

# Package ‘lilikoï’

January 15, 2021

**Type** Package

**Title** Metabolomics Personalized Pathway Analysis Tool

**Version** 2.0.4

## **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>>

## **biocViews**

**Depends** R (>= 3.5.0)

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

**License** GPL-2

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.0.2

**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>

**Repository** CRAN

**Date/Publication** 2021-01-15 09:40:09 UTC

## R topics documented:

lilikoi.explr . . . . .	2
lilikoi.featuresSelection . . . . .	3
lilikoi.KEGGplot . . . . .	4
lilikoi.Loaddata . . . . .	5
lilikoi.machine_learning . . . . .	6
lilikoi.MetaTOpathway . . . . .	7
lilikoi.meta_path . . . . .	8
lilikoi.PDSfun . . . . .	9
lilikoi.preproc_knn . . . . .	10
lilikoi.preproc_norm . . . . .	10
lilikoi.prognosis . . . . .	11
<b>Index</b>	<b>13</b>

---

lilikoi.explr	<i>Exploratory analysis</i>
---------------	-----------------------------

---

## 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 dimension 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= 0.54,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

<code>metamat</code>	metabolite expression data matrix
<code>sampleinfo</code>	is a vector of sample group, with element names as sample IDs.
<code>grouporder</code>	<code>grouporder</code> 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'.
<code>pathid</code>	character variable, Pathway ID, usually 5 digits.
<code>specie</code>	character, scientific name of the targeted species.
<code>filesuffix</code>	output file suffix
<code>Metabolite_pathway_table</code>	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.MetaT0pathway('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')
#lilikoiloaddata(metamat = metamat, sampleinfo = sampleinfo, grouporder = grouporder,
#pathid = '00250', specie = 'hsa',filesuffix = 'GSE16873',
#Metabolite_pathway_table = Metabolite_pathway_table)
```

---

lilikoiloaddata      *A Loaddata Function*

---

## Description

This function allows you to load your metabolomics data.

## Usage

```
lilikoiloaddata(filename)
```

## Arguments

filename      file name.

## Value

A data frame named Metadata.

## Examples

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

---

`lilikoimachine_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 specificity

### Usage

```
lilikoimachine_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

<code>MLmatrix</code>	selected pathway deregulation score or metabolites expression matrix
<code>measurementLabels</code>	measurement label for samples
<code>significantPathways</code>	selected pathway names
<code>trainportion</code>	train percentage of the total sample size
<code>cvnum</code>	number of folds
<code>dlround</code>	epoch number for the deep learning method
<code>nrun</code>	denotes the total number of runs of each method to get their averaged performance metrics
<code>Rpart</code>	TRUE if run Rpart method
<code>LDA</code>	TRUE if run LDA method
<code>SVM</code>	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.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: `compd_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](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 conversion results from metabolites ids to ids in different metabolomics databases and pathway ids and names.

**Examples**

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

---

`lilikoimeta_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 metabolites.

**Usage**

```
lilikoimeta_path(
  PDSmatrix,
  selected_Pathways_Weka,
  Metabolite_pathway_table,
  pathway = "Alanine, Aspartate And Glutamate 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.PDSfun	<i>A PDSfun Function</i>
----------------	--------------------------

---

**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.

**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)
```

---

`lilikoï.preproc_knn` *An imputation function.*

---

### Description

This function is used to preprocess data via knn imputation.

### Usage

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

---

`lilikoï.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

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

---

lilikoiprognosis      *Pathway-based prognosis model*

---

**Description**

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

**Usage**

```
lilikoiprognosis(
  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.
ifold	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 lilikoiprognosis
coxnnnet	if TRUE, coxnnnet will be used.
coxnnnet_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
diffptest	Test results of the survival curve difference test
survp	Kaplan Meier plot

**Examples**

```
# inst.path = path.package('lilikoiprognosis', quiet = FALSE) # path = "lilikoiprognosis/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.
# lilikoiprognosis(event, time, exprdata, percent=NULL, alpha=0, ifold=5, method="median",
#   cvlambda=NULL, python.path=NULL, path=inst.path, python.path=python.path,
#   coxnnnet=FALSE, coxnnnet_method="gradient")
```

# Index

- \* **Load**
  - lilikoi.Loaddata, 5
- \* **Match**
  - lilikoi.MetaT0pathway, 7
- \* **PDS**
  - lilikoi.PDSfun, 9
- \* **classification**
  - 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, 8  
lilikoi.MetaT0pathway, 7  
lilikoi.PDSfun, 9  
lilikoi.preproc\_knn, 10  
lilikoi.preproc\_norm, 10  
lilikoi.prognosis, 11