Package: lilikoi (via r-universe)

October 27, 2026

Type Package

Title Metabolomics Personalized Pathway Analysis Tool

Version 2.1.1

Description A comprehensive analysis tool for metabolomics data. It consists a variety of functional modules, including several new modules: a pre-processing module for normalization and imputation, an exploratory data analysis module for dimension reduction and source of variation analysis, a classification module with the new deep-learning method and other machine-learning methods, a prognosis module with cox-PH and neural-network based Cox-nnet methods, and pathway analysis module to visualize the pathway and interpret metabolite-pathway relationships. References: H. Paul Benton <http://www.metabolomics-forum.com/index.php?topic=281.0> Jeff Xia <https://github.com/cangfengzhe/Metabo/blob/master/MetaboAnalyst/website/name_match.R> Travers Ching, Xun Zhu, Lana X. Garmire (2018) <doi:10.1371/journal.pcbi.1006076>.

Depends R (>= 3.5.0)

Imports car, caret, dplyr, gbm, ggplot2, glmnet, h2o, impute, infotheo, limma, M3C, Metrics, MLmetrics, parallel, princurve, pathview, plyr, preprocessCore, pROC, RCy3, reticulate, reshape, RWeka, scales, stats, stringr, survminer, survival

License GPL-2

Encoding UTF-8

RoxygenNote 7.2.1

VignetteBuilder knitr

Suggests knitr, rmarkdown

NeedsCompilation no

Author Xinying Fang [aut], Yu Liu [aut], Zhijie Ren [aut], Fadhl Alakwaa [aut], Sijia Huang [aut], Lana Garmire [aut, cre]

Maintainer Lana Garmire <lana.garmire.group@gmail.com>

Config/pak/sysreqs default-jdk zlib1g-dev

Repository https://cran.r-universe.dev

Date/Publication 2022-10-05 18:00:02 UTC

RemoteUrl https://github.com/cran/lilikoi

RemoteRef HEAD

RemoteSha 1caf63620ae215bfc9fa696b5c96ca8a4191f53f

Contents

lilikoi.explr 2
lilikoi.featuresSelection 3
lilikoi.KEGGplot 4
lilikoi.Loaddata 5
lilikoi.machine_learning 6
lilikoi.meta_path 7
lilikoi.MetaTOpathway 8
lilikoi.PDSfun 9
lilikoi.preproc_knn 9
lilikoi.preproc_norm 10
lilikoi.prognosis 11

Index **13**

`lilikoi.explr` *Exploratory analysis*

Description

Performs source of variation test and build PCA and t-SNE plots to visualize important information.

Usage

```
lilikoi.explr(data, demo.data, pca = FALSE, tsne = FALSE)
```

Arguments

`data`	is a input data frame for analysis with sample ids as row names and metabolite names or pathway names as column names.
`demo.data`	is a demographic data frame with sample ids as row names, sample groups and demographic variable names as column names.
`pca`	if TRUE, PCA plot will be out.
`tsne`	if TRUE, T-SNE plot will be out.

Value

Source of variation test results and PCA and t-SNE plot

Examples

```
# lilikoi.explr(data, demo.data, pca=TRUE, tsne=FALSE)
```

`lilikoi.featuresSelection`

A featuresSelection Function

Description

This function allows you to reduce the pathway diemsion using xxxx

Usage

```
lilikoi.featuresSelection(PDSmatrix, threshold = 0.5, method = "info")
```

Arguments

`PDSmatrix`	from PDSfun function
`threshold`	to select the top pathways
`method`	information gain ("info") or gain ratio ("gain")

Value

A list of top metabolites or pathways.

Examples

```
dt <- lilikoi.Loaddata(file=system.file("extdata",
  "plasma_breast_cancer.csv", package = "lilikoi"))
Metadata <- dt$Metadata
dataSet <- dt$dataSet
# Metabolite_pathway_table=lilikoi.MetaTOpathway('name')
# PDSmatrix= lilikoi.PDSfun(Metabolite_pathway_table)
# selected_Pathways_Weka= lilikoi.featuresSelection(PDSmatrix,threshold= 0.50,method="gain")
```

lilikoi.KEGGplot *lilikoi.KEGGplot*

Description

Visualizes selected pathways based on their metabolites expression data.

Usage

```
lilikoi.KEGGplot(
  metamat,
  sampleinfo,
  grouporder,
  pathid = "00250",
  specie = "hsa",
  filesuffix = "GSE16873",
  Metabolite_pathway_table = Metabolite_pathway_table
)
```

Arguments

metamat	metabolite expression data matrix
sampleinfo	is a vector of sample group, with element names as sample IDs.
grouporder	grouporder is a vector with 2 elements, the first element is the reference group name, like 'Normal', the second one is the experimental group name like 'Cancer'.
pathid	character variable, Pathway ID, usually 5 digits.
specie	character, scientific name of the targeted species.
filesuffix	output file suffix
Metabolite_pathway_table	Metabolites mapping table

Value

Pathview visualization output

Examples

```
dt = lilikoi.Loaddata(file=system.file("extdata","plasma_breast_cancer.csv", package = "lilikoi"))
Metadata <- dt$Metadata
dataSet <- dt$dataSet
# convertResults=lilikoi.MetaTOpathway('name')
# Metabolite_pathway_table = convertResults$table

# data_dir=system.file("extdata", "plasma_breast_cancer.csv", package = "lilikoi")
# plasma_data <- read.csv(data_dir, check.names=FALSE, row.names=1, stringsAsFactors = FALSE)
```

```
# sampleinfo <- plasma_data$Label
# names(sampleinfo) <- row.names(plasma_data)

# metamat <- t(t(plasma_data[-1]))
# metamat <- log2(metamat)
# grouporder <- c('Normal', 'Cancer')
# make sure install pathview package first before running the following code.
# library(pathview)
# data("bods", package = "pathview")
# options(bitmapType='cairo')
 #lilikoi.KEGGplot(metamat = metamat, sampleinfo = sampleinfo, grouporder = grouporder,
  #pathid = '00250', specie = 'hsa',filesuffix = 'GSE16873',
  #Metabolite_pathway_table = Metabolite_pathway_table)
```

`lilikoi.Loaddata` *A Loaddata Function*

Description

This function allows you to load your metabolomics data.

Usage

```
lilikoi.Loaddata(filename)
```

Arguments

`filename`	file name.

Value

A data frame named Metadata.

Examples

```
lilikoi.Loaddata(file=system.file("extdata", "plasma_breast_cancer.csv", package = "lilikoi"))
```

`lilikoi.machine_learning` *A machine learning Function*

Description

This function for classification using 8 different machine learning algorithms and it plots the ROC curves and the AUC, SEN, and specificty

Usage

```
lilikoi.machine_learning(
  MLmatrix = PDSmatrix,
  measurementLabels = Label,
  significantPathways = selected_Pathways_Weka,
  trainportion = 0.8,
  cvnum = 10,
  dlround = 50,
  nrun = 10,
  Rpart = TRUE,
  LDA = TRUE,
  SVM = TRUE,
  RF = TRUE,
  GBM = TRUE,
  PAM = TRUE,
  LOG = TRUE,
  DL = TRUE
)
```

Arguments

`MLmatrix`	selected pathway deregulation score or metabolites expression matrix
`measurementLabels`	measurement label for samples
`significantPathways`	selected pathway names
`trainportion`	train percentage of the total sample size
`cvnum`	number of folds
`dlround`	epoch number for the deep learning method
`nrun`	denotes the total number of runs of each method to get their averaged performance metrics
`Rpart`	TRUE if run Rpart method
`LDA`	TRUE if run LDA method
`SVM`	TRUE if run SVM method

RF	TRUE if run random forest method
GBM	TRUE if run GBM method
PAM	TRUE if run PAM method
LOG	TRUE if run LOG method
DL	TRUE if run deep learning method

Value

Evaluation results and plots of all 8 machine learning algorithms, along with variable importance plots.

Examples

```
dt = lilikoi.Loaddata(file=system.file("extdata","plasma_breast_cancer.csv", package = "lilikoi"))
Metadata <- dt$Metadata
# lilikoi.machine_learning(MLmatrix = Metadata, measurementLabels = Metadata$Label,
# significantPathways = 0,
# trainportion = 0.8, cvnum = 10, dlround=50,Rpart=TRUE,
# LDA=FALSE,SVM=FALSE,RF=FALSE,GBM=FALSE,PAM=FALSE,LOG=FALSE,DL=FALSE)
```

---

lilikoi.meta_path *Metabolite-pathway regression*

Description

Performs single variate linear regression between selected pathways and each of their metabolites. Output the network plot between pathways and metabolties.

Usage

```
lilikoi.meta_path(
  PDSmatrix,
  selected_Pathways_Weka,
  Metabolite_pathway_table,
  pathway = "Pyruvate Metabolism"
)
```

Arguments

PDSmatrix	Pathway deregulation score matrix
selected_Pathways_Weka	Selected top pathways from the featureSelection function
Metabolite_pathway_table	Metabolites mapping table
pathway	interested pathway name

Value

A bipartite graph of the relationships between pathways and their corresponding metabolites.

`lilikoi.MetaTOpathway` *A MetaTOpathway Function*

Description

This function allows you to convert your metabolites id such as names, kegg ids, pubchem ids. into pathways. Metabolites which have not pathways will be excluded from any downstream analysis make sure that you have three database files which are used for exact and fuzzy matching: cmpd_db.rda, syn_nms_db.rda and Sijia_pathway.rda This function was modified version of the name.match function in the below link: https://github.com/cangfengzhe/Metabo/blob/master/MetaboAnalyst/website/name_r

Usage

```
lilikoi.MetaTOpathway(
  q.type,
  hmdb = TRUE,
  pubchem = TRUE,
  chebi = FALSE,
  kegg = TRUE,
  metlin = FALSE
)
```

Arguments

`q.type`	The type of the metabolites id such as 'name', 'kegg', 'hmdb','pubchem'
`hmdb`	if TRUE, match metabolites id to the HMDB database.
`pubchem`	if TRUE, match metabolites id to the PubChem database.
`chebi`	if TRUE, match metabolites id to the ChEBI database.
`kegg`	if TRUE, match metabolites id to the KEGG database.
`metlin`	if TRUE, match metabolites id to the METLIN database.

Value

A table showing the convertion results from metabolites ids to ids in different metabolomics databases and pathway ids and names.

Examples

```
dt <- lilikoi.Loaddata(file=system.file("extdata",
  "plasma_breast_cancer.csv", package = "lilikoi"))
Metadata <- dt$Metadata
dataSet <- dt$dataSet
# Metabolite_pathway_table=lilikoi.MetaTOpathway('name')
```

lilikoi.PDSfun *A PDSfun Function*

Description

This function allows you to compute Pathway Desregulation Score deriving make sure that you have the below database for the metabolites and pathway list: meta_path.RData

Usage

```
lilikoi.PDSfun(qvec)
```

Arguments

qvec	This is the Metabolite_pathway_table from MetaTOpathway function. This table includes the metabolites ids and the its corssponding hmdb ids

Value

A large matrix of the pathway deregulation scores for each pathway in different samples.

References

Nygård, S., Lingjærde, O.C., Caldas, C. et al. PathTracer: High-sensitivity detection of differential pathway activity in tumours. Sci Rep 9, 16332 (2019). https://doi.org/10.1038/s41598-019-52529-3

Examples

```
dt <- lilikoi.Loaddata(file=system.file("extdata",
  "plasma_breast_cancer.csv", package = "lilikoi"))
Metadata <- dt$Metadata
dataSet <- dt$dataSet
convertResults=lilikoi.MetaTOpathway('name')
Metabolite_pathway_table = convertResults$table
# PDSmatrix= lilikoi.PDSfun(Metabolite_pathway_table)
```

lilikoi.preproc_knn *An imputation function.*

Description

This function is used to preprocess data via knn imputation.

Usage

```
lilikoi.preproc_knn(inputdata = Metadata, method = c("knn"))
```

Arguments

inputdata	An expression data frame with samples in the rows, metabolites in the columns
method	The method to be used to process data, including

Value

A KNN imputed dataset with samples in the rows, metabolites in the columns.

Examples

```
dt <- lilikoi.Loaddata(file=system.file("extdata",
  "plasma_breast_cancer.csv", package = "lilikoi"))
Metadata <- dt$Metadata
dataSet <- dt$dataSet
lilikoi.preproc_knn(inputdata=Metadata, method="knn")
```

---

lilikoi.preproc_norm *A Normalization function.*

Description

This function is used to preprocess data via normalization. It provides three normalization methods: standard normalization, quantile normalization and median fold normalization. The median fold normalization is adapted from http://www.metabolomics-forum.com/index.php?topic=281.0.

Usage

```
lilikoi.preproc_norm(
  inputdata = Metadata,
  method = c("standard", "quantile", "median")
)
```

Arguments

inputdata	An expression data frame with samples in the rows, metabolites in the columns
method	The method to be used to process data, including standard normalization (standard), quantile normalization (quantile) and median fold normalization (median).

Value

A normalized dataset with samples in the rows, metabolites in the columns.

Examples

```
dt <- lilikoi.Loaddata(file=system.file("extdata",
 "plasma_breast_cancer.csv", package = "lilikoi"))
Metadata <- dt$Metadata
dataSet <- dt$dataSet
lilikoi.preproc_norm(inputdata=Metadata, method="standard")
```

`lilikoi.prognosis` *Pathway-based prognosis model*

Description

Fits a Cox proportional hazards regression model or a Cox neural network model to predict survival results.

Usage

```
lilikoi.prognosis(
  event,
  time,
  exprdata,
  percent = NULL,
  alpha = 1,
  nfold = 5,
  method = "median",
  cvlambda = "lambda.1se",
  python.path = NULL,
  path = NULL,
  coxnnet = FALSE,
  coxnnet_method = "gradient"
)
```

Arguments

`event`	survival event
`time`	survival time
`exprdata`	dataset for penalization, with id in the rownames and pathway or metabolites names in the column names.
`percent`	train-test separation percentage
`alpha`	denote which penalization method to use.
`nfold`	fold number for cross validation
`method`	determine the prognosis index, "quantile", "quantile" or "ratio".
`cvlambda`	determine the lambda for prediction, "lambda.min" or "lambda.1se".
`python.path`	saved path for python3

path	saved path for the L2cross_nopercent.py and L2cross.py files in lilikoi
coxnnet	if TRUE, coxnnet will be used.
coxnnet_method	the algorithm for gradient descent. Includes standard gradient descent ("gradient"), Nesterov accelerated gradient "nesterov" and momentum gradient descent ("momentum").

Value

A list of components:

c_index	C-index of the Cox-PH model
difftest	Test results of the survival curve difference test
survp	Kaplan Meier plot

Examples

```
# inst.path = path.package('lilikoi', quiet = FALSE) # path = "lilikoi/inst/", use R to run
# inst.path = file.path(inst.path, 'inst')
# python.path = "/Library/Frameworks/Python.framework/Versions/3.8/bin/python3"
# Prepare survival event, survival time and exprdata from your dataset.
# lilikoi.prognosis(event, time, exprdata, percent=NULL, alpha=0, nfold=5, method="median",
#   cvlambda=NULL,python.path=NULL, path=inst.path, python.path=python.path,
#   coxnnet=FALSE,coxnnet_method="gradient")
```

Index

∗ **Load**
    lilikoi.Loaddata, 5
∗ **Match**
    lilikoi.MetaTOpathway, 8
∗ **PDS**
    lilikoi.PDSfun, 9
∗ **classifcation**
    lilikoi.machine_learning, 6
∗ **features**
    lilikoi.featuresSelection, 3
∗ **selection**
    lilikoi.featuresSelection, 3

lilikoi.explr, 2
lilikoi.featuresSelection, 3
lilikoi.KEGGplot, 4
lilikoi.Loaddata, 5
lilikoi.machine_learning, 6
lilikoi.meta_path, 7
lilikoi.MetaTOpathway, 8
lilikoi.PDSfun, 9
lilikoi.preproc_knn, 9
lilikoi.preproc_norm, 10
lilikoi.prognosis, 11