discordant {fishersTrans}

Description

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

Usage

z <- fishersTrans(rho)

Arguments

|  |  |
| --- | --- |
| rho | Integer or numeric list of Pearson’s correlation coefficients |

Value

|  |  |
| --- | --- |
| z | Integer or numeric list transformed z scores |

Examples

rhoV <- rnorm(1000, 0, 0.2)

zV <- fisherTrans(rhoV)

print(head(zV))

discordant {createVectors}

Description

Creates vectors of correlation coefficients based on two groups of –omics bivariate data.

Usage

vectors <- createVectors(data1, data2, multOmics = TRUE/FALSE, featureSize)

Arguments

|  |  |
| --- | --- |
| data1 | 1st group of bivariate normal data |
| data2 | 2nd group of bivariate normal data |
| multOmics | Boolean value indicating if single or multiple –omics is being analyzed |
| featureSize | Integer of feature size length of first-omics in dataset. Value necessary if multOmics is TRUE. |

Value

|  |  |
| --- | --- |
| v1 | List of correlation coefficients for group 1 |
| v2 | List of correlation coefficients for group 2 |

Examples

library(MASS)

# for single -omics

data1 <- mvrnorm(20,rep(0,20),diag(20))

data2 <- mvrnorm(20,rep(0,20),diag(20))

vectors <- createVectors(data1, data2, multOmics = FALSE)

head(print(vectors$v1))

# for multiple –omics

data1 <- mvrnorm(20,rep(0,20),diag(20))

data2 <- mvrnorm(20,rep(0,20),diag(20))

vectors <- createVectors(data1, data2, multOmics = TRUE, featureSize = 10)

head(print(vectors$v1))

discordant {discordantRun}

Description

Runs discordant algorithm on two vectors of correlation coefficients.

Usage

test <- discordantRun(v1, v2, multOmics = TRUE/FALSE, featureSize)

Arguments

|  |  |
| --- | --- |
| data1 | 1st group of bivariate normal data |
| data2 | 2nd group of bivariate normal data |
| multOmics | Boolean value indicating if single or multiple –omics is being analyzed |
| featureSize | Numeric variable of feature size length of first-omics in dataset if multiple –omics, total feature size if single -omics. |

Details

Go over references??

Value

|  |  |
| --- | --- |
| discordPPMatrix | Matrix of posterior probabilities where rows and columns reflect features |
| discordPPV | Vector of posterior probabilities. |
| class | Vector of classes |
| probMatrix | Matrix of posterior probabilities where rows are each molecular feature pair and columns are nine different classes |
| Convergence | Number of iterations for method to converge. |
| loglik | Final log likelihood. |

Examples

library(MASS)

# for single -omics

data1 <- mvrnorm(20,rep(0,20),diag(20))

data2 <- mvrnorm(20,rep(0,20),diag(20))

vectors <- createVectors(data1, data2, multOmics = FALSE)

result <- discordRun(vectors$v1, vectors$v2, multOmics = FALSE, 20)

# for multiple –omics

data1 <- mvrnorm(20,rep(0,20),diag(20))

data2 <- mvrnorm(20,rep(0,20),diag(20))

vectors <- createVectors(data1, data2, multOmics = TRUE, featureSize = 10)

result <- discordRun(vectors$v1, vectors$v2, multOmics = TRUE, 10)

discordant {makeTable}

Description

Creates a table that where the first two columns are feature pairs and the third column is the posterior probability of discordance.

Usage

table <- makeTable(discordPPMatrix, featureNames1, featureNames2)

Arguments

|  |  |
| --- | --- |
| discordPPMatrix | Matrix of posterior probabilities taken from discordRun |
| featureNames1 | List of feature names for first –omics analyzed (multiple –omics), or total list of feature names (single –omics). |
| featureNames2 | List of feature names for second –omics analyzed (multiple –omics). |

Value

|  |  |
| --- | --- |
| outMatrix | Matrix of posterior probabilities for all possible pairs. |

Examples

library(MASS)

# for single -omics

data1 <- mvrnorm(20,rep(0,20),diag(20))

data2 <- mvrnorm(20,rep(0,20),diag(20))

vectors <- createVectors(data1, data2, multOmics = FALSE)

result <- discordRun(vectors$v1, vectors$v2, multOmics = FALSE, 20)

resultsTable <- makeTable(result$discordPPMatrix, featureNames)

# for multiple –omics

data1 <- mvrnorm(20,rep(0,20),diag(20))

data2 <- mvrnorm(20,rep(0,20),diag(20))

vectors <- createVectors(data1, data2, multOmics = TRUE, featureSize = 10)

result <- discordRun(vectors$v1, vectors$v2, multOmics = TRUE, 10)

resultsTable <- makeTable(result$discordPPMatrix, featureNames1, featureNames2)