Package 'TOAST'

April 15, 2020

Type Package

Title Tools for the analysis of heterogeneous tissues

Version 1.0.0

Description This package is devoted to analyzing high-throughput data (e.g. gene expression microarray, DNA methylation microarray, RNA-seq) from complex tissues. Current functionalities include 1. detect cell-type specific or cross-cell type differential signals 2. improve variable selection in reference-free deconvolution.

Author Ziyi Li and Hao Wu

Maintainer Ziyi Li <ziyi.li@emory.edu>

License GPL-2

Encoding UTF-8

LazyData false

Depends R (>= 3.6), RefFreeEWAS, EpiDISH

biocViews DNAMethylation, GeneExpression, DifferentialExpression, DifferentialMethylation, Microarray, GeneTarget, Epigenetics, MethylationArray

BugReports https://github.com/ziyili20/TOAST/issues

Imports stats, methods, SummarizedExperiment

Suggests BiocStyle, knitr, rmarkdown, csSAM, gplots, matrixStats, Matrix

VignetteBuilder knitr

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

git_branch RELEASE_3_10

git_last_commit 5525311

git_last_commit_date 2019-10-29

Date/Publication 2020-04-14

R topics documented:

`assignCellType` *Align cell types when reference proportions are known*

Description

Align target proportions with reference proportions by pearson correlation coefficients.

Usage

```
assignCellType(input,reference)
```

Arguments

`input`	Input proportiona matrix of dimension N by K.
`reference`	Reference proportion matrix of dimension N by K.

Value

The aligned proportion matrix, following the cell type ordering of reference proportion matrix.

Author(s)

Ziyi Li <ziyi.li@emory.edu>

References

Ziyi Li, Zhijin Wu, Peng Jin, Hao Wu. "Dissecting differential signals in high-throughput data from complex tissues."

Examples

```
## generate estimated proportion matrix
estProp <- matrix(abs(runif(50*4,0,1)), 50, 4)
estProp <- sweep(estProp, 1, rowSums(estProp), "/")

## generate reference proportion matrix
refProp <- matrix(abs(runif(50*4,0,1)), 50, 4)
refProp <- sweep(refProp, 1, rowSums(refProp), "/")

estProp_aligned = assignCellType(input = estProp,
reference = refProp)
```

csDeconv *Improve reference-free deconvolution using cross-cell type differential analysis*

Description

This function improve the feature selection in reference-free deconvolution through cross-cell type differential analysis

Usage

```
csDeconv(Y_raw, K, FUN, nMarker = 1000,
InitMarker = NULL, TotalIter = 30, bound_negative = FALSE)
```

Arguments

Y_raw	A G*N matrix, G is the number of features, N is the number of subjects; or a SummarizedExperiment object.
K	The number of cell types. Need to be specified a priori.
FUN	The reference-free deconvolution function, this function should take Y_raw and K, and the return values should be a N by K proportion matrix. N is the number of samples and K is the number of cell types. Default function is a wrapper of the RefFreeCellMix() function from CRAN package RefFreeEWAS.
nMarker	The number of markers used in the deconvolution. Default is 1000.
InitMarker	A vector of length L to represent the selection of inital markers. L should be equal or smaller than G. If G is large, it is recommended that L is much smaller than G. If not specified, the most variable nMarker features will be used.
TotalIter	The total number of iterations of applying cross-cell type differential analysis. Default is 30.
bound_negative	Whether to bound all negative parameter estimators to zero.

Value

allProp	A list of estimated proportions from all iterations.
allRMSE	A vector of root mean squared errors (RMSE) from all iteratoins.
estProp	A N*K matrix representing the mixture proportions of K cell types in N subjects, chosen from allProp with the smallest RMSE.

Author(s)

Ziyi Li <ziyi.li@emory.edu>

References

Ziyi Li and Hao Wu. "Improving reference-free cell composition estimation by cross-cell type differential analysis".

Examples

```
Y_raw <- abs(matrix(runif(10000*20, 0,1),10000,20))
K <- 3

## wrap your reference-free
## deconvolution method into a function
## this function should take Y and K as input
## and output a N by K proprotion matrix
## here we use RefFreeCellMix() as an example
outT <- csDeconv(Y_raw, K)

RefFreeCellMix_wrapper <- function(Y, K){
     outY = RefFreeEWAS::RefFreeCellMix(Y,
     mu0=RefFreeEWAS::RefFreeCellMixInitialize(Y,
     K = K))
     Prop0 = outY$Omega
     return(Prop0)
}

outT <- csDeconv(Y_raw, K,
            FUN = RefFreeCellMix_wrapper)
```

---

csTest *Testing differential signals for specified phenotype and cell type(s).*

Description

This function conducts statistical tests for specified phenotype and cell type(s).

Usage

```
csTest(fitted_model, coef = NULL, cell_type = NULL,
       contrast_matrix = NULL, var_shrinkage = TRUE,
       verbose = TRUE, sort = TRUE)
```

Arguments

fitted_model	The output from fitModel() function.
coef	A phenotype name, e.g. "disease", or a vector of contrast terms, e.g. c("disease", "case", "control").
cell_type	A cell type name, e.g. "celltype1", or "neuron". If cell_type is NULL or specified as "ALL", compound effect of coef in all cell types will be tested.
contrast_matrix	If contrast_matrix is specified, coef and cell_type will be ignored! A matrix (or a vector) to specify contrast, e.g., cmat <- matrix(0, 2, 6); cmat[1,3] <- 1: cmat[2,4] <- 1 is to test whether the 3rd parameter and 4th parameter are zero simultaneously i.e. beta3 = beta4 = 0.
var_shrinkage	Whether to apply shrinkage on estimated MSE or not. Applying shrinkage helps remove extremely small variance estimation and stablize statistics.
verbose	A boolean parameter. Testing information will be printed if verbose = TRUE.
sort	A boolean parameter. The output results will be sorted by p value if sort = TRUE.

Value

A matrix including the results from testing the phenotype in specified cell type(s).

Author(s)

Ziyi Li <ziyi.li@emory.edu>

References

Ziyi Li, Zhijin Wu, Peng Jin, Hao Wu. "Dissecting differential signals in high-throughput data from complex tissues."

Examples

```
N <- 300 # simulation a dataset with 300 samples
K <- 3 # 3 cell types
P <- 500 # 500 features

### simulate proportion matrix
Prop <- matrix(runif(N*K, 10,60), ncol=K)
Prop <- sweep(Prop, 1, rowSums(Prop), FUN="/")
colnames(Prop) <- c("Neuron", "Astrocyte", "Microglia")

### simulate phenotype names
design <- data.frame(disease=factor(sample(0:1,
                              size = N,replace=TRUE)),
                     age=round(runif(N, 30,50)),
                     race=factor(sample(1:3, size = N,replace=TRUE)))
Y <- matrix(rnorm(N*P, N, P), ncol = N)

### generate design matrix and fit model
Design_out <- makeDesign(design, Prop)
fitted_model <- fitModel(Design_out, Y)

### check the names of cell types and phenotypes
fitted_model$all_cell_types
fitted_model$all_coefs

### detect age effect in neuron
test <- csTest(fitted_model, coef = "age",
cell_type = "Neuron", contrast_matrix = NULL)

## coef can be specified in different ways:
#### jointly test a phenotype:
test <- csTest(fitted_model, coef = "age",
cell_type = "joint", contrast_matrix = NULL)

#### if I do not specify cell_type
test <- csTest(fitted_model, coef = "age",
cell_type = NULL, contrast_matrix = NULL)
## this is exactly the same as
test <- csTest(fitted_model, coef = "age",
contrast_matrix = NULL)

#### other examples
test <- csTest(fitted_model, coef = "race",
```

```
cell_type = "Astrocyte", contrast_matrix = NULL)
test <- csTest(fitted_model, coef = "age",
cell_type = "Microglia", contrast_matrix = NULL)

#### specify contrast levels
test <- csTest(fitted_model, coef = c("race", 3, 2),
cell_type = "Neuron", contrast_matrix = NULL)
#### specify contrast levels in all cell types
test <- csTest(fitted_model, coef = c("race", 3, 2),
cell_type = "joint", contrast_matrix = NULL)

#### csTest can tolerate different ways of specifying contrast level
#### note race=1 is used as reference when fitting model
#### we can here specify race=2 as reference
test <- csTest(fitted_model, coef = c("race", 1, 2),
cell_type = "Neuron", contrast_matrix = NULL)
## get exactly the same results as
test <- csTest(fitted_model, coef = c("race", 2, 1),
cell_type = "Neuron", contrast_matrix = NULL)

#### specify a contrast matrix:
cmatrix = rep(0,15)
cmatrix[c(4,5)] = c(1,-1)
test <- csTest(fitted_model, coef = NULL,
cell_type = NULL, contrast_matrix = cmatrix)
#### specific a contrast matrix with two rows:
cmatrix = matrix(rep(0,30),2,15)
cmatrix[1,4] = 1
cmatrix[2,5] = 1
test <- csTest(fitted_model, coef = NULL,
contrast_matrix = cmatrix)
```

`DEVarSelect` *Feature selection for reference-free deconvolution using cross-cell type differential analysis*

Description

This function selects cross-cell type differential features for reference-free deconvolution.

Usage

```
DEVarSelect(Y_raw, Prop0, nMarker, bound_negative)
```

Arguments

`Y_raw`	A data matrix containing P features and N samples; or a SummarizedExperiment object.
`Prop0`	A N by K proportion matrix with K as number of cell types.
`nMarker`	Number of markers selected.
`bound_negative`	Whether to bound all negative parameter estimators to zero.

Value

Selected markers using cross-cell type differential analysis.

Author(s)

Ziyi Li <ziyi.li@emory.edu>

References

Ziyi Li, Zhijin Wu, Peng Jin, Hao Wu. "Dissecting differential signals in high-throughput data from complex tissues."

Examples

```
Y_raw <- matrix(runif(5000*20, 0, 1), 5000, 20)
tmp <- matrix(runif(20*4), 20, 4)
Prop0 <- sweep(tmp, 1, rowSums(tmp), "/")
varlist <- DEVarSelect(Y_raw, Prop0,
                       nMarker=1000,
                       bound_negative=FALSE)
```

`findRefinx` *findRefinx*

Description

Find index for marker genes with largest coefficient of variation based on raw data.

Usage

```
findRefinx(rawdata, nmarker=1000, sortBy = "var")
```

Arguments

`rawdata`	A data matrix with rows representing features and columns represeting samples; or a SummarizedExperiment object.
`nmarker`	Desired number of markers after selection. Default is 1000.
`sortBy`	Desired method to select features. "var" represents selecting by largest variance. "cv" represents selecting by largest coefficients of variation. Default is "var".

Value

A vector of index for the selected markers.

Author(s)

Ziyi Li <ziyi.li@emory.edu>

References

Ziyi Li, Zhijin Wu, Peng Jin, Hao Wu. "Dissecting differential signals in high-throughput data from complex tissues."

Examples

```
Y_raw <- matrix(runif(5000*20, 0, 1), 5000, 20)
idx <- findRefinx(Y_raw, nmarker=500)
idx2 <- findRefinx(Y_raw, nmarker=500, sortBy = "cv")
```

`fitModel` *Fit model with proportions and phenotypes.*

Description

This function receives design matrix from makeDesign() and fits the model including all cell types and phenotypes.

Usage

```
fitModel(Design_out, Y)
```

Arguments

`Design_out`	The output from function makeDesign().
`Y`	A G*N matrix, G is the number of features, N is the number of subjects; or a SummarizedExperiment object.

Value

`Design_out`	The input Design_out object.
`N`	Number of samples from matrix Y.
`coefs`	Estimated coefficients (beta) in the model.
`coefs_var`	Estimated variance of the coefficients (beta variance) in the model.
`Y`	Observation Y matrix.
`Ypred`	Predicted Y from the fitted model.
`all_coefs`	The names of all phenotypes.
`all_cell_types`	The names of all cell types.
`MSE`	Estimated mean squared error.
`model_names`	The names of all terms in the fitted model.

Author(s)

Ziyi Li <ziyi.li@emory.edu>

References

Ziyi Li, Zhijin Wu, Peng Jin, Hao Wu. "Dissecting differential signals in high-throughput data from complex tissues."

Examples

```
N = 300 # simulation a dataset with 300 samples
K = 3 # 3 cell types
P <- 500 # 500 features

### simulate proportion matrix
Prop = matrix(runif(N*K, 10,60), ncol=K)
Prop = sweep(Prop, 1, rowSums(Prop), FUN="/")
colnames(Prop) = c("Neuron", "Astrocyte", "Microglia")
Y <- matrix(rnorm(N*P, N, P), ncol = N)

### simulate phenotype names
design <- data.frame(disease=factor(sample(0:1,
                    size = N,replace=TRUE)),
                    age=round(runif(N, 30,50)),
                    race=factor(sample(1:3, size = N,replace=TRUE)))
Design_out <- makeDesign(design, Prop)

### fit model
fitted_model <- fitModel(Design_out, Y)
```

---

makeDesign *Generate design matrix from input phenotypes and proportions.*

Description

This function generate design matrix and make preparations for following fitModel and csTest.

Usage

```
makeDesign(design, Prop)
```

Arguments

design	A N by P phenotype matrix, with rows as samples and columns as phenotypes (e.g. age, gender, disease, etc.).
Prop	A N by K proportion matrix, with rows as samples and columns as cell types

Value

design_matrix	A comprehensive design matrix incorporated phenotype and proportion information.
Prop	The input proportion matrix.
design	The input design/phenotype matrix.
all_coefs	The names of all phenotypes.
all_cell_types	The names of all cell types.
formula	The formula of the tested model, including all phenotypes, cell types and interaction terms.

Author(s)

Ziyi Li <ziyi.li@emory.edu>

References

Ziyi Li, Zhijin Wu, Peng Jin, Hao Wu. "Dissecting differential signals in high-throughput data from complex tissues."

Examples

```
N = 300 # simulation a dataset with 300 samples
K = 3 # 3 cell types

### simulate proportion matrix
Prop = matrix(runif(N*K, 10,60), ncol=K)
Prop = sweep(Prop, 1, rowSums(Prop), FUN="/")
colnames(Prop) = c("Neuron", "Astrocyte", "Microglia")

### simulate phenotype names
design <- data.frame(disease=factor(sample(0:1, size = N,replace=TRUE)),
                     age=round(runif(N, 30,50)),
                     race=factor(sample(1:3, size = N,replace=TRUE)))
Design_out <- makeDesign(design, Prop)
```

RA_100samples *An example dataset for cellular proportion estimation and multiple factor design*

Description

The dataset contains normalized beta valutes for 3000 CpGs from 100 samples (50 Rheumatoid arthritis patients and 50 controls) and their phenotypes (disease status, age, and gender). The dataset also contains a sub-setted blood reference matrix for the matched 3000 CpGs. This data was obtained and processed based on GSE42861.

Usage

```
data("RA_100samples")
```

References

Liu Y, Aryee MJ, Padyukov L, Fallin MD et al. Epigenome-wide association data implicate DNA methylation as an intermediary of genetic risk in rheumatoid arthritis. Nat Biotechnol 2013 Feb;31(2):142-7. PMID: 23334450

Examples

```
data(RA_100samples)
RA_100samples$Y_raw[1:5,1:5]
head(RA_100samples$Pheno)
head(RA_100samples$Blood_ref)
```

Index