Package ‘discordant’

January 31, 2024

Version 1.26.0

Title The Discordant Method: A Novel Approach for Differential Correlation

Depends R (&gt;= 4.1.0)

Description Discordant is an R package that identifies pairs of features that correlate differently between phenotypic groups, with application to -omics data sets. Discordant uses a mixture model that “bins” molecular feature pairs based on their type of coexpression or coabundance. Algorithm is explained further in "Differential Correlation for Sequencing Data" (Siska et al. 2016).

Encoding UTF-8

biocViews ImmunoOncology, BiologicalQuestion, StatisticalMethod, mRNAMicroarray, Microarray, Genetics, RNASeq

Suggests BiocStyle, knitr, testthat (&gt;= 3.0.0)

Imports Rcpp, Biobase, stats, biwt, gtools, MASS, tools, dplyr, methods, utils

License GPL-3

URL https://github.com/siskac/discordant

NeedsCompilation yes

LinkingTo Rcpp

VignetteBuilder knitr

RoxygenNote 7.1.2

Config/testthat/edition 3

git_url https://git.bioconductor.org/packages/discordant
git_branch RELEASE_3_18
git_last_commit d10110d
git_last_commit_date 2023-10-24

Repository Bioconductor 3.18

Date/Publication 2024-01-31
createVectors

Description

Calculates correlation coefficients based on two groups of -omics bivariate data. Currently, only two groups of samples can be specified. Used to make input for discordantRun().

Usage

createVectors(  
x,  
y = NULL,  
groups,  
cor.method = c("spearman", "pearson", "bwmc", "sparcc")  
)

Arguments

x ExpressionSet of -omics data
y Optional second ExpressionSet of -omics data, induces dual -omics analysis
groups n-length vector of 1s and 2s matching samples belonging to groups 1 and 2
cor.method Correlation method to measure association. Options are "spearman", "pearson", "bwmc" and "sparcc"
Details

Creates vectors of correlation coefficients based on feature pairs within x or between x and y. The names of the vectors are the feature pairs taken from x and y.

Value

List of two named numeric vectors. Vectors give the correlation coefficients for groups 1 and 2 respectively, and vector names give the each feature for the respective feature pair separated by an underscore.

Author(s)

Charlotte Siska <siska.charlotte@gmail.com>
Max McGrath <max.mcgrath@ucdenver.edu>

References


Examples

```r
## load data
data("TCGA_GBM_miRNA_microarray")
data("TCGA_GBM_transcript_microarray")
print(colnames(TCGA_GBM_transcript_microarray)) # look at groups
groups <- c(rep(1,10), rep(2,20)) # transcript-transcript pairs
vectors <- createVectors(TCGA_GBM_transcript_microarray,
                          groups = groups, cor.method = c("pearson"))

# miRNA-transcript pairs
vectors <- createVectors(TCGA_GBM_transcript_microarray,
                          TCGA_GBM_miRNA_microarray, groups = groups)
```

Description

Discordant is a method to determine differential correlation of molecular feature pairs from -omics data using mixture models. Algorithm is explained further in Siska et al.
discordantRun

Author(s)
Charlotte Siska
Max McGrath
Katerina Kechris

discordantRun  Run Discordant Algorithm

Description
Runs discordant algorithm on two vectors of correlation coefficients.

Usage

discordantRun(
  v1,
  v2,
  x,
  y = NULL,
  transform = TRUE,
  subsampling = FALSE,
  subSize = NULL,
  iter = 100,
  components = 3
)

Arguments

  v1  Vector of correlation coefficients in group 1
  v2  Vector of correlation coefficients in group 2
  x   ExpressionSet of -omics data
  y   ExpressionSet of -omics data, induces dual -omics analysis
  transform  If TRUE v1 and v2 will be Fisher transformed
  subsampling  If TRUE subsampling will be run
  subSize   Indicates how many feature pairs to be used for subsampling. Default is the feature size in x
  iter     Number of iterations for subsampling. Default is 100
  components  Number of components in mixture model.
Details

The discordant algorithm is based on a Gaussian mixture model. If there are three components, correlation coefficients are clustered into negative correlations (−), positive correlations (+) and no correlation (0). If there are five components, then there are two more classes for very negative correlation (−−) and very positive correlations (++). All possible combinations for these components are made into classes. If there are three components, there are 9 classes. If there are five components, there are 25 classes.

The posterior probabilities for each class are generated and outputted into the value probMatrix. The value probMatrix is a matrix where each column is a class and each row is a feature pair. The values discordPPVector and discordPPMatrix are the summed differential correlation posterior probability for each feature pair. The values classVector and classMatrix are the class with the highest posterior probability for each feature pair.

Value

- **discordPPVector** Vector of differentially correlated posterior probabilities.
- **discordPPMatrix** Matrix of differentially correlated posterior probabilities where rows and columns reflect features
- **classVector** Vector of classes that have the highest posterior probability
- **classMatrix** Matrix of classes that have the highest posterior probability where rows and columns reflect features
- **probMatrix** Matrix of posterior probabilities where rows are each molecular feature pair and columns are nine different classes
- **loglik** Final log likelihood

Author(s)

Charlotte Siska <siska.charlotte@gmail.com>
Max McGrath <max.mcgrath@ucdenver.edu>

References


Lai Y, Adam B-I, Podolsky R, She J-X. A mixture model approach to the tests of concordance and discordance between two large-scale experiments with two sample groups. (2007) Bioinformatics 23, 1243-1250.

Examples

```r
# Load Data
data(TCGA_GBM_miRNA_microarray)
data(TCGA_GBM_transcript_microarray)
```
print(colnames(TCGA_GBM_transcript_microarray)) # look at groups
groups <- c(rep(1,10), rep(2,20))

## DC analysis on only transcripts pairs
vectors <- createVectors(TCGA_GBM_transcript_microarray,
groups = groups)
result <- discordantRun(vectors$v1, vectors$v2,
TCGA_GBM_transcript_microarray)

## DC analysis on miRNA-transcript pairs
vectors <- createVectors(TCGA_GBM_transcript_microarray,
TCGA_GBM_miRNA_microarray, groups = groups,
cor.method = c("pearson"))
result <- discordantRun(vectors$v1, vectors$v2,
TCGA_GBM_transcript_microarray,
TCGA_GBM_miRNA_microarray)

---

**fishersTrans**  
*Fisher Transformation of Pearson Correlation Coefficients to Z Scores*

**Description**
Transforms Pearson’s correlation coefficients into z scores using Fisher’s method.

**Usage**
`fishersTrans(rho)`

**Arguments**
- `rho` Integer or numeric vector of Pearson’s correlation coefficients

**Details**
Fisher’s transformation is when correlation coefficients are transformed into a z score. These z scores have an approximately normal distribution.

**Value**
Returns Fisher-transformed correlation coefficients

**References**
Fisher, R.A. (1915). "Frequency distribution of the values of the correlation coefficient in samples of an indefinitely large population". Biometrika (Biometrika Trust) 10 (4).
splitMADOutlier

Examples

```r
## Create integer or list of Pearson's correlation coefficients.
library(MASS)
rhoV <- as.vector(cor(t(mvrnorm(10, rep(3,100), diag(100)))))

## Determine Fisher-Transformed z scores of rho
zV <- fishersTrans(rhoV)
```

Description

Identify features with outliers using left and right median absolute deviation (MAD).

Usage

```r
splitMADOutlier(mat, filter0 = TRUE, threshold = 2)
```

Arguments

- `mat`: m by n matrix of -omics data, where rows are features and columns samples.
- `filter0`: Option to filter out features if they have at least one 0 value. Default is TRUE.
- `threshold`: Threshold of how many MADs outside the left or right median is used to determine features with outliers.

Details

The purpose of this function is to determine outliers in non-symmetric distributions. The distribution is split by the median. Outliers are identified by being however many median absolute deviations (MAD) from either split distribution.

Value

- `mat.filtered`: Input matrix where features with outliers filtered out.
- `index`: Index of features that have no outliers.

References


Examples

```r
## Simulate matrix of continuous -omics data.
data(TCGA_Breast_miRNASeq)

## Filter matrix based on outliers.
mat.filtered <- splitMADOutlier(TCGA_Breast_miRNASeq)$mat.filtered
```

---

**TCGA_Breast_miRNASeq**  
*Example breast miRNA-Seq count dataset.*

---

**Description**

This dataset contains TMM normalized voom-transformed miRNA count values from miRNASeq that was taken from the Cancer Genome Atlas, or TCGA. The dataset has 100 miRNA and 57 samples. The original dataset has 212 miRNA and 57 samples.

**Usage**

```r
data(TCGA_Breast_miRNASeq)
```

**Format**

An ExpressionSet with 100 features, 57 samples

**Source**


**References**


**Examples**

```r
data(TCGA_Breast_miRNASeq)
```
**Description**

This dataset contains TMM normalized voom-transformed miRNA count values from miRNASeq that was taken from the Cancer Genome Atlas, or TCGA. The dataset has 100 miRNA and 57 samples. The original dataset has 212 miRNA and 57 samples.

**Usage**

```r
data(TCGA_Breast_miRNASeq_voom)
```

**Format**

An ExpressionSet with 100 features and 57 samples

**Source**

http://cancergenome.nih.gov/

**References**


**Examples**

```r
data(TCGA_Breast_miRNASeq_voom)
```

---

**Description**

This dataset contains TMM normalized RNA count values from RNASeq that was taken from the Cancer Genome Atlas, or TCGA. It has 100 features and 57 samples. The original dataset had 17972 features and 57 samples.

**Usage**

```r
data(TCGA_Breast_RNASeq)
```
Description

This dataset contains TMM normalized voom-transformed RNA count values from RNASeq that was taken from the Cancer Genome Atlas, or TCGA.

Usage

data(TCGA_Breast_RNASeq_voom)

Format

An ExpressionSet with 100 features and 57 samples

Source

http://cancergenome.nih.gov/

References


Examples

data(TCGA_Breast_RNASeq_voom)
TCGA_GBM_miRNA_microarray

TCGA Glioblastoma Multiforme miRNA Sample Dataset

Description
This dataset contains miRNA expression values from a microarray that was taken from the Cancer Genome Atlas, or TCGA. It has 10 features and 30 samples. The original dataset had 331 features and 30 samples.

Usage
data(TCGA_GBM_miRNA_microarray)

Format
An ExpressionSet with 10 features, 30 samples

Source
http://cancergenome.nih.gov/

References

Examples
data(TCGA_GBM_miRNA_microarray)

TCGA_GBM_transcript_microarray

TCGA Glioblastoma Multiforme Transcript Sample Dataset

Description
This dataset contains transcript expression values from a microarray that was taken from the Cancer Genome Atlas, or TCGA. It has 10 features and 30 samples. The original dataset had 72656 features and 30 samples.

Usage
data(TCGA_GBM_transcript_microarray)

Format
An ExpressionSet with 10 features, 30 samples
Source

http://cancergenome.nih.gov/

References


Examples

data(TCGA_GBM_transcript_microarray)
Index

* datasets
  TCGA_Breast_miRNASeq, 8
  TCGA_Breast_miRNASeq_voom, 9
  TCGA_Breast_RNASeq, 9
  TCGA_Breast_RNASeq_voom, 10
  TCGA_GBM_miRNA_microarray, 11
  TCGA_GBM_transcript_microarray, 11
createVectors, 2
discordant, 3
discordantRun, 4
fishersTrans, 6
splitMADOutlier, 7
TCGA_Breast_miRNASeq, 8
TCGA_Breast_miRNASeq_voom, 9
TCGA_Breast_RNASeq, 9
TCGA_Breast_RNASeq_voom, 10
TCGA_GBM_miRNA_microarray, 11
TCGA_GBM_transcript_microarray, 11