Package ‘EWCE’

March 30, 2017

Type Package

Title Expression Weighted Celltype Enrichment

Version 1.2.0

Date 2016-12-04

Author Dr Nathan Skene

Maintainer Nathan Skene <nathan.skene@gmail.com>

Description Used to determine which cell types are enriched within gene lists. The package provides tools for testing enrichments within simple gene lists (such as human disease associated genes) and those resulting from differential expression studies. The package does not depend upon any particular Single Cell Transcriptome dataset and user defined datasets can be loaded in and used in the analyses.

License Artistic-2.0

LazyData TRUE

Depends R(>= 3.3)

VignetteBuilder knitr

Imports ggplot2, reshape2, biomaRt

Suggests knitr, BiocStyle

biocViews GeneExpression, Transcription, DifferentialExpression, GeneSetEnrichment, Genetics, Microarray, mRNAMicroarray, OneChannel, RNASeq, BiomedicalInformatics, Proteomics, Visualization, FunctionalGenomics

RoxygenNote 5.0.1

NeedsCompilation no

R topics documented:

`add.res.to.merging.list` *Add to results to merging list*

Description

`add.res.to.merging.list` Adds EWCE results to a list for merging analysis

Usage

```
add.res.to.merging.list(full_res, existing_results = NULL)
```

Arguments

`full_res` results list generated using `bootstrap.enrichment.test` or `ewce_expression_data` functions. Multiple results tables can be merged into one results table, as long as the 'list' column is set to distinguish them.

`existing_results` Output of previous rounds from adding results to list. Leave empty if this is the first item in the list.

Value

merged results list

Examples

```
# Load the single cell data
data(celltype_data)

# Load the data
data(tt_alzh)
data(tt_alzh_BA36)
data(tt_alzh_BA44)

# Run EWCE analysis
tt_results = ewce_expression_data(sct_data=celltype_data,tt=tt_alzh)
tt_results_36 = ewce_expression_data(sct_data=celltype_data,tt=tt_alzh_BA36)
tt_results_44 = ewce_expression_data(sct_data=celltype_data,tt=tt_alzh_BA44)

# Fill a list with the results
results = add.res.to.merging.list(tt_alzh)
results = add.res.to.merging.list(tt_alzh_BA36,results)
results = add.res.to.merging.list(tt_alzh_BA44,results)
```

bootstrap.enrichment.test

Bootstrap celltype enrichment test

Description

`bootstrap.enrichment.test` takes a genelist and a single cell type transcriptome dataset and determines the probability of enrichment and fold changes for each cell type.

Usage

```
bootstrap.enrichment.test(sct_data = NA, mouse.hits = NA, mouse.bg = NA,
  human.hits = NA, human.bg = NA, reps = 100, sub = FALSE,
  geneSizeControl = FALSE)
```

Arguments

`sct_data`	List generated using `read_celltype_data`
`mouse.hits`	Array of MGI gene symbols containing the target gene list. Not required if geneSizeControl=TRUE
`mouse.bg`	Array of MGI gene symbols containing the background gene list. Not required if geneSizeControl=TRUE
`human.hits`	Array of HGNC gene symbols containing the target gene list. Not required if geneSizeControl=FALSE
`human.bg`	Array of HGNC gene symbols containing the background gene list. Not required if geneSizeControl=FALSE
`reps`	Number of random gene lists to generate (default=100 but should be over 10000 for publication quality results)
`sub`	a logical indicating whether to analyse sub-cell type annotations (TRUE) or cell-type annotations (FALSE). Default is FALSE.
`geneSizeControl`	a logical indicating whether you want to control for GC content and transcript length. Recommended if the gene list originates from genetic studies. Default is FALSE. If set to TRUE then human gene lists should be used rather than mouse.

Value

A list containing three data frames:

- `results`: dataframe in which each row gives the statistics (p-value, fold change and number of standard deviations from the mean) associated with the enrichment of the stated cell type in the gene list
- `hit.cells`: vector containing the summed proportion of expression in each cell type for the target list
- `bootstrap_data`: matrix in which each row represents the summed proportion of expression in each cell type for one of the random lists

Examples

```
# Load the single cell data
data(celltype_data)

# Set the parameters for the analysis
reps=100 # <- Use 100 bootstrap lists so it runs quickly, for publishable analysis use >10000
subCellStatus=0 # <- Use subcell level annotations (i.e. Interneuron type 3)
if(subCellStatus==1){subCellStatus=TRUE;cellTag="SubCells"}
if(subCellStatus==0){subCellStatus=FALSE;cellTag="FullCells"}

# Load the gene list and get human orthologs
data("example_genelist")
data("mouse_to_human_homologs")
m2h = unique(mouse_to_human_homologs[,c("HGNC.symbol","MGI.symbol")])
mouse.hits = unique(m2h[m2h$HGNC.symbol %in% example_genelist,"MGI.symbol"])
human.hits = unique(m2h[m2h$HGNC.symbol %in% example_genelist,"HGNC.symbol"])
human.bg = unique(setdiff(m2h$HGNC.symbol,human.hits))
mouse.bg  = unique(setdiff(m2h$MGI.symbol,mouse.hits))

# Bootstrap significance testing, without controlling for transcript length and GC content
full_results = bootstrap.enrichment.test(sct_data=celltype_data,mouse.hits=mouse.hits,
  mouse.bg=mouse.bg,reps=reps,sub=subCellStatus)

# Bootstrap significance testing controlling for transcript length and GC content
full_results = bootstrap.enrichment.test(sct_data=celltype_data,human.hits=human.hits,
  human.bg=human.bg,reps=reps,sub=subCellStatus,geneSizeControl=TRUE)
```

---

celltype_data *Subset of genes from Linnarsson lab's cortex single cell transcriptome dataset*

Description

The first 200 genes from the SCT dataset

Usage

```
celltype_data
```

Format

An object of class `list` of length 3.

Source

The table was downloaded from the website associated with the paper and loaded using read_celltype_data PMID:25700174

ewce.plot *Plot EWCE results*

Description

ewce.plot generates plots of EWCE enrichment results

Usage

```
ewce.plot(total_res, mtc_method = "bonferroni")
```

Arguments

total_res	results dataframe generated using bootstrap.enrichment.test or ewce_expression_data functions. Multiple results tables can be merged into one results table, as long as the 'list' column is set to distinguish them.
mtc_method	method to be used for multiple testing correction. Argument is passed to p.adjust. Valid options are "holm", "hochberg", "hommel", "bonferroni", "BH", "BY",

Value

A ggplot containing the plot

Examples

```
# Load the single cell data
data(celltype_data)

# Set the parameters for the analysis
reps=100 # <- Use 100 bootstrap lists so it runs quickly, for publishable analysis use >10000
subCellStatus=0 # <- Use subcell level annotations (i.e. Interneuron type 3)

# Load the gene list and get human orthologs
data("example_genelist")
data("mouse_to_human_homologs")
m2h = unique(mouse_to_human_homologs[,c("HGNC.symbol","MGI.symbol")])
mouse.hits = unique(m2h[m2h$HGNC.symbol %in% example_genelist,"MGI.symbol"])
mouse.bg  = unique(setdiff(m2h$MGI.symbol,mouse.hits))

# Bootstrap significance testing, without controlling for transcript length and GC content
full_results = bootstrap.enrichment.test(sct_data=celltype_data,mouse.hits=mouse.hits,
 mouse.bg=mouse.bg,reps=reps,sub=subCellStatus)

# Generate the plot
print(ewce.plot(full_results$results,mtc_method="BH"))
```

ewce_expression_data *Bootstrap celltype enrichment test for transcriptome data*

Description

ewce_expression_data takes a differential expression table and determines the probability of cell-type enrichment in the up & down regulated genes

Usage

```
ewce_expression_data(sct_data, tt, sortBy = "t", thresh = 250, reps = 100,
  sub = FALSE, useHGNC = TRUE)
```

Arguments

sct_data	List generated using read_celltype_data
tt	Differential expression table. Can be output of limma::topTable function. Minimum requirement is that one column stores a metric of increased/decreased expression (i.e. log fold change, t-statistic for differential expression etc) and another contains either HGNC or MGI symbols.
sortBy	Column name of metric in tt which should be used to sort up- from down- regulated genes. Default="t"
thresh	The number of up- and down- regulated genes to be included in each analysis. Dafault=250
reps	Number of random gene lists to generate (default=100 but should be over 10000 for publication quality results)
sub	a logical indicating whether to analyse sub-cell type annotations (TRUE) or cell-type annotations (FALSE). Default is FALSE.
useHGNC	a logical indicating whether HGNC or MGI gene symbols are provided. Dafault=TRUE

Value

A list containing five data frames:

- results: dataframe in which each row gives the statistics (p-value, fold change and number of standard deviations from the mean) associated with the enrichment of the stated cell type in the gene list. An additional column *Direction* stores whether it the result is from the up or downregulated set.
- hit.cells.up: vector containing the summed proportion of expression in each cell type for the target list
- hit.cells.down: vector containing the summed proportion of expression in each cell type for the target list#'
- bootstrap_data.up: matrix in which each row represents the summed proportion of expression in each cell type for one of the random lists
- bootstrap_data.down: matrix in which each row represents the summed proportion of expression in each cell type for one of the random lists

Examples

```
# Load the single cell data
data(celltype_data)

# Set the parameters for the analysis
reps=100 # <- Use 100 bootstrap lists so it runs quickly, for publishable analysis use >10000
subCellStatus=0 # <- Use subcell level annotations (i.e. Interneuron type 3)
if(subCellStatus==1){subCellStatus=TRUE;cellTag="SubCells"}
if(subCellStatus==0){subCellStatus=FALSE;cellTag="FullCells"}

# Load the gene list and get human orthologs
data("tt_alzh")

# Bootstrap significance testing, without controlling for transcript length and GC content
tt_results = ewce_expression_data(sct_data=celltype_data,tt=tt_alzh)
```

example_genelist *Example Gene list*

Description

A list of genes genetically associated with Alzheimer's disease.

Usage

```
example_genelist
```

Format

An object of class `character` of length 22.

Source

These were obtained from two sources: http://www.alzgene.org/TopResults.asp and PMID24162737

generate.bootstrap.plots *Generate bootstrap plots*

Description

`generate.bootstrap.plots` takes a genelist and a single cell type transcriptome dataset and generates plots which show how the expression of the genes in the list compares to those in randomly generated gene lists

Usage

```
generate.bootstrap.plots(sct_data, mouse.hits, mouse.bg, reps, sub = FALSE,
  full_results = NA, listFileName = "")
```

Arguments

`sct_data` List generated using `read_celltype_data`

`mouse.hits` Array of MGI gene symbols containing the target gene list.

`mouse.bg` Array of MGI gene symbols containing the background gene list.

`reps` Number of random gene lists to generate (default=100 but should be over 10000 for publication quality results)

`sub` a logical indicating whether to analyse sub-cell type annotations (TRUE) or celltype annotations (FALSE). Default is FALSE.

`full_results` The full output of `bootstrap.enrichment.test` for the same genelist

`listFileName` String used as the root for files saved using this function

Value

Saves a set of pdf files containing graphs. These will be saved with the filename adjusted using the value of listFileName. The files are saved into the 'BootstrapPlot' folder. The files start with one of the following:

- `qqplot_noText`: sorts the gene list according to how enriched it is in the relevant celltype. Plots the value in the target list against the mean value in the bootstrapped lists.
- `qqplot_wtGSym`: as above but labels the gene symbols for the highest expressed genes.
- `bootDists`: rather than just showing the mean of the bootstrapped lists, a boxplot shows the distribution of values
- `bootDists_LOG`: shows the bootstrapped distributions with the y-axis shown on a log scale

Examples

```
# Load the single cell data
data(celltype_data)

# Set the parameters for the analysis
reps=100 # <- Use 100 bootstrap lists so it runs quickly, for publishable analysis use >10000
subCellStatus=0 # <- Use subcell level annotations (i.e. Interneuron type 3)
if(subCellStatus==1){subCellStatus=TRUE;cellTag="SubCells"}
if(subCellStatus==0){subCellStatus=FALSE;cellTag="FullCells"}

# Load the gene list and get human orthologs
data("example_genelist")
data("mouse_to_human_homologs")
m2h = unique(mouse_to_human_homologs[,c("HGNC.symbol","MGI.symbol")])
mouse.hits = unique(m2h[m2h$HGNC.symbol %in% example_genelist,"MGI.symbol"])
mouse.bg  = unique(setdiff(m2h$MGI.symbol,mouse.hits))

# Bootstrap significance testing, without controlling for transcript length and GC content
full_results = bootstrap.enrichment.test(sct_data=celltype_data,mouse.hits=mouse.hits,
     mouse.bg=mouse.bg,reps=reps,sub=subCellStatus)

generate.bootstrap.plots(sct_data=celltype_data,mouse.hits=mouse.hits, mouse.bg=mouse.bg,
     reps=reps,sub=FALSE,full_results=full_results,listFileName="Example")
```

merged_ewce *Multiple EWCE results from multiple studies*

Description

merged_ewce combines enrichment results from multiple studies targetting the same scientific problem

Usage

```
merged_ewce(results, reps = 100)
```

Arguments

results	a list of EWCE results generated using add.res.to.merging.list
reps	Number of random gene lists to generate (default=100 but should be over 10000 for publication quality results)

Value

dataframe in which each row gives the statistics (p-value, fold change and number of standard deviations from the mean) associated with the enrichment of the stated cell type in the gene list

Examples

```
# Load the single cell data
data(celltype_data)

# Set the parameters for the analysis
reps=100 # <- Use 100 bootstrap lists so it runs quickly, for publishable analysis use >10000
subCellStatus=0 # <- Use subcell level annotations (i.e. Interneuron type 3)

# Load the gene list and get human orthologs
data("example_genelist")
data("mouse_to_human_homologs")
m2h = unique(mouse_to_human_homologs[,c("HGNC.symbol","MGI.symbol")])
mouse.hits = unique(m2h[m2h$HGNC.symbol %in% example_genelist,"MGI.symbol"])
mouse.bg  = unique(setdiff(m2h$MGI.symbol,mouse.hits))

# Bootstrap significance testing, without controlling for transcript length and GC content
full_results = bootstrap.enrichment.test(sct_data=celltype_data,mouse.hits=mouse.hits,
     mouse.bg=mouse.bg,reps=reps,sub=subCellStatus)

# Generate the plot
print(ewce.plot(full_results$results,mtc_method="BH"))
```

mouse_to_human_homologs *Table of Human–>Mouse orthologs for all human genes*

Description

A dataset containing the MGI and HGNC symbols, Human and Mouse Entrez and Ensembl gene IDs for all human orthologs for mouse genes. Whenin the mouse genes are defined based on a list of all MGI markers from the MGI website (downloaded as MRK_List2.rpt file from http://goo.gl/mjf2GQ)

Usage

```
mouse_to_human_homologs
```

Format

An object of class `data.frame` with 21508 rows and 6 columns.

Examples

```
## Not run:
## The code to prepare the .Rda file file from the marker file is:
markers = read.csv("MRK_List2.rpt",sep="\t")
genes = markers[markers$Feature.Type=="protein coding gene",]
listMarts(host="www.ensembl.org")
human <- useMart(host="www.ensembl.org", "ENSEMBL_MART_ENSEMBL", dataset="hsapiens_gene_ensembl")
mouse <- useMart(host="www.ensembl.org", "ENSEMBL_MART_ENSEMBL", dataset="mmusculus_gene_ensembl")
mouse_to_human_homologs = getLDS(attributes = c("mgi_symbol","entrezgene","ensembl_gene_id"),
                                 filters = "mgi_symbol", values = genes$Marker.Symbol,
                                 mart = mouse,
                          attributesL = c("hgnc_symbol","ensembl_gene_id","entrezgene"), martL = human)
save(mouse_to_human_homologs,file="mouse_to_human_homologs.Rda")

## End(Not run)
```

read_celltype_data *Read single cell transcriptome data*

Description

`read_celltype_data` loads single cell transcriptome data and determines for each gene, the proportion of expression found in each celltype

Usage

```
read_celltype_data(path)
```

Arguments

`path`	Path to the file containing the single cell transcriptome data

Value

A list containing three data frames:

- `all_scts`: stores the average value for each subcell type found across cells
- `cell_dists`: proportion of expression for each gene in each cell type
- `subcell_dists`: proportion of expression for each gene in each subcell type

Examples

```
celltype_data = read_celltype_data("expression_mRNA_17-Aug-2014.txt")
```

`tt_alzh` *Example table of differential expression (Alzheimer's BA46)*

Description

A list of genes found to be differentially expressed in the BA46 in Alzheimer's disease.

Usage

```
tt_alzh
```

Format

An object of class `data.frame` with 21745 rows and 7 columns.

Source

The table was determined based on data associated with the paper with PMID:17845826

`tt_alzh_BA36` *Example table of differential expression (Alzheimer's BA36)*

Description

A list of genes found to be differentially expressed in the BA36 in Alzheimer's disease.

Usage

```
tt_alzh_BA36
```

Format

An object of class `data.frame` with 21745 rows and 7 columns.

Source

The table was determined based on data associated with the paper with PMID:17845826 tt_alzh_BA44 *Example table of differential expression (Alzheimer's BA44)*

Description

A list of genes found to be differentially expressed in the BA44 in Alzheimer's disease.

Usage

```
tt_alzh_BA44
```

Format

An object of class `data.frame` with 21745 rows and 7 columns.

Source

The table was determined based on data associated with the paper with PMID:17845826

Index