R documentation

of all in ‘man’

October 5, 2022

R topics documented:

`batches_analysis` *batches_analysis*

Description

function to analyze each batch. A batch is a fragment of the original dataset. Internal function.

Usage

```
batches_analysis(all_groups, n_cores, final_df, path_dir, thr_score,
  progress_bar = T, nboots)
```

Arguments

`all_groups`	All batches names to analyze
`n_cores`	Number to use
`final_df`	Dataframe of features reduced
`path_dir`	Path to input directory. Mandatory.
`thr_score`	Threshold for filtering the similarity scores
`progress_bar`	If False,disable progress bar.
`nboots`	Number of bootstrap simulations for the density identiy test. Default to 6

Value

List of results batch analysis

`contract_data` *contract_data*

Description

function to contract the visnetdata object (containing nodes and edges info). Internal function.

Usage

```
contract_data(data)
```

Arguments

`data`	List of the full VisNetwork dataframe containing the edges and nodes information

Value

A list of two VisNetwork dataframes containing the contracted information of nodes and edges `exports_filtered_plots` *exports_filtered_plots*

Description

function to export only the files to keep for each group

Usage

```
exports_filtered_plots(visnetdata, df_features, path_expr_data,
  path_output = NULL, n_cores = 1, type = "dens", filter_thr = 2.1,
  plot_dendrogram = F, seed_n = 40)
```

Arguments

`visnetdata`	list of dataframes containing the edges and nodes information in VisNetwork format
`df_features`	A dataframe containing the original features set (before the tsne reduction) of all input files
`path_expr_data`	Path to directory containing the expression data of the files analyzed
`path_output`	Path to the directory where to export the plots for each group If path_output=NULL, no plots are generated. Default to NULL
`n_cores`	Number of cores to use. Default to 1
`filter_thr`	float number indicating the cut height for the dendrogram generated by the hierarchical clustering. Default to 2.1
`plot_dendrogram`	If True, the plot of the dendrogram is generated (It may fail for some groups). Default to False.
`seed_n`	Set seed number. Default to 40.

Value

A list of the selected files names to keep for each group

Examples

```
exports_filtered_plots(visnetdata=visnetdata,df_features=df_features,
path_expr_data="path to input directory",path_output=NULL,n_cores=1,filter_thr=2.1,plot_dendrogram=F)
```

exports_plots *exports_plots*

Description

function to export all plots for each group

Usage

```
exports_plots(visnetdata, path_expr_data, path_output = NULL,
  n_cores = 1, type = "dens", path_gates = NULL)
```

Arguments

`visnetdata` list of dataframes containing the edges and nodes information in VisNetwork format

`path_expr_data` Path to directory containing the expression data of the files analyzed

`path_output` Path to the directory where to export the plots for each group If path_output=NULL, no plots are generated. Default to NULL

`n_cores` Number of cores to use. Default to 1

`path_gates` Path of gates data, default to NULL

Value

A list of all file names for each group.

Examples

```
exports_plots(visnetdata=visnetdata,path_expr_data="path to input directory",path_output=NULL,n_cores=1)
```

flowSim_plot *flowSim_plot*

Description

function to generate the scatter plot with colored density of the events. Internal function.

Usage

```
flowSim_plot(df, plot_gate = F)
```

Arguments

`df` Dataframe containing the two markers expression data.

`plot_gate` show gates?,default to False

Value

Plot object

`flowSim_plot_dens` *flowSim_plot_dens*

Description

function to generate the denity scatter plot with density on the sides. Internal function.

Usage

```
flowSim_plot_dens(df)
```

Arguments

`df`	Dataframe containing the two markers expression data.

Value

Plot object

`gen_igraph_network` *Generate igraph network*

Description

function to generate the igraph network from similiraty scores

Usage

```
gen_igraph_network(similarity_output, method_clust = "louvain")
```

Arguments

`similarity_output`	The distance matrix generated by the get_similarity_all_plots functions
`method_clust`	clustering method: louvain, infomap. Default to louvain

Value

two objects $igraph_network : igraph network object $partition : vector of the clusters association

Examples

```
gen_igraph_network(similarity_output=out,method_clust="louvain")
```

gen_visnetwork_data *gen_visnetwork_data*

Description

function to convert igraph network to a list of two VisNetwork dataframes containig the edges and nodes information needed to generate a VisNetwork object

Usage

```
gen_visnetwork_data(igraph_network, contract_net = F,
  single_nodes_black = T)
```

Arguments

`igraph_network`	igraph network object
`contract_net`	If True, it generates a contracted version of the VisNetwork dataframes (to visualize large datasets). Default to False
`single_nodes_black`	Isolated files/nodes colored in black? Default to True.

Value

If contracted_net=False, a list of full VisNetwork dataframes is generated, if contracted_net=True, a list of contracted VisNetwork dataframes is returned. $nodes : dataframe containing the nodes information (full or contracted) $edges : dataframe containing the edges information (full or contracted)

Examples

```
gen_visnetwork_data(igraph_network=igraph_network,contract_net=F)
```

get_csv *get_csv*

Description

function to convert fcs files in bivariate csv files

Usage

```
get_csv(path, channels = NULL, markers = NULL, path.output)
```

Arguments

`path` path of the directory with fcs files to convert

`channels` Vector reporting the two channels to consider

`markers` Vector reporting the two markers to consider. Considered only if channels==NULL

`path.output` directory path to export the converted csv files

Examples

```
get_csv(path="path/to/directory",channels=c("FSC-A","FSC-H"),path.output="path/to/directory")
```

`get_densRange` *get_densRange*

Description

function to get the correct coordinates (correct range) based on the density coordinates x and y generated by the density() function. Internal function.

Usage

```
get_densRange(x, y, thr, direction = FALSE)
```

Arguments

`x` x axis density coordinates

`y` y axis density coordinates

`thr` Treshold of expression where to start the density calculation

`direction` On the left (FALSE) or on the right (TRUE) of the threshold

Value

Coordinates density points for the x and y axis

get_distance_loc_vs_test

get distance selected file vs another local file

Description

function to compare the similiraty between selected file and another file of the input directory based on the density of marker expression values. Internal function.

Usage

```
get_distance_loc_vs_test(test_df, loc_df, show_plot = "none",
  nboot = 50)
```

Arguments

`test_df`	Dataframe of selected file
`loc_df`	Dataframe of another file of the input directory
`show_plot`	show plot of the density comparison
`nboot`	Number of simulations for the permuation identity test between the densities.

Value

A list of two float numbers $pvalue_mark_1 : pvalue first marker comparison $pvalue_mark_2 : pvalue second marker comparison

get_groups_based_on_clustering

Generate groups based on a clustering method

Description

function to clusterize the samples using different methods. Internal function.

Usage

```
get_groups_based_on_clustering(igraph_network, method = "louvain")
```

Arguments

`igraph_network`	iGraph network object
`method_clust`	clustering method: louvain, infomap. Default to louvain

Value

dataframe reporting groups association

get_heterogeneity_score *Get heterogeneity score*

Description

function to get the heterogeneity score of a dataset

Usage

```
get_heterogeneity_score(igraph_network, partition_vec, visnetdata)
```

Arguments

`igraph_network`	Igraph network object
`partition_vec`	Vectors of files and clusters associations
`visnetdata`	list of dataframes containing the edges and nodes information in VisNetwork format

Value

A single float number indicating the final heterogeneity score

Examples

```
gen_igraph_network(igraph_network=igraph_network,partition_vec=partition_vec,visnetdata=visnetdata)
```

get_hull_all_gates *get_hull_all_gates*

Description

function to get the convex hull of all gates. Internal function

Usage

```
get_hull_all_gates(gated_df)
```

Arguments

`df_groups`	Dataframe containing the groups associations

Value

A list containing the convex hull for all classes

get_inds_files_selected *get_inds_files_selected*

Description

function to get indices for the paths of the files selected

Usage

```
get_inds_files_selected(path_dir, files_selected, return_path = F)
```

Arguments

`path_dir`	path of the directory to explore
`files_selected`	Vector containing the names of the files whose indices need to be found
`return_path`	If True, return paths instead of indices. Default to False

Value

Vector containing the indices of the files

Examples

```
get_inds_files_selected(path_dir="path/to/directory",files_selected=vector_names)
```

get_new_colors_based_on_groups *get_new_colors_based_on_groups*

Description

function to get new colors based on groups. Internal function

Usage

```
get_new_colors_based_on_groups(df_groups)
```

Arguments

`df_groups`	Dataframe containing the groups associations

Value

A dataframe containing the association between groups,files and colors

get_similarity_all_plots

Generate similarity scores all files V1 mode

Description

function to generate the similarity scores for all the files within the input directory. V1 mode (for small datasets only,N<500).

Usage

```
get_similarity_all_plots(n_samples = NULL, path_dir, thr_score = 0.6,
  n_cores = 1, nboots = NULL)
```

Arguments

`n_samples`	Number of files to analyze. If NULL, all files are analyzed. Default to NULL.
`path_dir`	Path to tthe input directory
`thr_score`	Threshold for filtering only the most significant similarity score, it ranges from 0 to 1. Default to 0.6
`n_cores`	Number of cores to use, default to 1
`nboots`	Number of bootstrap simulations for the density identiy test. If NULL, it is set based on number of files anayzed. Default to NULL

Value

A list of two dataframes $edges : A dataframe containing edges information of the network. The weight of each edge is the similarity score between the the two files/nodes $nodes : A dataframe containing several nodes information (Node ID,cluster color etc...)

Examples

```
get_similarity_all_plots(path_dir="path to input directory",thr_score=0.9,n_cores=1)
```

get_similarity_all_plots_v2

Generate similarity scores all files V2 mode

Description

function to generate the similarity scores for all the files within the input directory. V2 mode (for big datasets)

Usage

```
get_similarity_all_plots_v2(n_samples = NULL, path_dir,
  thr_score = 0.9, n_cores = 1, n_batches = NULL,
  show_tsne_plot = T, progress_bar = T, size_points_tsne = 15,
  nboots = 6)
```

Arguments

`n_samples` Number of files to analyze. If NULL, all input files are anayzed. Default to NULL.

`path_dir` Path to input directory. Mandatory.

`thr_score` Treshold for filtering only the most significant similarity score, it ranges from 0 to 1. Default to 0.9

`n_cores` Number of cores to use, default to 1

`n_batches` Number of batches to use = number of parts in which the input datasets must be divided to increase speed and memory efficiency. If number of files <= 2000 files, this integer is automatically estimated, otherwise it needs to be defined. As the value increases, less memory and time is required but also more residual homogeneity is generated.

`show_tsne_plot` show scatter plot of tsne reduced dataset of the density features. Default to True.

`progress_bar` if False,disable progress bar. Default to True.

`size_points_tsne` Size of points in t-sne plot. Default to 2

`nboots` Number of bootstrap simulations for the density identiy test. Default to 6

Value

A list of three dataframes $edges : A dataframe containing edges information of the network. The weight of each edge is the similarity score between the the two files/nodes $nodes : A dataframe containing several nodes information (Node ID,cluster color etc...) $df_features: A dataframe containing the original features set (before the tsne reduction) of all input files

Examples

```
get_similarity_all_plots_v2(n_samples=1:2000,path_dir="path to input directory",thr_score=0.9,n_cores=1,n_batch
```

get_similarity_selected_plot

Get similarity score selected file

Description

function to get the similarity of one file vs the rest. Internal usage only.

Usage

```
get_similarity_selected_plot(list_all_input_dfs, name_ref_plot, n_cores,
  names_analyzed_vec, n_boots)
```

Arguments

`list_all_input_dfs`	List of the dataset of the input directory generated by the import_all_dfs function
`name_ref_plot`	file to analyze
`n_cores`	Number of cores to use
`names_analyzed_vec`	Vector containing the names of the files already analyzed in a previous iteration
`n_boots`	Number of simulations for the permutation identity test of marker expression densities

Value

Dataframe of the similarity scores for the selected file

`get_size_based_on_groups` *get_size_based_on_groups*

Description

function to get size of groups based on groups. Internal function

Usage

```
get_size_based_on_groups(df_groups)
```

Arguments

`df_groups`	Dataframe containing the groups associations

Value

A dataframe containing the association between groups,files and size

`import_all_dfs` *Import files to analyze*

Description

function to import all datasets of the input directory

Usage

```
import_all_dfs(n_samples = NULL, paths_dir = NULL,
  paths_plots = NULL, n_cores = 1)
```

Arguments

`n_samples` Number of files to analyze. If NULL, all files are analyzed. Default to NULL.

`paths_dir` directory path of the input files to analyze

`paths_plots` Mandatory if paths_dir==NULL, it is possible to feed a list of the paths of the input files to analyze

`n_cores` The number of cores to use,default to 1

Value

list of all input dfs

Examples

```
import_all_dfs(paths_dir="path_to_input_directory",paths_plots=NULL,n_cores=1)
```

`import_df_features` *Import features set of all files*

Description

function to import the features set of all the files under analysis

Usage

```
import_df_features(n_samples = NULL, paths_dir, n_cores = 1,
  progress_bar = T)
```

Arguments

`n_samples` number of files to analyze

`paths_dir` directory path of the input files to analyze

`n_cores` The number of cores to use, default to 1

`progress_bar` If False,progress bar is disabled. Default to True

Value

a dataframe of the features set (columns) for each file (row)

Examples

```
import_df_features(n_samples=1:5,paths_dir="path_to_input_directory",n_cores=1)
```

myKmeans *kmeans function*

Description

function to perform k-means clustering on the tsne reduced dataset of density features. Internal function.

Usage

```
myKmeans(dist, k)
```

Arguments

dist	dataframe of the tsne reduced features set between the files
k	Number of centers

Value

k-means clustering output

plot_visnet *plot_visnet*

Description

function to plot the visnetwork based on the similarity scores

Usage

```
plot_visnet(visnetdata, show_legend = F, remove_single_files = F,
  select_group = NULL, size_nodes = NULL)
```

Arguments

`visnetdata` list of dataframes containing the edges and nodes information in VisNetwork format

`show_legend` show legend?

`remove_single_files` If True, single files (black nodes) are removed from the analysis.

`select_group` A vector indicating the names of the groups to visualize. If NULL, all groups are visualized. Default to NULL

`size_nodes` Set size of the nodes. Default to NULL.

Value

VisNetwork object

Examples

```
plot_visnet(visnetdata=visnetdata,show_legend=F,remove_single_files=F,select_group=NULL)
```

rename_files *rename_files*

Description

Function to rename files within a directory

Usage

```
rename_files(path_dir, to_replace, string, to_replace_fixed = T)
```

Arguments

`path_dir` path of the directory with files to rename

`to_replace` Pattern to replace. if to_replace_fixed=F, regex expression can be used.

`string` string to replace the pattern selected

`to_replace_fixed` if False, to_replace can be also a regex expression. Default to True.

Examples

```
rename_files(path_dir="path/to/directory",to_replace="pattern to replace",string="string replaced",regex_expr=F
```

`show_plot` *show_plot*

Description

function to visualize the selected plot using the flowSim_plot or flowSim_plot_dens function

Usage

```
show_plot(path_data_file, type = "classic")
```

Arguments

`path_data_file`	Path to the csv files to visualize
`type`	type of visualization: "classic" or "dens". Default to classic. the "dens" version has the density of the markers on the side of each axis.

Value

Plot object

Examples

```
show_plot(path_data="path to input directory",type="classic")
```

Index