Package ‘ACSNMineR’

September 1, 2016

**Type** Package

**Title** Gene Enrichment Analysis from ACSN Maps or GMT Files

**Version** 0.16.8.25

**Description** Compute and represent gene set enrichment or depletion from your data based on pre-saved maps from the Atlas of Cancer Signalling Networks (ACSN) or user imported maps. User imported maps must be complying with the GMT format as defined by the Broad Institute, that is to say that the file should be tab-separated, that the first column should contain the module name, the second column can contain comments that will be overwritten with the number of genes in the module, and subsequent columns must contain the list of genes (HUGO symbols; tab-separated) inside the module. The gene set enrichment can be run with hypergeometric test or Fisher exact test, and can use multiple corrections. Visualization of data can be done either by barplots or heatmaps.

**Depends** R (>= 3.1.0), ggplot2, gridExtra, scales,

**Suggests** rmarkdown, knitr

**License** GPL-2

**LazyData** true

**VignetteBuilder** knitr

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Author** Paul Deveau [aut, cre], Eric Bonnet [aut]

**Maintainer** Paul Deveau <paul.deveau@curie.fr>

**Repository** CRAN

**Date/Publication** 2016-09-01 17:30:55

### R topics documented:

- ACSN_maps ............................. 2
- cnum .................................... 3
- Create_master_map ...................... 3
Description

A dataset containing the six maps of ACSN: apoptosis, cell cycle, DNA repair, EMT motility, survival, and the master map.

Usage

ACSN_maps

Format

A list of dataframes

- **Apoptosis**  Map of apoptosis pathways
- **CellCycle**  Map of the cell cycle pathways
- **DNA_repair**  Map of DNA repair
- **EMT_motility**  Map of the Epithelial Mesenchymal Transition
- **Master**  Map grouping all modules from other maps, without a master module for each map
- **Survival**  Map of cellular survival pathways

Source

https://acsn.curie.fr/downloads.html
### cnum

**Convert to numeric**

<table>
<thead>
<tr>
<th>cnum</th>
<th>Convert to numeric</th>
</tr>
</thead>
</table>

#### Description

Convert to numeric

#### Usage

```r
cnum(x)
```

#### Arguments

- `x`: A vector of numbers which is not in numeric format

### create_master_map

**From a list of maps, create or replace a master**

<table>
<thead>
<tr>
<th>create_master_map</th>
<th>From a list of maps, create or replace a master</th>
</tr>
</thead>
</table>

#### Description

From a list of maps, create or replace a master

#### Usage

```r
create_master_map(maps)
```

#### Arguments

- `maps`: A list of molecular maps created by `format_from_gmt`

#### Value

Returns a list with previous maps and the master map, i.e. a concatenation of previous maps.

#### Examples

```r
create_master_map(list(Cycle = ACSNMineR::ACSN_maps$CellCycle, Apoptosis = ACSNMineR::ACSN_maps$Apoptosis))
```
**enrichment**

*Gene set enrichment analysis*

**Description**

Compute and represent gene set enrichment from your data based on pre-saved maps from ACSN or user imported maps. The gene set enrichment can be run with hypergeometric test or Fisher exact test, and can use multiple corrections. Visualization of data can be done either by barplots or heatmaps.

**Usage**

```r
enrichment(Genes = NULL, maps = c("Apoptosis", "CellCycle", "DNA_repair", "EMT_motility", "Survival"), correction_multitest = "BH", statistical_test = "fisher", min_module_size = 5, universe = "map_defined", Remove_from_universe = NULL, threshold = 0.05, alternative = "greater")
```

**Arguments**

- **Genes** Character vector of genes that should be tested for enrichment
- **maps** list of maps generated by `format_from_gmt`. Names of element of list will be used to track modules. Default: tests on the master map.
- **correction_multitest** either FALSE, "bonferroni", "holm", "hochberg", "hommel", "BH", "fdr" (identical to BH), or "BY"
- **statistical_test** one of "fisher", "hypergeom"
- **min_module_size** will remove from the analysis all modules which are (strictly) smaller than threshold
- **universe** Universe on which the statistical analysis should be performed. Can be either "HUGO", "ACSN", "map_defined", or a character vector of genes.
- **Remove_from_universe** Default is NULL. A list of genes that should not be considered for enrichment (will be removed from input, maps, and universe). The size of universe and map will be updated after removal.
- **threshold** maximal p-value (corrected if correction is enabled) that will be displayed
- **alternative** One of "greater", "less", "both" or "two.sided" Greater will check for enrichment, less will check for depletion, and both will look for both and will keep track of the side, while two-sided (only for fisher test) checks if there is a difference.
enrichment_test

Value

Output is a dataframe with the following columns:

module  The name of the map or the module preceded by the map
module_size  The number of genes in the module after taking into account universe reduction
nb_genes_in_module  The number of genes from input list in the module
genesis_in_module  Names of the genes from input list in the module, space separated
universe_size  size of the input universe
nb_genes_in_universe  number of genes from the input list that are found in the universe
test  the kind of test that was looked for. "greater" when enrichment is tested, "less" when depletion is tested, or "two.sided"

Examples

enrichment(genes_test,min_module_size = 10,
threshold = 0.05,
maps = list(cellcycle = ACSNMineR::ACSN_maps$CellCycle),
universe = "ACSN")

<table>
<thead>
<tr>
<th>enrichment_test</th>
<th>Result from enrichment test of &quot;genes_test&quot; on the ACSN maps</th>
</tr>
</thead>
</table>

Description

Parameters: bonferroni correction, min module size = 5

Usage

enrichment_test

Format

data.frame

module  Name of module
genes_in_module  Genes from genes_test in module
p.value  Uncorrected p-value
p.value.corrected  p-value corrected for multiple testing by Bonferroni correction
format_from_gmt  
*Import data from gmt files Convert gmt file to dataframe that can be used for analysis*

### Description

Import data from gmt files Convert gmt file to dataframe that can be used for analysis

### Usage

`format_from_gmt(path = "")`

### Arguments

- **path**
  
  Path to the gmt file to be imported

### Value

Returns a dataframe with the module - first column -, module length - second column - and gene names

### Examples

```r
file <- system.file("extdata", "cellcycle_short.gmt", package = "ACSNMineR")
format_from_gmt(file)
```

---

*genes_test  
*Set of genes to test map*

### Description

Genes of high importance in oncogenesis

### Usage

`genes_test`

### Format

A character vector
multisample_enrichment

Automated gene set analysis for multiple sets

Description
Automated gene set analysis for multiple sets

Usage

multisample_enrichment(Genes_by_sample = NULL, maps = c("Apoptosis", "CellCycle", "DNA_repair", "EMT_motility", "Survival"),
correction_multitest = "BH", statistical_test = "fisher",
min_module_size = 5, universe = "map_defined",
Remove_from_universe = NULL, threshold = 0.05, cohort_threshold = TRUE,
alternative = "greater")

Arguments

Genes_by_sample
List of character vectors. Each list element name should be a sample name, and
each character vector the set of genes to test for the sample.

maps
list of maps generated by format_from_gmt. Default: tests on all acsn maps

correction_multitest
either FALSE, "bonferroni", "holm", "hochberg", "hommel", "BH", "fdr" (iden-
tical to BH), or "BY"

statistical_test
one of "fisher", "hypergeom"

min_module_size
will remove from the analysis all modules which are (strictly) smaller than

universe
Universe on which the statistical analysis should be performed. Can be either
"HUGO","ACSN","map_defined", or a character vector of genes.

Remove_from_universe
Default is NULL. A list of genes that should not be considered for enrichment
will be removed from input, maps, and universe). The size of universe and map
will be updated after removal.

threshold
maximal p-value (corrected if correction is enabled) that will be displayed

cohort_threshold
if TRUE modules will be kept in all samples if at least one sample has p-value
lower than threshold, otherwise the threshold is applied for each sample inde-
pendently.

alternative
One of "greater", "less", "both", or "two.sided" (only for fisher test). Greater
will check for enrichment, less will check for depletion, and both will look for
both.
Value

Output is a list of dataframes with names the names given in 'Genes_by_sample' with the following columns:

- **module**: The name of the map or the module preceded by the map
- **module_size**: The number of genes in the module after taking into account universe reduction
- **nb_genes_in_module**: The number of genes from input list in the module
- **genes_in_module**: Names of the genes from input list in the module, space separated
- **universe_size**: Size of the input universe
- **nb_genes_in_universe**: Number of genes from the input list that are found in the universe
- **test**: The kind of test that was looked for. "greater" when enrichment is tested, "less" when depletion is tested, or "two.sided"

Examples

```r
multisample_enrichment(Genes_by_sample = list(set1 = genes_test, set2 = c(genes_test, "PTPRD")),
maps = list(cellcycle = ACSNMineR::ACSN_maps\$CellCycle),
min_module_size = 10,
universe = "ACSN", cohort_threshold = FALSE)
```

---

**p.val.calc**

*Calculate p-value*

Description

Calculate p-value

Usage

```r
p.val.calc(x, y, z, a, stat_test, alt)
```

Arguments

- **x**: First value
- **y**: Second value
- **z**: Third value
- **a**: Fourth value
- **stat_test**: Statistical test to be used
- **alt**: Alternative: one of two-sided, greater, less or both
represent_enrichment

---

represent_enrichment  Graphic representation of enrichment

Description

Graphic representation of enrichment

Usage

```r
represent_enrichment(enrichment, plot = "heatmap", scale = "log",
  low = "steelblue", high = "white", nrow = 1, sample_name = "Sample",
  na.value = "grey")
```

Arguments

- **enrichment**: Data frame or list of dataframes with p-values or corrected p-values (whenever available) and module names for representation. The name of the dataframe will be used as sample name.
- **plot**: Any of "heatmap" or "bar"
- **scale**: Any of "log", "identity" or "reverselog" (i.e. -log10(p-value))
- **low**: Color to be used in heatmap mode corresponding to lowest value
- **high**: Color to be used in heatmap mode corresponding to highest value
- **nrow**: Number of rows of the grid for display in bar mode.
- **sample_name**: used only if enrichment is a dataframe
- **na.value**: color for the missing values in the heatmap

Value

Function returns a ggplot2 object if input is a dataframe or a gridExtra object if the output is a list.

Examples

```r
represent_enrichment(enrichment = enrichment_test, scale = "reverselog",
  sample_name = "test", plot = "bar")

represent_enrichment(enrichment = list(SampleA = enrichment_test,
  SampleB = enrichment_test[1:3,]),
  plot = "heatmap", scale = "log")
```
reverse_log_trans

Scale for barplots and heatmaps

Description
Outputs the "-log" of a scale

Usage
reverse_log_trans(base = 10)

Arguments
base : base for the log, default is 10
Index

*Topic datasets
   ACSN_maps, 2
   enrichment_test, 5
   genes_test, 6

ACSN_maps, 2

cnum, 3
Create_master_map, 3

enrichment, 4
enrichment_test, 5

format_from_gmt, 6

genes_test, 6

multisample_enrichment, 7

p.val.calc, 8

represent_enrichment, 9
reverselog_trans, 10