Package 'nebula'

January 21, 2022

Type Package

Title Negative Binomial Mixed Models Using Large-Sample Approximation
for Differential Expression Analysis of ScRNA-Seq Data

Version 1.2.0

Date 2022-01-19

Author Liang He

Maintainer Liang He <liang.he@duke.edu>

Description

A fast negative binomial mixed model for conducting association analysis of multi-subject single-
cell data. It can be used for identifying marker genes, differential expression and co-
expression analyses. The model includes subject-
level random effects to account for the hierarchical structure in multi-subject single-
cell data. See He et al. (2021) <doi:10.1038/s42003-021-02146-6>.

License GPL-3

Encoding UTF-8

LazyData true

Imports Rcpp (>= 1.0.7), nloptr, MASS, stats, Matrix, methods, utils,
Rfast, trust

LinkingTo Rcpp, RcppEigen

Depends R (>= 4.1)

RoxygenNote 7.1.2

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation yes

Repository CRAN

Date/Publication 2022-01-21 09:02:45 UTC

R topics documented:

---

nebula-package	*Negative Binomial Mixed Models Using Large-Sample Approximation for Differential Expression Analysis of ScRNA-Seq Data*

Description

A fast negative binomial mixed model for conducting association analysis of multi-subject single-cell data. It can be used for identifying marker genes, differential expression and co-expression analyses. The model includes subject-level random effects to account for the hierarchical structure in multi-subject single-cell data. See He et al. (2021) <doi:10.1038/s42003-021-02146-6>.

Details

nebula is an R package for performing association analysis using a fast negative binomial mixed model for multi-subject single-cell data.

Author(s)

Liang He

Maintainer: Liang He <liang.he@duke.edu>

References

He. L, Kulminski, A.M., NEBULA: a fast negative binomial mixed model for differential expression and co-expression analysis of large-scale single-cell data. 2020

Examples

```
library(nebula)
data(sample_data)
pred = model.matrix(~X1+X2+cc,data=sample_data$pred)
re = nebula(count=sample_data$count,id=sample_data$sid,pred=pred)
```

---

group_cell *Group cells according to subject IDs*

Description

Group cells according to subject IDs

Usage

```
group_cell(count, id, pred = NULL, offset = NULL)
```

Arguments

count | A raw count matrix of the single-cell data. The rows are the genes, and the columns are the cells. The matrix can be a matrix object or a sparse dgCMatrix object.

id | A vector of subject IDs. The length should be the same as the number of columns of the count matrix.

pred | A design matrix of the predictors. The rows are the cells and the columns are the predictors. If not specified, an intercept column will be generated by default.

offset | A vector of the scaling factor. The values must be strictly positive. If not specified, a vector of all ones will be generated by default.

Value

count: A reordered count matrix. If the cells are already grouped, the function returns NULL.

id: A reordered subject ID vector.

pred: A reordered design matrix of the predictors.

offset: A reordered vector of the scaling factor.

Examples

```
library(nebula)
data(sample_data)
pred = model.matrix(~X1+X2+cc,data=sample_data$pred)
df_order = group_cell(count=sample_data$count,id=sample_data$sid,pred=pred)
```

---

nbresidual   *Extract Pearson residuals from the results of NEBULA*

Description

Extract Pearson residuals from the results of NEBULA

Usage

```
nbresidual(nebula, count, id, pred = NULL, offset = NULL)
```

Arguments

nebula	An object of the result obtained from running the function nebula.
count	A raw count matrix of the single-cell data. The rows are the genes, and the columns are the cells. The matrix can be a matrix object or a sparse dgCMatrix object.
id	A vector of subject IDs. The length should be the same as the number of columns of the count matrix.
pred	A design matrix of the predictors. The rows are the cells and the columns are the predictors. If not specified, an intercept column will be generated by default.
offset	A vector of the scaling factor. The values must be strictly positive. If not specified, a vector of all ones will be generated by default.

Value

residuals: A matrix of Pearson residuals. The number of columns is the number of cells in the count matrix. The rows correspond to gene IDs reported in the result from nebula.

gene: Gene names corresponding to the row names of the count matrix.

Examples

```
library(nebula)
data(sample_data)
pred = model.matrix(~X1+X2+cc,data=sample_data$pred)
re = nebula(count=sample_data$count,id=sample_data$sid,pred=pred)
resid = nbresidual(re,count=sample_data$count,id=sample_data$sid,pred=pred)
```

---

nebula	*Association analysis of a multi-subject single-cell data set using a fast negative binomial mixed model*

Description

Association analysis of a multi-subject single-cell data set using a fast negative binomial mixed model

Usage

```
nebula(
  count,
  id,
  pred = NULL,
  offset = NULL,
  min = c(1e-04, 1e-04),
  max = c(10, 1000),
  model = "NBGMM",
  method = "LN",
  cutoff_cell = 20,
  kappa = 800,
  opt = "lbfgs",
  verbose = TRUE,
  cpc = 0.005,
  covariance = FALSE
)
```

Arguments

count	A raw count matrix of the single-cell data. The rows are the genes, and the columns are the cells. The matrix can be a matrix object or a sparse dgCMatrix object.
id	A vector of subject IDs. The length should be the same as the number of columns of the count matrix.
pred	A design matrix of the predictors. The rows are the cells and the columns are the predictors. If not specified, an intercept column will be generated by default.
offset	A vector of the scaling factor. The values must be strictly positive. If not specified, a vector of all ones will be generated by default.
min	Minimum values for the overdispersions parameters σ^2 and ϕ. Must be positive. The default is c(1e-4,1e-4).
max	Maximum values for the overdispersions parameters σ^2 and ϕ. Must be positive. The default is c(10,1000).

model	'NBGMM', 'PMM' or 'NBLMM'. 'NBGMM' is for fitting a negative binomial gamma mixed model. 'PMM' is for fitting a Poisson gamma mixed model. 'NGLMM' is for fitting a negative binomial lognormal mixed model (the same model as that in the lme4 package). The default is 'NBGMM'.
method	'LN' or 'HL'. 'LN' is to use NEBULA-LN and 'HL' is to use NEBULA-HL. The default is 'LN'.
cutoff_cell	The data will be refit using NEBULA-HL to estimate both overdispersions if the product of the cells per subject and the estimated cell-level overdispersion parameter ϕ is smaller than cutoff_cell. The default is 20.
kappa	Please see the vignettes for more details. The default is 800.
opt	'lbfgs' or 'trust'. Specifying the optimization algorithm used in NEBULA-LN. The default is 'lbfgs'. If it is 'trust', a trust region algorithm based on the Hessian matrix will be used for optimization.
verbose	An optional logical scalar indicating whether to print additional messages. Default is FALSE.
cpc	A non-negative threshold for filtering low-expressed genes. Genes with counts per cell smaller than the specified value will not be analyzed.
covariance	If TRUE, nebula will output the covariance matrix for the estimated log(FC), which can be used for testing contrasts.

Value

summary: The estimated coefficient, standard error and p-value for each predictor.

overdispersion: The estimated cell-level and subject-level overdispersions σ^2 and ϕ^{-1}.

convergence: More information about the convergence of the algorithm for each gene. A value of -20 or -30 indicates a potential failure of the convergence.

algorithm: The algorithm used for analyzing the gene. More information can be found in the vignettes.

Examples

```
library(nebula)
data(sample_data)
pred = model.matrix(~X1+X2+cc,data=sample_data$pred)
re = nebula(count=sample_data$count,id=sample_data$sid,pred=pred)
```

---

sample_data   *An example data set for testing nebula*

Description

A dataset containing a count matrix, subject IDs, a data frame of predictors and scaling factors.

Usage

```
sample_data
```

Format

A list of four objects:

count A raw count matrix

sid A vector of subject IDs

pred A data frame of three predictors

offset A vector of scaling factors

Index