Package 'TROM'

July 2, 2018

Type Package

Title Transcriptome Overlap Measure

Version 1.3

Date 2018-06-29

Author Jingyi Jessica Li <jli@stat.ucla.edu>, Wei Vivian Li <liw@ucla.edu>

Maintainer Jingyi Jessica Li <jli@stat.ucla.edu>

Depends R (>= 3.1.0), lattice, methods

Imports AnnotationDbi, openxlsx, GO.db, gtools, RColorBrewer, gplots, topGO

Description A new bioinformatic tool for comparing transcriptomes of two biological samples from the same or different species. The mapping is conducted based on the overlap of the associated genes of different samples. More examples and detailed explanations are available in the vignette. For detailed description of the method, please refer to Li, Wei Vivian, Yiling Chen, and Jingyi Jessica Li. ``TROM: A testing-based method for finding transcriptomic similarity of biological samples." Statistics in biosciences 9.1 (2017): 105-136 <doi:10.1007/s12561-016-9163-y>.

License GPL-2

URL http://www.stat.ucla.edu/~jingyi.li/software-and-data/trom.html, http://www.stat.ucla.edu/~jingyi.li/packages/TROM/vignette.pdf

NeedsCompilation no

Repository CRAN

Date/Publication 2018-07-02 14:20:07 UTC

R topics documented:

`TROM-package` *Transcriptome Overlap Measure*

Description

`select.associated.genes`	Select the associated genes of different biological samples
`select.associated.orthologs`	Select the associated genes of different biological samples among the genes with orthologs in the other species
`choose.z`	Choose the suggested threshold for Z-scores
`ws.trom`	Within-species transcriptome mapping
`ws.trom.orthologs`	Within-species transcriptome mapping using only ortholog genes
`bs.trom`	Between-species transcriptome mapping
`heatmap.3`	Plot the resulting within- or between-species TROM scores in a heatmap
`find.top.GO.terms`	Find top enriched GO terms
`find.top.GO.slim.terms`	Find top enriched GO slim terms

Details

Package:	TROM
Type:	Package
Version:	1.2
Date:	2016-08-29
License:	GPL-2

Author(s)

Jingyi Jessica Li, Wei Vivian Li

Maintainer: Jingyi Jessica Li <jli@stat.ucla.edu>

References

Li WV, Chen Y and Li JJ (2016). TROM: A Testing-Based Method for Finding Transcriptomic Similarity of Biological Samples. Statistics in Biosciences. DOI: 10.1007/s12561-016-9163-y

Li JJ, Huang H, Bickel PJ, & Brenner SE (2014). Comparison of *D. melanogaster* and *C. elegans* developmental stages, tissues, and cells by modENCODE RNA-seq data. *Genome Research*, 24(7), 1086-1101.

Examples

```
## using TROM to map developmental stages of D. melanogaster and C. elegans

## The .rda files used in this example can be downloaded and unzipped from
## http://www.stat.ucla.edu/~jingyi.li/packages/TROM/TROM_Rdata.zip.
## Not run:
load("dm_gene_expr.rda")
load("ce_gene_expr.rda")
load("dm_ce_orthologs.rda")
dm_ce_trom <- bs.trom(sp1_gene_expr = dm_gene_expr,
                      sp2_gene_expr = ce_gene_expr,
                      sp1_sp2_orthologs = dm_ce_orthologs, z_thre=1.5,
                      provide=FALSE)
heatmap.3(   dm_ce_trom,
             max_score = 6,
             Rowv = NULL,
             Colv = NULL,
             dendrogram = c("none"),
             distfun = dist,
             hclustfun = hclust,
             xlab = "worm stages",
             ylab = "fly stages",
             main = "D. melanogaster vs. C. elegans Stage Mapping",
             key = TRUE,
             keysize = 1,
             trace = "none",
             density.info = c("none"),
             col = terrain.colors(120),
  )

## End(Not run)
```

`bs.trom` *Between-species transcriptome mapping*

Description

This function calculates the TROM scores in comparing samples from two different species. TROM score = -$\log_{10}$(Bonferroni-corrected *p*-value from a hypergeometric test), with a minimum value of 0.

Usage

```
bs.trom(sp1_gene_expr = NULL, sp2_gene_expr = NULL, sp1_sp2_orthologs,
        z_thre = 1.5, provide = FALSE, gene_lists = NULL,
        save_overlap_genes = FALSE)
```

Arguments

sp1_gene_expr	a data frame containing gene expression estimates of species 1; rows correspond to genes; columns (from the second to the last) correspond to samples, with the first column as gene IDs. Not needed if provide = TRUE.
sp2_gene_expr	a data frame containing gene expression estimates of species 2; rows correspond to genes; columns (from the second to the last) correspond to samples, with the first column as gene IDs. Not needed if provide = TRUE.
sp1_sp2_orthologs	a data frame containing ortholog gene pairs between species 1 and 2: rows are ortholog pairs; columns are the two species.
z_thre	a numeric value specifiying the Z-score threshold used to select associated genes, whose Z-scores $\geq$ z_thre. This can be specified by users or calculated using choose.z().
provide	a boolean value indicating whether associated genes are user-provided. If provide = TRUE, the users need to provide lists of genes that they think can represent the transcriptome characteristics of different samples.
gene_lists	an .xlsx file containing user-provided gene lists. It is required when provide = TRUE.
save_overlap_genes	a Boolean value indicating whether the users want to save overlapping gene orthologs between every two samples from species 1 and 2 to an .xlsx file. If save_overlap_genes = TRUE, this function outputs the genes of species 1 in the overlapping orthologs to a file named "between-species overlapping genes (of species 1) between sample pairs.xlsx", and outputs the genes of species 2 in the overlappig orthologs to a file named "between-species overlapping genes (of species 2) between sample pairs.xlsx".

Details

If provide = TRUE, the users are required to specify gene_lists as a path to an.xlsx file containing gene lists to be used for transcriptome mapping and calculating the TROM scores; otherwise, the function will automatically select associated genes based on the criterion: Z-scores $\geq$ z_thre.

If specified, gene_lists should be a two-sheet Excel file with the first sheet for species 1 and the second sheet for species 2. In each sheet, rows represent gene ids and columns represent biological samples. Each column of the file stores the user-provided genes corresponding to the sample of that column. Please note that different columns may have different numbers of rows.

This function outputs the between-species TROM scores into an .xlsx file named "between-species TROM scores.xlsx".

Value

A matrix of between-species TROM scores, where rows correspond to the samples of species 1 and columns correspond to the samples of species 2.

Author(s)

Jingyi Jessica Li and Wei Vivian Li

References

Li WV, Chen Y and Li JJ (2016). TROM: A Testing-Based Method for Finding Transcriptomic Similarity of Biological Samples. Statistics in Biosciences. DOI: 10.1007/s12561-016-9163-y

Li JJ, Huang H, Bickel PJ, & Brenner SE (2014). Comparison of *D. melanogaster* and *C. elegans* developmental stages, tissues, and cells by modENCODE RNA-seq data. *Genome Research*, 24(7), 1086-1101.

See Also

ws.trom

Examples

```
## Calculating transcriptome overlap measure between
## D. melanogaster and C .elegans

## The .rda files used in this example can be downloaded and unzipped from
## http://www.stat.ucla.edu/~jingyi.li/packages/TROM/TROM_Rdata.zip.
## Not run:
load("dm_gene_expr.rda")
load("ce_gene_expr.rda")
load("dm_ce_orthologs.rda")

## Without user-provided gene lists
dm_ce_trom <- bs.trom(sp1_gene_expr = dm_gene_expr,
                      sp2_gene_expr = ce_gene_expr,
                      sp1_sp2_orthologs = dm_ce_orthologs,
                      z_thre = 1.5,
                      provide = FALSE, save_overlap_genes = FALSE)

## With user-provided gene lists
## compare the first four stages of D. melanogaster and C .elegans
genelists <- system.file("dm_ce_genelists.xlsx", package = "TROM")
dm_ce_trom2 <- bs.trom(sp1_sp2_orthologs = dm_ce_orthologs, provide = TRUE,
                      gene_lists = genelists)
## End(Not run)
```

---

choose.z *Choose the suggested threshold of Z-scores*

Description

This function calculates the suggested `z_thre` for within-species comparison. The suggested `z_thre` gives the most sparse but still stable correspondance map of the transcriptomes.

Usage

```
choose.z(sp_gene_expr, mode = FALSE)
```

Arguments

sp_gene_expr	a data frame containing gene expression estimates of the species; rows correspond to genes; columns (from the second to the last) correspond to samples, with the first column as gene IDs. Required.
mode	a Boolean value specifying the method used to select the threshold. If mode = TRUE, threshold is selected based on the mode of a density function; otherwise, (mode + sd) is used. Please refer to the reference for more details.

Details

This function calculates the suggested z_thre for the species corresponding to sp_gene_expr to select associated genes and calculate TROM scores. The users can tune z_thre based on the mapping results calculated from the suggested z_thre.

Value

A numeric vector of length one containing the suggested z_thre.

Author(s)

Jingyi Jessica Li, Wei Vivian Li

References

https://arxiv.org/abs/1601.05158

Li WV, Chen Y and Li JJ (2016). TROM: A Testing-Based Method for Finding Transcriptomic Similarity of Biological Samples. Statistics in Biosciences. DOI: 10.1007/s12561-016-9163-y

See Also

ws.trom, ws.trom.orthologs

Examples

```
## Calculating the suggested z_thre for first 15 stages of D. melanogaster

## dm_gene_expr.rda can be downloaded and unzipped from
## http://www.stat.ucla.edu/~jingyi.li/packages/TROM/TROM_Rdata.zip
## Not run:
load("dm_gene_expr.rda")
z_thre1 <- choose.z(dm_gene_expr[, 1:16], mode = FALSE)
z_thre2 <- choose.z(dm_gene_expr[, 1:16], mode = TRUE)

## End(Not run)
```

`find.top.GO.slim.terms` *Find top enriched GO slim terms*

Description

This function finds the top enriched Gene Ontology (functional annotation) slim terms in gene lists.

Usage

```
find.top.GO.slim.terms(gene_lists, all_genes, GOmappingfile, output_file,
                       topNum = 20, GO_slim_id, heatmap = FALSE)
```

Arguments

`gene_lists`	an .xlsx file giving the lists of user-provided genes, either user-specified genes or associated genes found by `select.associated.genes()` or `select. associated.orthologs()`.
`all_genes`	a character verctor giving the population of all genes.
`GOmappingfile`	a character giving the path of GO mapping file, which contains the information of the mapping of gene IDs to GO terms.
`output_file`	a character specifying the name of a .txt file to store the output of this function: top enriched GO slim terms in the input gene lists.
`topNum`	a integer specifying the number of top GO terms to be included in the results. Defaults to 20.
`GO_slim_id`	a character vector containing the GO IDs of all GO slim terms.
`heatmap`	a Boolean value specifying whether to output the heatmap for the top enriched GO slim terms. The heatmap gives the enrichment results across all samples of the GO slim terms that are at least top enriched in one biological sample. If `heatmap = TRUE`, this function outputs a pdf file named "Top enriched GO slim terms across samples.pdf".

Details

To use this function, please download the GO mapping file of the species of interest from http://geneontology.org/page/download-annotations. Please make sure that this file is in R's working directory and set `GOmappingfile` to the file's name.

`gene_lists` can be either the output .xlsx file of `select.associated.orthologs()`, the output .xlsx file of `select.associated.genes()` or an .xlsx file of the same format that contains the user-provided gene lists. If users want to use the overlap genes or overlap orthologs, they can find them in the output .xlsx files of `ws.trom()`, `ws.trom.orthologs()` or `bs.trom()`. Users can select the columns they are interested in and compact them into a new .xlsx file, and then pass the name of the new .xlsx file to `gene_lists`.

Users can check the .txt file `output_file` for the results of top enriched GO slim terms.

Value

A list of length 6×(number of biological samples). List elements are ordered in correspondence with the biological samples, e.g., the first 6 elements in the list correspond to the first sample, etc. For each sample, there are

a character vector giving the top GO slim IDs.
a character vector giving the corresponding the top GO slim terms.
a vector giving the number of occurences of the top GO slim IDs in the population.
a vector giving the observed number of occurences of the top GO slim IDs in the sample.
a vector giving the expected number of occurences of the top GO slim IDs in the sample.
a character vector giving the *p*-values from a hypergeometric test.

Author(s)

Jingyi Jessica Li, Wei Vivian Li

References

Li WV, Chen Y and Li JJ (2016). TROM: A Testing-Based Method for Finding Transcriptomic Similarity of Biological Samples. Statistics in Biosciences. DOI: 10.1007/s12561-016-9163-y

Li JJ, Huang H, Bickel PJ, & Brenner SE (2014). Comparison of *D. melanogaster* and *C. elegans* developmental stages, tissues, and cells by modENCODE RNA-seq data. *Genome Research*, 24(7), 1086-1101.

See Also

find.top.GO.terms

Examples

```
## Find top enriched GO terms in the developmental stages of D. melanogaster

## To run this example, please download the file "gene_association_fb_example.txt" from
## https://ucla.box.com/GO-mapping-file.
## Please move "gene_association_fb_example.txt" to R's working directory.

## dm_gene_expr.rda can be downloaded and unzipped from
## http://www.stat.ucla.edu/~jingyi.li/packages/TROM/TROM_Rdata.zip.

## Not run:
load("dm_gene_expr.rda")
dm_genes_all <- as.character(dm_gene_expr[,1])
data(GO_slim_id)
gene_lists <- system.file("dm_associated_genes.xlsx", package = "TROM")
dm_stage_GO_slim <- find.top.GO.slim.terms(
gene_lists = gene_lists,
all_genes = dm_genes_all,
GOmappingfile = "gene_association_fb_example.txt",
output_file = "top 20 enriched GO slim terms in fly stage-associated genes.txt",
```

```
GO_slim_id = GO_slim_id,
topNum = 20,
heatmap = FALSE)
## End(Not run)
```

`find.top.GO.terms` *Find top enriched GO terms*

Description

This function finds the top enriched Gene Ontology (functional annotation) terms in gene lists.

Usage

```
find.top.GO.terms(gene_lists, all_genes, GOmappingfile, output_file,
                  topNum = 20, heatmap = FALSE)
```

Arguments

`gene_lists`	an .xlsx file giving the lists of user-provided genes, either user-specified genes or associated genes found by `select.associated.genes()` or `select. associated.orthologs()`.
`all_genes`	a character verctor giving the population of all genes.
`GOmappingfile`	a character giving the path of GO mapping file, which contains the information of the mapping of gene IDs to GO terms.
`output_file`	a character specifying the name of a .txt file to store the output of this function: top enriched GO terms on the input gene lists.
`topNum`	a integer specifying the number of top GO terms to be included in the results. Defaults to 20.
`heatmap`	a Boolean value specifying whether to output the heatmap for the top enriched GO terms. The heatmap gives the enrichment results across all samples of the GO terms that are at least top enriched in one biological sample. If `heatmap = TRUE`, this function outputs a pdf file named "Top enriched GO terms across samples.pdf".

Details

To use this function, please download the GO mapping file of the species of interest from http://geneontology.org/page/download-annotations. Please make sure that this file is in R's working directory and set `GOmappingfile` to the file's name.

`gene_lists` can be either the output .xlsx file of `select.associated.orthologs()`, the output .xlsx file of `select.associated.genes()` or an .xlsx file of the same format that contains the user-provided gene lists. If users want to use the overlap genes or overlap orthologs, they can find them in the output .xlsx files of `ws.trom()`, `ws.trom.orthologs()` or `bs.trom()`. Users can select the columns they are interested in and compact them into a new .xlsx file, and then pass the name of the new .xlsx file to `gene_lists`.

Users can check the .txt file `output_file` for the results of top enriched GO terms.

Value

A list of length 6×(number of biological samples). List elements are ordered in correspondence with the biological samples, e.g., the first 6 elements in the list correspond to the first sample, etc. For each sample, there are

a character vector giving the top GO IDs.
a character vector giving the corresponding the top GO terms.
a vector giving the number of occurences of the top GO IDs in the population.
a vector giving the observed number of occurences of the top GO IDs in the sample.
a vector giving the expected number of occurences of the top GO IDs in the sample.
a character vector giving the *p*-values from a hypergeometric test.

Author(s)

Jingyi Jessica Li, Wei Vivian Li

References

Li WV, Chen Y and Li JJ (2016). TROM: A Testing-Based Method for Finding Transcriptomic Similarity of Biological Samples. Statistics in Biosciences. DOI: 10.1007/s12561-016-9163-y

Li JJ, Huang H, Bickel PJ, & Brenner SE (2014). Comparison of *D. melanogaster* and *C. elegans* developmental stages, tissues, and cells by modENCODE RNA-seq data. *Genome Research*, 24(7), 1086-1101.

See Also

find.top.GO.terms

Examples

```
## Find top enriched GO terms in the developmental stages of D. melanogaster

## To run this example, please download the file "gene_association_fb_example.txt" from
## https://ucla.box.com/GO-mapping-file.
## Please move "gene_association_fb_example.txt" to R's working directory.
## Please remove "#" to run the following commands.

## dm_gene_expr.rda can be downloaded and unzipped from
## http://www.stat.ucla.edu/~jingyi.li/packages/TROM/TROM_Rdata.zip.

## Not run:
load("dm_gene_expr.rda")
dm_genes_all <- as.character(dm_gene_expr[,1])
gene_lists <- system.file("dm_associated_genes.xlsx", package = "TROM")
dm_stage_GO <- find.top.GO.terms(
gene_lists = gene_lists,
all_genes = dm_genes_all,
GOmappingfile = "gene_association_fb_example.txt",
output_file = "top 20 enriched GO terms in fly stage-associated genes.txt",
```

```
topNum = 20,
heatmap = FALSE)
## End(Not run)
```

GO_slim_id *GO slim IDs*

Description

GO_slim_id stores the GO IDs of all the GO slim terms.

Usage

```
data("GO_slim_id")
```

Format

The format is: chr [1:205] "GO:0000003" "GO:0019952" "GO:0050876" "GO:0000228" "GO:0000229" ...

Source

http://www.geneontology.org/ontology/subsets/goslim_generic.obo

References

Li JJ, Huang H, Bickel PJ, & Brenner SE (2014). Comparison of *D. melanogaster* and *C. elegans* developmental stages, tissues, and cells by modENCODE RNA-seq data. *Genome Research*, 24(7), 1086-1101.

Examples

```
data(GO_slim_id)
head(GO_slim_id)
```

---

heatmap.3 *Plot TROM scores in a heatmap.*

Description

heatmap.3 plots the TROM scores obtained from mapping different biological samples. Larger TROM scores are shown in darker colors, corresponding to a scale showing -$\log_{10}$ (transformed Bonferroni corrected *p*-values) saturated at a default value of 6. The TROM scores are calculated from function ws.trom() or bs.trom().

Usage

```
heatmap.3(x, max_score=6, Rowv = TRUE,
          Colv = if (symm) "Rowv" else TRUE,
          distfun = dist, hclustfun = hclust,
          dendrogram = c("both", "row", "column", "none"),
          symm = FALSE, scale = c("none", "row", "column"),
          na.rm = TRUE, revC = identical(Colv,"Rowv"),
          add.expr, breaks,
          symbreaks = min(x < 0, na.rm = TRUE) || scale != "none",
          col = "heat.colors", colsep, rowsep, sepcolor = "white",
          sepwidth = c(0.05, 0.05), cellnote,
          notecex = 1, notecol = "cyan",
          na.color = par("bg"),
          trace = c("none"),
          tracecol = "cyan", hline = median(breaks),
          vline = median(breaks), linecol = tracecol,
          margins = c(5, 5), ColSideColors, RowSideColors,
          cexRow = 0.2 + 1/log10(nr), cexCol = 0.2 + 1/log10(nc),
          labRow = NULL, labCol = NULL, key = TRUE, keysize = 1.5,
          density.info = c("histogram", "density", "none"),
          denscol = tracecol,
          symkey = min(x < 0, na.rm = TRUE) || symbreaks, densadj = 0.25,
          main = NULL, xlab = NULL, ylab = NULL, lmat = NULL,
          lhei = NULL, lwid = NULL,
          leftMargin = 7, bottomMargin = 7, reverse = FALSE, ...)
```

Arguments

x	numeric matrix of the TROM scores (-$\log_{10}$ transformed Bonferroni corrected *p*-values) to be plotted.
max_score	numeric value specifying the saturated value of TROM scores, i.e., -$\log_{10}$ (transformed Bonferroni corrected *p*-values). The TROM scores are saturated at max_score: TROM score=min(TROM score, max_score). The defualt value of max_score is 6.
Rowv	determines if and how the row dendrogram should be computed and reordered. Either a dendrogram or a vector of values used to reorder the row dendrogram or NA to suppress any row dendrogram (and reordering) or by default, NULL.
Colv	determines if and how the column dendrogram should be reordered. Has the same options as the Rowv argument above and additionally when x is a square matrix, Colv = "Rowv" means that columns should be treated identically to the rows (and so if there is to be no row dendrogram there will not be a column one either).
distfun	function used to compute the distance (dissimilarity) between both rows and columns. Defaults to dist.
hclustfun	function used to compute the hierarchical clustering when Rowv or Colv are not dendrograms. Defaults to hclust. Should take as argument a result of distfun and return an object to which as.dendrogram can be applied.

`dendrogram` character string indicating whether to draw 'none', 'row', 'column' or 'both' dendrograms. Defaults to 'both'. However, if Rowv (or Colv) is FALSE or NULL and dendrogram is 'both', then a warning is issued and Rowv (or Colv) arguments are honored.

`symm` logical indicating if x should be treated symmetrically; can only be true when x is a square matrix.

`scale` character indicating if the values should be centered and scaled in either the row direction or the column direction, or none. The default is "none".

`na.rm` logical indicating whether NA's should be removed.

`revC` logical indicating if the column order should be reversed for plotting, such that e.g., for the symmetric case, the symmetry axis is from lower left to upper right.

`add.expr` expression that will be evaluated after the call to image. Can be used to add components to the plot, eg., `add.expr=abline(...)`

`breaks` (optional) either a numeric vector indicating the splitting points for binning x into colors, or a integer number of break points to be used, in which case the break points will be spaced equally between min(x) and max(x).

`symbreaks` Boolean indicating whether breaks should be made symmetric about 0. Defaults to TRUE if the data includes negative values, and to FALSE otherwise.

`col` colors used for the image. Defaults to heat colors (`heat.colors`).

`colsep, rowsep, sepcolor`
(optional) vector of integers indicating which columns or rows, should be separated from the preceding columns or rows by a box of the color `sepcolor`.

`sepwidth` (optional) vector of length 2 giving the line width of the 2 vertical lines and the line width of the 2 horizontal lines of the separation box to be drawn, as a proportion of the width or height of a cell. Defaults to c(0.05, 0.05)

`cellnote` (optional) matrix of character strings which will be placed inside some cells, e.g. *p*-value symbols.

`notecex` (optional) numeric scaling factor for cellnote. Defaults to 1.

`notecol` (optional) character string specifying the color for cellnote. Defaults to "cyan".

`na.color` color to use for missing value (NA). Defaults to the background color.

`trace` character string indicating whether a solid "trace" line should be drawn along "row"s or down "column"s, "both" or "none". The distance of the line from the center of each colored cell is proportional to the size of the measurement. Defaults to "column".

`tracecol` character string giving the color for "trace" line. Defaults to "cyan".

`hline,vline,linecol`
(optional) vector of values within cells where a horizontal or vertical dotted line should be drawn. The color of the line is controlled by linecol. Horizontal lines are only plotted if trace is 'row' or 'both'. Vertical lines are only drawn if trace is "column" or "both". hline and vline default to the median of the breaks, linecol defaults to the value of tracecol.

`margins` numeric vector of length 2 containing the margins (see `par(mar= *)`) for column and row names, respectively.

ColSideColors	(optional) character vector of length ncol(x) containing the color names for a horizontal side bar that may be used to annotate the columns of x.
RowSideColors	(optional) character vector of length nrow(x) containing the color names for a vertical side bar that may be used to annotate the rows of x.
cexRow, cexCol	positive numbers, used as cex.axis in for the row or column axis labeling. Defaults to 0.2 + 1/$\log_{10}$(nr) and 0.2 + 1/$\log_{10}$(ncol), respectively.
labRow, labCol	character vectors with row and column labels to use. Defaults to rownames(x) and colnames(x), respectively.
key	logical indicating whether a color-key should be shown.
keysize	numeric value indicating the size of the key.
density.info	character string indicating whether to superimpose a 'histogram', a 'density' plot, or no plot ('none') on the color-key.
denscol	character string giving the color for the density display specified by density.info, defaults to the same value as tracecol.
symkey	Boolean indicating whether the color key should be made symmetric about 0. Defaults to TRUE if the data includes negative values, and to FALSE otherwise.
densadj	numeric scaling value for tuning the kernel width when a density plot is drawn on the color key. (See the adjust parameter for the density function for details.) Defaults to 0.25.
main, xlab, ylab	main, x- and y-axis titles; defaults to "".
lmat, lhei, lwid	visual layout: position matrix, column height, column width.
leftMargin, bottomMargin	sets the left and bottom margins respectively of the plot region.
reverse	Boolean indicating whether to reverse the rows of x.
...	additional arguments passed on to image.

Value

Invisibly, a list with components

rowInd	row index permutation vector as returned by order.dendrogram.
colInd	column index permutation vector as returned by order.dendrogram.
call	the matched call
carpet	reordered and scaled 'x' values used to generate the main 'carpet'
breaks	values used for break points in the color key
col	a character vector giving all the color IDs used in the heatmap
colorTable	A three-column data frame providing the lower and upper bounds and a color for each bin

Author(s)

Jingyi Jessica Li, Wei Vivian Li

References

Li WV, Chen Y and Li JJ (2016). TROM: A Testing-Based Method for Finding Transcriptomic Similarity of Biological Samples. Statistics in Biosciences. DOI: 10.1007/s12561-016-9163-y

Li JJ, Huang H, Bickel PJ, & Brenner SE (2014). Comparison of *D. melanogaster* and *C. elegans* developmental stages, tissues, and cells by modENCODE RNA-seq data. *Genome Research*, 24(7), 1086-1101.

https://gist.github.com/nachocab/3853004

See Also

image, heatmap

Examples

```
## using TROM to map developmental stages of D. melanogaster

## dm_gene_expr.rda can be downloaded and unzipped from
## http://www.stat.ucla.edu/~jingyi.li/packages/TROM/TROM_Rdata.zip.

## Not run:
load("dm_gene_expr.rda")
dm_trom <- ws.trom(sp_gene_expr = dm_gene_expr, z_thre = 1.5,
                   provide = FALSE)

## Draw heatmap without dendrogram
pdf("D. melanogaster Stage Mapping (without dendrogram).pdf", width = 10, height =8)
heatmap.3(   dm_trom,
             Rowv = NULL,
             Colv = NULL,
             dendrogram = c("none"),
             distfun = dist,
             hclustfun = hclust,
             xlab = "",
             ylab = "",
             main = "D. melanogaster Stage Mapping",
             key = TRUE,
             keysize = 1,
             trace = "none",
             density.info = c("none"),
             col = terrain.colors(120)
  )
dev.off()

## Draw heatmap with dendrogram
pdf("D. melanogaster Stage Mapping (with dendrogram).pdf", width = 10, height= 8)
heatmap.3(   dm_trom,
             Rowv = TRUE,
             Colv = TRUE,
             dendrogram = c("row"),
             distfun = dist,
             hclustfun = hclust,
```

```
            xlab = "",
            ylab = "",
            main = "D. melanogaster Stage Mapping",
            key = TRUE,
            keysize = 1,
            trace = "none",
            density.info = c("none"),
            col = terrain.colors(120),
  )
dev.off()
## End(Not run)
```

`select.associated.genes` *Select the associated genes for biological samples of a species*

Description

`select.associated.genes()` finds the ids of associated genes of a species based on criterion: Z-scores $\geq$ `z_thre` and saves the results to an .xlsx file.

Usage

```
select.associated.genes(sp_gene_expr, z_thre = 1.5,
                        save = TRUE, plot_distribution = FALSE)
```

Arguments

`sp_gene_expr`	a data frame containing gene expression estimates of the species; rows correspond to genes; columns (from the second to the last) correspond to samples, with the first column as gene IDs.
`z_thre`	a numeric value specifiying the Z-score threshold used to select associated genes, whose Z-scores $\geq$ `z_thre`. Defaults to 1.5. This can be specified by users or calculated using `choose.z()`.
`save`	a Boolean value specifying whether to save the associated genes to an Excel file. If `save = TRUE`, this function saves the results to an Excel file named "associated genes.xlsx".
`plot_distribution`	a Boolean value specifying whether to output the distribution of the number of associated genes across different samples. If `plot_distribution = TRUE`, this function outputs a barplot of the number of associated genes for each sample in a pdf file named "number of sample associated genes.pdf".

Value

a data frame containing the associated genes of every sample from the species. Every column in the data frame stores the associated gene IDs for the corresponding sample.

Author(s)

Jingyi Jessica Li, Wei Vivian Li

References

Li WV, Chen Y and Li JJ (2016). TROM: A Testing-Based Method for Finding Transcriptomic Similarity of Biological Samples. Statistics in Biosciences. DOI: 10.1007/s12561-016-9163-y

Li JJ, Huang H, Bickel PJ, & Brenner SE (2014). Comparison of *D. melanogaster* and *C. elegans* developmental stages, tissues, and cells by modENCODE RNA-seq data. *Genome Research*, 24(7), 1086-1101.

See Also

`select.associated.orthologs`

Examples

```
## Find the associaed genes of D. melanogaster

## dm_gene_expr.rda can be downloaded and unzipped from
## http://www.stat.ucla.edu/~jingyi.li/packages/TROM/TROM_Rdata.zip.

## Not run:
load("dm_gene_expr.rda")
dm_associated_genes <- select.associated.genes(sp_gene_expr = dm_gene_expr,
                       z_thre = 1.5, save = TRUE,
                       plot_distribution = TRUE)

## End(Not run)
```

`select.associated.orthologs` *Select the associated genes for biological samples of a species among the genes with orthologs in the other species*

Description

`select.associated.orthologs()` finds the ids of associated genes of a species within orthologous genes based on criterion: Z-scores $\geq$ `z_thre` and saves the results to an .xlsx file.

Usage

```
select.associated.orthologs(sp_gene_expr, sp1_sp2_orthologs,
                            z_thre = 1.5, i, save = TRUE,
                            plot_distribution = FALSE)
```

Arguments

sp_gene_expr	a data frame containing gene expression estimates of the species; rows correspond to genes; columns (from the second to the last) correspond to samples, with the first column as gene IDs.
sp1_sp2_orthologs	a data frame containing ortholog gene pairs between species 1 and 2.
z_thre	a numeric value specifiying the Z-score threshold used to select associated genes, whose Z-scores $\geq$ z_thre. Defaults to 1.5. This can be specified by users or calculated using choose.z().
i	an integer specifying which column of sp1_sp2_orthologs the species corresponds to. 1 for the first column and 2 for the second column.
save	a Boolean value specifying whether to save the associated orthologs to an Excel file. If save = TRUE, this function saves the results to an Excel file named "associated genes within ortholog genes.xlsx".
plot_distribution	a Boolean value specifying whether to output the distribution of the number of associated orthologs across different samples. If plot_distribution = TRUE, this function outputs a barplot of the number of associated orthologs for each sample in a pdf file named "number of sample associated orthologous genes.pdf".

Value

A data frame containing the associated genes within orthologous genes of each samples from the specified species. Every column in the data frame stores the associated gene IDs for the corresponding sample.

Author(s)

Jingyi Jessica Li, Wei Vivian Li

References

Li WV, Chen Y and Li JJ (2016). TROM: A Testing-Based Method for Finding Transcriptomic Similarity of Biological Samples. Statistics in Biosciences. DOI: 10.1007/s12561-016-9163-y

Li JJ., Huang H, Bickel PJ, & Brenner SE (2014). Comparison of *D. melanogaster* and *C. elegans* developmental stages, tissues, and cells by modENCODE RNA-seq data. *Genome Research*, 24(7), 1086-1101.

See Also

select.associated.genes

Examples

```
## Find the associaed genes of D. melanogaster samples
## among the D. melanogaster genes having orthologs in C.elegans

## dm_gene_expr.rda and dm_ce_orthologs.rda can be downloaded and unzipped from
```

```
## http://www.stat.ucla.edu/~jingyi.li/packages/TROM/TROM_Rdata.zip.
## Not run:
load("dm_gene_expr.rda")
load("dm_ce_orthologs.rda")
dm_associated_orthologs <- select.associated.orthologs(sp_gene_expr = dm_gene_expr,
                           sp1_sp2_orthologs = dm_ce_orthologs, z_thre = 1.5,
                           i = 1, save = TRUE, plot_distribution = TRUE)
## End(Not run)
```

---

ws.trom *Within-species transcriptome mapping*

Description

This function calculates the TROM scores in comparing/mapping samples from the same species. TROM score = -$\log_{10}$(Bonferroni-corrected *p*-value from a hypergeometric test), with a minimum value of 0.

Usage

```
ws.trom(sp_gene_expr = data.frame(), single = TRUE, sp_gene_expr2 = NULL,
        z_thre = 1.5, provide = FALSE, gene_lists = "",
        save_overlap_genes = FALSE)
```

Arguments

sp_gene_expr	a data frame containing gene expression estimates of the species; rows correspond to genes; columns (from the second to the last) correspond to samples, with the first column as gene IDs. Not needed if provide = TRUE.
single	a Boolean value indicating whether the within-species comparison will be conducted on a single dataset. If single = FALSE, the users need to provide sp_gene_expr2. Or if provide = TRUE, users need have a second sheet in the .xlsx file (that is to be supplied into gene_lists) to specify the second set of gene lists corresponding to another set of biological samples of the same species. Defaults to TRUE.
sp_gene_expr2	NULL (default) or a data frame containing gene expression estimates of the species; rows correspond to genes; columns (from the second to the last) correspond to samples, with the first column as gene IDs.
z_thre	a numeric value specifiying the Z-score threshold used to select associated genes, whose Z-scores $\geq$ z_thre. Defaults to 1.5. This can be specified by users or calculated using choose.z().
provide	a Boolean value indicating whether associated genes are user-provided. If provide = TRUE, the users need to provide lists of genes that they think can represent the transcriptome characteristics of different samples.
gene_lists	an .xlsx file containing user-provided gene lists. It is required when provide = TRUE.

save_overlap_genes
: a Boolean value indicating whether the users want to save overlap genes between every two samples from the species to an .xlsx file. If save_overlap_genes = TRUE, this function outputs the overlap genes to "within-species overlapping genes between sample pairs.xlsx". Defaults to FALSE.

Details

If provide = TRUE, users provide gene lists with gene_lists to calculate the TROM scores; otherwise, the function will automatically select associated genes based on the criterion: Z-scores $\geq$ z_thre.

If single = TRUE, TROM socres are calculated from single dataset of the species. Users should either specify sp_gene_expr or provide gene_lists. If provided, gene_lists should be a one-sheet Excel file. In the Excel file, rows represent gene ids and columns represent biological samples. Each column of the file stores the user-provided genes corresponding to the sample of that column. Please note that different columns may have different numbers of rows.

If single = FALSE, TROM socres are calculated from two different datasets of the species. Users should either specify sp_gene_expr and sp_gene_expr2 or provide gene_lists. If provided, gene_lists should be a two-sheet Excel file with the first sheet for one dataset and the second sheet for the other dataset. In each sheet, rows represent gene ids and columns represent biological samples. Each column of the file stores the user-provided genes corresponding to the sample of that column.

This function outputs the within-species TROM scores into an .xlsx file named "within-species TROM scores.xlsx".

Value

A matrix of within-species TROM scores, where rows and columns correspond to the samples of the species respectively.

Author(s)

Jingyi Jessica Li, Wei Vivian Li

References

Li WV, Chen Y and Li JJ (2016). TROM: A Testing-Based Method for Finding Transcriptomic Similarity of Biological Samples. Statistics in Biosciences. DOI: 10.1007/s12561-016-9163-y

Li JJ, Huang H, Bickel PJ, & Brenner SE (2014). Comparison of *D. melanogaster* and *C. elegans* developmental stages, tissues, and cells by modENCODE RNA-seq data. *Genome Research*, 24(7), 1086-1101.

See Also

bs.trom, ws.trom.orthologs

Examples

```
## Calculating transcriptome overlap measure within D. melanogaster

## dm_gene_expr.rda and dm_tissue_expr.rda
## can be downloaded and unzipped from
## http://www.stat.ucla.edu/~jingyi.li/packages/TROM/TROM_Rdata.zip

## Not run:
load("dm_gene_expr.rda")
## without user-provided gene lists
# use single dataset
dm_trom <- ws.trom(sp_gene_expr = dm_gene_expr, z_thre = 1.5,
                    provide = FALSE, save_overlap_genes = TRUE)

# use two different dataset
# calculate TROM scores between timecourse and tissue/cell line
# data within D. melanogaster
load("dm_tissue_expr.rda")
dm_trom2 <- ws.trom(sp_gene_expr = dm_gene_expr, single = FALSE,
                   sp_gene_expr2 = dm_tissue_expr, z_thre = 1.5,
                   provide = FALSE, save_overlap_genes = FALSE)

## with user-provided gene lists
gene_lists <- system.file("dm_associated_genes.xlsx", package = "TROM")
dm_trom3 <- ws.trom(provide = TRUE, gene_lists = gene_lists)

## End(Not run)
```

`ws.trom.orthologs` *Within-species transcriptome mapping using ortholog genes*

Description

This function calculates the TROM scores in comparing samples from the same species. TROM score = -$\log_{10}$(Bonferroni-corrected p-value from a hypergeometric test), with a minimum value of 0. And the hypergeometric test is performed on the genes having orthologs in the other species.

Usage

```
ws.trom.orthologs(sp1_sp2_orthologs, sp_gene_expr = NULL, single = TRUE,
                   sp_gene_expr2 = NULL, z_thre = 1.5, i,
                   provide = FALSE, gene_lists = NULL,
                   save_overlap_genes = FALSE)
```

Arguments

`sp1_sp2_orthologs`
: a data frame containing ortholog gene pairs between species 1 and 2.

`sp_gene_expr` a data frame containing gene expression estimates of the species; rows correspond to genes; columns (from the second to the last) correspond to samples, with the first column as gene IDs. Not needed if `provide = TRUE`.

`single` a Boolean value indicating whether the within-species comparison will be conducted on a single dataset. If `single = FALSE`, the users need to provide `sp_gene_expr2`. Or if `provide = TRUE`, users need have a second sheet in the .xlsx file (that is to be supplied into `gene_lists`) to specify the second set of gene lists corresponding to another set of biological samples of the same species. Defaults to `TRUE`.

`sp_gene_expr2` `NULL` (default) or a data frame containing gene expression estimates of the species; rows correspond to genes; columns (from the second to the last) correspond to samples, with the first column as gene IDs.

`z_thre` a numeric value specifiying the Z-score threshold used to select associated orthologous genes, whose Z-scores $\geq$ `z_thre`. Defaults to 1.5. This can be specified by users or calculated using `choose.z()`.

`i` an integer specifying which column of `sp1_sp2_orthologs` the species corresponds to. 1 for the first column and 2 for the second column.

`provide` a Boolean value indicating whether associated geness are user-provided. If `provide = TRUE`, the users need to provide lists of genes that they think can represent the transcriptome characteristics of different samples.

`gene_lists` an .xlsx file containing user-provided gene lists. It is required when `provide = TRUE`.

`save_overlap_genes`
a Boolean value indicating whether the users want to save overlap genes between every two samples from the species to a .xlsx file. If `save_overlap_genes = TRUE`, this function outputs the overlap genes (within ortholog genes) to "within-species overlapping genes (within ortholog genes) between sample pairs.xlsx". Defaults to FALSE.

Details

If `provide = TRUE`, users provide gene lists with `gene_lists` to calculate the TROM scores; otherwise, the function will automatically select associated orthologous genes based on the criterion: Z-scores $\geq$ `z_thre`.

If `single = TRUE`, TROM socres are calculated from single dataset of the species. Users should either specify `sp_gene_expr` or provide `gene_lists`. If provided, `gene_lists` should be a one-sheet Excel file. In the Excel file, rows represent gene ids and columns represent biological samples. Each column of the file stores the user-provided genes corresponding to the sample of that column. Please note that different columns may have different numbers of rows.

If `single = FALSE`, TROM socres are calculated from two different datasets of the species. Users should either specify `sp_gene_expr` and `sp_gene_expr2` or provide `gene_lists`. If provided, `gene_lists` should be a two-sheet Excel file with the first sheet for one dataset and the second sheet for the other dataset. In each sheet, rows represent gene ids and columns represent biological samples. Each column of the file stores the user-provided genes corresponding to the sample of that column.

This function outputs the within-species TROM scores into an .xlsx file named "within-species TROM scores (with ortholog genes).xlsx".

Value

A matrix of within-species TROM scores using orthologous genes, where rows and columns correspond to the samples of the species respectively.

Author(s)

Jingyi Jessica Li, Wei Vivian Li

References

Li WV, Chen Y and Li JJ (2016). TROM: A Testing-Based Method for Finding Transcriptomic Similarity of Biological Samples. Statistics in Biosciences. DOI: 10.1007/s12561-016-9163-y

Li JJ, Huang H, Bickel PJ, & Brenner SE (2014). Comparison of *D. melanogaster* and *C. elegans* developmental stages, tissues, and cells by modENCODE RNA-seq data. *Genome Research*, 24(7), 1086-1101.

See Also

ws.trom, bs.trom.

Examples

```
## Calculating transcriptome overlap measure within D. melanogaster
## using orthologous genes

## The .rda files used in this example can be downloaded and unzipped from
## http://www.stat.ucla.edu/~jingyi.li/packages/TROM/TROM_Rdata.zip.
## Not run:
load("dm_gene_expr.rda")
load("dm_ce_orthologs.rda")
## use single dataset of D. melanogaster
# without user-provided gene lists
dm_trom_orth <- ws.trom.orthologs(sp1_sp2_orthologs = dm_ce_orthologs,
                                  sp_gene_expr = dm_gene_expr, single = TRUE,
                                  z_thre = 1.5, i = 1, provide = FALSE,
                                  save_overlap_genes = FALSE)

# with user-provided gene lists
gene_lists <- system.file("dm_associated_orthologs.xlsx", package = "TROM")
dm_trom_orth2 <- ws.trom.orthologs(sp1_sp2_orthologs = dm_ce_orthologs,
                                   i = 1, provide = TRUE, gene_lists = gene_lists,
                                   save_overlap_genes = FALSE)

## use two different datasets of D. melanogaster
# without user-provided gene lists
dm_gene_expr2 <- dm_gene_expr[,1:13]
dm_trom_orth3 <- ws.trom.orthologs(sp1_sp2_orthologs = dm_ce_orthologs,
                                  sp_gene_expr = dm_gene_expr,
                                  single = FALSE, sp_gene_expr2 = dm_gene_expr2,
                                  z_thre = 1.5, i = 1,
                                  provide = FALSE, save_overlap_genes = FALSE)
```

```
# with user-provided gene lists
dm_trom_orth4 <- ws.trom.orthologs(sp1_sp2_orthologs = dm_ce_orthologs, single = FALSE,
                                   i = 1, provide = TRUE, gene_lists = gene_lists,
                                   save_overlap_genes = FALSE)


## End(Not run)
```

Index