Package 'levi'

April 3, 2025

Type Package

Title Landscape Expression Visualization Interface

Version 1.25.0

Maintainer Jose Luiz Rybarczyk Filho <jose.luiz@unesp.br>

Description The tool integrates data from biological networks with transcriptomes, displaying a heatmap with surface curves to evidence the altered regions.

Encoding UTF-8

LazyData true

RoxygenNote 7.0.1

Suggests rmarkdown, BiocStyle

Imports DT(>= 0.4), RColorBrewer(>= 1.1-2), colorspace(>= 1.3-2), dplyr(>= 0.7.4), ggplot2(>= 2.2.1), httr(>= 1.3.1), igraph(>= 1.2.1), reshape2(>= 1.4.3), shiny(>= 1.0.5), shinydashboard(>= 0.7.0), shinyjs(>= 1.0), xml2(>= 1.2.0), knitr, Rcpp (>= 0.12.18), grid, grDevices, stats, utils, testthat, methods, rmarkdown

LinkingTo Rcpp

License GPL (>= 2)

biocViews GeneExpression, Sequencing, Network, Software

VignetteBuilder knitr

LazyLoad yes

NeedsCompilation yes

git_url https://git.bioconductor.org/packages/levi

git_branch devel

git_last_commit ccbbbd4

git_last_commit_date 2024-10-29

Repository Bioconductor 3.21

Date/Publication 2025-04-03

Author Rafael Pilan [aut],
Isabelle Silva [ctb],
Agnes Takeda [ctb],
Jose Rybarczyk Filho [ctb, cre, ths]

Contents

`levi` *levi*

Description

This is the Levi script mode. It allows you to create the integration of networks and gene expression levels as batch processing

Usage

```
levi(expressionInput, fileTypeInput, networkCoordinatesInput,
networkInteractionsInput, geneSymbolnput, readExpColumn,
contrastValueInput, zoomValueInput, resolutionValueInput,
smoothValueInput, expressionLog, contourLevi, setcolor)
```

Arguments

`expressionInput`
: Filename of gene expression data, which is a numeric data.frame or matrix. The rows represent genes/proteins and the columns represent the experiment (RNA-seq, microarray, etc).

`fileTypeInput`
: Filename of biological network. Levi can read files written in the following formats: Medusa (DAT), RedeR (DYN), Pajek (NET) and STRING/STITCH

`networkCoordinatesInput`
: It allows the user to load the coordinate of the nodes the network.

`networkInteractionsInput`
: Parameter available only to STRING/STITCH data format. It allows the user to load the interaction data file of the network.

`geneSymbolnput`
: Column name from gene expression data containing the identifier (gene Symbol, Entrez ID, EMSEMBL, etc).

`readExpColumn`
: Variable from readExpColumn function containing the comparisons of the experiments

`contrastValueInput`
: Numeric value for image contrast. The variable range is 0 to 100. The default value is 50

`zoomValueInput`
: Numeric value for image zoom. The variable range is 0 to 100. The default value is 50.

`resolutionValueInput`
: Numeric value for image resolution. The variable range is 0 to 100. The default value is 50.

`smoothValueInput`
: Numeric value for image smoothness. The variable range is 0 to 100. The default is 50.

`expressionLog`
: Logical variable to indicate Log2 normalization in the expression levels. The default is FALSE

`contourLevi`
: Logical variable to allow contour lines. The default is TRUE.

`setcolor`
: Select the color palette to build the heatmat. There is two options the **Multicolor** has 20 color levels combined. The **Two colors** has two types of color and the options available are: *purple_pink*, *green_blue*, *blue_yellow*, *pink_green*, *orange_purple*, *green_marine*.

Details

Integrates the biological network and gene expression levels (or other type of data)

Value

Return a ggplot object and print a image (heatmat).

Author(s)

Isabelle Mira da Silva (isabelle.silva@unesp.br), José Rafael Pilan (rafael.pilan@unesp.br)

Examples

```
template_network <- file.path(system.file(package="levi"),"extdata",
   "medusa.dat", fsep = .Platform$file.sep)

template_expression <- file.path(system.file(package="levi"),
   "extdata","expression.dat", fsep = .Platform$file.sep)

multicolor <- levi(networkCoordinatesInput = template_network,
   expressionInput = template_expression, fileTypeInput = "dat",
   geneSymbolnput = "ID",
   readExpColumn=readExpColumn("TumorCurrentSmoker-NormalNeverSmoker"),
   contrastValueInput = 50, resolutionValueInput  = 50, zoomValueInput = 50,
   smoothValueInput = 50, expressionLog = FALSE, contourLevi = TRUE)

twocolors <- levi(networkCoordinatesInput = template_network,
   expressionInput = template_expression, fileTypeInput = "dat",
   geneSymbolnput = "ID",
   readExpColumn = readExpColumn("TumorCurrentSmoker-NormalNeverSmoker"),
   setcolor = "pink_green", contourLevi = FALSE)
```

LEVIui *LEVIui*

Description

Launch the Levi Graphical User Interface (GUI) in local machine.

Usage

```
LEVIui(browser)
```

Arguments

browser	This argument is necessary to launch Levi GUI. To launch Levi in the web browser the argument required "TRUE". To launch Levi in the R environment the argument required "FALSE". The default is "FALSE"

Details

This function launch the LEVI Graphical User Interface. The interface provides the same tools available in the script mode. There are two tools only available in the user interface: 1) Selection of area from heatmap to calculate the gene expression levels in the area selected; 2) Selection of the genes in some specific area from the image.

Value

return a GUI

Author(s)

José Rafael Pilan <rafael.pilan@unesp.br> & Isabelle Mira da Silva (isabelle.silva@unesp.br)

Examples

```
LEVIui(browser)
#LEVIui(browser=TRUE)  #Launch Levi to Browser.
#LEVIui(borwser=FALSE) #Launch Levi to R environment.
```

readExpColumn *readExpColumn*

Description

This function helps to prepare the data in the script mode. It also allows the obtention of dataset plot as a batch processing.

Usage

```
readExpColumn(x,...)
```

Arguments

x	Names of two expression datasets to be compared. They should be separated by hyphen (-)
...	To add more comparisons, each combination must be separated by comma (,).

Details

List the names of the expression datasets that will be used for comparison

Value

Returns the names of comparisons to be used by Levi

Note

To generate a plot from a single dataset, the name of the sample must be informed twice (Ex. "CaseA-CaseA")

Author(s)

José Rafael Pilan (rafael.pilan@unesp.br)

Examples

```
base <- readExpColumn(a="NormalNeverSmoker-NormalNeverSmoker")
```

Index