MetabAnalyse

July 4, 2022

`add_screening_vars` *Function to add measurements taken at screening time for samples to be added to all timepoints*

Description

Function to add measurements taken at screening time for samples to be added to all timepoints

Usage

```
add_screening_vars(object, vars)
```

Arguments

`object`	An object of class metab_analyser
`vars`	A character naming the vars of interest

Value

phenotype data which can be replaced into the original object or use it separately with a different object

---

`adni_add_index` *Function to add indices i.e. RID and timepoints to data matrix*

Description

add ids to check for data with ease

Usage

```
adni_add_index(data)
```

Arguments

data	data matrix with rownames as adni_ids

Value

data matrix with new added columns

---

adni_filter_full_tp *Function to extract timepoints of ineterest for longitudnal analysis*

Description

function to extract the timepoints that are of importance in longitudnal analysis similar to split_acc_to_time except is applied on data matrix rather than the object itself

Usage

```
adni_filter_full_tp(x.data, tp, by = c("timepoint"))
```

Arguments

x.data	data matrix of interest
tp	timepoints of interest

Value

data matrix with only timepoints of interest

---

adni_ggm_calc_ggm_dynamic *Function to calculate a dynamic GeneNet GGM*

Description

calculates GGM on longitudnal data matrix and returns a dataframe with edges, partial correlation and associated p-values

Usage

```
adni_ggm_calc_ggm_dynamic(data, threshhold = c("bonferoni", "FDR", "li"))
```

Arguments

data	data matrix in a longitudnal format(see adni_convert_ggm_longitudnal)
threshold	type of multiple hypothesis correction. Available are Bonferoni("bonferoni"), Benjamini-Hochberg("FDR") and independent tests method("li", also see Li et al ....)

Value

a dataframe with edges, partial correlation and associated p-values

---

`adni_ggm_convert_longitudinal` *Function to convert data into a longitudnal format for GeneNet ggms*

Description

converts a dataset with full data into a longitudnal version

Usage

```
adni_ggm_convert_longitudinal(data)
```

Arguments

`data`	data matrix with metabolite concentrations

Value

data matrix converted into a longitudnal format

`adni_rm_index` *Function to remove indices i.e. RID and timepoints from data matrix*

Description

remove ids that are used to check for data with ease(see adni_add_index())

Usage

```
adni_rm_index(data)
```

Arguments

`data`	data matrix with rownames as adni_ids

Value

data matrix without index columns

automated_ggm_genenet *An automated fucntion to calculate GGM from genenet*

Description

automated funtion that can be applied on s3 object obtained after prep_data_for_ggms() to obtain geneNet network along with threshold used

Usage

```
automated_ggm_genenet(object, which_data, threshhold, timepoints)
```

Arguments

object	object obtained after applying prep_data_for_ggms() on S4 object of class metab_analyse with type="single"
which_data	a character or a character vector naming the datasets of interest
timepoints	timepoints of interest that are to be used to build networks
threshold	type of threshold to be used for extracting significant edges

Value

Network data with edgelist, partial correlation values and associated p-values and corrected p-values

automated_ggm_mlp *An automated fucntion to calculate GGM from genenet*

Description

automated funtion that can be applied on s3 object obtained after prep_data_for_ggms() to obtain geneNet network along with threshold used. This function is not applicable for singular datasets

Usage

```
automated_ggm_mlp(object, which_data, rho, nfolds, timepoints)
```

Arguments

object	object obtained after applying prep_data_for_ggms() on S4 object of class metab_analyse with type="multi" not applicable to apply on singular sets
which_data	a character or a character vector naming the datasets of interest
rho	tuning parameter for regression
nfolds	check
timepoints	timepoints of interest that are to be used to build networks

Value

Network data with edgelist, partial correlation values and associated p-values and corrected p-values

---

`automated_temporal_network` *An automated function to caluclate temporal network with lagged model*

Description

calculates temporal networks for each dataset with a lagged model as used in graphical VAR

Usage

```
automated_temporal_network(object, lag, rho, timepoints, which_data)
```

Arguments

`object`	object obtained after applying prep_data_for_ggms() on S4 object of class metab_analyser
`lag`	which lagged model to use. 1 means one-lagged model, similary 2,3,..etc
`rho`	parameter for regression coefficient
`timepoints`	timepoints of interest that are to be used to build networks(in the order of measurement)
`which_data`	dataset or datasets to be used

Value

temporal network data with edgelist and regression values

`common_sample_extractor` *Function to get only common samples from the dataframes in list_of_data*

Description

Function to get only common samples from the dataframes in list_of_data

Usage

```
common_sample_extractor(object, time_splitter = FALSE)
```

Arguments

object	An object of class metab_anaylser
time_splitter	A boolean input: True leads to splitting of the data wrt time, False returns all the dataframes as they are with common rows

Value

list_of_data with common samples across all time points

---

convert_s4_to_s3 *Function to Convert S4 object of class metab_analyser to an S3 object with same architecture*

Description

Function to Convert S4 object of class metab_analyser to an S3 object with same architecture

Usage

```
convert_s4_to_s3(object)
```

Arguments

object	An object of class metab_analyser

Value

An S3 object of the same data as metab_analyser in other words all slots are now converted into nested lists

---

distribution_plotter *Function for Plotting distributions of phenotypic variables*

Description

Function for Plotting distributions of phenotypic variables

Usage

```
distribution_plotter(object, colname, which_data, strats)
```

Arguments

object	An object of class metab_analyser
colname	Name of the variable whose distribution is of interest
which_data	Name of the dataset from which the samples will be extracted

Value

a list with either 1) density plot, mean table acc to timepoint and variable type or 2) bar plot, line plot, and variable type

---

`get_betas` *Function to perform regression on list of matrics either divided based on time(temporal net) or data type(Multibipartite Lasso)*

Description

Function to perform regression on list of matrics either divided based on time(temporal net) or data type(Multibipartite Lasso)

Usage

```
get_betas(list_of_mats, alpha, nfolds)
```

Arguments

`list_of_mats`	list of matrices that are divided based on platform or timepoints
`alpha`	parameter for glmnet alpha=1 represents ridge regression and alpha=0 represents lasso regression and anything in between results in a mixed penalty regression
`nfolds`	nfolds parameter for glmnet

Value

a list with different combinations used and each combination is a nested list with regression result data

---

`get_files_and_names` *Function to pack all the data into a single object of class "metab_analyser"*

Description

This function loads all the files from the parent directory. It assumes a certain naming pattern as follows: "datatype_NULL|col|row_data.rds" Any other naming pattern is not allowed. The function first writes all files into a list and each type of data is packed into its respective class i.e. col_data, row_data or data

Usage

```
get_files_and_names(path, annotations_index)
```

Arguments

path	Path to the parent directory
annotations_index	a list to be filled as follows = list(phenotype="Name or index of the files", medication="Name or index of the files")

Value

An object of class metab_analyser

---

get_metadata_for_plotting

Function to extract metadata of the metabolites

Description

Function to extract metadata of the metabolites

Usage

```
get_metadata_for_plotting(object, which_data, metab_groups, metab_ids)
```

Arguments

object	S4 object of class metab_analyse
which_data	choose the dataset from which metabolites will be extracted for metadata
metab_groups	choose the column that has metabolite groups
metab_ids	chodse the column that has metabolite names

Value

metadata dataframe with names, groups and class

`get_palette` *Function to get a palette of distinct colorblind friendly colors.*

Description

Function to get a palette of distinct colorblind friendly colors.

Usage

```
get_palette(n)
```

Arguments

`n`	number of colors wanted in the palette

Value

a color palette vector with colors in the form of hex codes

`get_samples_and_timepoints` *Function to know the number of timepoints and the total number of samples available at that point*

Description

Function to know the number of timepoints and the total number of samples available at that point

Usage

```
get_samples_and_timepoints(object, which_data)
```

Arguments

`object`	An object of class metab_analyser
`which_data`	Name of the dataset in context

Value

A data table with timepoints and number of samples at each timepoint

get_text *Function to Obtain textual information for visualization in interactive plots*

Description

Function to Obtain textual information for visualization in interactive plots

Usage

```
get_text(data, colnames)
```

Arguments

data	a dataframe with plotting data along with other variables for visualization
colnames	a character vector with the names of the variables that you want to see on the plot

Value

a vector with strings that can be parsed into plot_ly text.

ggm_visualizer *Function to plot data from network and object after calculating a certain ggm*

Description

A function to plot ggms of different kinds namely visNetwork and cytoscape. Make sure that cytoscape is running before running this function

Usage

```
ggm_visualizer(
  network,
  type_of_plot,
  metadata,
  main,
  type_of_data,
  timepoints_fold
)
```

Arguments

network	dataframe with information of the network colnames should be in this format node1, node2, corr_val and other columns
metadata	dataframe with three columns and another optional column 1) metabs("name") and 2) Group("group") they belong to 3) and the class("class") they belong to see get_metadata_for_plotting() for more information. The optional column is for colors
type_of_data	character defining the type of network applied to a single network or a multi dataset network. Available options = c("single","multi")
type	Type of visualization options = c("visNetwork", "Cytoscape")

Value

network plot based on the type chosen by the user

---

metab_analyser-class *Constructor to generate an object of class metab_analyser. contains slots - list_of_data: For the list of all data matrices. - list_of_col_data: list of all the col data files in the same order. - list_of_row_data: list of all the row data files in the same order. - annotations: list with phenotype and medication. Each of which is character that represents the name of the aforementioned dataset types.*

Description

Constructor to generate an object of class metab_analyser. contains slots - list_of_data: For the list of all data matrices. - list_of_col_data: list of all the col data files in the same order. - list_of_row_data: list of all the row data files in the same order. - annotations: list with phenotype and medication. Each of which is character that represents the name of the aforementioned dataset types.

---

pca_plotter_general *Function to Plot PCA for one dataset with samples as data points*

Description

Function to Plot PCA for one dataset with samples as data points

Usage

```
pca_plotter_general(object, which_data, cols_for_vis)
```

Arguments

object	An object of class metab_analyser
which_data	Name of the dataset from which the samples will be extracted
cols_for_vis	choose the colnames from phenotype data that you want to show in the text labels

Value

an interactive pca plot with text that can be modified.

---

pca_plotter_wrt_common

Function to plot PCA of the individuals and metabolites of common data

---

Description

Function to plot PCA of the individuals and metabolites of common data

Usage

```
pca_plotter_wrt_common(
  object,
  which_data,
  metab_groups,
  metab_ids,
  cols_for_vis,
  phenotype_index
)
```

Arguments

object	An object of class metab_analyser
which_data	a character vector - Names of the dataset from which the samples will be extracted
metab_groups	a character vector - names of the column which has the pathway information of the metabolites(please make sure they are in the same order as the above ones)
metab_ids	a character vector - names of the column with metabolite names in the col data matix(please make sure they are in the same order as the above ones)
cols_for_vis	choose the colnames from phenotype data that you want to show in the text labels
phenotype_index	index of the phenotype data. Can input either the name of the phenotype dataset or the index of the same

Value

a list with two plot objects 1) samples - PCA plot of the individuals(".$samples") 2) metabs - PCA plot of the metabolites(".$metabs")

`prep_data_for_ggms` *Function to prepare and preprocess S4 objects to use it for gaussian gaphical models. Also converts S4 to S3*

Description

Function to prepare and preprocess S4 objects to use it for gaussian gaphical models. Also converts S4 to S3

Usage

```
prep_data_for_ggms(object, which_type, mlp_or_temp)
```

Arguments

`object`	An object of class metab_analyser
`which_type`	two choices either: 1) single - converts S4 to S3 and returns the nested list 2) multi - extracts common samples across the dataframes and returns an S3 nested list

Value

An S3 object(nested list) with the same architecture as that of class metab_analyser

---

`split_acc_time` *Function to split the list of dataframes into a nested list with each dataframe being split into into dataframes of different timepoints*

Description

Function to split the list of dataframes into a nested list with each dataframe being split into into dataframes of different timepoints

Usage

```
split_acc_time(object)
```

Arguments

`object`	An object of class metab_analyser

Value

list_of_data with each dataframe being broken into a list of dataframes with respect to the timepoint they belong to

---

tsne_plotter_general *Function to plot tsne plots for one data set*

Description

Function to plot tsne plots for one data set

Usage

```
tsne_plotter_general(object, which_data, metab_ids, metab_groups, cols_for_vis)
```

Arguments

object	An object of class metab_analyser
which_data	Name of the dataset from which the samples will be extracted
metab_ids	name of the column with metabolite names in the col data matix
metab_groups	name of the column which has the pathway information of the metabolites
cols_for_vis	choose the colnames from phenotype data that you want to show in the text labels

Value

a list with two plot objects 1) samples - tSNE plot of the individuals(samples) 2) metabs - tSNE plot of the metabolites(metabs)

---

tsne_plotter_wrt_common *Function to plot tSNE plots for multiple datasets*

Description

Function to plot tSNE plots for multiple datasets

Usage

```
tsne_plotter_wrt_common(
  object,
  which_data,
  metab_groups,
  metab_ids,
  cols_for_vis
)
```

Arguments

object	An object of class metab_analyser
which_data	a character vector - Names of the dataset from which the samples will be extracted
metab_groups	a character vector - names of the column which has the pathway information of the metabolites(please make sure they are in the same order as the above ones)
metab_ids	a character vector - names of the column with metabolite names in the col data matix(please make sure they are in the same order as the above ones)
cols_for_vis	choose the colnames from phenotype data that you want to show in the text labels
phenotype_index	index of the phenotype data. Can input either the name of the phenotype dataset or the index of the same

Value

a list with two plot objects 1) samples - tSNE plot of the individuals(".$samples") 2) metabs - tSNE plot of the metabolites(".$metabs")

---

umap_plotter_general *Function to plot UMAP plots for one data set*

Description

Function to plot UMAP plots for one data set

Usage

```
umap_plotter_general(
  object,
  which_data,
  metab_ids,
  metab_groups,
  cols_for_vis,
  phenotype_index
)
```

Arguments

object	An object of class metab_analyser
which_data	Name of the dataset from which the samples will be extracted
metab_ids	name of the column with metabolite names in the col data matix
metab_groups	name of the column which has the pathway information of the metabolites
cols_for_vis	choose the colnames from phenotype data that you want to show in the text labels

Value

a list with two plot objects 1) samples - UMAP plot of the individuals(samples) 2) metabs - UMAP plot of the metabolites(metabs)

`umap_plotter_wrt_common`

Function to plot UMAP plots for multiple datasets

Description

Function to plot UMAP plots for multiple datasets

Usage

```
umap_plotter_wrt_common(
  object,
  which_data,
  metab_groups,
  metab_ids,
  cols_for_vis,
  phenotype_index
)
```

Arguments

`object`	An object of class metab_analyser
`which_data`	a character vector - Names of the dataset from which the samples will be extracted
`metab_groups`	a character vector - names of the column which has the pathway information of the metabolites(please make sure they are in the same order as the above ones)
`metab_ids`	a character vector - names of the column with metabolite names in the col data matix(please make sure they are in the same order as the above ones)
`cols_for_vis`	choose the colnames from phenotype data that you want to show in the text labels
`phenotype_index`	index of the phenotype data. Can input either the name of the phenotype dataset or the index of the same

Value

a list with two plot objects 1) samples - UMAP plot of the individuals(".$samples") 2) metabs - UMAP plot of the metabolites(".$metabs")

Index