Package 'SDAMS'

November 4, 2025

Type Package

Title Differential Abundant/Expression Analysis for Metabolomics, Proteomics and single-cell RNA sequencing Data

Version 1.31.0

Date 2023-07-21

Author Yuntong Li <liyuntong0704@gmail.com>, Chi Wang <chi.wang@uky.edu>, Li Chen <lichenuky@uky.edu>

Maintainer Yuntong Li <liyuntong0704@gmail.com>

Depends R(>= 3.5), SummarizedExperiment

Suggests testthat

Imports trust, qvalue, methods, stats, utils

Description This Package utilizes a Semi-parametric Differential Abundance/expression analysis (SDA) method for metabolomics and proteomics data from mass spectrometry as well as single-cell RNA sequencing data. SDA is able to robustly handle non-normally distributed data and provides a clear quantification of the effect size.

License GPL

LazyLoad no

NeedsCompilation no

biocViews ImmunoOncology, DifferentialExpression, Metabolomics, Proteomics, MassSpectrometry, SingleCell

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

git_branch devel

git_last_commit b82dfb1

git_last_commit_date 2025-10-29

Repository Bioconductor 3.23

Date/Publication 2025-11-03

Contents

`SDAMS-package` *SDAMS package for differential abundance/expression analysis of Metabolomics, Proteomics and single-cell RNA sequencing data*

Description

SDAMS is an R package for differential abundance/expression analysis of metabolomics, proteomics and single-cell RNA sequencing data, and the main function for differential abundance/expression analysis is `SDA`. See the examples at `SDA` for basic analysis steps. SDAMS considers a two-part model, a logistic regression for the zero proportion and a semi-parametric log-linear model for the non-zero values.

Author(s)

Yuntong Li <liyuntong0704@gmail.com>, Chi Wang <chi.wang@uky.edu>, Li Chen <lichenuky@uky.edu>

References

Li, Y., Fan, T.W., Lane, A.N. et al. SDA: a semi-parametric differential abundance analysis method for metabolomics and proteomics data. BMC Bioinformatics 20, 501 (2019).

`dataInput` *Mass spectrometry data input*

Description

Two ways to input metabolomics or proteomics data from mass spectrometry or single-cell RNA sequencing data as `SummarizedExperiment`:

1. `createSEFromCSV` creates SummarizedExperiment object from csv files;
2. `createSEFromMatrix` creates SummarizedExperiment object from separate matrices: one for feature/gene data and the other one for colData.

Usage

```
createSEFromCSV(featurePath, colDataPath, rownames1 = 1, rownames2 = 1,
                  header1 = TRUE, header2 = TRUE)

createSEFromMatrix(feature, colData)
```

Arguments

featurePath	path for feature/gene data.
colDataPath	path for colData.
rownames1	indicator for feature/gene data with row names. If NULL, row numbers are automatically generated.
rownames2	indicator for colData with row names. If NULL, row numbers are automatically generated.
header1	a logical value indicating whether the first row of feature/gene is column names. The default value is TRUE.
header2	a logical value indicating whether the first row of colData is column names. The default value is TRUE. If colData input is a vector, set to False.
feature	a matrix with row being features/genes and column being subjects/cells.
colData	a column type data containing information about the subjects/cells.

Value

An object of SummarizedExperiment class.

Author(s)

Yuntong Li <yuntong.li@uky.edu>, Chi Wang <chi.wang@uky.edu>, Li Chen <lichenuky@uky.edu>

See Also

SDA input requires an object of SummarizedExperiment class.

Examples

```
# ---------- csv input -------------
directory1 <- system.file("extdata", package = "SDAMS", mustWork = TRUE)
path1 <- file.path(directory1, "ProstateFeature.csv")
directory2 <- system.file("extdata", package = "SDAMS", mustWork = TRUE)
path2 <- file.path(directory2, "ProstateGroup.csv")

exampleSE <- createSEFromCSV(path1, path2)
exampleSE

# ---------- matrix input -------------
set.seed(100)
featureInfo <- matrix(runif(800, -2, 5), ncol = 40)
featureInfo[featureInfo<0] <- 0
rownames(featureInfo) <- paste("gene", 1:20, sep = '')
colnames(featureInfo) <- paste('cell', 1:40, sep = '')
groupInfo <- data.frame(grouping=matrix(sample(0:1, 40, replace = TRUE),
                        ncol = 1))
rownames(groupInfo) <- colnames(featureInfo)

exampleSE <- createSEFromMatrix(feature = featureInfo, colData = groupInfo)
exampleSE
```

`exampleData` *Two example datasets for SDAMS package*

Description

SDAMS package provides two types of example datasets: one is prostate cancer proteomics data from mass spectrometry and the other one is single-cell RNA sequencing data.

1. For prostate cancer proteomics data, it is from the human urinary proteome database(http://mosaiques-diagnostics.de/mosaiques-diagnostics/human-urinary-proteom-database). There are 526 prostate cancer subjects and 1503 healthy subjects. A total of 5605 proteomic features were measured for each subject. For illustration purpose, we took a 10% subsample randomly from this real data. This example data contains 560 proteomic features for 202 experimental subjects with 49 prostate cancer subjects and 153 healthy subjects. SDAMS package provides two different kinds of data formats for prostate cancer proteomics data. `exampleSumExp.rda` is an object of `SummarizedExperiment` class which stores the information of both proteomic features and experimental subjects. `ProstateFeature.csv` contains a matrix-like proteomic feature data and `ProstateGroup.csv` contains a single column of experimental subject group data.
2. For single cell RNA sequencing data, it is in the form of transcripts per kilobase million (TPM). The count data can be found at Gene Expression Omnibus (GEO) database with Accession No. GSE29087. There are 92 single cells (48 mouse embryonic stem (ES) cells and 44 mouse embryonic fibroblasts (MEF)) that were analyzed. The example data provided by SDAMS contains 10% of genes which are randomly sampled from the raw dataset. `exampleSingleCell.rda` is an object of `SummarizedExperiment` class which stores the information of both gene expression and cell information.

Usage

```
data(exampleSumExp)
data(exampleSingleCell)
```

Value

An object of `SummarizedExperiment` class.

References

Siwy, J., Mullen, W., Golovko, I., Franke, J., and Zurbig, P. (2011). Human urinary peptide database for multiple disease biomarker discovery. PROTEOMICS-Clinical Applications 5, 367-374.

Islam, S., Kjallquist, U., Moliner, A., Zajac, P., Fan, J. B., Lonnerberg, P., & Linnarsson, S. (2011). Characterization of the single-cell transcriptional landscape by highly multiplex RNA-seq. Genome research, 21(7), 1160-1167.

See Also

SDA

Examples

```
#------ load data --------
data(exampleSumExp)
exampleSumExp
feature = assay(exampleSumExp) # access feature data
group = colData(exampleSumExp)$grouping # access grouping information
SDA(exampleSumExp)
```

SDA *Semi-parametric differential abuandance/expression analysis*

Description

This function considers a two-part semi-parametric model for metabolomics, proteomics and single-cell RNA sequencing data. A kernel-smoothed method is applied to estimate the regression coefficients. And likelihood ratio test is constructed for differential abundance/expression analysis.

Usage

```
SDA(sumExp, VOI = NULL, ...)
```

Arguments

sumExp	An object of 'SummarizedExperiment' class.
VOI	Variable of interest. Default is NULL, when there is only one covariate, otherwise it must be one of the column names in colData.
...	Additional arguments passed to qvalue.

Details

The differential abundance/expression analysis is to compare metabolomic or proteomic profiles or gene expression between different experimental groups, which utilizes a two-part model: a logistic regression model to characterize the zero proportion and a semi-parametric model to characterize non-zero values. Let Y_i be the random variable and X_i is a vector of covariates. This two-part model has the following form:

$$\log(\frac{\pi_i}{1-\pi_i}) = \gamma_0 + \boldsymbol{\gamma}\boldsymbol{X}_i$$
$$\log(Y_i) = \boldsymbol{\beta}\boldsymbol{X}_i + \varepsilon_i$$

where $\pi_i = Pr(Y_i = 0)$. The model parameters $\boldsymbol{\gamma}$ quantify the covariates effects on the fraction of zero values and γ_0 is the intercept. $\boldsymbol{\beta}$ are the model parameters quantifying the covariates effects on the non-zero values, ε_i are independent error terms with a common but completely unspecified density function f.

For differential abundant analysis on data from mass spectrometry, Y_i represents the abundance of certain feature for subject i, π_i is the probability of point mass. $\boldsymbol{X}_i = (X_{i1}, X_{i2}, ..., X_{iQ})^T$ is a

Q-vector of covariates that specifies the treatment conditions applied to subject i. The corresponding Q-vector of model parameters $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, ..., \gamma_Q)^T$ and $\boldsymbol{\beta} = (\beta_1, \beta_2, ..., \beta_Q)^T$ quantify the covariates effects for certain feature. Hypothesis testing on the effect of the qth covariate on certain feature is performed by assessing γ_q and β_q. Consider the null hypothesis H_0: $\gamma_q = 0$ and $\beta_q = 0$ against alternative hypothesis H_1: at least one of the two parameters is non-zero. We also consider the hypotheses for testing $\gamma_q = 0$ and $\beta_q = 0$ individually.

For differential expression analysis on single-cell RNA sequencing data, Y_i represents represents the expression (TPM value) of certain gene in ith cell, π_i is the drop-out probability. $\boldsymbol{X}_i = (Z_i, \boldsymbol{W}_i)^T$ is a vector of covariates with Z_i being a binary indicator of the cell population under comparison and $\boldsymbol{W}_i$ being a vector of other covariates, e.g. cell size, and $\boldsymbol{\gamma} = (\gamma_Z, \boldsymbol{\gamma}_W)$ and $\boldsymbol{\beta} = (\beta_Z, \boldsymbol{\beta}_W)$ are model parameters. Hypothesis testing on the effect of different cell subpopulations on certain gene is performed by assessing γ_Z and β_Z. For each gene, the likelihood ratio test is performed on the null hypothesis H_0: $\gamma_Z = 0$ and $\beta_Z = 0$ against alternative hypothesis H_1: at least one of the two parameters is non-zero. We also consider the hypotheses for testing $\gamma_Z = 0$ and $\beta_Z = 0$ individually.

The p-value is calculated based on an asympotic chi-squared distribution. To adjust for multiple comparisons across features, the false discovery discovery rate (FDR) q-value is calculated based on the qvalue function in R/Bioconductor.

Value

A list containing the following components:

`gamma`	a matrix of point estimators for γ_g in the logistic model (binary part)
`beta`	a matrix of point estimators for β_g in the semi-parametric model (non-zero part)
`pv_gamma`	a matrix of one-part p-values for γ_g
`pv_beta`	a matrix of one-part p-values for β_g
`qv_gamma`	a matrix of one-part q-values for γ_g
`qv_beta`	a matrix of one-part q-values for β_g
`pv_2part`	a matrix of two-part p-values for overall test
`qv_2part`	a matrix of two-part q-values for overall test
`feat.names`	a vector of feature/gene names

Author(s)

Yuntong Li <yuntong.li@uky.edu>, Chi Wang <chi.wang@uky.edu>, Li Chen <lichenuky@uky.edu>

Examples

```
##--------- load data ------------
data(exampleSumExp)

results = SDA(exampleSumExp)

##------ two part q-values -------
results$qv_2part
```

Index