R documentation

of 'GSCA/man/GSCA-package.Rd' etc.

June 8, 2009

R topics documented:

`GSCA-package` *Gene Set Co-expression Analysis*

Details

Package:	GSCA
Type:	Package
Version:	1.0
Date:	2009-04-26
License:	LGPL >= 2.0
LazyLoad:	yes

Author(s)

YounJeong Choi

Maintainer: YounJeong Choi <ychoi@biostat.wisc.edu>

References

Choi and Kendziorski (2009)

`LungCancer3` *Three lung cancer microarray data sets and matched annotation*

Description

The three human lung cancer microarray data sets, from Stanford (Garber et al., 2001), Harvard (Bhattacharjee et al., 2001), and Michigan (Beer et al., 2002). The 3,924 common Entrez Gene IDs are represented, matched across the three studies. The 3,649 common gene sets (GO categories and KEGG pathways) are represented, defined in Entrez Gene ID.

Usage

```
data(LungCancer3)
```

Format

A list of two sub-lists named 'data' and 'info', respectively.

Details

`LungCancer3$data` is a list of 3 R matrices named 'Harvard', 'Stanford', and 'Michigan', respectively. Each matrix contains 3,924 rows as the genes have been matched across three studies. Rows are named by Entrez Gene IDs.

The number of columns corresponds to the number of arrays used in each study. Columns are named `Tumor1`, `Tumor2`, ..., and `Normal1`, `Normal2`, ..., similarly for three matrices. `LungCancer3$data$Harvard` (Harvard study) has 156 columns of 139 tumor vs. 17 normal samples; `LungCancer3$data$Stanford` (Stanford study) has 46 columns of 41 tumor vs. 5 normal samples; and `LungCancer3$data$Michigan` (Michigan study) has 96 columns of 86 tumor vs. 10 normal samples. For tumor samples, only adenocarcinoma samples have been included for consistency.

`LungCancer3$info` is again a list of two objects, named `GSdef` and `Name`, respectively.

`LungCancer3$info$GSdef` is a list of 3,649 gene set definitions for the 3,924 genes by GO categories and KEGG pathways. This list is named by the gene set IDs such as `"GO:0007169"` for GO categories and `"00920"` for KEGG pathways. Each entry of this list is a character vector of Entrez Gene IDs, which the gene set consists of. For example, `LungCancer3$info$GSdef[[147]]` returns `c("348", "25", "27")`.

`LungCancer3$info$Name` is a character vector of length 3,649 corresponding to the gene sets defined in `LungCancer3$info$GSdef`, named by the gene set IDs. `GSdef` and `Name` have the gene sets in the same order. For example, `LungCancer3$info$GSdef[[123]]` and `LungCancer3$info$Name[123]` are both `"GO:0032774"`.

Source

Harvard data (Bhattacharjee et al.) http://www.broad.mit.edu/mpr/lung/

Stanford data (Garber et al.) http://smd.stanford.edu/cgi-bin/data/viewDetails.pl?exptid=12827&viewSet=1

Michigan data (Beer et al.) http://dot.ped.med.umich.edu:2000/ourimage/pub/Lung/index.html

References

Beer, D. G. et al. (2002) Gene-expression profiles predict survival of patients with lung adenocarcinoma. Nature Medicine, 8, 816-824.

Bhattacharjee, A. et al. (2001) Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. Proc. Natl Acad. Sci., 98, 13790-13795.

Garber, M. E. et al. (2001) Diversity of gene expression in adenocarcinoma of the lung. Proc. Natl Acad. Sci., 98, 13784-13789.

Examples

```
data(LungCancer3)
str(LungCancer3$data)
str(LungCancer3$info$GSdef[1:3])
LungCancer3$info$Name[1:3]
```

`metaDI` *The function to run a meta-GSCA.*

Description

This function can be used to run a meta-GSCA, described in Choi and Kendziorski (2009). Unlike `singleDC`, the study-specific values (e.g., gene-gene pairwise correlations, difference in condition-specific correlation signs) need to be pre-calculated and provided as input. For each gene set defined in `GSdefList`, the dispersion index is calculated between two given studies and returned.

Gene pairs are randomly permuted across gene sets for `nperm` times. Permutation-based p-values are calculated, based on the rank of observed `DI` among permuted index values.

Usage

```
metaDI(corr1, corr2, GSdefList, nperm, permDI = FALSE)
```

Arguments

`corr1, corr2` Two symmetric matrices from two studies of interest. For meta-GSCA, a difference matrix is used, of two condition-specific correlation sign matrices within each study. The lower triangle is used. `rownames(data)` is used to subset a sub-matrix from `data` for each gene set. (Rows must be named by gene IDs used in `GSdefList`. For example, if `GSdefList` defines gene sets in Entrez Gene IDs, `rownames(data)` should be Entrez Gene IDs.

`GSdefList` A list of character vectors that define gene sets. Each entry of this list is a gene set.

`nperm` The desired number of permutations.

`permDI` TRUE/FALSE. If set TRUE, dispersion index values from permutation are saved and returned; if FALSE, permutation-based dispersion index values are not returned. Default is FALSE.

Details

Gene pairs are permuted across gene sets, as described in Choi and Kendziorski (2009). For each permutation, annotated gene pairs (gene pairs which belong to at least one gene set) are randomly re-assigned to gene sets, and dispersion indices (DIs) are calculated based on those random gene sets. As focus is on preservation, the p-value for each gene set is calculated as:

p = sum(permutation DIs <= observed DI) / nperm .

Value

`DI` The dispersion index vector for each gene set.

`pvalue` The permutation-based p-value for each gene set.

`permv` The permutation-based DI matrix, of `nperm` columns. The first column is identical to what is returned by `DI`.

Note

Currently, `metaDI` implements meta-analysis of gene-gene pairwise correlations from two studies. In addition to meta-GSCA described in Choi and Kendziorski (2009), which uses the sign difference, raw correlations can be input to investigate preservation of them.

Author(s)

YounJeong Choi

References

Choi and Kendziorski, submitted

Examples

```
data(LungCancer3)
GS <- LungCancer3$info$GSdef
GSdesc <- LungCancer3$info$Name

data.grouped <- list(
Tumor = list(Harvard = LungCancer3$data$Harvard[, 1:139],
Michigan = LungCancer3$data$Michigan[, 1:86]),
Normal = list(Harvard = LungCancer3$data$Harvard[, 140:156],
Michigan = LungCancer3$data$Michigan[, 87:96]))

corr.t <- lapply(lapply(data.grouped$Tumor, t), cor, use = "pairwise.complete.obs")
corr.n <- lapply(lapply(data.grouped$Normal, t), cor, use = "pairwise.complete.obs")
cor.diff <- list(Harvard = corr.t$Harvard - corr.n$Harvard,
Michigan = corr.t$Michigan - corr.n$Michigan)

cor.diff.sign <- list(
Harvard = apply((cor.diff$Harvard > 0), 2, as.numeric) -
apply((cor.diff$Harvard < 0), 2, as.numeric),
Michigan = apply((cor.diff$Michigan > 0), 2, as.numeric) -
apply((cor.diff$Michigan < 0), 2, as.numeric))

for (i in 1:length(cor.diff.sign)) {
rownames(cor.diff.sign[[i]]) <- colnames(cor.diff.sign[[i]])
}
dist.HM <- metaDI(cor.diff.sign$Harvard, cor.diff.sign$Michigan, GS, 3, permDI = TRUE)
```

`plotMNW` *The function to draw a multi-edge network from a list of correlation matrices from multiple studies.*

Description

This function draws a network with multiple lines per edge, representing multiple study-specific correlations for the corresponding gene pairs. Nodes can be colored by DE information, and each line in edges can be colored based on the correlation magnitude and direction.

Usage

```
plotMNW(cormatrix.list, genes = NULL, mycolors = bluered(201), ncolor = "white", nc
```

Arguments

`cormatrix.list`
A list of symmetric correlation matrices from multiple studies.

`genes` Node names to be displayed in the network plot. If `NULL` (default), `rownames(cormatrix.list[[1` is used.

`mycolors` Edge color palette. The default is `bluered(201)` implemented by `gplots`, which sets c("blue", "white", "red") for the correlation of c(-1, 0, 1).

`ncolor` Node color for selected genes specified by `node.de`.

`node.de` A vector of 0’s and 1’s. The "1" nodes are colored in `ncolor`.

`r` The radius of the big circle where nodes are placed on

`nr` The radius of each node

`jt` Distance between adjacent lines in an edge

`lwd` The edge line thickness

`xlab1` Sub-text to be place below the network

`text.cex` Node font size

`...` The rest of arguments are passed to `plot`.

Details

Edge lines are sorted first and put in order separately for every edge, for effective display of study agreement.

Note

Because edges tend to be thicker as the number of studies grow larger, including too many genes (nodes) may result in a noninformative network plot.

Author(s)

YounJeong Choi

References

Choi and Kendziorski (2009)

Examples

```
data(LungCancer3)
GS <- LungCancer3$info$GSdef
GSdesc <- LungCancer3$info$Name

setid <- "GO:0008033"
gid <- GS[[setid]]
ss <- c("KARS", "SARS", "AARS", "SSB", "POP1", "RPP30")

data.list <- list(Harvard = LungCancer3$data$Harvard[gid, 140:156],
Stanford = LungCancer3$data$Stanford[gid, 42:46],
Michigan = LungCancer3$data$Michigan[gid, 87:96])

cormatrix.list <- lapply(lapply(data.list, t), cor,
use = "pairwise.complete.obs")

plotMNW(cormatrix.list, genes = ss, mycolors = bluered(201), ncolor =
"yellow", node.de = c(rep(1, 3), rep(0, 3)), lwd = 5, jt = 0.3)
```

plotNW *The function to draw a network from a correlation matrix*

Description

This function is a custom wrapper of `plot.graph` implemented in `Rgraphviz`. It draws a network from a given correlation matrix. Nodes can be colored by DE information, and edges can be colored based on the correlation magnitude and direction.

Usage

```
plotNW(genes = NULL, cormatrix, node.de = NULL, ncolor = "red", ecolor = NULL, ewid
```

Arguments

`genes` Node names to be displayed in the network plot. If `NULL` (default), `rownames(cormatrix)` is used.

`cormatrix` A symmetric correlation matrix to draw a network from.

`node.de` A vector of 0's and 1's. The "1" nodes are colored in `ncolor`.

`ncolor` Node color for selected genes specified by `node.de`.

`ecolor` Edge color palette. The default is `bluered(201)` implemented by `gplots`, which sets c("blue", "white", "red") for the correlation of c(-1, 0, 1).

`ewidth` Edge width. Default is set to 1.

`gtype` Graph type (`layout`) to be passed to `plot.graph`. One of `dot`, `neato`, `twopi`, `circo`, and `fdp`. The default is `neato`.

`...` The rest of arguments are passed to `plot.graph`.

Author(s)

YounJeong Choi

References

Choi and Kendziorski (2009)

See Also

`plot.graph`, `Rgraphviz`

Examples

```
data(LungCancer3)
GS <- LungCancer3$info$GSdef
GSdesc <- LungCancer3$info$Name

setid <- "GO:0019216"
gid <- GS[[setid]]
ss <- c("SERPINA3", "SOD1", "SCAP", "NPC2", "ADIPOQ", "PRKAA1", "AGT",
"PPARA", "BMP6", "BRCA1")

plotNW(genes = ss, cormatrix = cor(t(LungCancer3$data$Michigan[gid, 87:96]), use = "pairwise
```

`singleDC` *The function to run a single-study GSCA, differential co-expression (DC) analysis*

Description

This function runs a single-study GSCA, differential co-expression (DC) analysis, described in Choi and Kendziorski (2009). The condition-specific gene-gene pairwise correlations are first calculated; then for each gene set defined in `GSdefList`, the dispersion index is calculated across condition-specific correlations.

Samples are randomly permuted across conditions for `nperm` times. Permutation-based p-values are calculated, based on the rank of observed `DI` among permuted index values.

Usage

```
singleDC(data, group, GSdefList, nperm, permDI = FALSE)
```

Arguments

`data` A data matrix of rows representing genes and columns representing arrays. `rownames(data)` is used to subset a sub-matrix from `data` for each gene set. (Rows must be named by gene IDs used in `GSdefList`. For example, if `GSdefList` defines gene sets in Entrez Gene IDs, `rownames(data)` should be Entrez Gene IDs.

`group` A numeric vector that specifies the number of arrays (columns) in each condition. For example, if `c(10, 5)` is provided, first 10 columns of the `data` matrix are used for one condition and the next 5 are used for the other condition.

`GSdefList` A list of character vectors that define gene sets. Each entry of this list is a gene set.

`nperm` The desired number of permutations.

`permDI` TRUE/FALSE. If set TRUE, dispersion index values from permutation are saved and returned; if FALSE, permutation-based dispersion index values are not returned. Default is FALSE.

Details

Samples (columns) are permuted across conditions. For each permutation, condition-specific correlations are re-calculated based on permuted samples, and dispersion indices (DIs) are calculated based on those permutation-based correlations. As focus is on difference, the p-value for each gene set is calculated as:

p = sum(permutation DIs >= observed DI) / nperm .

Value

`DI` The dispersion index vector for each gene set.

`pvalue` The permutation-based p-value for each gene set.

`permv` The permutation-based DI matrix, of `nperm` columns. The first column is identical to what is returned by `DI`.

Note

Currently, `singleDC` implements DC analysis for two conditions (e.g., tumor vs. normal) and three conditions (e.g., AA, AB, and BB genotypes). For three conditions, pairwise DIs are first calculated and averaged (internally).

Author(s)

YounJeong Choi

References

Choi and Kendziorski, submitted.

Examples

```
data(LungCancer3)
GS <- LungCancer3$info$GSdef
GSdesc <- LungCancer3$info$Name
dc.M <- singleDC(data = LungCancer3$data$Michigan, group = c(86, 10),
GSdefList = GS, nperm = 3, permDI = TRUE)
```

Index