Package ‘DEG.comparison’

June 10, 2015

Type Package

Title DEG.comparison: A comparison of methods for DEG analysis of RNA-seq data

Version 1.0

Date 2015-06-05

Author Daniela Cassol

Maintainer Daniela Cassol <danicassol@gmail.com>

Description A comparison of methods for DEG analysis of RNA-seq data

Depends NBPSeq, baySeq, systemPipeR, ggplot2, ROCR, ape

Imports NBPSeq, baySeq, systemPipeR, ggplot2, ROCR, ape

URL https://github.com/dcassol/DEG.comparison

License What license is it under?

R topics documented:

filterDEGnew *Filter and plot DEG results*

Description

Function adapted from systemPipeR. Filters and plots DEG results for a given set of sample comparisons. The gene idenifiers of all (i) Up_or_Down, (ii) Up and (iii) Down regulated genes are stored as separate list components and the corresponding summary statistics, stored in a fourth list component, is plotted in form of a stacked bar plot.

Usage

```
filterDEGnew(degDF, filter, plot = TRUE, method)
```

Arguments

`degDF`	`data.frame` generated by `run_edgeR`
`filter`	Named vector with filter cutoffs of format `c(Fold=2, FDR=1)` where `Fold` refers to the fold change cutoff (unlogged) and `FDR` to the p-value cutoff.
`plot`	Allows to turn plotting behavior on and off with default set to `TRUE`.
`method`	Defines the method name in the plot.

Value

Returns `list` with four components

`UporDown`	List of up or down regulated gene/transcript indentifiers meeting the chosen filter settings for all comparisons defined in data frames `pval` and `log2FC`.
`Up`	Same as above but only for up regulated genes/transcript.
`Down`	Same as above but only for down regulated genes/transcript.

Author(s)

Daniela Cassol

References

Thomas Girke (2015). systemPipeR: systemPipeR: NGS workflow and report generation environment. R package version 1.0.12. https://github.com/tgirke/systemPipeR

See Also

`run_edgeR, run_DESeq2, run_NBPSeq_glm, run_NBPSeq_nbp, run_TSPM`

Examples

```
targetspath <- system.file("extdata", "targets.txt", package="DEG.comparison")
targets <- read.delim(targetspath, comment="#")
cmp <- readComp(file=targetspath, format="matrix", delim="-")
countDFeByg <- system.file("extdata", "countDFeByg.xls", package="DEG.comparison")
countDFeByg <- read.delim(countDFeByg, row.names=1)
Comp3 <- list(AP1.4_AP1.67=c("AP1.4A","AP1.4B", "AP1.67A", "AP1.67B"), AP3.4_AP3.67=c("AP3.4A","AP3.4B", "AP3.67
edgeDF <- run_edgeR(countDF=countDFeByg, targets=targets, cmp=cmp[[1]], independent=FALSE, mdsplot="")
DEG_list_edgeR <- filterDEGnew(degDF=edgeDF, filter=c(Fold=2, FDR=1), method="edgeR")
DEG_list_edgeR$Summary[1:4,]
```

filterDEG_FDR *Filter FDR and plot DEG results*

Description

Function adapted from systemPipeR. Filters and plots DEG results for a given set of sample comparisons. The gene idenifiers of all (i) Up_or_Down regulated genes are stored as separate list components and the corresponding summary statistics, stored in a one list component, is plotted in form of a stacked bar plot.

Usage

```
filterDEG_FDR(degDF, filter, plot = TRUE, method)
```

Arguments

`degDF` `data.frame` generated by `run_BaySeq`

`filter` Named vector with filter cutoffs of format `c(FDR=1)` where `FDR` to the p-value cutoff.

`plot` Allows to turn plotting behavior on and off with default set to `TRUE`.

`method` Defines the method name in the plot.

Value

Returns `list` with one components

`UporDown` List of up or down regulated gene/transcript indentifiers meeting the chosen filter settings for all comparisons defined in data frames `pval`.

Author(s)

Daniela Cassol

References

Thomas Girke (2015). systemPipeR: systemPipeR: NGS workflow and report generation environment. R package version 1.0.12. https://github.com/tgirke/systemPipeR

See Also

`run_BaySeq`

`filterDEG_logFC` *Filter logFC and plot DEG results*

Description

Function adapted from systemPipeR. Filters and plots DEG results for a given set of sample comparisons. The gene idenifiers of all (i) Up_or_Down, (ii) Up and (iii) Down regulated genes are stored as separate list components and the corresponding summary statistics, stored in a fourth list component, is plotted in form of a stacked bar plot.

Usage

```
filterDEG_logFC(degDF, filter, plot = TRUE, method)
```

Arguments

`degDF`	`data.frame` generated by `run_RPKM`
`filter`	Named vector with filter cutoffs of format `c(Fold=2)` where `Fold` refers to the fold change cutoff (unlogged).
`plot`	Allows to turn plotting behavior on and off with default set to `TRUE`.
`method`	Defines the method name in the plot.

Value

Returns `list` with four components

`UporDown`	List of up or down regulated gene/transcript indentifiers meeting the chosen filter settings for all comparisons defined in data frames `log2FC`.
`Up`	Same as above but only for up regulated genes/transcript.
`Down`	Same as above but only for down regulated genes/transcript.

Author(s)

Daniela Cassol

References

Thomas Girke (2015). systemPipeR: systemPipeR: NGS workflow and report generation environment. R package version 1.0.12. https://github.com/tgirke/systemPipeR

See Also

`run_RPKM`

Examples

```
targetspath <- system.file("extdata", "targets.txt", package="DEG.comparison")
targets <- read.delim(targetspath, comment="#")
cmp <- readComp(file=targetspath, format="matrix", delim="-")
rpkmDFeByg <- system.file("extdata", "rpkmDFeByg.xls", package="DEG.comparison")
rpkmDFeByg <- read.delim(rpkmDFeByg, row.names=1)
#Settings
Comp1 <- list(Factor=(Reduce(union, targets$Factor)), Sample=c(colnames(rpkmDFeByg)),
              group=c(1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8))

Comp2 <- list(AP1.4_AP1.67=c("AP1.4", "AP1.67"), AP3.4_AP3.67=c("AP3.4", "AP3.67"),
              AG.4_AG.67=c("AG.4", "AG.67"), AP1.4_AP3.4=c("AP1.4", "AP3.4"),
              AP1.4_AG.4=c("AP1.4", "AG.4"), AP3.4_AG.4=c("AP3.4", "AG.4"),
              AP1.67_AP3.67=c("AP1.67", "AP3.67"), AP1.67_AG.67=c("AP1.67", "AG.67"),
              AP3.67_AG.67=c("AP3.67", "AG.67"))

##Compute mean values for replicates and logFC for comparisons
RPKM_FC <- run_RPKM (rpkmDFeByg, Comp1, Comp2)
DEG_list_RPKM <- filterDEG_logFC(degDF=RPKM_FC, filter=c(Fold=2), method="RPKM")
DEG_list_RPKM$Summary[1:4,]
```

olBarplot *olBarplot - Define Bar Plot Function*

Description

Detailed instructions for using the functions of this script are available on this page: http://faculty.ucr.edu/~tgirke/Documents/

Usage

```
olBarplot(OLlist = OLlist, mycol = "default", margins = c(6, 10, 3, 2), mincount = 0, mysub = "default"
```

Details

Utilities: (1) Venn Intersects Computation of Venn intersects among 2-20 or more sample sets using the typical 'only in' intersect logic of Venn comparisons, such as: objects present only in set A, objects present only in the intersect of A & B, etc. Due to this restrictive intersect logic, the combined Venn sets contain no duplicates. (2) Regular Intersects Computation of regular intersects among 2-20 or more sample sets using the following intersect logic: objects present in the intersect of A & B, objects present in the intersect of A & B & C, etc. The approach results usually in many duplications of objects among the intersect sets. (3) Graphical Utilities - Venn diagrams of 2-5 sample sets. - Bar plots for the results of Venn intersect and all intersect approaches derived from many samples sets.

Detailed instructions for using the functions of this script are available on this page: http://faculty.ucr.edu/~tgirke/Documents/

Revision history: March 24, 2012: fixed substring problem in plotVenn function

Author(s)

Thomas Girke

References

Thomas Girke (2015). systemPipeR: systemPipeR: NGS workflow and report generation environment. R package version 1.0.12. https://github.com/tgirke/systemPipeR

See Also

`vennPlot`, `overLapper`

`overLapper` *overLapper - Define Generic Intersect Function*

Description

Detailed instructions for using the functions of this script are available on this page: http://faculty.ucr.edu/~tgirke/Documents/

Usage

```
overLapper(setlist = setlist, complexity = 1:length(setlist), sep = "-", cleanup = FALSE, keepdups = F
```

Details

Utilities: (1) Venn Intersects Computation of Venn intersects among 2-20 or more sample sets using the typical 'only in' intersect logic of Venn comparisons, such as: objects present only in set A, objects present only in the intersect of A & B, etc. Due to this restrictive intersect logic, the combined Venn sets contain no duplicates. (2) Regular Intersects Computation of regular intersects among 2-20 or more sample sets using the following intersect logic: objects present in the intersect of A & B, objects present in the intersect of A & B & C, etc. The approach results usually in many duplications of objects among the intersect sets. (3) Graphical Utilities - Venn diagrams of 2-5 sample sets. - Bar plots for the results of Venn intersect and all intersect approaches derived from many samples sets.

Detailed instructions for using the functions of this script are available on this page: http://faculty.ucr.edu/~tgirke/Documents/

Revision history: March 24, 2012: fixed substring problem in plotVenn function

Author(s)

Thomas Girke

References

Thomas Girke (2015). systemPipeR: systemPipeR: NGS workflow and report generation environment. R package version 1.0.12. https://github.com/tgirke/systemPipeR

See Also

`olBarplot`, `vennPlot`

`panel.cor` *panel.cor - Scatterplot*

Description

panel.cor puts correlation in upper panels, size proportional to correlation.

Usage

```
panel.cor(x, y, digits = 2, prefix = "", cex.cor, ...)
```

Value

Returns `plot` with scatterplot matrix.

Author(s)

Daniela Cassol

References

http://www.gettinggeneticsdone.com/2011/07/scatterplot-matrices-in-r.html

See Also

`pairs`

`run_BaySeq` *run_BaySeq - empirical Bayesian methods.*

Description

BaySeq package identifies differential expression in high-throughput 'count' data, such as that derived from next-generation sequencing machines, calculating estimated posterior likelihoods of differential expression (or more complex hypotheses) via empirical Bayesian methods.

Usage

```
run_BaySeq(counts, mycomp3, number)
```

Arguments

`counts`	`date.frame` containing raw read counts.
`mycomp3`	`list` where comparisons are defined in a list.
`number`	`number` number the rows in the counts. ex: dim(counts).

Value

`data.frame` containing `baySeq` results from all comparisons. Comparison labels are appended to column titles for tracking.

Author(s)

Daniela Cassol

References

Hardcastle, T.J. & Kelly, K.A., 2010. baySeq: empirical Bayesian methods for identifying differential expression in sequence count data. BMC bioinformatics, 11, p.422.

See Also

`filterDEG_FDR`

`run_NBPSeq_glm` *run_NBPSeq_glm - Negative Binomial (NB) models for two-group comparisons and regression inferences from RNA-Sequencing Data.*

Description

For each row of the input data matrix, nb.glm.test fits an NB log-linear regression model and performs large-sample tests for a one-dimensional regression coefficient.

Usage

```
run_NBPSeq_glm(counts, mycomp3)
```

Arguments

`counts`	`date.frame` containing raw read counts.
`mycomp3`	`list` where comparisons are defined in a list.

Value

`data.frame` containing `NBPSeq_glm` results from all comparisons. Comparison labels are appended to column titles for tracking.

Author(s)

Daniela Cassol

References

Di, Y. et al., The NBP Negative Binomial Model for Assessing Differential Gene Expression from RNA-Seq. Statistical applications in genetics and molecular biology, 10(1), pp.1???28.

See Also

`run_NBPSeq_nbp` and `NBPSeq` vignette

Examples

```
targetspath <- system.file("extdata", "targets.txt", package="DEG.comparison")
targets <- read.delim(targetspath, comment="#")
cmp <- readComp(file=targetspath, format="matrix", delim="-")
countDFeByg <- system.file("extdata", "countDFeByg.xls", package="DEG.comparison")
countDFeByg <- read.delim(countDFeByg, row.names=1)
Comp3 <- list(AP1.4_AP1.67=c("AP1.4A","AP1.4B", "AP1.67A", "AP1.67B"), AP3.4_AP3.67=c("AP3.4A","AP3.4B", "AP3.67
NBPSeq.glmDF <- run_NBPSeq_glm (countDFeByg, Comp3)
DEG_list_NBPSeq.glmDF <- filterDEGnew(degDF=NBPSeq.glmDF, filter=c(Fold=2, FDR=1), method="NBPSeq.glm")
DEG_list_NBPSeq.glmDF$Summary[1:4,]
```

run_NBPSeq_nbp — *run_NBPSeq_nbp - Negative Binomial (NB) models for two-group comparisons and regression inferences from RNA-Sequencing Data.*

Description

nbp.test fits an NBP model to the RNA-Seq counts and performs Robinson and Smyth's exact NB test on each gene to assess differential gene expression between two groups.

Usage

```
run_NBPSeq_nbp(counts, mycomp3)
```

Arguments

`counts`	`date.frame` containing raw read counts.
`mycomp3`	`list` where comparisons are defined in a list.

Value

`data.frame` containing `NBPSeq_nbp` results from all comparisons. Comparison labels are appended to column titles for tracking.

Author(s)

Daniela Cassol

References

Di, Y. et al., The NBP Negative Binomial Model for Assessing Differential Gene Expression from RNA-Seq. Statistical applications in genetics and molecular biology, 10(1), pp.1???28.

See Also

`run_NBPSeq_glm` and `NBPSeq` vignette

Examples

```
targetspath <- system.file("extdata", "targets.txt", package="DEG.comparison")
targets <- read.delim(targetspath, comment="#")
cmp <- readComp(file=targetspath, format="matrix", delim="-")
countDFeByg <- system.file("extdata", "countDFeByg.xls", package="DEG.comparison")
countDFeByg <- read.delim(countDFeByg, row.names=1)
Comp3 <- list(AP1.4_AP1.67=c("AP1.4A","AP1.4B", "AP1.67A", "AP1.67B"), AP3.4_AP3.67=c("AP3.4A","AP3.4B", "AP3.67
NBPSeq.nbpDF <- run_NBPSeq_nbp (countDFeByg, Comp3)
DEG_list_NBPSeq.nbpDF <- filterDEGnew(degDF=NBPSeq.nbpDF, filter=c(Fold=2, FDR=1), method="NBPSeq.nbp")
DEG_list_NBPSeq.nbpDF$Summary[1:4,]
```

`run_RPKM` *run_RPKM*

Description

Simple Fold Change Method - RPKM

Usage

```
run_RPKM(counts, mycomp1, mycomp2)
```

Arguments

`counts`	`date.frame` containing raw read counts.
`mycomp1`	codelist where Factor, Names and groups are defined in a list.
`mycomp2`	`list` where comparisons are defined in a list.

Value

`data.frame` containing `RPKM` results from all comparisons. Comparison labels are appended to column titles for tracking.

Author(s)

Daniela Cassol

References

Mortazavi, A., Williams, B. A., McCue, K., Schaeffer, L., and Wold, B. (2008). Mapping and quantifying mammalian transcriptomes by rna-seq. Nat Methods, 5(7):621-628.

See Also

`filterDEG_logFC`

Examples

```
targetspath <- system.file("extdata", "targets.txt", package="DEG.comparison")
targets <- read.delim(targetspath, comment="#")
cmp <- readComp(file=targetspath, format="matrix", delim="-")
rpkmDFeByg <- system.file("extdata", "rpkmDFeByg.xls", package="DEG.comparison")
rpkmDFeByg <- read.delim(rpkmDFeByg, row.names=1)
#Settings
Comp1 <- list(Factor=(Reduce(union, targets$Factor)), Sample=c(colnames(rpkmDFeByg)),
              group=c(1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8))

Comp2 <- list(AP1.4_AP1.67=c("AP1.4", "AP1.67"), AP3.4_AP3.67=c("AP3.4", "AP3.67"),
              AG.4_AG.67=c("AG.4", "AG.67"), AP1.4_AP3.4=c("AP1.4", "AP3.4"),
              AP1.4_AG.4=c("AP1.4", "AG.4"), AP3.4_AG.4=c("AP3.4", "AG.4"),
              AP1.67_AP3.67=c("AP1.67", "AP3.67"), AP1.67_AG.67=c("AP1.67", "AG.67"),
              AP3.67_AG.67=c("AP3.67", "AG.67"))
##Compute mean values for replicates and logFC for comparisons
RPKM_FC <- run_RPKM (rpkmDFeByg, Comp1, Comp2)
```

run_TSPM *run_TSPM - "A Two-Stage Poisson Model for Testing RNA-Seq Data"*

Description

Simple and powerful statistical approach, based on a two-stage Poisson model, for modeling RNA sequencing data and testing for biologically important changes in gene expression. Users are strongly encouraged to consult the Auer and Doerge (2011) for more detailed information on this topic and how to properly run `TSPM` on data sets with more complex experimental designs.

Usage

```
run_TSPM(counts, mycomp3)
```

Arguments

`counts` `date.frame` containing raw read counts.

`mycomp3` `list` where comparisons are defined in a list.

Value

`data.frame` containing `TSPM` results from all comparisons. Comparison labels are appended to column titles for tracking.

Author(s)

Daniela Cassol

References

Paul L. Auer, Rebecca W Doerge: A Two-Stage Poisson Model for Testing RNA-Seq Data. Statistical Applications in Genetics and Molecular Biology 2011, 10(1):26.

See Also

TSPM

Examples

```
targetspath <- system.file("extdata", "targets.txt", package="DEG.comparison")
targets <- read.delim(targetspath, comment="#")
cmp <- readComp(file=targetspath, format="matrix", delim="-")
countDFeByg <- system.file("extdata", "countDFeByg.xls", package="DEG.comparison")
countDFeByg <- read.delim(countDFeByg, row.names=1)
Comp3 <- list(AP1.4_AP1.67=c("AP1.4A","AP1.4B", "AP1.67A", "AP1.67B"), AP3.4_AP3.67=c("AP3.4A","AP3.4B", "AP3.6
TSPMDF <- run_TSPM(countDFeByg, Comp3)
DEG_list_TSPM <- filterDEGnew(degDF=TSPMDF, filter=c(Fold=2, FDR=1), method="TSPM")
DEG_list_TSPM$Summary[1:4,]
```

TSPM *TSPM - "A Two-Stage Poisson Model for Testing RNA-Seq Data"*

Description

Simple and powerful statistical approach, based on a two-stage Poisson model, for modeling RNA sequencing data and testing for biologically important changes in gene expression. Users are strongly encouraged to consult the Auer and Doerge (2011) for more detailed information on this topic and how to properly run TSPM on data sets with more complex experimental designs.

Usage

```
TSPM(counts, x1, x0, lib.size, alpha.wh = 0.05)
```

Arguments

counts — date.frame containing raw read counts

x1 — x1a vector of treatment group factors (under the alternative hypothesis)

x0 — x0 a vector of treatment group factors (under the null hypothesis)

lib.size — lib.size a vector of RNA-Seq library sizes. This could simply be obtained by specifying lib.size <- apply(counts,2,sum). It may also be any other appropriate scaling factor.

alpha.wh — alpha.wh the significance threshold to use for deciding whether a gene is overdispersed. Defaults to 0.05.

Value

Returns `list` with five components

`log.fold.change`
: List of a vector containing the estimated log fold changes for each gene.

`pvalues`
: A vector containing the raw p-values testing differential expression for each gene.

`index.over.disp`
: a vector of integer values containing the indices of the over-dispersed genes.

`index.not.over.dis`
: A vector of integer values containing the indices of the non-over-dispersed genes.

`padj`
: A vector containing the p-values after adjusting for multiple testing using the method of Benjamini-Hochberg.

Author(s)

Paul Auer (plivermo@fhcrc.org) and R.W. Doerge (doerge@purdue.edu)

References

Paul L. Auer, Rebecca W Doerge: A Two-Stage Poisson Model for Testing RNA-Seq Data. Statistical Applications in Genetics and Molecular Biology 2011, 10(1):26.

See Also

`run_TSPM`

`vennPlot` *vennPlot - Define Venn Diagram Plotting Function*

Description

Detailed instructions for using the functions of this script are available on this page: http://faculty.ucr.edu/~tgirke/Documents/

Details

Utilities: (1) Venn Intersects Computation of Venn intersects among 2-20 or more sample sets using the typical 'only in' intersect logic of Venn comparisons, such as: objects present only in set A, objects present only in the intersect of A & B, etc. Due to this restrictive intersect logic, the combined Venn sets contain no duplicates. (2) Regular Intersects Computation of regular intersects among 2-20 or more sample sets using the following intersect logic: objects present in the intersect of A & B, objects present in the intersect of A & B & C, etc. The approach results usually in many duplications of objects among the intersect sets. (3) Graphical Utilities - Venn diagrams of 2-5 sample sets. - Bar plots for the results of Venn intersect and all intersect approaches derived from many samples sets.

Detailed instructions for using the functions of this script are available on this page: http://faculty.ucr.edu/~tgirke/Documents/

Revision history: March 24, 2012: fixed substring problem in plotVenn function

Author(s)

Thomas Girke

References

Thomas Girke (2015). systemPipeR: systemPipeR: NGS workflow and report generation environment. R package version 1.0.12. https://github.com/tgirke/systemPipeR

See Also

`olBarplot`, `overLapper`

Index