MuscleData — *Inflammatory Myopathy Dataset*

Description

MuscleData is a R workspace containing MuscleData, index.DM, index.IBM, and index.NORM. The `MuscleData` object contains data from 49 samples hybridized onto Affymetrix HG-U133A arrays; the row names for the data frame contain the corresponding probe set IDs. The 'index' objects contain indices that describe the phenotype for the sample columns in `MuscleData`.

Usage

```
data(MuscleData)
```

Format

1 data frame and 3 vectors

Source

http://www.chip.org/~ppark/PNAS05

References

Tian L., Greenberg S.A., Kong S.W., Altschuler J., Kohane I.S., Park P.J. (2005) Discovering statistically significant pathways in expression profiling studies. *Proceedings of the National Academy of Sciences of the USA*, **102**, 13544-9.

http://www.pnas.org/cgi/doi/10.1073/pnas.0506577102

runSigPathway — *Perform pathway analysis*

Description

Performs pathway analysis

Usage

```
runSigPathway(G, minNPS = 20, maxNPS = 500,
              tab, phenotype, nsim = 1000,
              weightType = c("constant", "variable"), ngroups = 2,
              npath = 25, verbose = FALSE)
```

Arguments

G	a list containing the source, title, and probe sets associated with each curated pathway
minNPS	an integer specifying the minimum number of probe sets in tab that should be in a gene set
maxNPS	an integer specifying the maximum number of probe sets in tab that should be in a gene set
tab	a numeric matrix of expression values, with the rows and columns representing probe sets and sample arrays, respectively
phenotype	a numeric vector indicating the phenotype
nsim	an integer indicating the number of permutations to use
weightType	a character string specifying the type of weight to use when calculating NEk statistics
ngroups	an integer indicating the number of groups in the matrix
npath	an integer indicating the number of top gene sets to consider from each statistic when ranking the top pathways
verbose	a boolean to indicate whether to print debugging messages to the R console

Details

runSigPathway is a wrapper function that

(1) Selects the gene sets to analyze using selectGeneSets

(2) Calculates NTk and NEk statistics using calculate.NTk and calculate.NEK

(3) Ranks the top npath pathways from each statistic using rankPathways

(4) Summarizes the means, standard deviation, and individual statistics of each probe set in each of the above pathways using getPathwayStatistics

Value

A list containing

gsList	a list containing three vectors from the output of the selectGeneSets function
list.NTk	a list from the output of calculate.NTk
list.NEk	a list from the output of calculate.NEk
df.pathways	a data frame from rankPathways which contains the top pathways' indices in G, gene set category, pathway title, set size, NTk statistics, NEk statistics, the corresponding q-values, and the ranks.
list.gPS	a list from getPathwayStatistics containing nrow(df.pathways) data frames corresponding to the pathways listed in df.pathways. Each data frame contains the name, mean, standard deviation, the test statistic (e.g., t-test), and the corresponding unadjusted p-value. If ngroups = 1, the Pearson correlation coefficient is also returned.

Author(s)

Lu Tian and Peter Park, with contributions from Weil Lai

References

Tian L., Greenberg S.A., Kong S.W., Altschuler J., Kohane I.S., Park P.J. (2005) Discovering statistically significant pathways in expression profiling studies. *Proceedings of the National Academy of Sciences of the USA*, **102**, 13544-9.

http://www.pnas.org/cgi/doi/10.1073/pnas.0506577102

Examples

```
## Load in expression data and select the probe sets have expression
## values greater than the trimmed mean in at least 1 out of 49 arrays
data(MuscleData)
sf <- apply(MuscleData, 2, mean, tr = 0.025)
temp <- sweep(MuscleData, 2, sf, FUN = '/')
ind.pskeep <- which(rowSums(temp > 1) > 0)
tabMD <- MuscleData[ind.pskeep, ]
probeID <- names(ind.pskeep)

rm(temp)

## Select the data to study: IBM vs. NORM _or_ DM vs. NORM
compIBM <- TRUE

if( compIBM == TRUE )  {
  tab <- tabMD[,c(index.NORM, index.IBM)]
  phenotype <- c(rep.int(0,length(index.NORM)), rep.int(1,length(index.IBM)))
}else  {
  tab <- tabMD[,c(index.NORM, index.DM)]
  phenotype <- c(rep.int(0,length(index.NORM)), rep.int(1,length(index.DM)))
}

## Prepare the pathways to analyze
data(GenesetsU133a)

nsim <- 100
ngroups <- 2
verbose <- TRUE
weightType <- "constant"
npath = 25

res.muscle <- runSigPathway(G, 20, 500, tab, phenotype, nsim,
                            weightType, ngroups, npath, verbose)

## Summarize results
print(res.muscle$df.pathways)

## Get more information about the probe sets' means and other statistics
## for the top pathway in res.pathways
print(res.muscle$list.gPS[[1]])
```

GenesetsU133a — *Collection of Human Pathways from Gene Ontology, BioCyc, BioCarta, KEGG, and SuperArray*

Description

In this data set, the list `G` contain annotations for most human pathways as described in Gene Ontology, BioCyc, BioCarta, KEGG, and SuperArray. Each pathway in `G` contains the lookup ID (`src`), the Entrez Gene IDs (`locusID`), the pathway title (`title`), and the probe set IDs represented in the pathway (`probes`).

Usage

```
data(GenesetsU133a)
```

Format

A nested list

Source

http://www.chip.org/~ppark/PNAS05/

References

Tian L., Greenberg S.A., Kong S.W., Altschuler J., Kohane I.S., Park P.J. (2005) Discovering statistically significant pathways in expression profiling studies. *Proceedings of the National Academy of Sciences of the USA*, **102**, 13544-9.

http://www.pnas.org/cgi/doi/10.1073/pnas.0506577102

`calcTNullFast` — *Compute Null T Distribution for Each Gene*

Description

Computes a null t distribution for each gene by permuting the phenotypes.

Usage

```
calcTNullFast(tab, phenotype, nsim, ngroups = 2)
```

Arguments

`tab`	a numeric matrix of expression values, with the rows and columns representing probe sets and sample arrays, respectively
`phenotype`	a numeric vector indicating the phenotype
`nsim`	an integer indicating the number of permutations to use
`ngroups`	an integer indicating the number of groups in the expression matrix

Details

Similar to `calcTStatFast` but calculates t-statistics over permuted phenotypes. Please refer to the help file of `calcTStatFast` for more details.

Value

A matrix with `nsim` rows and `nrow(tab)` columns.

Author(s)

Weil Lai

`calculatePathwayStatistics`

Calculate the NTk and NEk statistics

Description

Calculates the NTk and NEk statistics and the corresponding p-values and q-values for each selected pathway.

Usage

```
calculate.NTk(tab, phenotype, gsList, nsim = 1000,
              ngroups = 2, verbose = FALSE)
calculate.NEk(tab, phenotype, gsList, nsim = 1000,
              weightType = c("constant", "variable"),
              ngroups = 2, verbose = FALSE)
```

Arguments

`tab`	a numeric matrix of expression values, with the rows and columns representing probe sets and sample arrays, respectively
`phenotype`	a numeric vector indicating the phenotype
`gsList`	a list containing three vectors from the output of the `selectGeneSets` function
`nsim`	an integer indicating the number of permutations to use

`weightType`	a character string specifying the type of weight to use when calculating NEk statistics
`ngroups`	an integer indicating the number of groups in the matrix
`verbose`	a boolean to indicate whether to print debugging messages to the R console

Details

These functions calculate the NTk and NEk statistics and the corresponding p-values and q-values for each selected pathway. The output of both functions should be together to rank top pathways with the `rankPathways` function.

Value

A list containing

`ngs`	number of gene sets
`nsim`	number of permutations performed
`t.set`	a numeric vector of Tk/Ek statistics
`t.set.new`	a numeric vector of NTk/NEk statistics
`p.null`	the proportion of nulls
`p.value`	a numeric vector of p-values
`q.value`	a numeric vector of q-values

Author(s)

Lu Tian and Peter Park, with contributions from Weil Lai

References

Tian L., Greenberg S.A., Kong S.W., Altschuler J., Kohane I.S., Park P.J. (2005) Discovering statistically significant pathways in expression profiling studies. *Proceedings of the National Academy of Sciences of the USA*, **102**, 13544-9.

http://www.pnas.org/cgi/doi/10.1073/pnas.0506577102

Examples

```
## Load in expression data and select the probe sets have expression
## values greater than the trimmed mean in at least 1 out of 49 arrays
data(MuscleData)
sf <- apply(MuscleData, 2, mean, tr = 0.025)
temp <- sweep(MuscleData, 2, sf, FUN = '/')
ind.pskeep <- which(rowSums(temp > 1) > 0)
tabMD <- MuscleData[ind.pskeep, ]
probeID <- names(ind.pskeep)

rm(temp)

## Select the data to study: IBM vs. NORM _or_ DM vs. NORM
```

```
compIBM <- TRUE

if( compIBM == TRUE )  {
  tab <- tabMD[,c(index.NORM, index.IBM)]
  phenotype <- c(rep.int(0,length(index.NORM)), rep.int(1,length(index.IBM)))
}else  {
  tab <- tabMD[,c(index.NORM, index.DM)]
  phenotype <- c(rep.int(0,length(index.NORM)), rep.int(1,length(index.DM)))
}

## Prepare the pathways to analyze
data(GenesetsU133a)
gsList <- selectGeneSets(G, probeID, 20, 500)

## Calculate NTk and weighted NEk for each gene set
## * Use a higher nsim (e.g., 2500) value for more reproducible results
nsim <- 100
ngroups <- 2
verbose <- TRUE
weightType <- "constant"
methodNames <- c("NTk", "NEk")
npath = 25
res.NTk <- calculate.NTk(tab, phenotype, gsList, nsim, ngroups, verbose)
res.NEk <- calculate.NEk(tab, phenotype, gsList, nsim, weightType,
                         ngroups, verbose)

## Summarize results
res.pathways <- rankPathways(res.NTk, res.NEk, G, gsList, methodNames,
                             npath)
print(res.pathways)

## Get more information about the probe sets' means and other statistics
## for the top pathway in res.pathways
topIndex <- res.pathways$IndexG[1]
res.topPathway <- getPathwayStatistics(tab, phenotype, G, topIndex,
                                       ngroups, NULL, FALSE)
print(res.topPathway[[1]])
```

`calcTStatFast` *Compute T-Statistics and Corresponding P-Values*

Description

Computes t-statistics and corresponding p-values.

Usage

```
calcTStatFast(tab, phenotype, ngroups = 2)
```

Arguments

`tab`	a numeric matrix of expression values, with the rows and columns representing probe sets and sample arrays, respectively
`phenotype`	a numeric vector indicating the phenotype
`ngroups`	an integer indicating the number of groups in the expression matrix

Details

If there are two groups in the matrix, then the phenotype vector should only consist of 0 and 1 to denote which sample columns belong to which group.

If `ngroups` = 2, the t-test done here is equivalent to a unpaired two-sample t-test, assuming unequal variances.

If there is only one group in the matrix (e.g., Alzheimer's data set as reanalyzed in Tian et al. (2005)), then the phenotype vector should consist of continuous values. In this case, the association between phenotype and expression values is first calculated as Pearson correlation coefficients, transformed to Fisher's z, and then rescaled so that its variance is 1:

z = 0.5*log((1+rho)/(1-rho))*sqrt(n-3), where n is the number of phenotypes.

Value

`pval`	A vector of unadjusted p-values
`tstat`	A vector of t-statistics (`ngroups` = 2) or rescaled Fisher's z (`ngroups` = 1)
`rho`	(Also returned when `ngroups` = 1) A vector of Pearson correlation coefficients

Author(s)

Weil Lai

Examples

```
## Load inflammatory myopathy data set
data(MuscleData)

## Create appropriate variables for
tab <- MuscleData[, c(index.IBM, index.NORM)]
phenotype <- c(rep(0,length(index.IBM)), rep(1,length(index.NORM)))
statList <- calcTStatFast(tab, phenotype, ngroups = 2)

## Generate histogram of p-values
hist(statList$pval, xlab = "Unadjusted p-values", ylab = "Frequency")
```

`getPathwayStatistics` *Give the statistics for the probe sets in a pathway*

Description

Gives the statistics for the probe sets associated with a pathway.

Usage

```
getPathwayStatistics(tab, phenotype, G, index, ngroups = 2,
                     statList = NULL, keepUnknownProbes = FALSE)
```

Arguments

`tab`	a numeric matrix of expression values, with the rows and columns representing probe sets and sample arrays, respectively
`phenotype`	a numeric vector indicating the phenotype
`G`	a list containing the source, title, and probe sets associated with each curated pathway
`index`	an integer vector specifying the pathway(s) to summarize in `G`
`ngroups`	an integer indicating the number of groups in the expression matrix
`statList`	a list containing results from `calcTStatFast`
`keepUnknownProbes`	a boolean indicating whether to keep the names of probe sets not represented in tab in the summary data frame

Details

This function gives the mean, standard deviation, and test statistic for each in the pathway as indicated in `G[[index]]`.

Value

A data frame indicating the name, mean, standard deviation, the test statistic (e.g., t-test), and the corresponding unadjusted p-value. If `ngroups` = 1, the Pearson correlation coefficient is also returned.

Note

See the help page for `calculate.NTk` or `calculate.NEk` for example code that uses `getPathwayStatistics`

Author(s)

Weil Lai

`rankPathways` *Summarizes Top Pathways from Pathway Analyses*

Description

Summarizes top pathways from pathway analyses.

Usage

```
rankPathways(res.A, res.B, G, gsList, methodNames = NULL, npath = 25)
```

Arguments

`res.A`	a list from the output of calculate.NTk or calculate.NEk
`res.B`	a list from the output of calculate.NTk or calculate.NEk
`G`	a list containing the source, title, and probe sets associated with each curated pathway
`gsList`	a list containing three vectors from the output of the `selectGeneSets` function
`methodNames`	a character vector of length 2 giving the names for res.A and res.B
`npath`	an integer indicating the number of top gene sets to consider from each statistic when ranking the top pathways

Details

This function ranks together the statistics given in res.A and res.B and summarizes the top gene sets in a tabular format similar to Table 2 in Tian et al. (2005)

Value

A data frame showing the pathways' indices in `G`, gene set category, pathway title, set size, res.A's statistics, res.B's statistics, the corresponding q-values, and the ranks for the top gene sets.

Note

See the help page for `calculate.NTk` or `calculate.NEk` for example code that uses `getPathwayStatistics`

Author(s)

Lu Tian and Peter Park, with contributions from Weil Lai

References

Tian L., Greenberg S.A., Kong S.W., Altschuler J., Kohane I.S., Park P.J. (2005) Discovering statistically significant pathways in expression profiling studies. *Proceedings of the National Academy of Sciences of the USA*, **102**, 13544-9.

http://www.pnas.org/cgi/doi/10.1073/pnas.0506577102

`selectGeneSets` *Select gene sets to be analyzed in pathway analysis*

Description

Selects gene sets to be analyzed in pathway analysis based on minimum and maximum number of probe sets to consider per pathway.

Usage

```
selectGeneSets(G, probeID, minNPS = 20, maxNPS = 500)
```

Arguments

`G`	a list containing the source, title, and probe sets associated with each curated pathway
`probeID`	a character vector containing the names of probe sets associated with a matrix of expression values
`minNPS`	an integer specifying the minimum number of probe sets in `probeID` that should be in a gene set
`maxNPS`	an integer specifying the maximum number of probe sets in `probeID` that should be in a gene set

Details

This function selects the appropriate pathways from a large, curated list based on the minimum and maximum number of probe sets that should be considered in a gene set. It creates three vectors: `nprobesV` and `indexV` representing a sparse indicator matrix and `indGused` indicating which gene sets were selected from `G`.

Value

A list containing

`nprobesV`	an integer vector indicating the number of probe sets in `probeID` that is in each selected gene set
`indexV`	an integer vector containing positions for each 1s in the sparse indicator matrix
`indGused`	an integer vector indicating which pathways in `G` were chosen

Note

See the help page for `calculate.NTk` or `calculate.NEk` for example code that uses `getPathwayStatistics`

Author(s)

Lu Tian and Peter Park, with contributions from Weil Lai

References

Tian L., Greenberg S.A., Kong S.W., Altschuler J., Kohane I.S., Park P.J. (2005) Discovering statistically significant pathways in expression profiling studies. *Proceedings of the National Academy of Sciences of the USA*, **102**, 13544-9.

http://www.pnas.org/cgi/doi/10.1073/pnas.0506577102

`calculate.GSEA` *Calculate 2-sided statistics based on the GSEA algorithm*

Description

Calculates the 2-sided statistics based on the GSEA algorithm.

Usage

```
calculate.GSEA(tab, phenotype, gsList, nsim = 1000,
               verbose = FALSE)
```

Arguments

`tab`	a numeric matrix of expression values, with the rows and columns representing probe sets and sample arrays, respectively
`phenotype`	a numeric vector indicating the phenotype
`gsList`	a list containing three vectors from the output of the `selectGeneSets` function
`nsim`	an integer indicating the number of permutations to use
`verbose`	a boolean to indicate whether to print debugging messages to the R console

Details

This function calculates a variant of the GSEA statistics (Mootha et al.) with the following modifications: (a) GSEA was changed from a 1-sided to a 2-sided approach. (b) The 2-group t-statistics is used as the difference metric.

The function also normalizes the GSEA statistic and calculates the corresponding q-values for each gene set as described in Tian et al. (2005)

Value

A list containing

`ngs`	number of gene sets
`nsim`	number of permutations performed
`t.set`	a numeric vector of Tk statistics
`t.set.new`	a numeric vector of NTk statistics
`p.null`	the proportion of nulls
`p.value`	a numeric vector of p-values
`q.value`	a numeric vector of q-values

Author(s)

Lu Tian and Peter Park, with contributions from Weil Lai

References

Mootha V.K., Lindgren C.M., Eriksson K.F., Subramanian A., Sihag S., Lehar J., Puigserver P., Carlsson E., Ridderstrale M., Laurila E., Houstis N., Daily M.J., Patterson N., Mesirov J.P., Golud T.R., Tamayo P., Spiegelman B., Lander E.S., Hirshhorn J.N., Altshuler D., Groop L.C. (2003) PGC-1alpha-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nature Genetics*, **34**, 267-73.

Tian L., Greenberg S.A., Kong S.W., Altschuler J., Kohane I.S., Park P.J. (2005) Discovering statistically significant pathways in expression profiling studies. *Proceedings of the National Academy of Sciences of the USA*, **102**, 13544-9.

http://www.pnas.org/cgi/doi/10.1073/pnas.0506577102