01,
  SIZE_upper = 30,
  parallel = FALSE,
  NCores = 5,
  FIX_MU = TRUE,
  GR = FALSE
)
```

Arguments

Data	A matrix of single-cell expression where rows are genes and columns are samples (cells). Data can be of class SummarizedExperiment (the assays slot contains the expression matrix, is named "Counts"), just matrix or sparse matrix.
BETA_vec	A vector of capture efficiencies (probabilities) of cells.
INITIAL_MU_vec	Mean expression of genes, can be estimated from EstPrior.

INITIAL_SIZE_vec
: size of genes (size is a parameter in NB distribution), can come from EstPrior.

MU_lower
: The lower bound for the mu.(Only need it when you want to do 2D optimization). Default is 0.01.

MU_upper
: The upper bound for the mu.(Only need it when you want to do 2D optimization). Default is 500.

SIZE_lower
: The lower bound for the size. Default is 0.01.

SIZE_upper
: The upper bound for the size. Default is 30.

parallel
: If TRUE, NCores cores will be used for parallelization. Default is TRUE.

NCores
: number of cores to use, default is 5. This will be used to set up a parallel environment using either MulticoreParam (Linux, Mac) or SnowParam (Windows) with NCores using the package BiocParallel.

FIX_MU
: If TRUE, then 1D optimization, otherwise 2D optimization (slow).

GR
: If TRUE, the gradient function will be used in optimization. However since the gradient function itself is very complicated, it does not help too much in speeding up. Default is FALSE.

Value

BB estimated size (1D optimization) or size and mu (2D optimization).

Examples

```
data('EXAMPLE_DATA_list')
BB_RESULT<-BB_Fun(Data=EXAMPLE_DATA_list$inputdata[,seq(1,30)],
BETA_vec = EXAMPLE_DATA_list$inputbeta[seq(1,30)],
INITIAL_MU_vec=EXAMPLE_DATA_list$mu,
INITIAL_SIZE_vec=EXAMPLE_DATA_list$size,
MU_lower=0.01,MU_upper=500,SIZE_lower=0.01,
SIZE_upper=30,parallel=FALSE,NCores=5,FIX_MU=TRUE,GR=FALSE)
```

BetaFun *Estimate capture efficiency for cells*

Description

This function estimates cell specific capture efficiencies (BETA_vec) using mean raw counts of a subset of genes that is an input for bayNorm. A specific method is used to exclude genes with high expression or high drop-out are excluded.

Usage

```
BetaFun(Data, MeanBETA)
```

Arguments

Data	A matrix of single-cell expression where rows are genes and columns are samples (cells). Data can be of class SummarizedExperiment (the assays slot contains the expression matrix, is named "Counts"), just matrix or sparse matrix.
MeanBETA	Mean capture efficiency of the scRNAseq data. This can be estimated via spike-ins or other methods.

Value

List containing: BETA: a vector of capture efficiencies, which is of length number of cells; Selected_genes: a subset of genes that are used for estimating BETA.

Examples

```
data('EXAMPLE_DATA_list')
BETA_out<-BetaFun(Data=EXAMPLE_DATA_list$inputdata,
MeanBETA=0.06)
```

Check_input *Check input*

Description

Check input

Usage

```
Check_input(Data)
```

Arguments

Data	Input data.

Value

Matrix (of type matrix in R).

Examples

```
aa<-matrix(seq(1,6),nrow=2,ncol=3)
Check_input(aa)
```

DownSampling *Binomial downsampling*

Description

For each element in the Data, randomly generate a number using Binomial distribution with probability equal to the specific capture efficiency.

Usage

```
DownSampling(Data, BETA_vec)
```

Arguments

Data	raw count Data
BETA_vec	A vector of capture efficiencies of cells

Value

A matrix of binomial downsampling data.

Examples

```
data("EXAMPLE_DATA_list")
Downsample_data<-DownSampling(Data=EXAMPLE_DATA_list$inputdata
,BETA_vec = EXAMPLE_DATA_list$inputbeta)
```

EstPrior *Estimate size and mu for Negative Binomial distribution for each gene using MME method*

Description

Input raw data and return estimated size and mu for each gene using the MME method.

Usage

```
EstPrior(Data, verbose = TRUE)
```

Arguments

Data	A matrix of single-cell expression where rows are genes and columns are samples (cells). Data can be of class SummarizedExperiment (the assays slot contains the expression matrix, is named "Counts"), just matrix or sparse matrix.
verbose	print out status messages. Default is TRUE.

Details

mu and size are two parameters of the prior that need to be specified for each gene in bayNorm. They are parameters of negative binomial distribution. The variance is $mu + mu^2/size$ in this parametrization.

Value

List containing estimated mu and size for each gene.

Examples

```
data('EXAMPLE_DATA_list')
#Return estimated mu and size for each gene using MME method.
MME_est<-EstPrior(Data=EXAMPLE_DATA_list$inputdata[,seq(1,30)],
verbose=TRUE)
```

`EstPrior_rcpp` *Estimate size and mu for Negative Binomial distribution for each gene using MME method (Rcpp version)*

Description

Input raw data and return estimated size and mu for each gene using the MME method.

Usage

```
EstPrior_rcpp(Data)
```

Arguments

`Data`	A matrix of single-cell expression where rows are genes and columns are samples (cells). `Data` can be of class `SummarizedExperiment` (the assays slot contains the expression matrix and is named "Counts") or just matrix.

Details

mu and size are two parameters of the prior that need to be specified for each gene in bayNorm. They are parameters of negative binomial distribution. The variance is $mu + mu^2/size$ in this parametrization.

Value

List containing estimated mu and size for each gene.

Examples

```
data("EXAMPLE_DATA_list")
#Should not run by the users, it is used in prior estimation.
## Not run:
```

`EstPrior_sprcpp` *Estimate size and mu for Negative Binomial distribution for each gene using MME method (Rcpp version, sp_mat)*

Description

Input raw data and return estimated size and mu for each gene using the MME method.

Usage

```
EstPrior_sprcpp(Data)
```

Arguments

`Data`	A matrix of single-cell expression where rows are genes and columns are samples (cells). `Data` can be of class `SummarizedExperiment` (the assays slot contains the expression matrix and is named "Counts") or just matrix.

Details

mu and size are two parameters of the prior that need to be specified for each gene in bayNorm. They are parameters of negative binomial distribution. The variance is $mu + mu^2/size$ in this parametrization.

Value

List containing estimated mu and size for each gene.

Examples

```
data("EXAMPLE_DATA_list")
#Should not run by the users, it is used in prior estimation.
## Not run:
```

`EXAMPLE_DATA_list` *A subset of Grun et al (2014) data: 2i samples*

Description

Small extract (20 genes and 74 cells) from the Grun et al (2014) data: 2i samples

Usage

```
EXAMPLE_DATA_list
```

Format

A list: EXAMPLE_DATA_list[[1]]: inputdata, EXAMPLE_DATA_list[[2]]: inputbeta (a vector of probabilities with length equal to the number of cells), EXAMPLE_DATA_list[[3]]: mu (MME method estimated mean expression for each gene), EXAMPLE_DATA_list[[4]]: size (adjusted MME size for each gene).

References

Grun, Kester and van Oudenaarden (2014). Nature Methods.

Examples

```
data(EXAMPLE_DATA_list)
```

NOISY_FUN *Noisy gene detection*

Description

This function detects noisy genes using trends observed in a set of synthetic controls. Input bayNorm normalized data of real data (bay_array_N) and synthetic control (bay_array_C) respectively.

Usage

```
NOISY_FUN(bay_array_N, bay_array_C, plot.out = FALSE)
```

Arguments

bay_array_N	A 2D matrix or 3D array of normalized data(real cells).
bay_array_C	A 2D matrix or 3D array of normalized data(synthetic control).
plot.out	If TRUE, show CV^2 vs Mean expression plot. Default is FALSE.

Details

bay_array_N and bay_array_C should be of the same dimension.

Value

A vector of adjusted P-values.

Examples

```
bay_array_N<-array(rpois(1000*50*2,17),dim=c(1000,50,2))
bay_array_C<-array(rpois(1000*50*2,58),dim=c(1000,50,2))
noisy_output<-NOISY_FUN(bay_array_N,bay_array_C)
```

noisy_gene_detection *A wrapper function for noisy gene detection from raw data. his produces synthetic control, performs bayNorm on both real cell data and synthetic controls and does noisy gene detection.*

Description

A wrapper function for noisy gene detection from raw data. his produces synthetic control, performs bayNorm on both real cell data and synthetic controls and does noisy gene detection.

Usage

```
noisy_gene_detection(
  Data,
  BETA_vec = NULL,
  mode_version = FALSE,
  mean_version = FALSE,
  S = 20,
  parallel = TRUE,
  NCores = 5,
  FIX_MU = TRUE,
  GR = FALSE,
  BB_SIZE = TRUE,
  verbose = TRUE,
  plot.out = FALSE,
  PRIORS = NULL,
  input_params = NULL
)
```

Arguments

Data	A matrix of single-cell expression where rows are genes and columns are samples (cells). Data can be of class SummarizedExperiment (the assays slot contains the expression matrix, is named "Counts"), just matrix or sparse matrix.
BETA_vec	A vector of capture efficiencies of cells.
mode_version	If TRUE, bayNorm return mode version normalized data which is of 2D matrix instead of 3D array. Default is FALSE.
mean_version	If TRUE, bayNorm return mean version normalized data which is of 2D matrix instead of 3D array. Default is FALSE.
S	The number of samples you would like to generate from estimated posterior distribution (The third dimension of 3D array). Default is 20. S needs to be specified if mode_version=FALSE.
parallel	If TRUE, NCores cores will be used for parallelization. Default is TRUE.
NCores	number of cores to use, default is 5. This will be used to set up a parallel environment using either MulticoreParam (Linux, Mac) or SnowParam (Windows) with NCores using the package BiocParallel.
FIX_MU	Whether fix mu when estimating parameters by maximizing marginal distribution. If TRUE, then 1D optimization, otherwise 2D optimization (slow).

GR	If TRUE, the gradient function will be used in optimization. However since the gradient function itself is very complicated, it does not help too much in speeding up. Default is FALSE.
BB_SIZE	If TRUE, estimate BB size, and then use it for adjusting MME SIZE. Use the adjusted MME size for bayNorm. Default is TRUE.
verbose	Print out status messages. Default is TRUE.
plot.out	If TRUE, show CV^2 vs Mean expression plot. Default is FALSE.
PRIORS	(Need to be specified for efficiency if bayNorm has already been applied) A list of estimated prior parameters obtained from bayNorm. Default is NULL.
input_params	(Need to be specified for efficiency if bayNorm has already been applied) A list of input parameters which have been used: BETA_vec, Conditions, UMI_sffl, Prior_type, FIX_MU, BB_SIZE and GR.

Details

A wrapper function for noisy gene detection from raw scRNA-seq data.

Value

A list of objects.

Examples

```
data("EXAMPLE_DATA_list")
noisy_out<-noisy_gene_detection(Data=
EXAMPLE_DATA_list$inputdata[,seq(1,30)],BETA_vec
=EXAMPLE_DATA_list$inputbeta[seq(1,30)], mode_version = FALSE,
mean_version=FALSE,
S = 20,parallel = FALSE, NCores = 5,
FIX_MU = TRUE, GR = FALSE,
PRIORS=NULL,
BB_SIZE = TRUE,
verbose = TRUE, plot.out = TRUE)
```

Prior_fun *A wrapper function of* EstPrior *and* AdjustSIZE_fun

---

Description

A wrapper function for estimating the parameters of prior using the hybrid method adjusted MME estimates based on maximization of marginal likelihood. Input raw data and a vector of capture efficiencies of cells.

Usage

```
Prior_fun(
  Data,
  BETA_vec,
  parallel = TRUE,
  NCores = 5,
```

```
  FIX_MU = TRUE,
  GR = FALSE,
  BB_SIZE = TRUE,
  verbose = TRUE
)
```

Arguments

Data	A matrix of single-cell expression where rows are genes and columns are samples (cells). Data can be of class SummarizedExperiment (the assays slot contains the expression matrix, is named "Counts"), just matrix or sparse matrix.
BETA_vec	A vector of capture efficiencies of cells.
parallel	If TRUE, 5 cores will be used for parallelization. Default is TRUE.
NCores	number of cores to use, default is 5. This will be used to set up a parallel environment using either MulticoreParam (Linux, Mac) or SnowParam (Windows) with NCores using the package BiocParallel.
FIX_MU	If TRUE, then 1D optimization, otherwise 2D optimization (slow). Default is TRUE.
GR	If TRUE, the gradient function will be used in optimization. However since the gradient function itself is very complicated, it does not help too much in speeding up. Default is FALSE.
BB_SIZE	If TRUE, estimate BB size, and then use it for adjusting MME SIZE. Use the adjusted MME size for bayNorm. Default is TRUE.
verbose	Print out status messages. Default is TRUE.

Details

By Default, this function will estimate mu and size for each gene using MME method. If BB_size is enable, spectral projected gradient method from BB package will be implemented to estimate 'BB size' by maximizing marginal likelihood function. MME estimated size will be adjusted according to BB size. BB size itself will not be used in bayNorm this is because that in our simulation we found that MME estimated mu and size have more accurate relationship, but MME estimated size deviates from the true value. BB size is overall more close to the true size but it does not possess a reasonable relationship with either MME estimated mu or BB estimated mu.

Value

List of estimated parameters: mean expression of genes and size of each gene.

Examples

```
data('EXAMPLE_DATA_list')
PRIOR_RESULT<-Prior_fun(Data=EXAMPLE_DATA_list$inputdata[,seq(1,30)],
BETA_vec = EXAMPLE_DATA_list$inputbeta[seq(1,30)],parallel=FALSE,
NCores=5,FIX_MU=TRUE,GR=FALSE,BB_SIZE=TRUE,verbose=TRUE)
```

SyntheticControl *Generate synthetic control*

Description

Input raw data and a vector of capture efficiencies of cells.

Usage

```
SyntheticControl(Data, BETA_vec)
```

Arguments

Data	A matrix of single-cell expression where rows are genes and columns are samples (cells). Data can be of class SummarizedExperiment (the assays slot contains the expression matrix and is named "Counts") or just matrix.
BETA_vec	A vector of capture efficiencies (probabilities) of cells.

Details

Simulate control data (based on Poisson distribution).

Value

List containing 2D matrix of synthetic control, BETA_vec used and lambda used in rpois.

Examples

```
data("EXAMPLE_DATA_list")
SC_output<-SyntheticControl(Data=
EXAMPLE_DATA_list$inputdata,
BETA_vec = EXAMPLE_DATA_list$inputbeta)
```

t_sp *Transpose of sparse matrix*

Description

Transpose of sparse matrix

Usage

```
t_sp(Data)
```

Arguments

Data	A matrix of single-cell expression where rows are genes and columns are samples (cells). Data can be of class SummarizedExperiment (the assays slot contains the expression matrix and is named "Counts") or just matrix.

Details

Transpose of sparse matrix.

Value

Transpose of sparse matrix.

Examples

```
data("EXAMPLE_DATA_list")
#Should not run by the users, it is used in prior estimation.
## Not run:
```

Index