

Package: CoFRA (via r-universe)

January 10, 2025

Title Complete Functional Regulation Analysis

Version 0.1002

Description Calculates complete functional regulation analysis and visualize the results in a single heatmap. The provided example data is for biological data but the methodology can be used for large data sets to compare quantitative entities that can be grouped. For example, a store might divide entities into cloth, food, car products etc and want to see how sales changes in the groups after some event. The theoretical background for the calculations are provided in New insights into functional regulation in MS-based drug profiling, Ana Sofia Carvalho, Henrik Molina & Rune Matthiesen, Scientific Reports <doi:10.1038/srep18826>.

Depends R (>= 3.2.3)

Imports gplots, grid, methods, parallel, stats, stringr

License GPL-2 | file LICENSE

LazyData true

RoxygenNote 5.0.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Rune Matthiesen [aut, cre]

Maintainer Rune Matthiesen <runem2009@gmail.com>

Date/Publication 2017-04-06 09:26:46 UTC

Additional_repositories <https://cranhaven.r-universe.dev>

Config/pak/sysreqs libicu-dev

Repository <https://cranhaven.r-universe.dev>

RemoteUrl <https://github.com/cranhaven/cranhaven.r-universe.dev>

RemoteRef package/CoFRA

RemoteSha 6a328d69c0140f7f329c648a1a80822b921d91a4

RemoteSubdir CoFRA

Contents

completeFunctionalRegulationAnalysis	2
filterData	3
getFunctionalCategories	3
HeatMapEnrichment	4
iBAQ	4
plot.CompleteEnrichment	6
summary.CompleteEnrichment	6
Index	7

completeFunctionalRegulationAnalysis

This function performs complete functional regulation analysis

Description

This function performs complete functional regulation analysis

Usage

```
completeFunctionalRegulationAnalysis(dfPro, func, Fac, dfComp,
  NbackGround = length(unique(unlist(func))), DataExtract = "FDR",
  minCounts = 10, Test = "t.test", no_cores = -1)
```

Arguments

dfPro	data frame with quantitative values
func	data frame defining which gene ontology to use "BP","CC","MF"
Fac	factor describing the sample groups
dfComp	df containing the comparisons to perform
NbackGround	integer number of total proteins
DataExtract	string which P value correction to use
minCounts	integer minimum number of matching genes for functional category
Test	"t.test" or "wilcox.test"
no_cores	=-1 (no parallel execution) =0 (number of available cores -1) >0 (use number of cores)

Examples

```

library(CoFRA)
data(iBAQ)
Fac=factor(c("MCCTT", "MCCTT", "MCCTT", "MCCT", "MCCT", "MCCT", "MC", "MC", "MC", "MCT", "MCT", "MCT",
"MTT", "MTT", "MTT", "MT", "MT", "MT", "sN", "sN", "sN", "sNT", "sNT", "sNT", "sNT", "iN", "iN", "iN", "iNT", "iNT", "iNT"))
dfComp=data.frame(Con=c("MCCTT", "MT", "MC", "iN", "sN", "AllC,MCCT,MT,MC,iN,sN"),Tre=c("MCCTT", "MTT",
"MCT", "iNT", "sNT", "AllT,MCCTT,MTT,MCT,iNT,sNT"))
Func=CoFRA::getFunctionalCategories("CC")
head(str(Func))
CC1=CoFRA::completeFunctionalRegulationAnalysis(iBAQ,Func[100:200],Fac,dfComp,NbackGround=142140)

```

filterData

This function filter a data frame on column named "pro"

Description

This function filter a data frame on column named "pro"

Usage

```
filterData(dfPro, filter)
```

Arguments

dfPro	data frame with iBAQ values
filter	character vector with items to remove

Examples

```

library(CoFRA)
data(iBAQ)
iBAQ2=filterData(iBAQ,">CON") # filter headers starting with >CON

```

getFunctionalCategories

get named vector with functional categories

Description

get named vector with functional categories

Usage

```
getFunctionalCategories(func = "CC")
```

Arguments

func string ("BP","CC","MF")

Examples

```
library(CoFRA)
Acc=getFunctionalCategories(func="CC")
```

HeatMapEnrichment *This function plot a heatmap to summarize the results from complete functional enrichment analysis*

Description

This function plot a heatmap to summarize the results from complete functional enrichment analysis

Usage

```
HeatMapEnrichment(Eres, title = "")
```

Arguments

Eres object from complete functional enrichment analysis
title string

iBAQ *data frame containing iBAQ values*

Description

A dataset containing iBAQ values for identified proteins

Usage

```
iBAQ
```

Format

A data frame with 18889 rows and 33 variables:

MCCTT1 iBAQ values

MCCTT2 iBAQ values

MCCTT3 iBAQ values

MCCT1 iBAQ values

MCCT2 iBAQ values
MCCT3 iBAQ values
MC1 iBAQ values
MC2 iBAQ values
MC3 iBAQ values
MCT1 iBAQ values
MCT2 iBAQ values
MCT3 iBAQ values
MTT1 iBAQ values
MTT2 iBAQ values
MTT3 iBAQ values
MT1 iBAQ values
MT2 iBAQ values
MT3 iBAQ values
sN1 iBAQ values
sN2 iBAQ values
sN3 iBAQ values
sNT1 iBAQ values
sNT2 iBAQ values
sNT3 iBAQ values
iN1 iBAQ values
iN2 iBAQ values
iN3 iBAQ values
iNT1 iBAQ values
iNT2 iBAQ values
iNT3 iBAQ values
pro FASTA header
E Evidence groups
FDR False Discovery Rate ...

plot.CompleteEnrichment

This function plot a heatmap to summarize the results from complete functional enrichment analysis

Description

This function plot a heatmap to summarize the results from complete functional enrichment analysis

Usage

```
## S3 method for class 'CompleteEnrichment'  
plot(x, ...)
```

Arguments

x object from complete functional enrichment analysis
... list of additional arguments

summary.CompleteEnrichment

This function summarize and print the results from complete functional enrichment analysis

Description

This function summarize and print the results from complete functional enrichment analysis

Usage

```
## S3 method for class 'CompleteEnrichment'  
summary(object, ...)
```

Arguments

object object from complete functional enrichment analysis
... list of additional arguments

- HeatMapEnrichment, 4
- plot.CompleteEnrichment, 6
- * **file**
 - HeatMapEnrichment, 4
 - plot.CompleteEnrichment, 6
- * **filter**
 - filterData, 3
- * **following**
 - HeatMapEnrichment, 4
 - plot.CompleteEnrichment, 6
- * **getwd()**
 - HeatMapEnrichment, 4
 - plot.CompleteEnrichment, 6
- * **graphic**
 - HeatMapEnrichment, 4
 - plot.CompleteEnrichment, 6
- * **head(str(Func))**
 - HeatMapEnrichment, 4
 - plot.CompleteEnrichment, 6
 - summary.CompleteEnrichment, 6
- * **heatmap**
 - completeFunctionalRegulationAnalysis, 2
 - HeatMapEnrichment, 4
 - plot.CompleteEnrichment, 6
- * **if**
 - HeatMapEnrichment, 4
 - plot.CompleteEnrichment, 6
- * **inside**
 - HeatMapEnrichment, 4
 - plot.CompleteEnrichment, 6
- * **is**
 - HeatMapEnrichment, 4
 - plot.CompleteEnrichment, 6
- * **library(CoFRA)**
 - HeatMapEnrichment, 4
 - plot.CompleteEnrichment, 6
 - summary.CompleteEnrichment, 6
- * **note**
 - HeatMapEnrichment, 4
 - plot.CompleteEnrichment, 6
- * **of**
 - HeatMapEnrichment, 4
 - plot.CompleteEnrichment, 6
- * **ontology**
 - getFunctionalCategories, 3
- * **overwrite**
 - HeatMapEnrichment, 4
- plot.CompleteEnrichment, 6
- * **pdf(CCxxxTest.pdf)**
 - HeatMapEnrichment, 4
 - plot.CompleteEnrichment, 6
- * **pdf**
 - HeatMapEnrichment, 4
 - plot.CompleteEnrichment, 6
- * **setup**
 - HeatMapEnrichment, 4
 - plot.CompleteEnrichment, 6
- * **studio**
 - HeatMapEnrichment, 4
 - plot.CompleteEnrichment, 6
- * **summary.CompleteEnrichment(CC1)**
 - summary.CompleteEnrichment, 6
- * **summary**
 - summary.CompleteEnrichment, 6
- * **that**
 - HeatMapEnrichment, 4
 - plot.CompleteEnrichment, 6
- * **the**
 - HeatMapEnrichment, 4
 - plot.CompleteEnrichment, 6
- * **this**
 - HeatMapEnrichment, 4
 - plot.CompleteEnrichment, 6
- * **to**
 - HeatMapEnrichment, 4
 - plot.CompleteEnrichment, 6
- * **use**
 - HeatMapEnrichment, 4
 - plot.CompleteEnrichment, 6
- * **using**
 - HeatMapEnrichment, 4
 - plot.CompleteEnrichment, 6
- * **way**
 - HeatMapEnrichment, 4
 - plot.CompleteEnrichment, 6
- * **write**
 - HeatMapEnrichment, 4
 - plot.CompleteEnrichment, 6
- completeFunctionalRegulationAnalysis, 2
- filterData, 3
- getFunctionalCategories, 3
- HeatMapEnrichment, 4

iBAQ, [4](#)

plot.CompleteEnrichment, [6](#)

summary.CompleteEnrichment, [6](#)