Package 'metadeconfoundR'

July 22, 2025

Type Package

Title Covariate-Sensitive Analysis of Cross-Sectional High-Dimensional Data

Version 1.0.2

Maintainer Till Birkner <metadeconf@till-birkner.de>

Description Using non-parametric tests, naive associations between omics features and metadata in cross-sectional data-sets are detected. In a second step, confounding effects between metadata associated to the same omics feature are detected and labeled using nested post-hoc model comparison tests, as first described in Forslund, Chakaroun, Zimmermann-Kogadeeva, et al. (2021) <doi:10.1038/s41586-021-04177-9>. The generated output can be graphically summarized using the built-in plotting function.

License GPL-2

Encoding UTF-8

LazyData true

URL https://github.com/TillBirkner/metadeconfoundR

BugReports https://github.com/TillBirkner/metadeconfoundR/issues

Imports lmtest, foreach, parallel, doParallel, stats, futile.logger, lme4, ggplot2, reshape2, methods, rlang

Depends R (>= 3.5.0), detectseparation

Suggests pander, knitr, gridExtra, kableExtra

VignetteBuilder knitr

RoxygenNote 7.3.1

NeedsCompilation no

Author Till Birkner [aut, cre] (ORCID: <https://orcid.org/0000-0003-2656-2821>), Sofia Kirke Forslund-Startceva [ctb] (ORCID: <https://orcid.org/0000-0003-4285-6993>)

Repository CRAN

Date/Publication 2024-06-25 14:40:02 UTC

Contents

`BuildHeatmap` *BuildHeatmap*

Description

BuildHeatmap summarizes MetaDeconfound output in a heatmap or cuneiform plot

Usage

```
BuildHeatmap(
  metaDeconfOutput,
  q_cutoff = 0.1,
  d_cutoff = 0.01,
  cuneiform = FALSE,
  coloring = 0,
  showConfounded = TRUE,
  intermedData = FALSE,
  featureNames = NULL,
  metaVariableNames = NULL,
  d_range = "fit",
  d_col = c("blue", "white", "red"),
  keepMeta = NULL,
  keepFeature = NULL,
  trusted = c("OK_sd", "OK_nc", "OK_d", "AD"),
  tileBordCol = "black",
  reOrder = "both"
)
```

Arguments

`metaDeconfOutput`	output of a metadeconfound run
`q_cutoff`	optional FDR-value cutoff used to remove low-significance entries from data
`d_cutoff`	optional effect size cutoff used to remove low effect size entries from data
`cuneiform`	optional logical parameter, plot cuneiform instead of heatmap when cuneiform = TRUE

coloring	optional, can be 0,1,2; 0: color all tiles according to effectsize ; 1: don't color not significant tiles 2: like 1 but also don't color confounded signal tiles
showConfounded	optional logical parameter; set to FALSE to remove significance markers from confounded signals
intermedData	only return intermediate data for plotting, default = FALSE
featureNames	optional two-column-dataframe containing corresponding "human-readable" names to the "machine-readable" feature names used as row.names in metaDeconfOutput. These human readable names will be displayed in the final plot. First column: machine-readable, second column: human-readable.
metaVariableNames	optional two-column-dataframe containing corresponding "human-readable" names to the "machine-readable" metadata names used as column names in metaDeconfOutput. These human readable names will be displayed in the final plot. First column: machine-readable, second column: human-readable.
d_range	range of effect sizes shown; "full": (default) range from -1 to +1; "fit": range reduced according to maximum and minimum effect size present in resulting plot
d_col	set color range for effect size as c(minimum, middle, maximum), default c("red", "white", "blue")
keepMeta	character vector of metavariable names (corresponding to names in metaDeconfOutput), that should be shown in resulting plot, even when they have no associations passing d_cutoff and q_cutoff
keepFeature	character vector of metavariable names (corresponding to names in metaDeconfOutput), that should be shown in resulting plot, even when they have no associations passing d_cutoff and q_cutoff
trusted	character vector of confounding status labels to be treated as trustworthy, not-confounded signal. default = c("OK_sd", "OK_nc", "OK_d", "AD")
tileBordCol	tile border color of heatmap tiles, default: "black"
reOrder	reorder features and/or metadata? possible options: c("both", "feat", "meta", "none"), default: "both"

Details

for more details and explanations please see the package vignette.

Value

ggplot2 object

Examples

```
data(reduced_feature)
data(metaMatMetformin)

example_output <- MetaDeconfound(featureMat = reduced_feature,
                                 metaMat = metaMatMetformin,
```

```
                                    logLevel = "ERROR")

plotObject <- BuildHeatmap(example_output)

alternativePlot <- BuildHeatmap(example_output, coloring = 2, showConfounded = FALSE)
```

ImportLongPrior *ImportLongPrior*

Description

ImportLongPrior imports prior knowledge of associations between individual features and metadata in form of a long-format dataframe.

Usage

```
ImportLongPrior(longPrior, featureMat, metaMat)
```

Arguments

longPrior	long-format dataframe as generated by Metadeconfound(returnLong = TRUE). Must contain at least one column containing feature names and one column containing associated metadata names, called "feature" and "metaVariable", respectively. Only associations between features and metadata present in featureMat and metaMat will be returned. Additionally, "Qs" and "status" (as produced by MetaDeconfound)columns can be supplied and will be parsed as well. If only "feature" and "metaVariable" columns are supplied, all listed associations are assumed to be significant. If "status" is supplied, only non-"NS" labeled associations will be kept.
featureMat	omics features to be analyzed by MetaDeconfound
metaMat	metadata to be analyzed by MetaDeconfound

Details

This function is meant to facilitate incorporation of prior knowledge about associations between measured omics features and available metadata both from earlier metadeconfoundR runs by supplying the long-format Metadeconfound(returnLong = TRUE) output directly or by supplying a simple list of known associations from other studies.

Value

wide-format dataframe that can be used as minQValues parameter in MetaDeconfound

Examples

```
data(reduced_feature)
data(metaMatMetformin)


# note that this example is only to demonstrate the process of integrating
 # prior knowledge into a MetaDeconfound() analysis. Using the output of a
 # MetaDeconfound() run as minQValues input for a second run with the exact
 # same features and metadata will not lead to any new insights since the set
 # of QValues calculated by MetaDeconfound() and the set supplied using the
 # minQValues parameter are identical in this case.

example_output <- MetaDeconfound(featureMat = reduced_feature,
                                 metaMat = metaMatMetformin,
                                 returnLong = TRUE,
                                 logLevel = "ERROR")

minQValues <- ImportLongPrior(longPrior = example_output,
                              featureMat = reduced_feature,
                              metaMat = metaMatMetformin)

example_output2 <- MetaDeconfound(featureMat = reduced_feature,
                                 metaMat = metaMatMetformin,
                                 minQValues = minQValues,
                                 logLevel = "ERROR")
```

MetaDeconfound *MetaDeconfound*

Description

MetaDeconfound checks all feature <-> covariate combinations for counfounding effects of covariates on feature <-> effect correlation

Usage

```
MetaDeconfound(
  featureMat,
  metaMat,
  nnodes = 1,
  adjustMethod = "fdr",
  robustCutoff = 5,
  QCutoff = 0.1,
  DCutoff = 0,
  PHS_cutoff = 0.05,
  logfile = NULL,
  logLevel = "INFO",
```

```
  startStop = NA,
  QValues = NA,
  DValues = NA,
  minQValues = NULL,
  deconfT = NULL,
  deconfF = NULL,
  doConfs = 0,
  doRanks = NA,
  randomVar = NA,
  fixedVar = NA,
  robustCutoffRho = NULL,
  typeCategorical = NULL,
  typeContinuous = NULL,
  logistic = FALSE,
  rawCounts = FALSE,
  returnLong = FALSE,
  collectMods = FALSE,
  ...
)
```

Arguments

featureMat	a data frame with row(sample ID) and column(feature such as metabolite or microbial OTU) names, listing features for all samples
metaMat	a data frame with row(sample ID) and column(meta data such as age,BMI and all possible confounders) names listing metadata for all samples. first column should be case status with case=1 and control=0. All binary variables need to be in 0/1 syntax!
nnodes	number of nodes/cores to be used for parallel processing
adjustMethod	multiple testing p-value correction using one of the methods of p.adjust.methods
robustCutoff	minimal number of sample size for each covariate in order to have sufficient power for association testing
QCutoff	significance cutoff for q-value, DEFAULT = 0.1
DCutoff	effect size cutoff (either cliff's delta or spearman correlation test estimate), DEFAULT = 0
PHS_cutoff	PostHoc Significance cutoff
logfile	name of optional logging file.
logLevel	logging verbosity, possible levels: TRACE, DEBUG, INFO, WARN, ERROR, FATAL, DEFAULT = INFO
startStop	vector of optional strings controlling which parts of the pipeline should be executed. ("naiveStop": only naive associations will be computed, no confounder analysis is done)
QValues	optional data.frame containing pre-computed multiple-testing corrected p-values for naive associations
DValues	optional data.frame containing pre-computed effect sizes for naive associations

minQValues	pessimistic qvalues, can be generated by ImportLongPrior. This dataframe of QValues is used to incorporate prior knowledge of potential associations between individual features and metadata by supplying QValues < QCutoff for these associations. All significant associations thus reported will be treated as potentially confounding influences.
deconfT	vector of metavariable names *always* to be included as potential confounder
deconfF	vector of metavariable names *never* to be included as potential confounder
doConfs	optional parameter for additional computation of confidence interval of linear models in the deconfounding step (0 = no , 1 = logging, 2 = strict)
doRanks	optional vector of metavariable names, that should be rank transformed when building linear models in the doconfounding step
randomVar	optional vector of metavariable names to be treated as random effect variables. These variables will not be tested for naive associations and will not be included as potential confounders, but will be added as random effects "+ (1|variable)" into any models being built. Any associations reducible to the supplied random effect(s) will be labeled as "NS". Note: Ps, Qs, Ds are computed independently and thereby not changed through inclusion of random effects.
fixedVar	optional vector of metavariable names to be treated as fixed effect variables. These variabels will not be tested for naive associations and will not be included as potential confounders, but will be added as fixed effects "+ variable" into any models being built. Any associations reducible to the supplied fixed effect(s) will be labeled as "NS". Note: Ps, Qs, Ds are computed independently and thereby not changed through inclusion of fixed effects.
robustCutoffRho	optional robustness cutoff for continuous variables
typeCategorical	optional character vector of metavariable names to always be treated as categorical
typeContinuous	optional character vector of metavariable names to always be treated as continuous
logistic	optional logical parameter; DEFAULT = FALSE; Set TRUE to treat supplied features as binary instead of continuous
rawCounts	optional logical parameter; DEFAULT = FALSE; Set TRUE to treat supplied features as not normalized/rarefied counts; metadeconfoundR will compute total read count per sample and include this information in the modelling steps. WARNING: naive associations computed in first part of metadeconfoundR are reliant on normalized/rarefied data. Please split your analysis up into 2 parts as shown in the documentation when using this mode..
returnLong	DEFAULT = FALSE; Set TRUE to get output in one long format data.frame instead of list of four wide format data.frames
collectMods	DEFAULT = FALSE; Set TRUE to collect all model objects generated by Metadeconfound and return them in a nested list alongside the standard Ps/Qs/Ds/status output.
...	for additional arguments used internally (development/debugging)

Details

for more details and explanations please see the vignette.

Value

list with elements (or data.frame with columns, when returnLong = TRUE) Ds = effectsize, Ps = uncorrected p-value for naive association, Qs = multiple testing corrected p-value/fdr, and status = confounding status for all feature <=> covariate combinations with following categories: (NS = not significant, OK_sd = strictly deconfounded, OK_nc = no covariates, OK_d = doubtful, AD = ambiguously deconfounded, C: followed by comma separated covariate names = confounded by listed covariates)

Can be plotted using BuildHeatmap.

Examples

```
data(reduced_feature)
data(metaMatMetformin)

example_output <- MetaDeconfound(featureMat = reduced_feature,
                                 metaMat = metaMatMetformin,
                                 logLevel = "ERROR")
```

---

metaMatMetformin *Documentation for the metaMatMetformin RData in /data*

Description

set of features from the metformin dataset (Forslund et al. (2015), DOI: https://doi.org/10.1038/nature15766), containing status for 5 different properties for 753 samples

---

reduced_feature *Documentation for the reduced_feature RData in /data*

Description

reduced set of features from the metformin dataset (Forslund et al. (2015), DOI: https://doi.org/10.1038/nature15766), containing feature measurements for 753 samples

Index