# Package

June 12, 2019

**Type** Package

**Title** TinderMIX: An R package to cluster gene expression by contour plots

**Version** 0.1.0

**Author** Angela Serra

**Maintainer** Angela Serra <angela.serra@tuni.fi>

**Description** The TinderMIX package allows to analyse toxicogenomics data with multiple dose levels and time-points. It allows to identify the expression patterns with respect to both variables and to cluster molecular features accordingly. It also identify enriched pathways/go terms that are associated to each cluster.

**Depends** R (>= 3.4),
stats,
utils,
AnnotationDbi,
gProfileR,
gtools,
reshape,
plotly,
clv,
gplots,
org.Hs.eg.db,
org.Mm.eg.db,
org.Rn.eg.db,
xlsx

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**Suggests** knitr,
rmarkdown,
testthat

**VignetteBuilder** knitr

## R topics documented:

---

`build_items_list` *This function computes the venn diagram of the genes associated to time, dose or their interaction*

---

## Description

This function computes the venn diagram of the genes associated to time, dose or their interaction

## Usage

```
build_items_list(PvalMat, p.val.th = 0.01)
```

## Arguments

| | |
|---|---|
| `PvalMat` | matrix with pvalue associated to the dose, timepoint and the dose*timepoint effect that is the output of the compute_anova_dose_time function |
| `p.val.th` | is the threshold at which p.values are considered significant. Default = 0.01 |

## Value

a list containing the genes in each position of the venn diagram

## Examples

```
data("WY14643")
exp_data = WY14643$exp_data
pheno_data = WY14643$pheno_data
PvalMat = compute_anova_dose_time(exp_data, pheno_data,dose_index = 2,time_point_index = 3)
ItemsList = build_items_list(PvalMat)
```

`clustering_summary` *This function evaluate the summary index for a clustering results by taking into account the within and between variability*

## Description

This function evaluate the summary index for a clustering results by taking into account the within and between variability

## Usage

```
clustering_summary(DB, cluster)
```

## Arguments

| | |
|---|---|
| `DB` | your matrix dataset |
| `cluster` | is a numeric vector of clustering results |

## Value

a vector of evaluation indexes

`compute_anova_dose_time` *This function computes a two way anova between dose and time for the expression value of every genes*

## Description

This function computes a two way anova between dose and time for the expression value of every genes

## Usage

```
compute_anova_dose_time(exp_data, pheno_data, dose_index, time_point_index)
```

## Arguments

| | |
|---|---|
| `exp_data` | is the expression matrix with genes on the rows and samples on the columns |
| `pheno_data` | is a dataframe with phenodata informations. Samples are on the rows. The columns should include the dose and time point information. |
| `dose_index` | numeric value specifing the column of the phenodata table containing the doses |
| `time_point_index` | numeric value specifing the column of the phenodata table containing the time points |

## Value

a matrix with pvalue associated to the dose, timepoint and the dose*timepoint effect

## Examples

```
data("WY14643")
exp_data = WY14643$exp_data
pheno_data = WY14643$pheno_data
PvalMat = compute_anova_dose_time(exp_data, pheno_data,dose_index = 2,time_point_index = 3)
```

---

`compute_enrichment` *This function perform enchment of the genes in each cluster*

---

## Description

This function perform enchment of the genes in each cluster

## Usage

```
compute_enrichment(optimal_clustering, corrType = "fdr",
  type_enrich = "KEGG", org_enrich = "rnorvegicus", pth = 0.05,
  sig = FALSE, mis = 0, only_annotated = FALSE)
```

## Arguments

| | |
|---|---|
| `optimal_clustering` | vector of final clustering |
| `corrType` | string specifing the algorithm used for determining the significance threshold, one of gSCS, fdr, bonferroni. Default: fdr |
| `type_enrich` | string specifying the enrichment type. Default = KEGG |
| `org_enrich` | string specifying the organism. Default = rnorvegicus |
| `pth` | numeric value specifyint the pvalue threshold. Default = 0.05 |
| `sig` | whether all or only statistically significant results should be returned |
| `mis` | minimum size of functional category, smaller categories are excluded |
| `only_annotated` | statistical domain size, one of "annotated", "known" |

## Value

a list with the enriched pathways for each cluster of genes

## Examples

```
 data("WY14643")
 exp_data = WY14643$exp_data
 pheno_data = WY14643$pheno_data
PvalMat = compute_anova_dose_time(exp_data, pheno_data,dose_index = 2,time_point_index = 3)
 ItemsList = build_items_list(PvalMat)
responsive_genes = unique(c(unlist(ItemsList$Dose),unlist(ItemsList$Time),unlist(ItemsList$`Dose:Time:Dose
contour_res = create_contour(exp_data, pheno_data, responsive_genes,dose_index = 2,time_point_index =3 ,gric
hls_res = hls_genes_clustering(contour_res$GenesMap,  nClust = c(5,10,15,20,25), method="pearson", hls.metho
clpr = create_prototypes(clust_res = hls_res,summaryMat = hls_res$summaryMat,contour_res )
enrRes = compute_enrichment(clpr$optcl,corrType = "fdr",type_enrich="KEGG", org_enrich = "rnorvegicus",pth =
```

`convert_genes` *This function convert genes identifiers*

## Description

This function convert genes identifiers

## Usage

```
convert_genes(organism = "hsapiens", GList, annType = "SYMBOL")
```

## Arguments

| | |
|---|---|
| `organism` | a string specifying the organism under analysis |
| `GList` | a list of genes identifier |
| `annType` | string specifying the wanted gene identifier |

## Value

a list with the converted genes identifiers

`create_contour` *This function fits a 3D regression model for every gene in the dataset and creates an N x N contour plot*

## Description

This function fits a 3D regression model for every gene in the dataset and creates an N x N contour plot

## Usage

```
create_contour(exp_data, pheno_data, responsive_genes, dose_index,
  time_point_index, gridSize = 50)
```

## Arguments

| | |
|---|---|
| `exp_data` | is the expression matrix with genes on the rows and samples on the columns |
| `pheno_data` | is a dataframe with phenodata informations. Samples are on the rows. The columns should include the dose and time point information. |
| `responsive_genes` | responsive_genes character vector with the genes statistically significant for the two-way anova |
| `dose_index` | numeric value specifing the column of the phenodata table containing the doses |
| `time_point_index` | numeric value specifing the column of the phenodata table containing the time points |
| `gridSize` | numeric value specifing size of the z-grid |

## Value

a list with list with estimated contour objects, 3D fitted objects, fitting statistics and feature values for time and dose

| | |
|---|---|
| GenesMap | a matrix with the z-maps computed for each gene |
| RPGenes | a list with the 3D fitted objects |
| Statis | a matrix witht the fitting statistics: PValue,Adj.R.Square,RMSE |
| DFList | a list with the data used for the fitting |

## Examples

```
 data("WY14643")
 exp_data = WY14643$exp_data
 pheno_data = WY14643$pheno_data
PvalMat = compute_anova_dose_time(exp_data, pheno_data,dose_index = 2,time_point_index = 3)
 ItemsList = build_items_list(PvalMat)
responsive_genes = unique(c(unlist(ItemsList$Dose), unlist(ItemsList$Time), unlist(ItemsList$`Dose:Time:Dos
contour_res = create_contour(exp_data, pheno_data, responsive_genes,dose_index = 2,time_point_index =3 ,gric
```

---

# create_prototypes *this function create the cluster prototypes as the mean values of the z maps of all the genes in the cluster*

---

## Description

this function create the cluster prototypes as the mean values of the z maps of all the genes in the cluster

## Usage

```
create_prototypes(clust_res, summaryMat, contour_res)
```

## Arguments

| | |
|---|---|
| clust_res | the clustering results object given in input by the function hls_genes_clustering |
| summaryMat | the matrix with the summary statistics computed for the different k values |
| contour_res | a list with the contours object returned in output by the create_contour function |

## Value

a list with the contour objects for each clustering prototype and the vector of the optimal clustering

| | |
|---|---|
| meanXYZ | a list the contour object for each prototype |
| optcl | a vector with the optimal clustering |

## Examples

```
 data("WY14643")
 exp_data = WY14643$exp_data
 pheno_data = WY14643$pheno_data
PvalMat = compute_anova_dose_time(exp_data, pheno_data,dose_index = 2,time_point_index = 3)
 ItemsList = build_items_list(PvalMat)
responsive_genes = unique(c(unlist(ItemsList$Dose),unlist(ItemsList$Time),unlist(ItemsList$`Dose:Time:Dose
contour_res = create_contour(exp_data, pheno_data, responsive_genes,dose_index = 2,time_point_index =3 ,gric
hls_res = hls_genes_clustering(contour_res$GenesMap,  nClust = c(5,10,15,20,25), method="pearson", hls.metho
clpr = create_prototypes(clust_res = hls_res,summaryMat = hls_res$summaryMat,contour_res )
```

---

`enrich` *This function perform enchment of the genes in each cluster*

---

## Description

This function perform enchment of the genes in each cluster

## Usage

```
enrich(x, type, org, pval, adjust_method, sig = FALSE, mis = 0,
  only_annotated = TRUE)
```

## Arguments

| | |
|---|---|
| `x` | a dataframe with the gene names on the first column |
| `type` | string specifying the enrichment type. Default = KEGG |
| `org` | string specifying the organism. Default = rnorvegicus |
| `pval` | numeric value specifyint the pvalue threshold. Default = 0.05 |
| `adjust_method` | string specifing the algorithm used for determining the significance threshold, one of gSCS, fdr, bonferroni. Default: fdr |
| `sig` | whether all or only statistically significant results should be returned |
| `mis` | minimum size of functional category, smaller categories are excluded |
| `only_annotated` | statistical domain size, one of "annotated", "known" |

## Value

a list with the enriched pathways for each cluster of genes

`findCenter` *This function select cluster prototypes. It select the prototype as the more correlated with the other elements in the cluster*

## Description

This function select cluster prototypes. It select the prototype as the more correlated with the other elements in the cluster

## Usage

```
findCenter(DB, clust_vector)
```

## Arguments

| | |
|---|---|
| `DB` | your matrix dataset |
| `clust_vector` | a numeric vector of clustering results |

## Value

a matrix of prototypes

`fisher_test` *This function construct confusion matrix between patient classes and the obtained clustering*

## Description

This function construct confusion matrix between patient classes and the obtained clustering

## Usage

```
fisher_test(classes, clustering, matrixRownames, nCluster)
```

## Arguments

| | |
|---|---|
| `classes` | is a vector of patient labels |
| `clustering` | is a vector of clustering results |
| `matrixRownames` | is a vector of names to assign as rownames of the confusion matrix |
| `nCluster` | is the number of obtained clusters |

## Value

the confusion matrix

`hls_genes_clustering` *It clusters the contour plots by also estimating the ideal number of clusters*

## Description

It clusters the contour plots by also estimating the ideal number of clusters

## Usage

```
hls_genes_clustering(GenesMap, nClust = c(5, 10, 25, 50, 75, 100, 125,
  150, 175, 200, 250, 300), method = "pearson", hls.method = "ward")
```

## Arguments

| | |
|---|---|
| `GenesMap` | a matrix with the z-maps computed with the function create_contour |
| `nClust` | vector of putative numbers used to determine number of clusters. Default = c(5,10,25,50,75,100,125,150,175,200,250,300) |
| `method` | string specifying the correlation method. Default = "pearson" |
| `hls.method` | string specifying method for the hierarchical clustering. Default = "ward" |

## Value

a list with the clustering results and statistics for the optimal number of clusters

| | |
|---|---|
| `hls_res` | a list containing the clustering results, the clustering vector and the centers for each k value given in input |
| `summaryMat` | a matrix with summary statistic for each k value given in input |
| `clusterList` | a list with the clustering vectors |

## Examples

```
 data("WY14643")
 exp_data = WY14643$exp_data
 pheno_data = WY14643$pheno_data
PvalMat = compute_anova_dose_time(exp_data, pheno_data,dose_index = 2,time_point_index = 3)
 ItemsList = build_items_list(PvalMat)
responsive_genes = unique(c(unlist(ItemsList$Dose),unlist(ItemsList$Time),unlist(ItemsList$`Dose:Time:Dose
contour_res = create_contour(exp_data, pheno_data, responsive_genes,dose_index = 2,time_point_index =3 ,gric
hls_res = hls_genes_clustering(contour_res$GenesMap,  nClust = c(5,10,15,20,25), method="pearson", hls.metho
```

## plot3d — *This function plots the fitted 3d surface for the expression value of a gene*

### Description

This function plots the fitted 3d surface for the expression value of a gene

### Usage

```
plot3d(toPlot = list(x, y, z), DF)
```

### Arguments

| | |
|---|---|
| toPlot | is a list containing the predicted value for the x, y and z axis |
| DF | is the data frame containing the information for the samples used in the fitting process |

### Value

a plotly object

### Examples

```
 data("WY14643")
 exp_data = WY14643$exp_data
 pheno_data = WY14643$pheno_data
PvalMat = compute_anova_dose_time(exp_data, pheno_data,dose_index = 2,time_point_index = 3)
 ItemsList = build_items_list(PvalMat)
responsive_genes = unique(c(unlist(ItemsList$Dose),unlist(ItemsList$Time),unlist(ItemsList$`Dose:Time:Dose
contour_res = create_contour(exp_data, pheno_data, responsive_genes,dose_index = 2,time_point_index =3 ,gric
 plot3d(toPlot = contour_res$RPGenes[["Pdk4"]],DF = contour_res$DFList[["Pdk4"]])
```

## plot_clusters_prototypes — *this function plots the clusters prototype*

### Description

this function plots the clusters prototype

### Usage

```
plot_clusters_prototypes(meanXYZ, nR = 2)
```

### Arguments

| | |
|---|---|
| meanXYZ | a list the contour object for each prototype computed with the function create_prototypes |
| nR | the number of rows to use in the plot |

## Examples

```
data("WY14643")
exp_data = WY14643$exp_data
pheno_data = WY14643$pheno_data
PvalMat = compute_anova_dose_time(exp_data, pheno_data,dose_index = 2,time_point_index = 3)
ItemsList = build_items_list(PvalMat)
responsive_genes = unique(c(unlist(ItemsList$Dose),unlist(ItemsList$Time),
                            unlist(ItemsList$`Dose:Time:DoseTime`),
                            unlist(ItemsList$`Dose:Time`)))
contour_res = create_contour(exp_data, pheno_data, responsive_genes,\cr
                            dose_index = 2,time_point_index =3 ,gridSize = 50)
hls_res = hls_genes_clustering(contour_res$GenesMap,  nClust = c(5,10,15,20,25),\cr
                               method="pearson", hls.method = "ward")

clpr = create_prototypes(clust_res = hls_res,summaryMat = hls_res$summaryMat,contour_res )
plot_clusters_prototypes(clpr$meanXYZ, nR = 2)
```

---

# write_xlsx_for_funmappone

*This function create an excel file with the same format of the input need by the FunMappOne tool*

---

## Description

This function create an excel file with the same format of the input need by the FunMappOne tool

## Usage

```
write_xlsx_for_funmappone(optimal_clustering,
  filePath = "../contour_clustering/gene_clustering.xlsx")
```

## Arguments

| | |
|---|---|
| optimal_clustering | is a numeric vector with the clustering result for every gene |
| filePath | is a string specifying the path of the xlsx file |

## Examples

```
data("WY14643")
exp_data = WY14643$exp_data
pheno_data = WY14643$pheno_data
PvalMat = compute_anova_dose_time(exp_data, pheno_data,dose_index = 2,time_point_index = 3)
ItemsList = build_items_list(PvalMat)
responsive_genes = unique(c(unlist(ItemsList$Dose),unlist(ItemsList$Time),unlist(ItemsList$`Dose:Time:Dose
contour_res = create_contour(exp_data, pheno_data, responsive_genes,dose_index = 2,time_point_index =3 ,gric
hls_res = hls_genes_clustering(contour_res$GenesMap,  nClust = c(5,10,15,20,25), method="pearson", hls.metho
enrRes = compute_enrichment(clpr$optcl,corrType = "fdr",type_enrich="KEGG", org_enrich = "rnorvegicus",pth =
write_xlsx_for_funmappone(clpr$optcl,filePath = "../contour_clustering/gene_clustering.xlsx")
```

# Index