Package: easyDifferentialGeneCoexpression (via r-universe)

September 3, 2024

Encoding UTF-8

Title Easily Performs Differential Coexpression Analysis

Version 1.4

Description A function that reads in the GEO code of a list of probesets or gene symbols, a gene expression dataset GEO accession code, the name of the dataset feature discriminating the two conditions for the differential coexpression, and the values of the two different conditions for the differential coexpression, and returns the significant pairs of genes/probesets with highest differential coexpression (p-value < 0.005). If the input gene list is made of gene symbols, this package associates the probesets to these gene symbols, if found. Platforms available: GPL80, GPL8300, GPL80, GPL96, GPL570, GPL571, GPL20115, GPL1293, GPL6102, GPL6104, GPL6883, GPL6884, GPL13497, GPL14550, GPL17077, GPL6480. GEO: Gene Expression Omnibus. ID: identifier code. The GEO datasets are downloaded from the URL <`https://ftp.ncbi.nlm.nih.gov/geo/series/`>.

BugReports `https://github.com/davidechicco/easyDifferentialGeneCoexpression/issues`

Depends R (>= 4.0.0)

License GPL-3

URL `https://github.com/davidechicco/easyDifferentialGeneCoexpression`

Imports annotate, Biobase, diffcoexp, geneExpressionFromGEO, GEOquery, jetset, limma, magrittr, xml2

Suggests knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 7.1.2

NeedsCompilation no

Author Davide Chicco [aut, cre] (<`https://orcid.org/0000-0001-9655-7142`>)

Maintainer Davide Chicco `<davidechicco@davidechicco.it>`

Repository CRAN

Date/Publication 2024-02-05 11:40:02 UTC

Contents

`easyDifferentialGeneCoexpression`

Function that computes the differential coexpression of a list of probesets in a specific dataset and returns the most significant pairs

---

Description

Function that computes the differential coexpression of a list of probesets in a specific dataset and returns the most significant pairs

Usage

```
easyDifferentialGeneCoexpression(
  list_of_probesets_to_select,
  GSE_code,
  featureNameToDiscriminateConditions,
  firstConditionName,
  secondConditionName,
  batchCorrection = TRUE,
  verbose = FALSE
)
```

Arguments

`list_of_probesets_to_select`	list of probesets for which the differential coexpression should be computed
`GSE_code`	GEO accession code of the dataset to analyze
`featureNameToDiscriminateConditions`	name of the feature of the dataset that contains the two conditions to investigate
`firstConditionName`	name of the first condition in the feature to discriminate (for example, "healthy")

`secondConditionName`	name of the second condition in the feature to discriminate (for example, "cancer")
`batchCorrection`	says if the script should perform the batch correction with limma::removeBatchEffect() or not
`verbose`	prints all the intermediate message to standard output or not

Value

a dataframe containing the significantly differentially co-expressed pairs of genes

Examples

```
probesetList <- c("200738_s_at", "217356_s_at", "206686_at")
verboseFlag <- "TRUE"
batchCorrection <- "TRUE"
signDiffCoexpressGenePairs <- easyDifferentialGeneCoexpression(probesetList,
"GSE3268", "description", "Normal", "Tumor", verboseFlag)
```

`fromProbesetToGeneSymbol` *Function that associates a gene symbol to a probeset for some Affymetrix platforms*

Description

Function that associates a gene symbol to a probeset for some Affymetrix platforms

Usage

```
fromProbesetToGeneSymbol(
  thisProbeset,
  thisPlatform,
  this_platform_ann_df,
  verbose = FALSE
)
```

Arguments

`thisProbeset`	probeset in input
`thisPlatform`	GEO platform accession code
`this_platform_ann_df`	annotation dataframe of the platform
`verbose`	prints all the intermediate message to standard output or not

Value

a gene symbol as string

`geoDataDownload` *Function that downloads gene expression data from GEO, after checking the connection*

Description

Function that downloads gene expression data from GEO, after checking the connection

Usage

```
geoDataDownload(GSE_code, verbose = FALSE)
```

Arguments

`GSE_code`	GEO code dataset
`verbose`	prints all the intermediate message to standard output or not

Value

a gene set gene expression AnnotationDataFrame

`geoPlatformAnnotationsDownload` *Function that downloads the annotations of a GEO platform*

Description

Function that downloads the annotations of a GEO platform

Usage

```
geoPlatformAnnotationsDownload(platformID, verbose = FALSE)
```

Arguments

`platformID`	GEO platform ID
`verbose`	prints all the intermediate message to standard output or not

Value

a dataframe containing the annotations of the GEO platform

`probesetRetrieval` *Function that reads a CSV file of probesets or gene symbols and, in the latter case, it retrieves the original probesets*

Description

Function that reads a CSV file of probesets or gene symbols and, in the latter case, it retrieves the original probesets

Usage

```
probesetRetrieval(
  probesets_or_gene_symbols,
  csv_file_name,
  platformCode,
  verbose = FALSE
)
```

Arguments

`probesets_or_gene_symbols`	flag saying if we're reading probesets or gene symbols
`csv_file_name`	complete name of CSV file containing the probesets or the gene symbols
`platformCode`	code of the microarray platform for which the probeset-gene symbol mapping should be done
`verbose`	prints all the intermediate message to standard output or not

Value

a vector of probesets

Index