Package 'SDAMS'

April 16, 2019

Type Package

Title Differential Abundant Analysis for Metabolomics and Proteomics Data

Version 1.2.1

Date 2018-10-17

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

Maintainer Yuntong Li <yuntong.li@uky.edu>

Depends R(>= 3.5), SummarizedExperiment

Suggests testthat

Imports trust, qvalue, methods, stats, utils

Description This Package utilizes a Semi-parametric Differential Abundance analysis (SDA) method for metabolomics and proteomics data from mass spectrometry. 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

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

git_branch RELEASE_3_8

git_last_commit ec66e1e

git_last_commit_date 2019-01-04

Date/Publication 2019-04-15

R topics documented:

SDAMS-package *SDAMS package for differential abundance analysis of Metabolomics and Proteomics data from mass spectrometry*

Description

SDAMS is an R package for differential abundance analysis of metabolomics and proteomics data, and the main function for differential abundance 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 <yuntong.li@uky.edu>, Chi Wang <chi.wang@uky.edu>, Li Chen <lichenuky@uky.edu>

References

Yuntong Li, Teresa W.M. Fan, Andrew N. Lane, Woo-Young Kang, Susanne M. Arnold, Arnold J. Stromberg, Chi Wang and Li Chen: A Two-Part Semi-Parametric Model for Metabolomics and Proteomics Data. (Manuscript)

dataInput *Mass spectrometry data input*

Description

Two ways to input metabolomics or proteomics data from mass spectrometry as SummarizedExperiment:

1. createSEFromCSV creates SummarizedExperiment object from csv files;
2. createSEFromMatrix creates SummarizedExperiment object from separate matrices: one for feature 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 data.
colDataPath	path for colData.
rownames1	indicator for feature 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 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 and column being subjects.
colData	a column type data containing information about the subjects.

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("feature", 1:20, sep = '')
colnames(featureInfo) <- paste('subject', 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 *An example data for the SDAMS package*

Description

SDAMS package provides two different data formats for prostate cancer proteomics data, which 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. `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.

Usage

```
data(exampleSumExp)
```

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.

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 analysis*

Description

This function considers a two-part semi-parametric model for metabolomics and proteomics data. A kernel-smoothed method is applied to estimate the regression coefficients. And likelihood ratio test is constructed for differential abundance 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 analysis is to compare metabolomic or proteomic profiles 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_{ig} be the random variable representing the abundance of feature g in subject i. This two-part model has the following form:

$$\log(\frac{\pi_{ig}}{1-\pi_{ig}}) = \gamma_{0g} + \boldsymbol{\gamma}_g \boldsymbol{X}_i$$

$$\log(Y_{ig}) = \boldsymbol{\beta}_g \boldsymbol{X}_i + \varepsilon_{ig}$$

where $\pi_{ig} = Pr(Y_{ig} = 0)$ be the probability of point mass, $\boldsymbol{X}_i = (X_{i1}, X_{i2}, ..., X_{iQ})^T$ is a Q-vector covariates that specifies the treatment conditions applied to subject i. The corresponding Q-vector of model parameters $\boldsymbol{\gamma}_g = (\gamma_{1g}, \gamma_{2g}, ..., \gamma_{Qg})^T$ quantify the covariates effects on the fraction of zero values for feature g and γ_{0g} is the intercept. $\boldsymbol{\beta}_g = (\beta_{1g}, \beta_{2g}, ..., \beta_{Qg})^T$ is a Q-vector of model parameters quantifying the covariates effects on the non-zero values for the feature. And ε_{ig} are independent error terms with a common but completely unspecified density function f_g.

Hypothesis testing on the effect of the qth covariate on the gth feature is performed by assessing γ_{qg} and β_{qg}. Consider the null hypothesis H_0: γ_{qg} and β_{qg} against alternative hypothesis H_1: at least one of the two parameters is non-zero. The p-value is calculated based on a chi-square distribution with 2 degrees of freedom. 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 vector of point estimators for γ_g in the logistic model (binary part)
`beta`	a vector of point estimators for β_g in the semi-parametric model (non-zero part)
`pv_gamma`	a vector of one-part p-values for γ_g
`pv_beta`	a vector of one-part p-values for β_g
`qv_gamma`	a vector of one-part q-values for γ_g
`qv_beta`	a vector of one-part q-values for β_g
`pv_2part`	a vector of two-part p-values for overall test
`qv_2part`	a vector of two-part q-values for overall test
`feat.names`	a vector of feature 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