

Vignette

Contents

GUI introductive vignette

1

GUI introductive vignette

Introduction of propaGUIation applied on a network-based propagated profile obtained from a gene expression profile. This vignette is recommended for who wants to understand how to start the GUI and explore the propagation scores (aka guilty scores) associated to the patient's features due to its overall biology and starting information.

What you will get:

- You will discover how to start the GUI

For further information about how to use the GUI and which information it provides, I suggest you to read the README page of the github repository which provides you all the details about the interface.

Let's clean and prepare the enviroment for the workflow

We remove every variables, clean the RAM memory and load GUI library We set the random seed in order to get always the same results out of this workflow

```
#Clean workspace and memory ----
rm(list=ls())
gc()
#>          used (Mb) gc trigger (Mb) max used (Mb)
#> Ncells 404669 21.7      845531 45.2    621331 33.2
#> Vcells 741328  5.7      8388608 64.0   1642323 12.6

#Set working directory ----
gps0=getwd()
gps0=paste(gps0,"/%s",sep="")
rootDir=gps0
setwd(gsub("%s","",rootDir))

#Load libraries ----
suppressWarnings(suppressMessages(
  library("propaGUIation", quietly = T)
))

#Set variables ----
#Set seed to get always the same results out of this vignette
set.seed(8)
```

The GUI works with the original patient's profile, the propagated patient's profile and the network that has been used to propagate the starting patient's values. In this vignette, we tested the GUI in the visualization of a dummy patient's profile that had genes as features and expression levels as values.

```
#Load and set up the example data ----
data("example")

# get the original patient profile
input_m = example$input_gm; print(input_m)
#>           profile
#> DDX11L1          9
#> WASH7P           8
#> MIR6859-1        7
#> MIR1302-2HG       6
#> MIR1302-2         5
#> FAM138A           4
#> OR4G4P            3
#> OR4G11P           2
# get the propagated profile
gg_prop = example$gg_prop; head(gg_prop)
#> Loading required package: Matrix
#>
#> Attaching package: 'Matrix'
#> The following object is masked from 'package:spam':
#>
#>      det
#> 6 x 1 sparse Matrix of class "dgCMatrix"
#>           profile
#> DDX11L1      8.582738
#> WASH7P       7.512264
#> MIR6859-1    6.712264
#> MIR1302-2HG  6.516548
#> MIR1302-2    5.078768
#> FAM138A      4.039055
# get the gene-gene interaction network
g_net = example$gg_net; head(g_net);
#>      [,1]      [,2]
#> [1,] "DDX11L1" "WASH7P"
#> [2,] "DDX11L1" "MIR6859-1"
#> [3,] "DDX11L1" "MIR1302-2HG"
#> [4,] "WASH7P"  "MIR1302-2"
#> [5,] "MIR6859-1" "MIR1302-2"
#> [6,] "MIR1302-2" "FAM138A"
```

The GUI accepts the annotation of the features. In case of genes (hsa,mmu), it offers the possibility to automatically get it with a function. The annotation will allow the GUI to connect the node names to the databases which offer more details and explanations about their function.

```
# download the annotation data for the genes in the propagated profile
g_ann <- download_genes_annotation(organism = "human", gene_names = rownames(gg_prop))
#> Warning: 'select_()' is deprecated as of dplyr 0.7.0.
#> Please use 'select_()' instead.
#> This warning is displayed once every 8 hours.
```

```

#> Call 'lifecycle::last_warnings()' to see where this warning was generated.
#> Warning: 'filter_()' is deprecated as of dplyr 0.7.0.
#> Please use 'filter()' instead.
#> See vignette('programming') for more help
#> This warning is displayed once every 8 hours.
#> Call 'lifecycle::last_warnings()' to see where this warning was generated.
#> Cache found
head(g_ann)
#>           ID chr start  end
#> 1    DDX11L1 chr1 11869 14409
#> 2    FAM138A chr1 34554 36081
#> 3  MIR1302-2 chr1 30366 30503
#> 4 MIR1302-2HG chr1 29554 31109
#> 5  MIR6859-1 chr1 17369 17436
#> 6    OR4G11P chr1 57598 64116

```

We give all the prepared data in a function which creates an igraph object including all the details for the interactive visualization

```

#Create igraph object with all the information included
net=create_net2plot(g_net = g_net, input_m = input_m, gf_prop = gg_prop, ann_net_b = g_ann)
head(net)
#> 6 x 8 sparse Matrix of class "dgCMatrix"
#>           DDX11L1 WASH7P MIR6859-1 MIR1302-2HG MIR1302-2 FAM138A OR4G11P
#> DDX11L1      .      1      1      1      .      .      .
#> WASH7P      1      .      .      .      1      .      1
#> MIR6859-1    1      .      .      .      1      .      1
#> MIR1302-2HG  1      .      .      .      .      .      .
#> MIR1302-2    .      1      1      .      .      1      .
#> FAM138A      .      .      .      .      1      .      .
#>           OR4G4P
#> DDX11L1      .
#> WASH7P      .
#> MIR6859-1    .
#> MIR1302-2HG  .
#> MIR1302-2    1
#> FAM138A      1

```

We can start the GUI

```

#start_GUI(net, g_ann, example$chr_len, example=T)

```