Package 'RcometsAnalytics'

March 26, 2024

Type Package

Title Comets Analytics for consortium based metabolomic analyses

Version 2.9.0.28

Author Ella Temprosa, Ewy Mathe

Maintainer Bill Wheeler <wheelerb@imsweb.com>

Description This R package support all cohort-specific analyses of the COMETS consortium. Data are not saved in the system but output must be downloaded and submitted for meta-analyses. This package can be used in several ways: supports the CBIIT implementation of comets-analytics, local shiny based add-in called shinycomets, or console based analysis. See example vignette. The version of the package is noted as: level 1 and 2 reflect the comets-analytics version, 3rd level reflects the number of major revision and 4th level for bug fixes.

Depends R (>= 3.5.0)

Imports readxl, rio, dplyr, plyr, plotly, tidyr, heatmaply, stringr, data.table, caret, subselect, broom, psych, MASS, ppcor, survival

Suggests Hmisc, knitr, testthat, rmarkdown, RaMP

VignetteBuilder knitr

License GPL-3

RoxygenNote 7.3.1

NeedsCompilation no

URL http://comets-analytics.org/

BugReports https://github.com/CBIIT/nci-webtools-comets-analytics/issues

R topics documented:

`RcometsAnalytics-package`

RcometsAnalytics R package

Description

This R package supports all cohort-specific analyses of the COMETS consortium https://www.comets-analytics.org/. Data are not saved in the system but output must be downloaded and submitted for meta-analyses.

Details

Functions for analysis:
runModel (correlation, glm, lm, coxph, or clogit)
runAllModels (run models in batch mode from models sheet)
Functions for graphics:
plotVar (metabolite variance distribution plot)
plotMinvalues (distribution of minimum or missing values)
showHeatmap (heat map of metabolite correlations)
showHClust (interactive heat map with hierarchical clustering)
Functions for saving results to files:
OutputCSVResults (write to .csv file)
OutputXLSResults (write to excel file)
OutputListToExcel (write list of data frames to excel file with multiple sheets)
Functions for meta-analysis:
runMeta (run a single meta-analysis)
runAllMeta (run multiple meta-analyses)
meta_calc (main calculation function for meta-analysis)

`checkIntegrity` *Checks integrity of sheets in the user input CSV file*

Description

Checks integrity of sheets in the user input CSV file

Usage

```
checkIntegrity(
  dta.metab,
  dta.smetab,
  dta.sdata,
  dta.vmap,
  dta.models,
  dict_metabnames,
  dta.op
)
```

Arguments

`dta.metab`	dta.metab
`dta.smetab`	dta.smetab
`dta.sdata`	dta.sdata
`dta.vmap`	dta.vmap
`dta.models`	dta.models
`dict_metabnames`	dict_metabnames
`dta.op`	dta.op

`chemClassEnrichment` *Chemical Class Enrichment*

Description

Chemical Class Enrichment

Usage

```
chemClassEnrichment(df, metabdata, db.version = NULL, pvalue.adj = 0.05)
```

Arguments

`df`	The "Effects" data frame returned from function `runModel`
`metabdata`	Return object from `readCOMETSinput`
`db.version`	The RaMP SQLite database version to use. The default is the most recent version.
`pvalue.adj`	The maximum adjusted p-value for a metabolite to be included in the analysis.

Details

The **RaMP** R package must be installed prior to calling this function. An attempt to load RaMP will be done through the `requireNamespace` function.

The set of all possible metbolite id types are: LIPIDMAPS, pubchem, hmdb, chemspider, chebi, CAS, wikidata, swisslipids, kegg, lipidbank, plantfa, and kegg_glycan. The **METABOLITES** sheet in the input Excel file will need to have at least one of the above id types as a column (case-insensitive). If the **METABOLITES** sheet contains more than one id type, then the metabolite ids passed into `chemicalClassEnrichment` depends on the order of the id type columns. For example, if the **METABOLITES** sheet contains columns HMDB (column F) and pubchem (column C), then the metbolite ids will taken from the pubchem column and any missing ids from the pubchem column will be assigned from the HMDB column.

Value

A data frame

Examples

```
dir <- system.file("extdata", package="RcometsAnalytics", mustWork=TRUE)
csvfile <- file.path(dir, "cometsInputAge.xlsx")
exmetabdata <- readCOMETSinput(csvfile)
modeldata <- getModelData(exmetabdata,exposures="age", modelspec="Interactive",
outcomes=c("lactose","lactate"))
obj <- runModel(modeldata,exmetabdata, cohortLabel="DPP")
# ret <- chemClassEnrichment(obj$Effects, exmetabdata, pvalue.adj=1)
```

clogit.options *options list for clogit*

Description

A list of 2:

- `method` One of: "exact", "approximate", "efron", "breslow". The default is "exact".
- `weights` A variable name to specify weights. The default is NULL.

Examples

```
model.options <- list(method="efron", weights="weightVarInData")
```

correlation.options *options list for* `model="correlation"`

Description

A list of 1:

- `method` Correlation method to use. It must be one of "spearman", "pearson", "kendall". The default value is "spearman".

Examples

```
model.options <- list(method="pearson")
```

coxph.options *options list for coxph*

Description

A list of 6:

- `ties` One of: "efron", "breslow", "exact". The default is "efron".
- `robust` TRUE or FALSE for computing a robust covariance matrix). The default is FALSE.
- `weights` A variable name to specify weights. The default is NULL.
- `singular.ok` See coxph. The default is TRUE.
- `Surv.type` See the `type` option in Surv. The default is NULL.

Examples

```
model.options <- list(robust=TRUE, weights="weightVarInData")
```

`Effects` *Effects table*

Description

The Effects data frame contains the estimates for each exposure, and will contain multiple rows for categorical exposure variables. Depending on the model run and options specified, all the below names may not appear in the data frame.

- `corr` The correlation between `term` and the outcome
- `estimate` The regression coefficient of `term`
- `estimate.lower` The lower confidence limit for `term`
- `estimate.upper` The upper confidence limit for `term`
- `exp.estimate` The exponentiated regression coefficient of `term`
- `exp.estimate.lower` The exponentiated lower confidence limit for `term`
- `exp.estimate.upper` The exponentiated upper confidence limit for `term`
- `exp.std.error` The standard error of `exp.estimate` from the delta method
- `exposurespec` Exposure variable
- `model` Model label from `getModelData`
- `model_number` Model number used in `runAllModels`
- `outcomespec` Outcome variable
- `pvalue` The p-value of the test
- `pvalue.adj` Adjusted p-values for exposure variables only
- `run` Run number that can be used to link with the `ModelSummary` table
- `statistic` The test statistic for `term`
- `std.error` The standard error of `estimate`
- `stratavar` Stratum variable(s)
- `strata` Stratum level(s)
- `term` Variable in the model

Details

Missing values will appear if a model did not converge, produced an error, or not run because of too many missing values in the outcome. Adjusted p-values are only computed for models with a single outcome variable or single exposure variable.

Errors_Warnings *Errors and Warnings table*

Description

Columns in the Errors_Warnings table.

- `type` WARNING or ERROR
- `object` The object that produced the warning or error. This is typically a variable or a particular stratum.
- `message` Message describing the warning or error
- `model` Model label from `getModelData`
- `model_number` Model number used in `runAllModels`

Details

The kinds of warnings and errors stored in this matrix are ones that apply to all models or all outcomes for an exposure variable. An error message for a particular exposure-outcome pair will be stored in the `message` column of the `ModelSummary` table.

file.list *list to describe or transform a file for meta-analysis*

Description

A list of 20:

- `file` The complete path to the file. Must be specified.
- `cohort` The name of the cohort. Use this option to set or change the name of the cohort. Must be specified for files not obtained from version 3.0 of RcometsAnalytics.
- `sep` The file delimiter. For example "," for csv files, "\t" for tab-delimited files. The default is NULL.
- `correlation` 0 or 1 for correlation results. The default is 0, so that the file is assumed to be from a non-correlation model.
- `estimate.col` Column name for the beta estimate or correlation. The default value is "estimate".
- `se.col` Column name for the standard error. Needed for non-correlation models. The default value is "std.error".
- `nobs.col` Column name for the number of subjects. If such a column does not exist in the file, set `nobs` to a value. The default is "nobs".
- `nobs` Number of observations. Must be specified if `nobs.col` is not specified. The default is NULL.
- `outcome.col` Column name for the outcome. If such a column does not exist in the file, then set `outcome.name`. The default is "outcome_uid".
- `outcome.name` The name of the outcome variable. Must be specified if `outcome.col` is not specified. No default.

- `exposure.col` Column name for the exposure. If such a column does not exist in the file, then set `exposure.name`. The default is "exposure_uid".
- `exposure.name` The name of the exposure variable. Must be specified if `exposure.col` is not specified. No default.
- `stratavar.col` Column name for the stratification variable. The default is NULL.
- `strata.col` Column name for the stratification values. The default is NULL.
- `strata.name` The name of the stratification variable. Use this option to change the name of the stratification variable or to add a stratification variable to the file. When adding a stratification variable, set `strata.value` also. The default is NULL.
- `strata.value` The constant value of the stratification. The default is NULL.
- `model.col` Column name for the model. The default is NULL.
- `model.name` The name of the model. Use this option to change or add the model name to the results. The default is NULL.
- `change.col.values` A list of sublists, where each sublist has elements "col", "old", and "new". Use this option to change the values of a (exposure or stratification) column to new values. For example, list(list(col="smoke", old=c("never", "current", "former"), new=c(0, 1, 2))) will change the values "never" to 0, "current" to 1, and "former" to 2 in the "smoke" column. The default is NULL.
- `where` Vector of strings with a variable name, a comparison operator (e.g. "<", ">", "<=", ">=", "!=", "="), and a value. The strings are combined by the logical AND (&) operator. For example, `where = c("study = A", "age > 50")` uses all subjects with age > 50 and in study A. This option would primarily be used if the file contained results from multiple cohorts and each cohort needed to be included in the meta-analysis.
- `new.ref.value` The new reference value for a categorical. exposure variable. This option will update the Effects data frame based on the new reference value. Note that the updated p-values will be based on a Wald test regardless of the original test. This option is applied before the option `change.col.values`. This option is only for results files from non-correlation models obtained using version 3.0 of RcometsAnalytics. The default is NULL.

Details

For any type of file, the parameter `file` must be specified. For files not obtained from version 3.0 of RcometsAnalytics, other parameters may need to be specified. This list can be used with the functions `runMeta` and `meta_transform`.

Examples

```
# Suppose \code{f} is a file containing results from a linear regression model with
#  the metabolites as the outcomes in column called "metabolite", and the exposure
#  was a variable called "age" which is not a column in \code{f}. Also, 572 subjects
# were included in the analysis and the cohort name is "study_XYZ".
f <- "path_to_file"
file.list <- list(file=f, sep=",", cohort="study_XYZ", outcome.col="metabolite",
                      exposure.name="age", nobs=572)
```

`filterCOMETSinput` *Function that subsets input data based on "where variable"*

Description

Function that subsets input data based on "where variable"

Usage

```
filterCOMETSinput(readData, where = NULL)
```

Arguments

`readData`	list from readComets or readData$subjdata
`where`	users can specify which subjects to perform the analysis by specifying this parameter. 'where' expects a vector with a variable name, a comparison operator ("<", ">", "=","<=",">="), and a value. For example, "where = c("Gender=Female")".

Value

filtered list

`fixData` *Fixes input data (e.g. takes care of factors, and other data frame conversions)*

Description

Fixes input data (e.g. takes care of factors, and other data frame conversions)

Usage

```
fixData(dta, compbl = FALSE)
```

Arguments

`dta`	any data frame
`compbl`	compress multiple blank spaces to single blank space for all character or factor variables in the dataset

getModelData *Prepares data for the models to be run as specified in the input. Can be run in interactive or batch mode. Each model is checked for validity (correlation between predictors, zero variance, etc.).*

Description

Prepares data for the models to be run as specified in the input. Can be run in interactive or batch mode. Each model is checked for validity (correlation between predictors, zero variance, etc.).

Usage

```
getModelData(
  readData,
  modelspec = "Batch",
  modlabel = "",
  outcomes = "All metabolites",
  exposures = "",
  adjvars = NULL,
  strvars = NULL,
  wgtvar = NULL,
  offvar = NULL,
  timevar = NULL,
  groupvar = NULL,
  where = NULL,
  exposurerefs = NULL
)
```

Arguments

readData	List from readCOMETSinput
modelspec	How model is specified (Interactive or Batch). The default is Batch
modlabel	If batch, chosen model specified by batch mode (the MODEL column in the Models sheet). If interactive, then the model label.
outcomes	If Interactive, a vector of outcome variables (see details), the default is All metabolites)
exposures	If Interactive, a vector of exposure variables (see details)
adjvars	If Interactive, a vector adjustment covariates (see details)
strvars	If Interactive, stratification covariates (see details)
wgtvar	If Interactive, a variable of weights (see details)
offvar	If Interactive, an offset variable (see details)
timevar	If Interactive, time variable(s) for survival models (see details)
groupvar	If Interactive, a group variable for conditional logistic models (see details)
where	users can specify which subjects to perform the analysis on by specifying this parameter. 'where' expects a vector of strings with a variable name, a comparison operator (e.g. "<", ">", "<=", ">=", "!=", "="), and a value. For example, where = c("age>50","bmi > 22") uses all subjects with age > 50 AND bmi > 22. Note that when running in Batch mode, rules in the WHERE column of the Models sheet must be separated by a comma.

`exposurerefs` If Interactive, a vector of exposure reference levels for categorical exposure variables. If specified, then this vector must have the same length as `exposures`.

Details

All metabolite variables specified should be listed in the `metabolite_name` column of the `Metabolites` sheet of the Excel file. All non-metabolite variables should be listed in the `VARREFERENCE` column of the `VarMap` sheet. The `wgtvar`, `offvar`, and `timevar` are only used for specific models. See the `model` option in `options`.

Value

a list comprising:
1: subset data: gdta
2: exposure variables: ccovs
3: outcome variables: rcovs
4: adjustment variables: acovs
5: stratification variable: scovs
6: model specification: modspec
7: model label: modlab
8: whether all metabolites vs all metabolites is run: allvsall
9: weight variables: wgtcov
10: offset variables: offcov

Examples

```
dir <- system.file("extdata", package="RcometsAnalytics", mustWork=TRUE)
csvfile <- file.path(dir, "cometsInputAge.xlsx")
exmetabdata <- readCOMETSinput(csvfile)
modeldata <- getModelData(exmetabdata,modlabel="1 Age")
```

---

glm.options *options list for glm*

Description

A list of 6:

- `family` One of: "binomial", "gaussian", "Gamma", "inverse.gaussian", "poisson", "quasi", "quasibinomial", "quasipoisson". The default is "gaussian".
- `link` NULL or a string for the link function to use (see `family`). The default is to use the canonical link for `family`.
- `weights` A variable name to specify weights. The default is NULL.
- `offset` A variable name to specify an offset. The default is NULL.
- `control` See `glm`. The default is `glm.control`.
- `singular.ok` See `glm`. The default is TRUE.

Examples

```
model.options <- list(family="binomial", weights="weightVarInData")
```

`Harmonize` *Harmonizes metabolites by looking up metabolites names from user input and finding the corresponding COMETS harmonized name.*

Description

Harmonizes metabolites by looking up metabolites names from user input and finding the corresponding COMETS harmonized name.

Usage

```
Harmonize(dtalist)
```

Arguments

`dtalist`	results of reading a CSV data sheet (with read_excel)

`lm.options` *options list for lm*

Description

A list of 4:

- `weights` A variable name to specify weights. The default is NULL.
- `offset` A variable name to specify an offset. The default is NULL.
- `tol` See `lm`. The default is 1e-7.
- `singular.ok` See `lm`. The default is TRUE.

Examples

```
model.options <- list(weights="weightVarInData")
```

`mergeResultsFiles` *This function will combine output files assumed to be from the same cohort and same model.*

Description

This function will combine output files assumed to be from the same cohort and same model.

Usage

```
mergeResultsFiles(filevec, out.file, op = NULL)
```

Arguments

filevec	Character vector of files that contain the output files from runModel or runAllModels. Valid file extensions are ".xlsx", and ".rda".
out.file	NULL or a file to save the results. The file extension must be either ".xlsx", or ".rda".
op	A list containing the options to use or NULL. The options are precedence and check.consistency with default values of "nobs" and TRUE respectively (see details). The default is NULL.

Details

This function will merge output files containing results from the same model and same cohort. Within a file, metabolites with missing results (estimate and std.error) will be excluded. The files can have different and overlapping metabolites. When there is an overlapping metabolite among the files, the option precedence="nobs" will choose the file with the largest number of subjects for that metabolite. With precedence="data", the first file in filevec that contains the metabolite will be chosen. The option check.consistency will check for the same outcome, exposure, and stratification variable names across the files.

Value

List of data frames containing the merged results

---

metaOutputColumns *Meta Analysis Output Columns*

Description

The Results data frame contains the estimates from the meta-analyses. Depending on the model run and options specified, all the below names may not appear in the data frame returned from runMeta.

- outcome_uid The harmonized outcome variable
- exposure_uid The harmonized exposure variable
- term For a continuous exposure, this will be the same as exposure_uid. For a categorical exposure, it will be the dummy variable for the exposure.
- n.cohort The number of included cohorts for the estimates.
- n.sub Total number of subjects.
- fixed.pvalue P-value for the fixed-effects model.
- random.pvalue P-value for the random-effects model.
- fixed.estimate Estimate for the fixed-effects model.
- fixed.std.error Estimated standard error of fixed.estimate. This will not appear for correlation models.
- random.estimate Estimate for the random-effects model.
- random.std.error Estimated standard error of random.estimate. This will not appear for correlation models.
- fixed.estimate.L Lower 95 percent confidence limit for fixed.estimate.
- fixed.estimate.U Upper 95 percent confidence limit for fixed.estimate.

- random.estimate.L Lower 95 percent confidence limit for random.estimate.
- random.estimate.U Upper 95 percent confidence limit for random.estimate.
- het.pvalue P-value for Cochran's Q test of heterogeneity.
- stratavar Stratum variable(s)
- strata Stratum level(s)
- strata.fixed.het.pvalue P-value for Cochran's Q test of heterogeneity across the strata using the fixed-effects meta-analysis estimates.
- strata.fixed.het.df Degrees of freedom for the above test.
- strata.random.het.pvalue P-value for Cochran's Q test of heterogeneity across the strata using the random-effects meta-analysis estimates.
- strata.random.het.df Degrees of freedom for the above test.

The Duplicates data frame contains the meta-analysis results for duplicated metabolites. It contains the same columns as the Results data frame plus the columns run, outcomespec, and exposurespec which are useful for differentiating the duplicated metabolites.

Details

Correlation models will not contain columns for fixed.std.error and random.std.error. Instead, there will be columns for lower and upper confidence limits of fixed.estimate and random.estimate. The meta-analysis options add.cohort.names and add.cohort.cols will also add cohort specific columns to the output (see meta_options).

---

meta_calc *Main function that performs the meta-analysis calculations.*

Description

Main function that performs the meta-analysis calculations.

Usage

```
meta_calc(beta, se)
```

Arguments

beta	matrix or vector of betas. If a matrix, then the rows represent the metabolites and the columns represent the cohorts.
se	matrix or vector of the standard errors for beta. This object must be in the same order and have the same dimension as beta.

Value

List containing the results from fixed-effect and random-effect meta-analyses along with Cochran's Q test for heterogeneity.

meta_opfile *Options file for meta-analyses*

Description

An Excel file containing models and options for the runAllMeta function

Details

The file can contain sheets **META_MODELS** and **META_TYPES**. Each sheet is optional. The **META_MODELS** sheet should have a column called MODEL containing the models that the user wants to run meta-analyses for. If the **MODELS** sheet is not given then meta-analyses for all models will be run. This sheet can also have an optional column called META_TYPE that links to the META_TYPE column in the **META_TYPES** sheet to define the options for each meta-analysis model. The **META_TYPES** sheet should contain columns META_TYPE, OPTION, and VALUE. See the example file in /inst/extdata/cometsMetaInput.xlsx. If the **META_TYPES** sheet is not specified, then the default values for all meta-analysis options will be used.

meta_options *Meta-analysis options*

Description

A list of options for the runMeta function

Details

- min.n.cohort Minimum number of cohorts to include when meta-analyzing each metabolite. The default is 2.
- min.nsub.cohort Minimum number of subjects in a cohort for a metabolite to be included from that cohort. The default is 25.
- min.nsub.total Minimum number of subjects in all cohorts for a metabolite to be meta-analyzed. The default is 50.
- cohorts.include Character vector of cohorts to include. The default is NULL.
- cohorts.exclude Character vector of cohorts to exclude. The default is NULL.
- output.type Type of output file, either "xlsx" for an Excel worksheet or "rda" for an R object file created with the save() function. The default is "xlsx".
- strata.exclude.het.test A list of stratification levels to be excluded from the test for heterogeneity. This list has the form list(var1=vec1, var2=vec2, ...), where var1, var2, ... are stratification variables, and vec1, vec2, ... are vectors of stratification levels to be removed from the test. The default is NULL.
- stopOnFileError TRUE or FALSE to stop processing when a problem with any one of the input files is encountered. If FALSE, then the files containing errors will be removed from the analysis. The default is TRUE.
- cohorts.indep Vector of cohort names that correspond to multiple input files, where each file will be treated as an independent cohort.

- `cohorts.merge` Vector of cohort names where each of these cohorts have multiple files that will be merged together. This will take the union of all metabolites in the files. A metabolite that occurs in multiple files will be chosen from the file with the maximum sample size for that metabolite.
- `add.cohort.names` TRUE or FALSE to add binary columns (one for each cohort) to the data frame of results to show which cohorts contributed to the meta-analysis The value will be 1 if the cohort contributed to the result or 0 if the cohort did not. The default is TRUE.
- `add.cohort.cols` Vector of column names to add to the output for each cohort. These must be column names from the ModelSummary or Effects tables in the meta-analysis input files. For example, if `add.cohort.cols = "pvalue"`, then the p-values from each cohort will be added to the output, and the columns will be of the form <cohort name>.pvalue. A missing value will appear if the cohort did not contribute to the result. The default is NULL.

`meta_transform` *Function for transforming a set of results data*

Description

Function for transforming a set of results data

Usage

```
meta_transform(file.obj, out.file = NULL)
```

Arguments

`file.obj`	list of type `file.list`
`out.file`	NULL or the name of an output file to save the results. The file extension must be ".xlsx" or ".rda".

Details

This function will transform or update a file of model results so that the results can be used for a meta-analysis. The types of changes that can be made are:
1. Column values can be modified. For example, if the smoking status values 'never', 'current', and 'former' need to be changed to 0, 1, 2.
2. A column with a constant value can be added. For example, a column with the number of subjects used in the analysis can be added. Or for instance, for a file from an all-female study, a stratification column containing the value 'female' can be added.
3. The file can be subsetted if it contains results from multiple models or multiple cohorts.
4. The results can be updated to reflect a different reference value of a categorical exposure variable.
5. The name of the cohort can be renamed.
6. The name of the model can be renamed.

Value

A list of objects with names `ModelSummary`, `Effects`, `Errors_Warnings`, `Table1`, `Info`.

ModelSummary *ModelSummary table*

Description

The ModelSummary data frame contains one row of model summary results for each exposure/outcome combination. Depending on the model run and options specified, all the below names may not appear in the data frame.

- adjspec Original adjustment variables specified
- adjvars Adjustment variables included in the model
- adjvars.removed Adjustment variables removed from the model
- adj_uid Adjustment variable universal ids
- adj.r.squared Adjusted R-squared
- aic Akaike information criterion
- bic Bayesian information criterion
- cohort String passed into runModel
- converged TRUE or FALSE for model convergence
- deviance Deviance of the fitted model
- df.null NULL model degrees of freedom
- df.residual Residual degrees of freedom
- exposure Exposure variable
- exposure_uid Exposure universal id
- exposurespec Exposure variable
- loglik Log-likelihood of the fitted model
- message Error message produced from the modeling function
- model Model label from getModelData
- model_function Model function used in runModel
- model_number Model number used in runAllModels
- nobs Number of observations used
- null.deviance Deviance of the NULL model
- outcome Outcome variable
- outcomespec Outcome variable
- outcome_uid Outcome universal id
- run Run number that can be used to link with the Effects table
- runmode "Batch" or "Interactive"
- r.squared R-squared, the fraction of variance explained by the model
- sigma Square root of the estimated variance of the random error
- stratavar Stratum variable(s)
- strata Stratum level(s)
- term Variable in the model

- `wald.pvalue` P-value from the Wald test of the exposure variable. Note that this test may be a multi-df test if the exposure is categorical.
- `wald.pvalue.adj` Adjusted p-values of `wald.pvalue` for exposure variables only
- `exposure.covariances` String of exposure reference level, dummy variable names and covariances for a categorical exposure variable in a non-correlation model. This covariances can be used for changing the reference level of the categorical exposure variable. The string has the form: "ref|v1,v2,...,vm|cov(v1,v2),cov(v1,v3),cov(v2,v3),cov(v1,v4),...,cov(vm-1, vm)", where ref is the reference level and v1, v2, ..., vm are the dummy variable names.

Details

Missing values will appear if a model did not converge, produced an error, or not run because of too many missing values in the outcome. Adjusted p-values are only computed for models with a single outcome variable or single exposure variable.

`options` *options list*

Description

A list of 19:

- `check.cor.cutoff` Cutoff value to remove highly correlated columns in the design matrix. The default value is 0.97.
- `check.cor.method` Correlation method to remove highly correlated columns in the design matrix. It must be one of "spearman", "pearson", "kendall". The default value is "spearman".
- `check.design` TRUE or FALSE to check the design matrix for linearly dependent columns, highly correlated columns, or for being ill-conditioned whenever it is updated. The default is TRUE.
- `check.illCond` TRUE or FALSE to check for an ill-conditioned design matrix. The default is TRUE.
- `check.nsubjects` Minimum number of subjects. The default is 25.
- `max.npairwise` The maximum number of metabolites to process the "all pairwise correlations" model. The default is 1000.
- `max.nstrata` The maximum number of strata for a stratified analysis. The default is 10.
- `model` String for the model function. Currently, it must be one of "correlation", "lm", "glm", "coxph", or "clogit". The default is "correlation".
- `model.options` List of options specific for the model. See `correlation.options`, `glm.options`, `lm.options`, `coxph.options` and `clogit.options` for options specific to `model="correlation"`, `"lm"`, `"glm"`, `"coxph"`, `"clogit"` respectively. The default is NULL.
- `output.ci_alpha` Confidence interval level for estimated from glm models. This option must be a number >= 0 and < 1, where 0 is for not creating confidence intervals. The default value is 0.95.
- `output.Effects` A string to define the terms output in the returned `Effects` and `ModelSummary` data frames. Currently, it must be "exposure" or "all". If set to "all", then summary statistics for the exposure and adjustment variables will be output. Otherwise, only summary statistics for the exposure will be output. This option is ignored with `model = "correlation"`. The default is "exposure".

- output.exp_parms TRUE, FALSE or NULL to exponentiate parameter estimates. Standard errors are obtained from the delta method. The default is NULL, so that estimates from logistic regression and survival models will be exponentiated, and not otherwise.
- output.metab.cols Character vector of column names in the METABOLITES sheet to be output in the ModelSummary and Effects data frames. Metabolite ids are matched first using the outcomespec column and then using the exposurespec column. The default is "metabolite_name".
- output.ModelSummary A string to defines the columns output in the returned ModelSummary data frame. Currently, it must be "anova" or "all". This option is ignored with model = "correlation". The default is "anova".
- output.type "rda" or "xlsx" to define the type of output file(s) when runAllModels is called. See output.common.cols and output.merge.
  The default is "xlsx".
- method.adjPvalue Method for adjusted p-values. It must be one of: "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none". Adjusted p-values are only computed for models with a single outcome variable or a single exposure variable. The default is 'fdr'.
- chemEnrich 0 or 1 to run a chemical class enrichment (0=no, 1=yes) using RaMP. The default is 0.
- chemEnrich.adjPvalue The BH-adjusted p-value cutoff to select metabolites for chemical class enrichment. The default is 0.05.

Details

Before any analysis is performed, an initial design matrix is constructed using the above options as follows:
1. Adjustment variables with less than two distinct non-missing values, or with less than check.nsubjects non-missing values are removed.
2. The design matrix is created from the remaining adjustment variables and any linearly dependent columns are removed.
3. If check.cor.cutoff > 0, then highly correlated columns are removed by computing the correlation matrix cor_matrix = cor(design_matrix, method=check.cor.method).
4. If check.illCond is TRUE, then the function subselect::trim.matrix is applied to the correlation matrix to determine if additional columns will be removed.

If check.design is TRUE, then steps 2-4 are repeated each time an exposure variable is added to the design matrix or when rows of the design matrix are removed due to missing values in an outcome variable.

Examples

```
# Logistic regression with all default options
model.op <- list(family="binomial")
op <- list(model="glm", model.options=model.op)
# Compute Pearson correlations requiring at least 100 subjects
model.op <- list(method="pearson")
op <- list(model="correlation", check.nsubjects=100, model.options=model.op)
```

OutputCSVResults *Create output CSV file*

Description

Create output CSV file

Usage

```
OutputCSVResults(filename, dataf, cohort = "")
```

Arguments

filename	name of CSV file and can include path
dataf	correlation output (from function runCorr())
cohort	cohort name

Value

the filename of the CSV file with results named with cohort

Examples

```
## Not run:
dir <- system.file("extdata", package="RcometsAnalytics", mustWork=TRUE)
csvfile <- file.path(dir, "cometsInputAge.xlsx")
exmetabdata <- readCOMETSinput(csvfile)
modeldata <- getModelData(exmetabdata,exposures="age",modlabel="1 Age",
outcomes=c("lactose","lactate"))
corrmatrix <-runCorr(modeldata,exmetabdata,"DPP")
# Get correlation results
OutputCSVResults(filename="corr",dataf=corrmatrix,cohort="DPP")
# Get harmonization results
OutputCSVResults(filename="harmonization",dataf=exmetabdata$metab,cohort="DPP")

## End(Not run)
```

OutputListToExcel *Create an excel xlsx file from a list of data frames*

Description

Create an excel xlsx file from a list of data frames

Usage

```
OutputListToExcel(filename, obj)
```

Arguments

filename	Name of file and can include path. It must have a ".xlsx" extension.
obj	List of data frames or matrices

OutputXLSResults *Create output XLSX file*

Description

Create output XLSX file

Usage

```
OutputXLSResults(filename, datal, cohort = "")
```

Arguments

filename	name of file and can include path
datal	data list to output (each item on list outputs to a worksheet)
cohort	cohort name

Value

the filename of the XLSX file with results named with cohort

Examples

```
dir <- system.file("extdata", package="RcometsAnalytics", mustWork=TRUE)
csvfile <- file.path(dir, "cometsInputAge.xlsx")
exmetabdata <- readCOMETSinput(csvfile)
modeldata <- getModelData(exmetabdata,exposures="age",modlabel="1 Age",
outcomes=c("lactose","lactate"))
# Get descriptive data
descdata <-runDescrip(exmetabdata)
OutputXLSResults(filename="corr",datal=descdata,cohort="DPP")
```

pathwayEnrichment *Pathway Enrichment*

Description

Pathway Enrichment

Usage

```
pathwayEnrichment(
  df,
  metabdata,
  db.version = NULL,
  pvalue.adj = 0.05,
  filter.p = 0.1,
  filter.p.type = "fdr",
  perc_analyte_overlap = 0.5,
  min_pathway_tocluster = 2,
  perc_pathway_overlap = 0.5
)
```

Arguments

df	The "Effects" data frame returned from function runModel
metabdata	Return object from readCOMETSinput
db.version	The RaMP SQLite database version to use. The default is the most recent version.
pvalue.adj	The maximum adjusted p-value for a metabolite to be included in the analysis.
filter.p	See argument pval_cutoff in RaMP::FilterFishersResults.
filter.p.type	See argument pval_type in RaMP::FilterFishersResults.
perc_analyte_overlap	See argument perc_analyte_overlap in RaMP::findCluster.
min_pathway_tocluster	See argument min_pathway_tocluster in RaMP::findCluster.
perc_pathway_overlap	See argument perc_pathway_overlap in RaMP::findCluster.

Details

The **RaMP** R package must be installed prior to calling this function. An attempt to load RaMP will be done through the requireNamespace function.

The set of all possible metbolite id types are: LIPIDMAPS, pubchem, hmdb, chemspider, chebi, CAS, wikidata, swisslipids, kegg, lipidbank, plantfa, and kegg_glycan. The **METABOLITES** sheet in the input Excel file will need to have at least one of the above id types as a column (case-insensitive). If the **METABOLITES** sheet contains more than one id type, then the metabolite ids passed into chemicalClassEnrichment depends on the order of the id type columns. For example, if the **METABOLITES** sheet contains columns HMDB (column F) and pubchem (column C), then the metbolite ids will taken from the pubchem column and any missing ids from the pubchem column will be assigned from the HMDB column.

Value

A data frame

Examples

```
dir <- system.file("extdata", package="RcometsAnalytics", mustWork=TRUE)
csvfile <- file.path(dir, "cometsInputAge.xlsx")
exmetabdata <- readCOMETSinput(csvfile)
modeldata <- getModelData(exmetabdata,exposures="age", modelspec="Interactive",
outcomes=c("lactose","lactate"))
obj <- runModel(modeldata,exmetabdata, cohortLabel="DPP")
# ret <- pathwayEnrichment(obj$Effects, exmetabdata, pvalue.adj=1)
```

plotMinvalues *Plot the distribution of minimum or missing values for each metabolite*

Description

Plot the distribution of minimum or missing values for each metabolite

Usage

```
plotMinvalues(
  cometsdata,
  title = NULL,
  xlabel = NULL,
  ylabel = "Frequency",
  xylabelsize = 12,
  titlesize = 16,
  missing = FALSE
)
```

Arguments

cometsdata	output of readCOMETSinput function
title	main title for the plot (default is "Distribution of the Number of Minimum Values")
xlabel	x-axis label (default is "Number of minimum values")
ylabel	y-axis label (default is "Frequency")
xylabelsize	size of x and y labels (default=8)
titlesize	size of title (default, 20)
missing	TRUE or FALSE to plot distribution of missing values (default=FALSE)

Value

a distribution plot

Examples

```
dir <- system.file("extdata", package="RcometsAnalytics", mustWork=TRUE)
csvfile <- file.path(dir, "cometsInputAge.xlsx")
exmetabdata <- readCOMETSinput(csvfile)
plotMinvalues(exmetabdata)
```

plotVar *Plot the variance distribution of transformed metabolite abundances*

Description

Plot the variance distribution of transformed metabolite abundances

Usage

```
plotVar(
  cometsdata,
  title = "Distribution of Variance",
  titlesize = 16,
  xlabel = "Variance of transformed metabolite abundances",
  ylabel = "Frequency",
  xylabelsize = 12
)
```

Arguments

cometsdata	output of readCOMETSinput function
title	main title for the plot (default is "Distribution of Variance")
titlesize	size of title (default, 20)
xlabel	x-axis label (default is "Variance of transformed metabolite abundances")
ylabel	y-axis label (default is "Frequency")
xylabelsize	size of x and y labels (default=8)

Value

a distribution plot

Examples

```
dir <- system.file("extdata", package="RcometsAnalytics", mustWork=TRUE)
csvfile <- file.path(dir, "cometsInputAge.xlsx")
exmetabdata <- readCOMETSinput(csvfile)
plotVar(exmetabdata)
```

prdebug *debug by printing object with time time*

Description

debug by printing object with time time

Usage

```
prdebug(lab, x)
```

Arguments

lab	label of object
x	object

`readCOMETSinput` *Read in Excel file that contains metabolite data, covariate data, models, and model options.*

Description

Read in Excel file that contains metabolite data, covariate data, models, and model options.

Usage

```
readCOMETSinput(file, mode = "Batch")
```

Arguments

file	path of Excel file to be read in. This file must contain sheets with names **Metabolites**, **SubjectMetabolites**, **SubjectData**, **VarMap**, and optionally **Models**, **Model_Types** (see details).
mode	How is this function being called (Interactive or Batch). When running in Batch mode, all models in the **Models** sheet must be valid; otherwise an error will be thrown. When running in Interactive mode, a warning will be given if the **Models** sheet contains errors. The default is Batch.

Details

Additional information regarding each sheet in the input Excel file is given below.

Metabolites
A table with the columns `METABID`, `METABOLITE_NAME`, and possibly other columns of information about the metabolites. The `METABID` column is used for harmonizing the metabolite names across different cohorts when meta-analyses are performed.

SubjectMetabolites
A table with the subject ids in the first column and metabolites as the other columns. Each cell must contain a numeric value or left empty (missing); otherwise an error will occur.

SubjectData
A table with the subject ids in the first column and covariates as the other columns. For continuous variables, each cell must contain a numeric value or left empty (missing); otherwise an error will occur.

VarMap
A table with at least the required columns `VARREFERENCE`, `COHORTVARIABLE`, `VARTYPE`, `VARDEFINITION`, and optionally `ACCEPTED_VALUES`. The `COHORTVARIABLE` column must contain names that match the column names in the **SubjectData** table. These names will be renamed to their corresponding name in the `VARREFERENCE` column. The `VARTYPE` column should have values `continuous` or `categorical` for each row. The `ACCEPTED_VALUES` column defines the allowed values that the variable should have, and an error will be thrown if the variable is outside of the accepeted values. For a categorical variable, `ACCEPTED_VALUES` should be a comma separated list of values such as '1, 2, 3'. For a continuous variable, `ACCEPTED_VALUES` should be a range of the form: (a, b), (a, b], [a, b), or [a, b], where parentheses denote exclusion of the value, and brackets denote inclusion of the

value. Use Inf or -Inf to denote infinity or minus infinity.

Models

A table where each row represents a model to be run, and with columns MODEL, OUTCOMES, EXPOSURE, ADJUSTMENT, STRATIFICATION, WHERE, and optionally MODEL_TYPE, TIME, GROUP, and EXPOSURE_REFERENCE. All variable names in this table must match variable names in the VARREFERENCE column of the **VarMap** sheet. The MODEL column is a label for the model. The OUTCOMES and EXPOSURE columns define the outcome and exposure variables for the model. Use `All metabolites` to specify that all metabolite variables are to be included as outcomes or exposures, otherwise use a space separated list of variable names. For any categorical exposure variable, the EXPOSURE_REFERENCE column is required to specify the reference level. The reference level must match one of the ACCEPTED_VALUES in the **VarMap** sheet. The ADJUSTMENT column contains a space separated list of covariate adjustment variables; use an empty cell for no covariate adjustment. The STRATIFICATION column is used for stratified analyses, with a space separated list of stratification variables. If more than one stratification variable is specified, then the strata are defined by all unique combinations of the stratification variables that appear in the data. The WHERE column is used to define a subset of subjects to include in the analysis, and has the form `variable operator value`, where `operator` can be one of the following `>`, `<`, `>=`, `<=` `!=`, `=`. An example WHERE condition is `age > 50`, which will include all subjects older than 50 in the analysis. Multiple WHERE conditions must be separated by a comma and are logically connected with the & operator. For example, the WHERE condition `age > 50 , bmi >= 22` will include the subjects older than 50 AND with bmi >= 22. Values in the MODEL_TYPE column must match with the MODEL_TYPE column in the **Model_Types** sheet. The TIME column is only required when survival models are run. This column can contain a single time variable or two time variables separated by a space. The GROUP column is only required when conditional logistic regression models are run. This column can contain only a single variable that defines the groups (sets, strata). This sheet is not required when running in interactive mode, but is required when running in batch mode.

Model_Types

A table where each row specifies an option and has columns MODEL_TYPE, FUNCTION, OPTION, and VALUE. For an example sheet and additional information about this sheet, see the Excel file `/extdata/cometsInput.xlsx`. This sheet is optional, but is required when the **Models** sheet contains the column MODEL_TYPE.

Value

a list comprising of data and information needed for `getModelData`.

Examples

```
dir <- system.file("extdata", package="RcometsAnalytics", mustWork=TRUE)
csvfile <- file.path(dir, "cometsInputAge.xlsx")
exmetabdata <- readCOMETSinput(csvfile)
```

`runAllMeta` *This function allows users to run meta-analyses on all models or a subset of models*

Description

This function allows users to run meta-analyses on all models or a subset of models

Usage

```
runAllMeta(filesFolders, out.dir, opfile = NULL)
```

Arguments

filesFolders	Character vector of files and/or folders that contain the output files from runAllModels. Valid file extensions are ".xlsx", ".rda", ".zip", ".tar", and ".tar.gz". Zip and tar files must contain files with extensions ".xlsx" or ".rda". See details.
out.dir	Output directory to save the results for each model.
opfile	Excel file containing the models and options. See meta_opfile. The default is NULL.

Details

The names of the files output from runAllModels will be of the form
<model name>__<cohort name>__<date>.xlsx or
<model name>__<cohort name>__<date>.rda.
These output files from runAllModels can be bundled together into zip or tar files. There is no file name requirement for the zip and tar files.

The steps used to obtain the initial set of files to include in the meta-analyses are:
1. Recursively list all files found in any folder of filesFolders.
2. Unzip or untar any zip or tar file.
3. Remove files that do not have a file name of the forms above.

---

runAllModels *This function allows users to run all models that are provided in the "Models" sheet of the input Excel file.*

Description

This function allows users to run all models that are provided in the "Models" sheet of the input Excel file.

Usage

```
runAllModels(readData, cohortLabel = "", writeTofile = TRUE)
```

Arguments

readData	list from readCOMETSinput
cohortLabel	cohort label (e.g. DPP, NCI, Shanghai)
writeTofile	TRUE/FALSE (whether or not to write results for each model into separate xlsx files). Files are written to current directory. Default is TRUE.

Value

A list of return objects from runModel or runCorr. The ith element in this list is the output from the ith model run.

Examples

```
## Not run:
dir <- system.file("extdata", package="RcometsAnalytics", mustWork=TRUE)
csvfile <- file.path(dir, "cometsInputAge.xlsx")
exmetabdata <- readCOMETSinput(csvfile)
allmodeloutput <- runAllModels(exmetabdata)

## End(Not run)
```

`runCorr` *Calculate correlations for input model.*

Description

Calculate correlations for input model.

Usage

```
runCorr(modeldata, metabdata, cohort = "")
```

Arguments

`modeldata`	list from function `getModelData`
`metabdata`	metabolite data list from `readCOMETSinput`
`cohort`	cohort label (e.g DPP, NCI, Shanghai)

Details

This function is a special case of `runModel` with the option `op$model = "correlation"`, however for backwards compatibility, it returns a data frame as in the original version of the **COMETS** R package.

Value

data frame with each row representing the correlation for each combination of outcomes and exposures represented as specified in the model (*spec), label (*lab), and universal id (*_uid) with additional columns for n, pvalue, method of model specification (Interactive or Batch), universal id for outcomes (outcome_uid) and exposures (exposure_uid) name of the cohort, adjustment (adjvars) and stratification (stratavar,strata) variables. Attribute of dataframe includes ptime for processing time of model run.

Examples

```
dir <- system.file("extdata", package="RcometsAnalytics", mustWork=TRUE)
csvfile <- file.path(dir, "cometsInputAge.xlsx")
exmetabdata <- readCOMETSinput(csvfile)
modeldata <- getModelData(exmetabdata,exposures="age",modlabel="1 Age",
outcomes=c("lactose","lactate"), modelspec="Interactive")
corrmatrix <- runCorr(modeldata,exmetabdata, "DPP")
```

`runDescrip` *This function provides a description of the input data (for categorical data, the number of samples of each type; for continous data, the median and other statistics for each variable)*

Description

This function provides a description of the input data (for categorical data, the number of samples of each type; for continous data, the median and other statistics for each variable)

Usage

```
runDescrip(readData)
```

Arguments

`readData`	list from readComets

Value

a list with 2 data frames, continuous and categorical summaries. Type of variable is defined in varmap

Examples

```
## Not run:
dir <- system.file("extdata", package="RcometsAnalytics", mustWork=TRUE)
csvfile <- file.path(dir, "cometsInputAge.xlsx")
exmetabdata <- readCOMETSinput(csvfile)
allmodeloutput <- runAllModels(exmetabdata)
# Get descriptive data
descdata <-runDescrip(exmetabdata)
OutputXLSResults(filename="corr",datal=descdata,cohort="DPP")

## End(Not run)
```

`runMeta` *This function allows users to run a single meta-analysis*

Description

This function allows users to run a single meta-analysis

Usage

```
runMeta(file.obj, op = NULL)
```

Arguments

file.obj	Character vector of file names or a list, where each element of the list is a file name or a list of type file.list. The files can be output files from runModel or runAllModels. If all the files are from runModel or runAllModels, file.obj can be a vector. A list is required if any of the files are not from runModel and runAllModels, or if any of the files requires changes.
op	A list containing the options to use. See meta_options. The default is NULL.

Value

List of data frames containing the results and information. See metaOutputColumns for of a description of columns in the results table.

---

runModel *Main function for running the models*

Description

Main function for running the models

Usage

```
runModel(
  modeldata,
  metabdata,
  cohortLabel = "",
  op = NULL,
  writeTofile = FALSE
)
```

Arguments

modeldata	list from function getModelData
metabdata	metabolite data list from readCOMETSinput
cohortLabel	cohort label (e.g DPP, NCI, Shanghai)
op	list of options when running in Interactive mode (see options).
writeTofile	TRUE/FALSE (whether or not to write results to the current directory). Default is FALSE.

Details

This function will check for stratification variables, and if present run within each stratum. The design matrix is checked for validity (see options). When running in Batch mode, the options are obtained from the Options sheet in the excel file.

Value

A list of objects with names ModelSummary, Effects, Errors_Warnings, Table1, Info. **Important: check the** adjvars.removed **column in the** ModelSummary **data frame to see if any adjustment variables were dropped from the model, and use caution interpreting results when variables are removed. The** Errors_Warnings **object may also contain additional variables removed.** Attribute of this list includes ptime for processing time of model run.

Examples

```
dir <- system.file("extdata", package="RcometsAnalytics", mustWork=TRUE)
csvfile <- file.path(dir, "cometsInputAge.xlsx")
exmetabdata <- readCOMETSinput(csvfile)
modeldata <- getModelData(exmetabdata,exposures="age",modlabel="1 Age",
outcomes=c("lactose","lactate"))
obj <- runModel(modeldata,exmetabdata, cohortLabel="DPP")
```

---

showCorr *Function that returns top N lines of the* runCorr *output*

Description

Function that returns top N lines of the runCorr output

Usage

```
showCorr(corr, nlines = 50)
```

Arguments

corr	returned object from runCorr
nlines	number of lines to return (default 50)

Value

first 50 lines of output

Examples

```
dir <- system.file("extdata", package="RcometsAnalytics", mustWork=TRUE)
csvfile <- file.path(dir, "cometsInputAge.xlsx")
exmetabdata <- readCOMETSinput(csvfile)
modeldata <- getModelData(exmetabdata,modlabel="1 Age")
corrmatrix <- runCorr(modeldata,exmetabdata,"DPP")
showCorr(corrmatrix)
```

showHClust *Show interactive heatmap using heatmaply_cor with hierarchical clustering*

Description

This function outputs a heatmap with hierarchical clustering. It thus requires you to have at least 2 outcome and 2 exposure variables in your models.

Usage

```
showHClust(
  ccorrList,
  strata = NULL,
  clust = TRUE,
  colscale = "RdYlBu",
  showticklabels = TRUE
)
```

Arguments

ccorrList	correlation object (output of runCorr)
strata	Only valid if ccorrList is from a stratified analysis. If NULL, then results from the first stratum will be used in the plot.
clust	TRUE or FALSE to show hierarchical clustering. The default is TRUE.
colscale	colorscale, can be custom or named ("Hots","Greens","Blues","Greys","Purples") see RColorBrewer_colors
showticklabels	TRUE or FALSE to show axis labels. The default is TRUE.

Value

a heatmap with outcomes as rows and exposures in columns.

References

For colorscale reference: RColorBrewer_colors

Examples

```
dir <- system.file("extdata", package="RcometsAnalytics", mustWork=TRUE)
csvfile <- file.path(dir, "cometsInputAge.xlsx")
exmetabdata <- readCOMETSinput(csvfile)
modeldata <- getModelData(exmetabdata, modelspec="Interactive",
exposures=c("age","bmi_grp"))
corrmatrix <-runCorr(modeldata,exmetabdata,"DPP")
#showHClust(corrmatrix)
```

showHeatmap *Show interactive heatmap using plot_ly*

Description

Show interactive heatmap using plot_ly

Usage

```
showHeatmap(
  ccorrList,
  strata = NULL,
  rowsortby = "estimate",
  plothgt = 700,
  plotwid = 800,
  colscale = "RdYlBu"
)
```

Arguments

ccorrList	correlation object (output of runCorr)
strata	Only valid if ccorrList is from a stratified analysis. If NULL, then results from the first stratum will be used in the plot.
rowsortby	How row labels are sorted
plothgt	Plot height default 700
plotwid	Plot width default 800
colscale	colorscale, can be custom or named ("Hots","Greens","Blues","Greys","Purples") see https://plot.ly/ipython-notebooks/color-scales/

Value

a heatmap with outcomes as rows and exposures in columns.

References

For colorscale reference: https://plot.ly/ipython-notebooks/color-scales/

Examples

```
## Not run:
dir <- system.file("extdata", package="RcometsAnalytics", mustWork=TRUE)
csvfile <- file.path(dir, "cometsInputAge.xlsx")
exmetabdata <- readCOMETSinput(csvfile)
modeldata <- getModelData(exmetabdata,modlabel="1 Age")
corrmatrix <-runCorr(modeldata,exmetabdata,"DPP")
showHeatmap(corrmatrix)

## End(Not run)
```

showModel *Function that displays the first N rows of each data frame in the the* runModel *output.*

Description

Function that displays the first N rows of each data frame in the the runModel output.

Usage

```
showModel(obj, nlines = 10)
```

Arguments

obj	returned object from runModel
nlines	number of lines to display (default 10)

Examples

```
dir         <- system.file("extdata", package="RcometsAnalytics", mustWork=TRUE)
csvfile     <- file.path(dir, "cometsInputAge.xlsx")
exmetabdata <- readCOMETSinput(csvfile)
modeldata   <- getModelData(exmetabdata,modlabel="1 Age")
result      <- runModel(modeldata,exmetabdata, cohortLabel="DPP")
showModel(result)
```

Table1 *Non-metabolite Variable Summary Table*

Description

Columns in Table1. Depending on the model run and options specified, all the below names may not appear in the data frame.

- category Category for categorical variables only
- in.model How variable enters the model (outcome, exposure, adjustment, time, group, weight, offset)
- max Maximum value
- mean Mean value
- median Median value
- min Minimum value
- n Number of non-missing observations
- n.missing Number of missing observations
- n.outcomeEqual0 Number of non-missing observations with outcome = 0
- n.outcomeEqual1 Number of non-missing observations with outcome = 1
- n.unique Number of unique non-missing observations

- quartile1 25th percentile
- quartile3 75th percentile
- stratavar Stratum variable(s)
- strata Stratum level(s)
- type Either continuous or categorical
- variable Variable name in the model

Details

The columns max, mean, median, min, n.missing, n.unique, quartile1, and quartile3 are for continuous variables only.

Index