Package 'ProbMetab'

October 31, 2013

Type Package

Title Probabilistic annotation of LC-MS based metabolomics

Version 1.0

Date 2013-05-23

Author Ricardo R. Silva, Diego Martinez Salvanha, Ricardo Vencio

Maintainer Ricardo Silva <rsilvabioinfo@usp.br>

Description Provides probability ranking to candidate compounds assigned to masses, with the prior assumption of connected sample and additional previous and spectral information modeled by the user.

License GPL (>= 2)

URL http://labpib.fmrp.usp.br/methods/probmetab/

LazyData true

ByteCompile true

Depends R (>= 2.10), RcppArmadillo, Rcpp, CAMERA, GeneNet, graph,RCytoscape, hwriter, RCurl, XML, multtest, rjson

Suggests mzmatch.R, bootstrap, leaps, mgcv

LinkingTo RcppArmadillo, Rcpp

Collate 'RCreateDataFrame.R' 'combineMolIon.R' 'comp.results.R' 'create.reactionM.R' 'createXGMML.R' 'design.connection.R' 'export.class.table.R' 'export2cytoscape.R' 'formula2mass.R' 'get.Mzmatch.annot.R' 'get.annot.R' 'get.compounds.by.pathway.biocyc.R' 'get.kegg.pathways.R' 'get.name.R' 'get.ncbi.id.R' 'get.pathway.by.organism.biocyc.R' 'get.reactions.by.compound.biocyc.R' 'gibbs.samp.R' 'incorporate.isotopes.R' 'reac2cor.R' 'rt.predict.R' 'sbml2table.R' 'weightM.R' 'weightRT.R' 'get.pathway.by.compound.kegg.R' 'create.pathway.node.attributes.R' 'create.reaction.edge.attributes.R' 'createJSONToCytoscape.R' 'get.pathway.by.organism.kegg.R' 'openGraph.R' 'get.kgml.positions.kegg.R' 'get.formula.kegg.R' 'build.database.kegg.R'

R topics documented:

`build.database.kegg` *build.database.kegg*

Description

Get KEGG compound information needed for exact mass searching and modeling.

Usage

```
build.database.kegg(orgID = NULL)
```

Arguments

orgID	KEGG's organism id. If NULL the function recovers information from all database compounds.

Value

A data.frame with unique id, name, formula and reactions as needed by ProbMetab.

---

combineMolIon *combineMolIon*

Description

This function combines ion annotations in different acquisition modes. It operates in two main modes, combining individual annotations given by get.annot function, using the retention time and mass/charge windows provided by the user or extracting annotations from a peak table provided by CAMERA's combinexsAnnos function.

Usage

```
combineMolIon(antPOS, antNEG, peaklist = NULL,
  cameraobj = NULL, polarity = NULL, rtwin = 5,
  mzwin = 0.05)
```

Arguments

antPOS	positive annotation list given by get.annot.
antNEG	negative annotation list given by get.annot.
peaklist	given by CAMERA's combinexsAnnos function. If this option is chosen the user has to set the acquisition mode to the same as in CAMERA's function, and provide the respective object for downstream analysis.
cameraobj	xsAnnotate object for downstream analysis.
polarity	the same CAMERA's function acquisition mode.
rtwin	retention time window to annotate a peak as present in both acquisition modes.
mzwin	mass to charge ratio window to annotate a peak as present in both acquisition modes.

Value

a list with a matrix of possible molecular ions with a trace of their annotation and the respective xsAnnotate object.

comp.results *comp.results*

Description

Compare two classification tables, given by export.class.table, and reports the difference between two different models.

Usage

```
comp.results(reactionM, w, ansLik, ansConn)
```

Arguments

reactionM	matrix with reactions of each candidate compound.
w	matrix of compound connections.
ansLik	a list of mass to compound assignment, based only on a likelihood.
ansConn	list of mass to compound assignment, with compound's connections contribution.

Value

a list with compound classification classes, table index of classes and a matrix of intensities of selected compounds.

---

create.pathway.node.attributes
create.pathway.node.attributes

Description

This function writes a standard Cytoscape node attribute list (http://www.cytoscape.org/) file. It takes the compound codes and retrieve all the known pathways, where the compound is known to be present. It only works for KEGG, but, a specification of database will be available soon.

Usage

```
create.pathway.node.attributes(classTable, graph, DB,
  filename1, filename2 = NULL, organismId = NULL)
```

Arguments

classTable	classification table created by export.class.table function.
graph	graphNEL object, with node indexes corresponding to mass indexes in classTable.
DB	database with compound names associated to unique ids, used by export.class.table function.
filename1	filename to attribute pathway list file.
filename2	optional filename to attribute pathway list discriminating compound/pathway associations.
organismId	KEGG organism id (http://www.kegg.jp/kegg/catalog/org_list.html) to filter possibibly pathwyas for known pathways for that organism.

Value

writes a standard Cytoscape attribute list to current working directory. Also creates a matrix containing putative compound counting to each pathway.

`create.reaction.edge.attributes`

create.reaction.edge.attributes

Description

This function writes a standard Cytoscape edge attribute list (http://www.cytoscape.org/) file. It takes the the possible mass connections, codified w matrix, retrieve the reactions where the compound is known to be present and associate then to a mass edge. It only works for KEGG, but, a specification of database will be available soon.

Usage

```
create.reaction.edge.attributes(classTable, graph, w,
  reactionM, DB, filename)
```

Arguments

classTable	classification table created by export.class.table function.
graph	graphNEL object, with node indexes corresponding to mass indexes in classTable.
w	matrix of compound connections.
reactionM	data.frame with compound annotation information.
DB	database with compound names associated to unique ids, used by export.class.table function.
filename	reaction attribute list file.

Value

Writes a standard Cytoscape attribute list to the current working directory.

`create.reactionM` *create.reactionM*

Description

This function matches a mass vector against a user provided database, inside an user provided mass tolerance window.

Usage

```
create.reactionM(DB, molIon, ppm.tol)
```

Arguments

`DB`	dataframe containing the mandatory fields id, formula and reactions.
`molIon`	list of annotations provided by get.annot function.
`ppm.tol`	parts per million mass tolerance allowed in the mass search.

Value

A matrix of reactions that each compound candidate, inside the mass window, can participate in the metabolism.

`createJSONToCytoscape` *createJSONToCytoscape*

Description

createJSONToCytoscape converts a graph-like file into a JSON-like file structure. It is mostly used to enable cytoscape.js render a graph.

Usage

```
createJSONToCytoscape(gr, node.label, node.form = NULL,
  edge.form = NULL, saveAsJSONFile = TRUE)
```

Arguments

`gr`	a graphNEL Object.
`node.label`	a vector of node labels (usually in the same node(gr) order).
`node.form`	data.frame file to set node parameters to cytoscape.js. First column should refers to node.names (or node.label) and will be ignored (as we use node.label to set node labels). Every other column must be a logical one (i.e. TRUE/FALSE values per row), and it's name should be separeted by a period (.) (e.g. shape.triangle or color.#FFFF00) One need to mind only valid parameters will be correctly rendered by cytoscape web. Valid parameres are: - width - color

edge.form	data.frame file to set edge parameters to cytoscape.js. First column must refers to edge.names (or edge.label) and it's value should be set as source~target (e.g. 819~821) (same pattern used for RCytoscape edgeNames() function) Every other column must be a logical one (i.e. TRUE/FALSE values per row), and it's name should be separeted by a period (.) (e.g. width.5 or color.#FFFF00) One need to mind: only valid parameters will be correctly rendered by cytoscape web. Valid parameres are: - shape (valid values: rectangle, roundrectangle, ellipse, triangle, pentagon, hexagon, heptagon, octagon) - height - color
saveAsJSONFile	if TRUE (default) saves a file named "network.json" user's current R Working Directory.

Value

createJSONToCytoscape returns a JSON-like string object.

---

createXGMML *createXGMML*

Description

Creates a Cytoscape xgmml readable file.

Usage

```
createXGMML(gr, node.label, cwName = "test",
  node.form = NULL, edge.form = NULL, attributes = NULL,
  path)
```

Arguments

gr	a graphNEL object generated by edge matrices provided by reac2cor.
node.label	a character vector containing the name of each node.
cwName	Cytoscape window name.
node.form	data.frame containing the names of the nodes in the first column, and TRUE/FALSE vectors for graph parameters in the additional columns. The column name should have the parameter type, color and shape and the value in the format "color.#FF0000".
edge.form	data.frame containing the names of the nodes in the first column, and TRUE/FALSE vectors for graph parameters in the additional columns. The column name should have the parameter type, color, width and style, and the value in the format "color.#FF0000".
attributes	classification table created by export.class.table function.
path	system path to save the file.

Value

Cytoscape xgmml readable file.

design.connection *design.connection*

Description

Design a connection matrix w from unique reaction identifiers from experimentally known reactions between candidate compound.

Usage

```
design.connection(reactionM)
```

Arguments

reactionM	a data.frame with compounds and its connections.

Value

A binary matrix w of connections between candidate compound.

export.class.table *export.class.table*

Description

Builds a matrix with the probability for all mass to candidate compounds assignments, by averaging the number of assignments obtained by the gibbs sampler algorithm or ordering the compound candidates with the likelihood matrix.

Usage

```
export.class.table(gibbsL = NULL, reactionM,
  molIon = NULL, probM = NULL, html = FALSE,
  filename = "test", burnIn = 3000, linkPattern = "kegg",
  m.test = "none", class1 = NULL, class2 = NULL,
  norm = FALSE, DB, prob = "count")
```

Arguments

gibbsL	a list of attributions and probabilities from gibbs.samp function.
reactionM	data.frame with compound annotation information.
molIon	non redundant ion annotation.
probM	optionally to gibbsL, a matrix of likelihoods.
html	logical, indicating whether a html file should be generated. This parameter uses the raw data to plot EICs and may be time consuming.

filename	html file name, the default is "test".
burnIn	how many samples of the gibbs sampler should be discarded.
linkPattern	which pattern should be linked to compound id, for now we have implemented "kegg", "pubchem" and "chebi" patterns.
m.test	statistical test to compare mean differences. This option is only available to single acquisition mode analysis, with options "t.test" and "anova".
class1	if the m.test is "t.test" first class to compare in the test, according with xcmsSet phenoData.
class2	if the m.test is "t.test" second class to compare in the test, according with xcmsSet phenoData.
norm	logical, if TRUE performs median normalization from (Anal. Chem. 2011, 83, 5864-5872).
DB	data.frame table used to search compounds, with the field name to be incorporated in the html table.
prob	how to calculate the probability to attribute a mass to a compound. Default is "count", which divide the number of times each identity was was attributed by the number of samples. Optionally the user could choose to use the mean of the probabilities of the identity, "mean".

Value

A list with a matrix "classTable" with attributions and probabilities and indexes of selected masses from xcms peak table.

`export2cytoscape` *export2cytoscape*

Description

Exports a graph to Cytoscape with optional node and edge attributes.

Usage

```
export2cytoscape(gr, node.label, cwName = "test",
  node.form = NULL, edge.form = NULL, cpdInfo = NULL,
  classTable = NULL, pos = NULL)
```

Arguments

gr	a graphNEL object generated by edge matrices provided by reac2cor.
node.label	a character vector containing the name of each node.
cwName	Cytoscape window name.

node.form	data.frame containing the names of the nodes in the first column, and TRUE/FALSE vectors for graph parameters in the additional columns. The column name should have the parameter type, color and shape and the value in the format "color.#FF0000".
edge.form	data.frame containing the names of the nodes in the first column, and TRUE/FALSE vectors for graph parameters in the additional columns. The column name should have the parameter type, color, width and style, and the value in the format "color.#FF0000".
cpdInfo	matrix of pathway information, with pathway code, name and the counting how many nodes each pathway has.
classTable	matrix of classification generated by export.class.table. If the user provides this matrix the information is sent to Cytoscape as node attributes, and the user can visualize it in the data panel.
pos	node position list given by get.kgml.positions function.

Value

Exports a graph to Cytoscape window, with additional parameters.

`formula2mass` *formula2mass*

Description

Calculates the exact mass of a given formula.

Usage

```
formula2mass(formula)
```

Arguments

formula	molecular formula, not allowing characters aside atom's alphabet.

Value

a float representing the sum of monoisotopic masses.

`get.annot` *get.annot*

Description

This function extracts annotation from CAMERA object, generating a matrix of non-redundant putative molecular ions.

Usage

```
get.annot(xsAnnotate, polarity = "positive",
  allowMiss = FALSE, xset = NULL, toexclude = NULL,
  minsamp = 0.6, minint = 5000)
```

Arguments

`xsAnnotate`	CAMERA's annotation object.
`polarity`	acquisition mode of mass spectrometer.
`allowMiss`	logical, optionally retrieves peaks with no evidence of adduct or isotope and annotate them as single charged molecules [M+/-H].
`toexclude`	samples to be excluded of peak counting to non-annotated peak selection.
`xset`	xcmsSet xcms object after missing data replacement, to retrieve SNR to isotopic peaks.
`minsamp`	minimum number of samples in which an ion should be present to be selected.
`minint`	minimum mean intensity that a ion should present to be selected.

Value

A list with a matrix of possible molecular ions with a trace of their annotation, and the used xsAnnotate object.

`get.compounds.by.pathway.biocyc` *get.compounds.by.pathway.biocyc*

Description

Get biocyc compound codes from its API.

Usage

```
get.compounds.by.pathway.biocyc(pathID)
```

Arguments

pathID	code.

Value

character matrix of compound codes, names and formulas.

---

get.formula.kegg *get.formula.kegg*

Description

Get KEGG compound formulas from its API.

Usage

```
get.formula.kegg(x)
```

Arguments

x	KEGG compound code.

Value

character formula.

---

get.kegg.pathways *get.kegg.pathways*

Description

Get pathway map links to all kegg pathways that have more than numCpds as nodes.

Usage

```
get.kegg.pathways(cpds, numCpds)
```

Arguments

cpds	KEGG compound ids, including "cpd:".
numCpds	minimum number of compounds as nodes of the pathway.

Value

list of the missing compounds of kegg pathways, and a table with map links and statistics.

get.kgml.positions.kegg *get.kgml.positions.kegg*

Description

Gets KEGG's pathway code, download the kgml file, and retrieves the pathway layout.

Usage

```
get.kgml.positions.kegg(path)
```

Arguments

path	KEGG's pathway code.

Value

a list with an adjacency matrix for pathway nodes, and a matrix with node positions.

get.Mzmatch.annot *get.Mzmatch.annot*

Description

This function extracts annotation from mzMatch PeakML file, generating a matrix of non-redundant putative molecular ions.

Usage

```
get.Mzmatch.annot(filename, onlyBP = TRUE)
```

Arguments

filename	PeakML file containing ion annotation.
onlyBP	logical, if TRUE retrieves only PeakML "bp" relationship, if FALSE also retrieves "potential bp" relationship.

Value

A list with a matrix of possible molecular ions with a trace of their annotation, and the used xsAnnotate object.

`get.name` *get.name*

Description

Get KEGG object names from its API.

Usage

```
get.name(x)
```

Arguments

x	KEGG code.

Value

character name.

`get.ncbi.id` *get.ncbi.id*

Description

Transtates KEGG's enzyme code to NCBI gene id.

Usage

```
get.ncbi.id(x)
```

Arguments

x	KEGG's enzime code.

Value

a character code of NCBI gene entry.

`get.pathway.by.compound.kegg` *get.pathway.by.compound.kegg*

Description

Takes KEGG id code and retrieves the pathways that the compound is possible involved.

Usage

```
get.pathway.by.compound.kegg(keggId)
```

Arguments

`keggId`	KEGG's compound id.

Value

A string with the pathways collapsed by ";" character.

`get.pathway.by.organism.biocyc` *get.pathway.by.organism.biocyc*

Description

Get biocyc pathway codes from its API.

Usage

```
get.pathway.by.organism.biocyc(orgID)
```

Arguments

`orgID`	code.

Value

character vector of pathway codes.

`get.pathway.by.organism.kegg` *get.pathway.by.organism.kegg*

Description

Get KEGG pathway codes from its API.

Usage

```
get.pathway.by.organism.kegg(organismId)
```

Arguments

`organismId`	KEGG's organism id.

Value

A vector of KEGG pathway codes.

`get.reactions.by.compound.biocyc` *get.reactions.by.compound.biocyc*

Description

Get biocyc reaction codes from its API.

Usage

```
get.reactions.by.compound.biocyc(cpdID)
```

Arguments

`cpdID`	code.

Value

character vector of reaction concatenated by ";" character.

gibbs.samp *gibbs.samp*

Description

Call a C++ function that runs a Gibbs Sampler algorithm to sample from the distribution of metabolite to compound attribution with the previous assumption that the connected combination of attributions makes more sense.

Usage

```
gibbs.samp(x, y, N, w, p)
```

Arguments

x	a vector of masses (unique from mass/retention time pairs).
y	a vector of candidate compounds for each mass.
N	number of iterations to sample.
w	matrix of compound connections.
p	matrix of likelihood probabilities.

Value

A list of matrices with attribution indexes and probabilities.

incorporate.isotopes *incorporate.isotopes*

Description

Calculates the theoretical pattern of first 13C isotope for each candidate formula.

Usage

```
incorporate.isotopes(ionAnnot, reactionM, comb = NULL,
  polarity = NULL, var = 2, samp = NULL, DB)
```

Arguments

ionAnnot	annotation list from get.annot function.
reactionM	compound's reaction matrix.
comb	1 for acquisition mode combination.
polarity	acquisition mode polarity.
var	1 to use standard mean/sd estimators to carbon number prediction, 2 for median/mad estimators.
samp	sample indexes, other than blanks, controls and QCs, according to xcms's phenoData.
DB	data.frame of compound information, with chemical formula.

Value

matrix of candidate compound theoretical isotope patterns.

KEGGcpds *Compounds of KEGG database*

Description

This is a illustrative database format for ProbMetab package, with the mandatory fields id, formula and reactions.

Usage

```
KEGGcpds
```

Format

A data.fram containing 14014 compound entries.

Source

http://www.kegg.jp/.

References

Kanehisa, Minoru and Goto, Susumu (2000) KEGG: kyoto encyclopedia of genes and genomes Nucleic acids research. 28(1):27-30.

NIST_relative_atomic_mass *Most abundant atomic mass From NIST database.*

Description

This is a periodic table used by ProbMetab::formula2mass to calculate the exact mass of a given molecular formula.

Usage

```
NIST_relative_atomic_mass
```

Format

A data.fram containing 118 atomic masses.

Source

http://www.nist.gov/pml/data/comp.cfm.

References

The National Institute of Standards and Technology (NIST) is an agency of the U.S. Department of Commerce.

openGraph *openGraph*

Description

Send the JSONFile to http://labpib.fmrp.usp.br/ web server and to visualize a dynamic graph in the web browser.

Usage

```
openGraph(JSONFile, classTable = NULL,
  openBrowser = FALSE)
```

Arguments

JSONFile	a file generated by createJSONToCytoscape function.
classTable	classification table generated by export.class.table function.
openBrowser	logical if FALSE, returns a URL link, if TRUE opens the web browser to access graph visualization.

RCreateDataFrame *RCreateDataFrame*

Description

Create a data frame with compound information from mzMatch open source project format.

Usage

```
RCreateDataFrame(file)
```

Arguments

file	xml input file.

Value

a data frame with compound's id, name and formula.

reac2cor *reac2cor*

Description

Use the intensity of putative molecules in repeated samples to calculate correlations and partial correlation in a user defined threshold of false discovery rate for significance testing. After the correlation test the function also overlay significant correlations with all putative reactions between two masses.

Usage

```
reac2cor(mw, classTable, opt = "cor", corths = 0.75,
  corprob = 0.8, pcorprob = 0.8)
```

Arguments

mw	two column of adjacency matrix indexes connecting compounds by reactions.
classTable	classification table, with intensities for repeated samples.
opt	correlation option, "cor" for correlation, and "pcor" for partial correlation.
corths	correlation intensity threshold.
corprob	probability that the correlation is considered significant.
pcorprob	probability that the partial correlation is considered significant.

Value

A list of estimated correlations and reactions.

`rt.predict` *rt.predict*

Description

Predict retention time from a given training set, and a matrix of compound descriptors.

Usage

```
rt.predict(testSet, predSet, descData = NULL,
  descList = NULL, columnPar = NULL, voidTime = NULL)
```

Arguments

`testSet`	is a training set provided by user.
`predSet`	is a set of candidate compound to have their times predicted.
`descData`	is the user provided list of matrices of compound descriptors.
`descList`	is the optional user provided list of descriptors name calculated by rcdk package.
`columnPar`	is the optional user provided list of column parameters.
`voidTime`	column "dead" time.

Value

A list of prediction parameters.

`sbml2table` *sbml2table*

Description

Retrieves compound and its reactions from SBML models.

Usage

```
sbml2table(file)
```

Arguments

`file`	SBML file.

Value

A database in the format required for ProbMetab, with mandatory fields id, formula and reactions.

weightM *weightM*

Description

Builds a c (number of compounds) by m (number of masses) matrix of compound likelihoods with the gaussian error function erf, based on mass accuracy and optionally on isotopic patterns.

Usage

```
weightM(isoPatt, useIso = TRUE, intervals = NULL,
  offset = NULL, massWeigth = 1, likelihood = "erfc",
  precision = 1)
```

Arguments

isoPatt	is the likelihood data.frame generated by incorporate.isotopes function, or reactionM the data.frame of compound's information.
useIso	logical indicating whether to use or not isotope information in the likelihood.
intervals	a vector of SNR numerical intervals, to which different carbon offset should be added to predicted C-number.
offset	vector of empirically estimated carbon offset to be added to predicted C-number.
massWeigth	is the contribution parameter of the probabilistic model.
likelihood	which noise model to use, "erfc" to complementary error function, or "gaussian" to standard gaussian with two sd corresponding to the given p.p.m precision.
precision	equipment mass accuracy, usually the same used in exact mass search.

Value

A matrix wm of likelihood weights.

weightRT *weightRT*

Description

Builds a c (number of compounds) by m (number of masses) matrix of weights

Usage

```
weightRT(rtObj, reactionM, userCuttoff = 0.95,
  rtWeigth = 0.1, plot = FALSE)
```

Arguments

rtObj	is a list of fitted values predicted by rt.predict.
reactionM	is the KEGG search list.
userCuttoff	is the user set expectation of error acceptance.
rtWeigth	is penalty to errors that fall bellow the threshold given b user.
plot	logical, wheter or not do plot the output of the function.

Value

A matrix wrt of likelihood.

Index