Package: TwoSampleMR (via r-universe)

June 30, 2024

Title Two Sample MR Functions and Interface to MR Base Database

Version 0.6.5

Description A package for performing Mendelian randomization using GWAS summary data. It uses the IEU GWAS database <https://gwas.mrcieu.ac.uk/> to automatically obtain data, and a wide range of methods to run the analysis. You can use the MR-Base web app <https://www.mrbase.org/> to try out a limited range of the functionality in this package, but for any serious work we strongly recommend using this R package.

License MIT + file LICENSE

URL https://github.com/MRCIEU/TwoSampleMR, https://mrcieu.github.io/TwoSampleMR/

BugReports https://github.com/MRCIEU/TwoSampleMR/issues/

Depends R (>= 4.0.0)

Imports cowplot, data.table, dplyr, ggplot2 (>= 3.4.0), glmnet, gridExtra, gtable, ieugwasr (>= 1.0.0), jsonlite, knitr, lattice, magrittr, MASS, meta, mr.raps, MRMix, MRPRESSO, pbapply, plyr, psych, RadialMR, reshape2, rmarkdown

Suggests Cairo, car, markdown, MendelianRandomization, MRInstruments, randomForest, testthat

VignetteBuilder knitr

Remotes gqi/MRMix, mrcieu/MRInstruments, MRPRESSO=rondolab/MR-PRESSO, WSpiller/RadialMR

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

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

RemoteUrl https://github.com/MRCIEU/TwoSampleMR

RemoteRef HEAD

RemoteSha 2f0736eab810311a61851c0477cc452cf2a07861

Contents

`add_metadata` *Add meta data to extracted data*

Description

Previously the meta data was returned alongside association information. This is mostly unnecessary as it is needlessly repeating information. This is a convenience function that reinstates that information. Can be applied to either exposure data, outcome data, or harmonised data

Usage

```
add_metadata(dat, cols = c("sample_size", "ncase", "ncontrol", "unit", "sd"))
```

Arguments

`dat`	Either exposure data, outcome data or harmonised data
`cols`	Which metadata fields to add. Default = `c("sample_size", "ncase", "ncontrol", "unit", "sd")`

Value

Data frame

add_rsq *Estimate r-square of each association*

Description

Can be applied to exposure_dat, outcome_dat or harmonised_data. Note that it will be beneficial in some circumstances to add the meta data to the data object using `add_metadata()` before running this function. Also adds effective sample size for case control data.

Usage

```
add_rsq(dat)
```

Arguments

`dat`	exposure_dat, outcome_dat or harmonised_data

Value

data frame

allele_frequency *Estimate allele frequency from SNP*

Description

Estimate allele frequency from SNP

Usage

```
allele_frequency(g)
```

Arguments

`g`	Vector of 0/1/2

Value

Allele frequency

available_outcomes *Get list of studies with available GWAS summary statistics through API*

Description

Get list of studies with available GWAS summary statistics through API

Usage

```
available_outcomes(opengwas_jwt = ieugwasr::get_opengwas_jwt())
```

Arguments

opengwas_jwt	Used to authenticate protected endpoints. Login to https://api.opengwas.io to obtain a jwt. Provide the jwt string here, or store in .Renviron under the keyname OPENGWAS_JWT.

Value

Dataframe of details for all available studies

clump_data *Perform LD clumping on SNP data*

Description

Uses PLINK clumping method, where SNPs in LD within a particular window will be pruned. The SNP with the lowest p-value is retained.

Usage

```
clump_data(
  dat,
  clump_kb = 10000,
  clump_r2 = 0.001,
  clump_p1 = 1,
  clump_p2 = 1,
  pop = "EUR",
  bfile = NULL,
  plink_bin = NULL
)
```

Arguments

dat	Output from format_data(). Must have a SNP name column (SNP), SNP chromosome column (chr_name), SNP position column (chrom_start). If id.exposure or pval.exposure not present they will be generated.
clump_kb	Clumping window, default is 10000.
clump_r2	Clumping r2 cutoff. Note that this default value has recently changed from 0.01 to 0.001.
clump_p1	Clumping sig level for index SNPs, default is 1.
clump_p2	Clumping sig level for secondary SNPs, default is 1.
pop	Super-population to use as reference panel. Default = "EUR". Options are "EUR", "SAS", "EAS", "AFR", "AMR". 'legacy' also available - which is a previously used version of the EUR panel with a slightly different set of markers
bfile	If this is provided then will use the API. Default = NULL
plink_bin	If NULL and bfile is not NULL then will detect packaged plink binary for specific OS. Otherwise specify path to plink binary. Default = NULL

Details

This function interacts with the OpenGWAS API, which houses LD reference panels for the 5 super-populations in the 1000 genomes reference panel. It includes only bi-allelic SNPs with MAF > 0.01, so it's quite possible that a variant you want to include in the clumping process will be absent. If it is absent, it will be automatically excluded from the results.

You can check if your variants are present in the LD reference panel using ieugwasr::ld_reflookup().

This function does put load on the OpenGWAS servers, which makes life more difficult for other users. We have implemented a method and made available the LD reference panels to perform clumping locally, see ieugwasr::ld_clump() and related vignettes for details.

Value

Data frame

---

combine_all_mrresults *Combine all mr results*

Description

This function combines results of mr(), mr_heterogeneity(), mr_pleiotropy_test() and mr_singlesnp() into a single data frame. It also merges the results with outcome study level characteristics in available_outcomes(). If desired it also exponentiates results (e.g. if the user wants log odds ratio converted into odds ratios with 95 percent confidence intervals). The exposure and outcome columns from the output from mr() contain both the trait names and trait ids. The combine_all_mrresults() function splits these into separate columns by default.

Usage

```
combine_all_mrresults(
  res,
  het,
  plt,
  sin,
  ao_slc = TRUE,
  Exp = FALSE,
  split.exposure = FALSE,
  split.outcome = FALSE
)
```

Arguments

res	Results from mr().
het	Results from mr_heterogeneity().
plt	Results from mr_pleiotropy_test().
sin	Results from mr_singlesnp().
ao_slc	Logical; if set to TRUE then outcome study level characteristics are retrieved from available_outcomes(). Default is TRUE.
Exp	Logical; if set to TRUE results are exponentiated. Useful if user wants log odds ratios expressed as odds ratios. Default is FALSE.
split.exposure	Logical; if set to TRUE the exposure column is split into separate columns for the exposure name and exposure ID. Default is FALSE.
split.outcome	Logical; if set to TRUE the outcome column is split into separate columns for the outcome name and outcome ID. Default is FALSE.

Value

data frame

---

combine_data *Combine data*

Description

Taking exposure or outcome data (returned from format_data()) combine multiple datasets together so they can be analysed in one batch. Removes duplicate SNPs, preferentially keeping those usable in MR analysis.

Usage

```
combine_data(x)
```

Arguments

x	List of data frames returned from format_data().

Value

data frame

---

contingency *Obtain 2x2 contingency table from marginal parameters and odds ratio*

Description

Columns are the case and control frequencies. Rows are the frequencies for allele 1 and allele 2.

Usage

```
contingency(af, prop, odds_ratio, eps = 1e-15)
```

Arguments

af	Allele frequency of effect allele.
prop	Proportion of cases.
odds_ratio	Odds ratio.
eps	tolerance, default is 1e-15.

Value

2x2 contingency table as matrix

---

convert_outcome_to_exposure *Convert outcome data to exposure data*

Description

Helper function to convert results from extract_outcome_data() to exposure_dat format.

Usage

```
convert_outcome_to_exposure(outcome_dat)
```

Arguments

outcome_dat	Output from extract_outcome_data().

Value

data frame

---

dat_to_MRInput *Convert TwoSampleMR format to MendelianRandomization format*

Description

The MendelianRandomization package offers MR methods that can be used with the same data used in the TwoSampleMR package. This function converts from the TwoSampleMR format to the MRInput class.

Usage

```
dat_to_MRInput(dat, get_correlations = FALSE, pop = "EUR")
```

Arguments

dat	Output from the harmonise_data() function.
get_correlations	Default FALSE. If TRUE then extract the LD matrix for the SNPs from the European 1000 genomes data on the MR-Base server.
pop	If get_correlations is TRUE then use the following

Value

List of MRInput objects for each exposure/outcome combination

---

dat_to_RadialMR *Convert dat to RadialMR format*

Description

Creates a list of RadialMR format datasets for each exposure-outcome pair.

Usage

```
dat_to_RadialMR(dat)
```

Arguments

dat	Output from harmonise_data().

Value

List of RadialMR format datasets

default_parameters *List of parameters for use with MR functions*

Description

The default is `list(test_dist = "z", nboot = 1000, Cov = 0, penk = 20, phi = 1, alpha = 0.05, Qthresh = 0.05, over.dispersion = TRUE, loss.function = "huber")`.

Usage

```
default_parameters()
```

directionality_test *Perform MR Steiger test of directionality*

Description

A statistical test for whether the assumption that exposure causes outcome is valid.

Usage

```
directionality_test(dat)
```

Arguments

dat	Harmonised exposure and outcome data. Output from `harmonise_data()`.

Value

List

effective_n *Estimate the effective sample size in a case control study*

Description

Taken from https://www.nature.com/articles/nprot.2014.071

Usage

```
effective_n(ncase, ncontrol)
```

Arguments

ncase	Vector of number of cases
ncontrol	Vector of number of controls

Value

Vector of effective sample size

---

enrichment *Perform enrichment analysis*

Description

Perform enrichment analysis

Usage

```
enrichment(dat, method_list = enrichment_method_list()$obj)
```

Arguments

dat	Harmonised exposure and outcome data. Output from harmonise_data().
method_list	List of methods to use in analysis. Default is enrichment_method_list()$obj. See enrichment_method_list() for details.

Value

data frame

---

enrichment_method_list *Get list of available p-value enrichment methods*

Description

Get list of available p-value enrichment methods

Usage

```
enrichment_method_list()
```

Value

Data frame

`estimate_trait_sd` *Estimate trait SD by obtaining beta estimates from z-scores and finding the ratio with original beta values*

Description

Assumes that sample size and allele frequency is correct for each SNP, and that allele frequency gives a reasonable estimate of the variance of the SNP.

Usage

```
estimate_trait_sd(b, se, n, p)
```

Arguments

`b`	vector of effect sizes.
`se`	vector of standard errors.
`n`	vector of sample sizes.
`p`	vector of allele frequencies.

Value

Vector of sd estimates for each association.

`extract_instruments` *Find instruments for use in MR from the MR Base database*

Description

This function searches for GWAS significant SNPs (for a given p-value) for a specified set of outcomes. It then performs LD based clumping to return only independent significant associations.

Usage

```
extract_instruments(
  outcomes,
  p1 = 5e-08,
  clump = TRUE,
  p2 = 5e-08,
  r2 = 0.001,
  kb = 10000,
  opengwas_jwt = ieugwasr::get_opengwas_jwt(),
  force_server = FALSE
)
```

Arguments

outcomes	Array of outcome IDs (see available_outcomes()).
p1	Significance threshold. The default is 5e-8.
clump	Logical; whether to clump results. The default is TRUE.
p2	Secondary clumping threshold. The default is 5e-8.
r2	Clumping r2 cut off. The default is 0.001.
kb	Clumping distance cutoff. The default is 10000.
opengwas_jwt	Used to authenticate protected endpoints. Login to https://api.opengwas.io to obtain a jwt. Provide the jwt string here, or store in .Renviron under the keyname OPENGWAS_JWT.
force_server	Force the analysis to extract results from the server rather than the MRInstruments package.

Value

data frame

---

extract_outcome_data *Supply the output from read_exposure_data() and all the SNPs therein will be queried against the requested outcomes in remote database using API.*

Description

Supply the output from read_exposure_data() and all the SNPs therein will be queried against the requested outcomes in remote database using API.

Usage

```
extract_outcome_data(
  snps,
  outcomes,
  proxies = TRUE,
  rsq = 0.8,
  align_alleles = 1,
  palindromes = 1,
  maf_threshold = 0.3,
  opengwas_jwt = ieugwasr::get_opengwas_jwt(),
  splitsize = 10000,
  proxy_splitsize = 500
)
```

Arguments

snps	Array of SNP rs IDs.
outcomes	Array of IDs (see id column in output from available_outcomes()).
proxies	Look for LD tags? Default is TRUE.
rsq	Minimum LD rsq value (if proxies = 1). Default = 0.8.
align_alleles	Try to align tag alleles to target alleles (if proxies = 1). 1 = yes, 0 = no. The default is 1.
palindromes	Allow palindromic SNPs (if proxies = 1). 1 = yes, 0 = no. The default is 1.
maf_threshold	MAF threshold to try to infer palindromic SNPs. The default is 0.3.
opengwas_jwt	Used to authenticate protected endpoints. Login to https://api.opengwas.io to obtain a jwt. Provide the jwt string here, or store in .Renviron under the keyname OPENGWAS_JWT.
splitsize	The default is 10000.
proxy_splitsize	The default is 500.

Value

Dataframe of summary statistics for all available outcomes

---

fishers_combined_test *Fisher's combined test*

Description

Fisher's combined test

Usage

```
fishers_combined_test(pval)
```

Arguments

pval	Vector of outcome p-values

Value

List with the following elements:

b MR estimate

se Standard error

pval p-value

forest_plot *Forest plot for multiple exposures and multiple outcomes*

Description

Perform MR of multiple exposures and multiple outcomes. This plots the results.

Usage

```
forest_plot(
  mr_res,
  exponentiate = FALSE,
  single_snp_method = "Wald ratio",
  multi_snp_method = "Inverse variance weighted",
  group_single_categories = TRUE,
  by_category = TRUE,
  in_columns = FALSE,
  threshold = NULL,
  xlab = "",
  xlim = NULL,
  trans = "identity",
  ao_slc = TRUE,
  priority = "Cardiometabolic"
)
```

Arguments

mr_res	Results from mr().
exponentiate	Convert effects to OR? Default is FALSE.
single_snp_method	Which of the single SNP methosd to use when only 1 SNP was used to estimate the causal effect? The default is "Wald ratio".
multi_snp_method	Which of the multi-SNP methods to use when there was more than 1 SNPs used to estimate the causal effect? The default is "Inverse variance weighted".
group_single_categories	If there are categories with only one outcome, group them together into an "Other" group. The default is TRUE.
by_category	Separate the results into sections by category? The default is TRUE.
in_columns	Separate the exposures into different columns. The default is FALSE.
threshold	p-value threshold to use for colouring points by significance level. If NULL (default) then colour layer won't be applied.
xlab	x-axis label. If in_columns=TRUE then the exposure values are appended to the end of xlab. e.g. if xlab="Effect of" then x-labels will read "Effect of exposure1", "Effect of exposure2" etc. Otherwise will be printed as is.

xlim	limit x-axis range. Provide vector of length 2, with lower and upper bounds. The default is NULL.
trans	Transformation to apply to x-axis. e.g. "identity", "log2", etc. The default is "identity".
ao_slc	retrieve sample size and subcategory from available_outcomes(). If set to FALSE then mr_res must contain the following additional columns: sample_size and subcategory. The default behaviour is to use available_outcomes() to retrieve sample size and subcategory.
priority	Name of category to prioritise at the top of the forest plot. The default is "Cardiometabolic".

Value

grid plot object

`forest_plot_1_to_many` *1-to-many forest plot*

Description

Plot results from an analysis of multiple exposures against a single outcome or a single exposure against multiple outcomes. Plots effect estimates and 95 percent confidence intervals. The ordering of results in the plot is determined by the order supplied by the user. Users may find sort_1_to_many() helpful for sorting their results prior to using the 1-to-many forest plot. The plot function works best for 50 results and is not designed to handle more than 100 results.

Usage

```
forest_plot_1_to_many(
  mr_res = "mr_res",
  b = "b",
  se = "se",
  TraitM = "outcome",
  col1_width = 1,
  col1_title = "",
  exponentiate = FALSE,
  trans = "identity",
  ao_slc = TRUE,
  lo = NULL,
  up = NULL,
  by = NULL,
  xlab = "Effect (95% confidence interval)",
  addcols = NULL,
  addcol_widths = NULL,
  addcol_titles = "",
  subheading_size = 6,
```

```
  shape_points = 15,
  colour_scheme = "black",
  col_text_size = 5,
  weight = NULL
)
```

Arguments

mr_res	Data frame of results supplied by the user. The default is "mr_res".
b	Name of the column specifying the effect of the exposure on the outcome. The default is "b".
se	Name of the column specifying the standard error for b. The default is "se".
TraitM	The column specifying the names of the traits. Corresponds to 'many' in the 1-to-many forest plot. The default is "outcome".
col1_width	Width of Y axis label for the column specified by the TraitM argument. The default is 1.
col1_title	Title for the column specified by the TraitM argument. The default is "".
exponentiate	Convert log odds ratios to odds ratios? Default is FALSE.
trans	Specify x-axis scale. e.g. "identity", "log2", etc. If set to "identity" an additive scale is used. If set to log2 the x-axis is plotted on a multiplicative / doubling scale (preferable when plotting odds ratios). Default is "identity".
ao_slc	Logical; retrieve trait subcategory information using available_outcomes(). Default is FALSE.
lo	Lower limit of X axis to plot.
up	upper limit of X axis to plot.
by	Name of the grouping variable to stratify results on. Default is NULL.
xlab	X-axis label, default is "Effect (95% confidence interval)".
addcols	Name of additional columns to plot. Character vector. The default is NULL.
addcol_widths	Widths of Y axis labels for additional columns specified by the addcols argument. Numeric vector. The default is NULL.
addcol_titles	Titles of additional columns specified by the addcols argument. Character vector. The default is NULL.
subheading_size	text size for the subheadings specified in by argument. The default is 6.
shape_points	the shape of the data points to pass to ggplot2::geom_point(). Default is set to 15 (filled square).
colour_scheme	the general colour scheme for the plot. Default is to make all text and data points "black".
col_text_size	The default is 5.
weight	The default is NULL.

Value

grid plot object

`forest_plot_basic2` *A basic forest plot*

Description

This function is used to create a basic forest plot. It requires the output from `format_1_to_many()`.

Usage

```
forest_plot_basic2(
  dat,
  section = NULL,
  colour_group = NULL,
  colour_group_first = TRUE,
  xlab = NULL,
  bottom = TRUE,
  trans = "identity",
  xlim = NULL,
  lo = lo,
  up = up,
  subheading_size = subheading_size,
  colour_scheme = "black",
  shape_points = 15
)
```

Arguments

`dat`	Output from `format_1_to_many()`
`section`	Which category in dat to plot. If `NULL` then prints everything.
`colour_group`	Which exposure to plot. If `NULL` then prints everything grouping by colour.
`colour_group_first`	The default is `TRUE`.
`xlab`	x-axis label. Default=`NULL`.
`bottom`	Show x-axis? Default=`FALSE`.
`trans`	x-axis scale.
`xlim`	x-axis limits.
`lo`	Lower limit of x axis.
`up`	Upper limit of x axis.
`subheading_size`	text size for the subheadings. The subheadings correspond to the values of the section argument.
`colour_scheme`	the general colour scheme for the plot. Default is to make all text and data points `"black"`.
`shape_points`	the shape of the data points to pass to `ggplot2::geom_point()`. Default is set to 15 (filled square).

Value

ggplot object

`format_1_to_many` *Format MR results for a 1-to-many forest plot*

Description

This function formats user-supplied results for the `forest_plot_1_to_many()` function. The user supplies their results in the form of a data frame. The data frame is assumed to contain at least three columns of data:

1. effect estimates, from an analysis of the effect of an exposure on an outcome;
2. standard errors for the effect estimates; and
3. a column of trait names, corresponding to the 'many' in a 1-to-many forest plot.

Usage

```
format_1_to_many(
  mr_res,
  b = "b",
  se = "se",
  exponentiate = FALSE,
  ao_slc = FALSE,
  by = NULL,
  TraitM = "outcome",
  addcols = NULL,
  weight = NULL
)
```

Arguments

`mr_res`	Data frame of results supplied by the user.
`b`	Name of the column specifying the effect of the exposure on the outcome. Default = `"b"`.
`se`	Name of the column specifying the standard error for b. Default = `"se"`.
`exponentiate`	Convert log odds ratios to odds ratios? Default=`FALSE`.
`ao_slc`	Logical; retrieve trait subcategory information using `available_outcomes()`. Default=`FALSE`.
`by`	Name of the column indicating a grouping variable to stratify results on. Default=`NULL`.
`TraitM`	The column specifying the names of the traits. Corresponds to 'many' in the 1-to-many forest plot. Default=`"outcome"`.
`addcols`	Name of any additional columns to add to the plot. Character vector. The default is `NULL`.
`weight`	The default is `NULL`.

Value

data frame.

`format_aries_mqtl` *Get data from methylation QTL results*

Description

See `format_data()`.

Usage

```
format_aries_mqtl(aries_mqtl_subset, type = "exposure")
```

Arguments

`aries_mqtl_subset`	Selected rows from `aries_mqtl` data loaded from `MRInstruments` package.
`type`	Are these data used as `"exposure"` or `"outcome"`? Default is `"exposure"`.

Value

Data frame

`format_data` *Read exposure or outcome data*

Description

Reads in exposure data. Checks and organises columns for use with MR or enrichment tests. Infers p-values when possible from beta and se.

Usage

```
format_data(
  dat,
  type = "exposure",
  snps = NULL,
  header = TRUE,
  phenotype_col = "Phenotype",
  snp_col = "SNP",
  beta_col = "beta",
  se_col = "se",
  eaf_col = "eaf",
  effect_allele_col = "effect_allele",
```

```
  other_allele_col = "other_allele",
  pval_col = "pval",
  units_col = "units",
  ncase_col = "ncase",
  ncontrol_col = "ncontrol",
  samplesize_col = "samplesize",
  gene_col = "gene",
  id_col = "id",
  min_pval = 1e-200,
  z_col = "z",
  info_col = "info",
  chr_col = "chr",
  pos_col = "pos",
  log_pval = FALSE
)
```

Arguments

dat	Data frame. Must have header with at least SNP column present.
type	Is this the exposure or the outcome data that is being read in? The default is "exposure".
snps	SNPs to extract. If NULL then doesn't extract any and keeps all. The default is NULL.
header	The default is TRUE.
phenotype_col	Optional column name for the column with phenotype name corresponding the the SNP. If not present then will be created with the value "Outcome". The default is "Phenotype".
snp_col	Required name of column with SNP rs IDs. The default is "SNP".
beta_col	Required for MR. Name of column with effect sizes. The default is "beta".
se_col	Required for MR. Name of column with standard errors. The default is "se".
eaf_col	Required for MR. Name of column with effect allele frequency. The default is "eaf".
effect_allele_col	Required for MR. Name of column with effect allele. Must contain only the characters "A", "C", "T" or "G". The default is "effect_allele".
other_allele_col	Required for MR. Name of column with non effect allele. Must contain only the characters "A", "C", "T" or "G". The default is "other_allele".
pval_col	Required for enrichment tests. Name of column with p-value. The default is "pval".
units_col	Optional column name for units. The default is "units".
ncase_col	Optional column name for number of cases. The default is "ncase".
ncontrol_col	Optional column name for number of controls. The default is "ncontrol".
samplesize_col	Optional column name for sample size. The default is "samplesize".

gene_col	Optional column name for gene name. The default is "gene".
id_col	The default is "id".
min_pval	Minimum allowed p-value. The default is 1e-200.
z_col	The default is "z".
info_col	The default is "info_col".
chr_col	The default is "chr_col".
pos_col	The default is "pos".
log_pval	The pval is -log10(P). The default is FALSE.

Value

data frame

`format_gtex_eqtl` *Get data from eQTL catalog into correct format*

Description

See `format_data()`.

Usage

```
format_gtex_eqtl(gtex_eqtl_subset, type = "exposure")
```

Arguments

gtex_eqtl_subset	Selected rows from gtex_eqtl data loaded from MRInstruments package.
type	Are these data used as "exposure" or "outcome"? Default is "exposure".

Value

Data frame

format_gwas_catalog *Get data selected from GWAS catalog into correct format*

Description

DEPRECATED. Please use `format_data()` instead.

Usage

```
format_gwas_catalog(gwas_catalog_subset, type = "exposure")
```

Arguments

gwas_catalog_subset	The GWAS catalog subset.
type	The default is "exposure".

Value

Data frame

Examples

```
## Not run:
require(MRInstruments)
data(gwas_catalog)
bmi <- subset(gwas_catalog, Phenotype=="Body mass index" & Year==2010 & grepl("kg", Units))
bmi <- format_data(bmi)

## End(Not run)
```

format_metab_qtls *Get data from metabolomic QTL results*

Description

See `format_data()`.

Usage

```
format_metab_qtls(metab_qtls_subset, type = "exposure")
```

Arguments

metab_qtls_subset	Selected rows from metab_qtls data loaded from MRInstruments package.
type	Are these data used as "exposure" or "outcome"? Default is "exposure".

Value

Data frame

`format_mr_results` *Format MR results for forest plot*

Description

This function takes the results from `mr()` and is particularly useful if the MR has been applied using multiple exposures and multiple outcomes. It creates a new data frame with the following:

- Variables: exposure, outcome, category, outcome sample size, effect, upper ci, lower ci, pval, nsnp
- only one estimate for each exposure-outcome
- exponentiated effects if required

Usage

```
format_mr_results(
  mr_res,
  exponentiate = FALSE,
  single_snp_method = "Wald ratio",
  multi_snp_method = "Inverse variance weighted",
  ao_slc = TRUE,
  priority = "Cardiometabolic"
)
```

Arguments

`mr_res`	Results from `mr()`.
`exponentiate`	Convert effects to OR? The default is `FALSE`.
`single_snp_method`	Which of the single SNP methods to use when only 1 SNP was used to estimate the causal effect? The default is `"Wald ratio"`.
`multi_snp_method`	Which of the multi-SNP methods to use when there was more than 1 SNPs used to estimate the causal effect? The default is `"Inverse variance weighted"`.
`ao_slc`	Logical; retrieve sample size and subcategory using `available_outcomes()`. If set to `FALSE` `mr_res` must contain the following additional columns: `subcategory` and `sample_size`.
`priority`	Name of category to prioritise at the top of the forest plot. The default is `"Cardiometabolic"`.

Details

By default it uses the available_outcomes() function to retrieve the study level characteristics for the outcome trait, including sample size and outcome category. This assumes the MR analysis was performed using outcome GWAS(s) contained in MR-Base.

If ao_slc is set to TRUE then the user must supply their own study level characteristics. This is useful when the user has supplied their own outcome GWAS results (i.e. they are not in MR-Base).

Value

data frame.

---

format_proteomic_qtls *Get data from proteomic QTL results*

Description

See format_data().

Usage

```
format_proteomic_qtls(proteomic_qtls_subset, type = "exposure")
```

Arguments

proteomic_qtls_subset	Selected rows from proteomic_qtls data loaded from MRInstruments package.
type	Are these data used as "exposure" or "outcome"? Default is "exposure".

Value

Data frame

---

generate_odds_ratios *Generate odds ratios*

Description

This function takes b and se from mr() and generates odds ratios and 95 percent confidence intervals.

Usage

```
generate_odds_ratios(mr_res)
```

Arguments

mr_res	Results from mr().

Value

data frame

---

get_population_allele_frequency *Estimate the allele frequency in population from case/control summary data*

Description

Estimate the allele frequency in population from case/control summary data

Usage

```
get_population_allele_frequency(af, prop, odds_ratio, prevalence)
```

Arguments

af	Effect allele frequency (or MAF)
prop	Proportion of samples that are cases
odds_ratio	Odds ratio
prevalence	Population disease prevalence

Value

Population allele frequency

---

get_p_from_r2n *Calculate p-value from R-squared and sample size*

Description

Calculate p-value from R-squared and sample size

Usage

```
get_p_from_r2n(r2, n)
```

Arguments

r2	Rsq
n	Sample size

Value

P-value

`get_r_from_bsen` *Estimate R-squared from beta, standard error and sample size*

Description

Estimate R-squared from beta, standard error and sample size

Usage

```
get_r_from_bsen(b, se, n)
```

Arguments

`b`	Array of effect sizes
`se`	Array of standard errors
`n`	Array of (effective) sample sizes

Value

Vector of signed r values

`get_r_from_lor` *Estimate proportion of variance of liability explained by SNP in general population*

Description

This uses equation 10 in Lee et al. A Better Coefficient of Determination for Genetic Profile Analysis. Genetic Epidemiology 36: 214–224 (2012) doi:10.1002/gepi.21614.

Usage

```
get_r_from_lor(
  lor,
  af,
  ncase,
  ncontrol,
  prevalence,
  model = "logit",
  correction = FALSE
)
```

Arguments

lor	Vector of Log odds ratio.
af	Vector of allele frequencies.
ncase	Vector of Number of cases.
ncontrol	Vector of Number of controls.
prevalence	Vector of Disease prevalence in the population.
model	Is the effect size estimated from the "logit" (default) or "probit" model.
correction	Scale the estimated r by correction value. The default is FALSE.

Value

Vector of signed r values

---

get_r_from_pn *Calculate variance explained from p-values and sample size*

Description

This method is an approximation, and may be numerically unstable. Ideally you should estimate r directly from independent replication samples. Use get_r_from_lor() for binary traits.

Usage

```
get_r_from_pn(p, n)
```

Arguments

p	Array of pvals
n	Array of sample sizes

Value

Vector of r values (all arbitrarily positive)

get_se *Get SE from effect size and p-value*

Description

Get SE from effect size and p-value

Usage

```
get_se(eff, pval)
```

Arguments

eff	effect size
pval	p-values

Value

array

harmonise_data *Harmonise the alleles and effects between the exposure and outcome*

Description

In order to perform MR the effect of a SNP on an outcome and exposure must be harmonised to be relative to the same allele.

Usage

```
harmonise_data(exposure_dat, outcome_dat, action = 2)
```

Arguments

exposure_dat	Output from read_exposure_data().
outcome_dat	Output from extract_outcome_data().
action	Level of strictness in dealing with SNPs. • action = 1: Assume all alleles are coded on the forward strand, i.e. do not attempt to flip alleles • action = 2: Try to infer positive strand alleles, using allele frequencies for palindromes (default, conservative); • action = 3: Correct strand for non-palindromic SNPs, and drop all palindromic SNPs from the analysis (more conservative). If a single value is passed then this action is applied to all outcomes. But multiple values can be supplied as a vector, each element relating to a different outcome.

Details

Expects data in the format generated by `read_exposure_data()` and `extract_outcome_data()`. This means the inputs must be dataframes with the following columns:

`outcome_dat`:

- `SNP`
- `beta.outcome`
- `se.outcome`
- `effect_allele.outcome`
- `other_allele.outcome`
- `eaf.outcome`
- `outcome`

`exposure_dat`:

- `SNP`
- `beta.exposure`
- `se.exposure`
- `effect_allele.exposure`
- `other_allele.exposure`
- `eaf.exposure`

The function tries to harmonise INDELs. If they are coded as sequence strings things work more smoothly. If they are coded as D/I in one dataset it will try to convert them to sequences if the other dataset has adequate information. If coded as D/I in one dataset and as a variant with equal length INDEL alleles in the other, the variant is dropped. If one or both the datasets only has one allele (i.e. the effect allele) then harmonisation is naturally going to be more ambiguous and more variants will be dropped.

Value

Data frame with harmonised effects and alleles

---

harmonise_ld_dat *Harmonise LD matrix against summary data*

Description

LD matrix returns with rsid_ea_oa identifiers. Make sure that they are oriented to the same effect allele as the summary dataset. Summary dataset can be exposure dataset or harmonised dartaset.

Usage

```
harmonise_ld_dat(x, ld)
```

Arguments

x	Exposure dataset or harmonised dataset
ld	Output from ld_matrix()

Value

List of exposure dataset and harmonised LD matrix

Isq *I-squared calculation*

Description

This function calculates the I^2 statistic. To use it for the I^2_{GX} metric ensure that the effects are all the same sign (e.g. abs(y)).

Usage

```
Isq(y, s)
```

Arguments

y	Vector of effects.
s	Vector of standard errors.

Value

Isq value

---

ldsc_h2 *Univariate LDSC*

Description

Imported here to help estimate sample overlap between studies

Usage

```
ldsc_h2(id, ancestry = "infer", snpinfo = NULL, splitsize = 20000)
```

Arguments

id	ID to analyse
ancestry	ancestry of traits 1 and 2 (AFR, AMR, EAS, EUR, SAS) or 'infer' (default) in which case it will try to guess based on allele frequencies
snpinfo	Output from `ieugwasr::afl2_list("hapmap3")`, or `NULL` for it to be done automatically
splitsize	How many SNPs to extract at one time. Default=`20000`

Value

model fit

References

Bulik-Sullivan,B.K. et al. (2015) An atlas of genetic correlations across human diseases and traits. Nat. Genet. 47, 1236–1241.

Guo,B. and Wu,B. (2018) Principal component based adaptive association test of multiple traits using GWAS summary statistics. bioRxiv 269597; doi: 10.1101/269597

Gua,B. and Wu,B. (2019) Integrate multiple traits to detect novel trait-gene association using GWAS summary data with an adaptive test approach. Bioinformatics. 2019 Jul 1;35(13):2251-2257. doi: 10.1093/bioinformatics/bty961.

https://github.com/baolinwu/MTAR

---

ldsc_rg *Bivariate LDSC*

Description

Imported here to help estimate sample overlap between studies

Usage

```
ldsc_rg(id1, id2, ancestry = "infer", snpinfo = NULL, splitsize = 20000)
```

Arguments

id1	ID 1 to analyse
id2	ID 2 to analyse
ancestry	ancestry of traits 1 and 2 (AFR, AMR, EAS, EUR, SAS) or 'infer' (default) in which case it will try to guess based on allele frequencies
snpinfo	Output from `ieugwasr::afl2_list("hapmap3")`, or `NULL` for it to be done automatically
splitsize	How many SNPs to extract at one time. Default=`20000`

Value

model fit

`ld_matrix` *Get LD matrix for list of SNPs*

Description

This function takes a list of SNPs and searches for them in a specified super-population in the 1000 Genomes phase 3 reference panel. It then creates an LD matrix of r values (signed, and not squared). All LD values are with respect to the major alleles in the 1000G dataset. You can specify whether the allele names are displayed.

Usage

```
ld_matrix(snps, with_alleles = TRUE, pop = "EUR")
```

Arguments

`snps`	List of SNPs.
`with_alleles`	Whether to append the allele names to the SNP names. The default is `TRUE`.
`pop`	Super-population to use as reference panel. Default = `"EUR"`. Options are `"EUR"`, `"SAS"`, `"EAS"`, `"AFR"`, `"AMR"`. `'legacy'` also available - which is a previously used version of the EUR panel with a slightly different set of markers.

Details

The data used for generating the LD matrix includes only bi-allelic SNPs with MAF > 0.01, so it's quite possible that a variant you want to include will be absent. If it is absent, it will be automatically excluded from the results.

You can check if your variants are present in the LD reference panel using `ieugwasr::ld_reflookup()`.

This function does put load on the OpenGWAS servers, which makes life more difficult for other users, and has been limited to analyse only up to 500 variants at a time. We have implemented a method and made available the LD reference panels to perform the operation locally, see `ieugwasr::ld_matrix()` and related vignettes for details.

Value

Matrix of LD r values

make_dat *Convenient function to create a harmonised dataset*

Description

Convenient function to create a harmonised dataset.

Usage

```
make_dat(
  exposures = c("ieu-a-2", "ieu-a-301"),
  outcomes = c("ieu-a-7", "ieu-a-1001"),
  proxies = TRUE
)
```

Arguments

exposures	The default is c("ieu-a-2", "ieu-a-301") (BMI and LDL).
outcomes	The default is c("ieu-a-7", "ieu-a-1001") (CHD and EDU).
proxies	Look for proxies? Default = TRUE

Value

Harmonised data frame

mr *Perform all Mendelian randomization tests*

Description

Perform all Mendelian randomization tests

Usage

```
mr(
  dat,
  parameters = default_parameters(),
  method_list = subset(mr_method_list(), use_by_default)$obj
)
```

Arguments

dat	Harmonised exposure and outcome data. Output from harmonise_data().
parameters	Parameters to be used for various MR methods. Default is output from default_parameters().
method_list	List of methods to use in analysis. See mr_method_list() for details.

Value

List with the following elements:

mr Table of MR results

extra Table of extra results

`mr_density_plot` *Density plot*

Description

Density plot

Usage

```
mr_density_plot(
  singlesnp_results,
  mr_results,
  exponentiate = FALSE,
  bandwidth = "nrd0"
)
```

Arguments

`singlesnp_results`	from `mr_singlesnp()`.
`mr_results`	Results from `mr()`.
`exponentiate`	Plot on exponentiated scale. The default is `FALSE`.
`bandwidth`	Density bandwidth parameter.

Value

List of plots

`mr_egger_regression` *Egger's regression for Mendelian randomization*

Description

Egger's regression for Mendelian randomization

Usage

```
mr_egger_regression(b_exp, b_out, se_exp, se_out, parameters)
```

Arguments

`b_exp`	Vector of genetic effects on exposure.
`b_out`	Vector of genetic effects on outcome.
`se_exp`	Standard errors of genetic effects on exposure.
`se_out`	Standard errors of genetic effects on outcome.
`parameters`	List of parameters.

Value

List of with the following elements:

b MR estimate

se Standard error of MR estimate

pval p-value of MR estimate

b_i Estimate of horizontal pleiotropy (intercept)

se_i Standard error of intercept

pval_i p-value of intercept

Q, Q_df, Q_pval Heterogeneity stats

mod Summary of regression

dat Original data used for MR Egger regression

`mr_egger_regression_bootstrap` *Run bootstrap to generate standard errors for MR*

Description

Run bootstrap to generate standard errors for MR

Usage

```
mr_egger_regression_bootstrap(b_exp, b_out, se_exp, se_out, parameters)
```

Arguments

`b_exp`	Vector of genetic effects on exposure.
`b_out`	Vector of genetic effects on outcome.
`se_exp`	Standard errors of genetic effects on exposure.
`se_out`	Standard errors of genetic effects on outcome.
`parameters`	List of parameters. Specifically, the `nboot` parameter can be specified for the number of bootstrap replications. The default is `parameters=list(nboot=1000)`.

Value

List of with the following elements:

b MR estimate

se Standard error of MR estimate

pval p-value of MR estimate

b_i Estimate of horizontal pleiotropy (intercept)

se_i Standard error of intercept

pval_i p-value of intercept

mod Summary of regression

dat Original data used for MR Egger regression

mr_forest_plot *Forest plot*

Description

Forest plot

Usage

```
mr_forest_plot(singlesnp_results, exponentiate = FALSE)
```

Arguments

singlesnp_results	from mr_singlesnp().
exponentiate	Plot on exponential scale. The default is FALSE.

Value

List of plots

mr_funnel_plot *Funnel plot*

Description

Create funnel plot from single SNP analyses.

Usage

```
mr_funnel_plot(singlesnp_results)
```

Arguments

singlesnp_results	from mr_singlesnp().

Value

List of plots

mr_heterogeneity *Get heterogeneity statistics*

Description

Get heterogeneity statistics.

Usage

```
mr_heterogeneity(
  dat,
  parameters = default_parameters(),
  method_list = subset(mr_method_list(), heterogeneity_test & use_by_default)$obj
)
```

Arguments

dat	Harmonised exposure and outcome data. Output from harmonise_data().
parameters	Parameters to be used for various MR methods. Default is output from default_parameters().
method_list	List of methods to use in analysis. See mr_method_list() for details.

Value

Data frame

mr_ivw *Inverse variance weighted regression*

Description

The default multiplicative random effects IVW estimate. The standard error is corrected for under dispersion Use the mr_ivw_mre() function for estimates that don't correct for under dispersion.

Usage

```
mr_ivw(b_exp, b_out, se_exp, se_out, parameters = default_parameters())
```

Arguments

b_exp	Vector of genetic effects on exposure.
b_out	Vector of genetic effects on outcome.
se_exp	Standard errors of genetic effects on exposure.
se_out	Standard errors of genetic effects on outcome.
parameters	List of parameters.

Value

List with the following elements:

b MR estimate

se Standard error

pval p-value

Q, Q_df, Q_pval Heterogeneity stats

`mr_ivw_fe` *Inverse variance weighted regression (fixed effects)*

Description

Inverse variance weighted regression (fixed effects)

Usage

```
mr_ivw_fe(b_exp, b_out, se_exp, se_out, parameters = default_parameters())
```

Arguments

`b_exp`	Vector of genetic effects on exposure.
`b_out`	Vector of genetic effects on outcome.
`se_exp`	Standard errors of genetic effects on exposure.
`se_out`	Standard errors of genetic effects on outcome.
`parameters`	List of parameters.

Value

List with the following elements:

b MR estimate

se Standard error

pval p-value

Q, Q_df, Q_pval Heterogeneity stats

mr_ivw_mre — *Inverse variance weighted regression (multiplicative random effects model)*

Description

Same as `mr_ivw()` but no correction for under dispersion.

Usage

```
mr_ivw_mre(b_exp, b_out, se_exp, se_out, parameters = default_parameters())
```

Arguments

b_exp	Vector of genetic effects on exposure.
b_out	Vector of genetic effects on outcome.
se_exp	Standard errors of genetic effects on exposure.
se_out	Standard errors of genetic effects on outcome.
parameters	List of parameters.

Value

List with the following elements:

b MR estimate

se Standard error

pval p-value

Q, Q_df, Q_pval Heterogeneity stats

mr_ivw_radial — *Radial IVW analysis*

Description

Radial IVW analysis

Usage

```
mr_ivw_radial(b_exp, b_out, se_exp, se_out, parameters = default_parameters())
```

Arguments

b_exp	Vector of genetic effects on exposure.
b_out	Vector of genetic effects on outcome.
se_exp	Standard errors of genetic effects on exposure.
se_out	Standard errors of genetic effects on outcome.
parameters	List of parameters.

Value

List with the following elements:

b causal effect estimate

se standard error

pval p-value

---

mr_leaveoneout *Leave one out sensitivity analysis*

Description

Leave one out sensitivity analysis

Usage

```
mr_leaveoneout(dat, parameters = default_parameters(), method = mr_ivw)
```

Arguments

dat	Output from harmonise_data().
parameters	List of parameters.
method	Choose which method to use. The default is mr_ivw.

Value

List of data frames

mr_leaveoneout_plot *Plot results from leaveoneout analysis*

Description

Plot results from leaveoneout analysis.

Usage

```
mr_leaveoneout_plot(leaveoneout_results)
```

Arguments

leaveoneout_results	Output from mr_leaveoneout().

Value

List of plots

mr_median *MR median estimators*

Description

MR median estimators

Usage

```
mr_median(dat, parameters = default_parameters())
```

Arguments

dat	Output from harmonise_data().
parameters	List of parameters. The default is default_parameters().

Value

data frame

mr_meta_fixed *Perform 2 sample IV using fixed effects meta analysis and delta method for standard errors*

Description

Perform 2 sample IV using fixed effects meta analysis and delta method for standard errors

Usage

```
mr_meta_fixed(b_exp, b_out, se_exp, se_out, parameters)
```

Arguments

b_exp	Vector of genetic effects on exposure.
b_out	Vector of genetic effects on outcome.
se_exp	Standard errors of genetic effects on exposure.
se_out	Standard errors of genetic effects on outcome.
parameters	List of parameters.

Value

List with the following elements:

b causal effect estimate

se standard error

pval p-value

Q, Q_df, Q_pval Heterogeneity stats

mr_meta_fixed_simple *Perform 2 sample IV using simple standard error*

Description

Perform 2 sample IV using simple standard error

Usage

```
mr_meta_fixed_simple(b_exp, b_out, se_exp, se_out, parameters)
```

Arguments

b_exp	Vector of genetic effects on exposure.
b_out	Vector of genetic effects on outcome.
se_exp	Standard errors of genetic effects on exposure.
se_out	Standard errors of genetic effects on outcome.
parameters	List of parameters.

Value

List with the following elements:

b causal effect estimate

se standard error

pval p-value

---

mr_meta_random — *Perform 2 sample IV using random effects meta analysis and delta method for standard errors*

Description

Perform 2 sample IV using random effects meta analysis and delta method for standard errors

Usage

```
mr_meta_random(b_exp, b_out, se_exp, se_out, parameters)
```

Arguments

b_exp	Vector of genetic effects on exposure.
b_out	Vector of genetic effects on outcome.
se_exp	Standard errors of genetic effects on exposure.
se_out	Standard errors of genetic effects on outcome.
parameters	List of parameters.

Value

List with the following elements:

b causal effect estimate

se standard error

pval p-value

Q, Q_df, Q_pval Heterogeneity stats

mr_method_list *Get list of available MR methods*

Description

Get list of available MR methods

Usage

```
mr_method_list()
```

Value

character vector of method names

mr_mode *MR mode estimators*

Description

Perform simple, weighted, penalised modes, as well as versions that use the NOME assumption.

Usage

```
mr_mode(dat, parameters = default_parameters(), mode_method = "all")
```

Arguments

dat	Output from harmonise_data().
parameters	List of parameters. The default is default_parameters().
mode_method	The default is "all". The other choices are 'Simple mode', 'Weighted mode', 'Penalised mode', 'Simple mode (NOME)', 'Weighted mode (NOME)'.

Value

data frame

mr_moe *Mixture of experts*

Description

Based on the method described here https://www.biorxiv.org/content/10.1101/173682v2. Once all MR methods have been applied to a summary set, you can then use the mixture of experts to predict the method most likely to be the most accurate.

Usage

```
mr_moe(res, rf)
```

Arguments

res	Output from mr_wrapper().
rf	The trained random forest for the methods. This is available to download at https://www.dropbox.com/s/5la7y38od95swcf/rf.rdata?dl=0.

Details

The mr_moe() function modifies the estimates item in the list of results from the mr_wrapper() function. It does three things:

1. Adds the MOE column, which is a predictor for each method for how well it performs in terms of high power and low type 1 error (scaled 0-1, where 1 is best performance).
2. It renames the methods to be the estimating method + the instrument selection method. There are 4 instrument selection methods: Tophits (i.e. no filtering), directional filtering (DF, an unthresholded version of Steiger filtering), heterogeneity filtering (HF, removing instruments that make substantial ($p < 0.05$) contributions to Cochran's Q statistic), and DF + HF which is where DF is applied and the HF applied on top of that.
3. It orders the table to be in order of best performing method.

Note that the mixture of experts has only been trained on datasets with at least 5 SNPs. If your dataset has fewer than 5 SNPs this function might return errors.

Value

List

Examples

```
## Not run:
# Example of body mass index on coronary heart disease
# Extract and harmonise data
a <- extract_instruments("ieu-a-2")
b <- extract_outcome_data(a$SNP, 7)
dat <- harmonise_data(a, b)
```

```
# Apply all MR methods
r <- mr_wrapper(dat)

# Load the rf object containing the trained models
load("rf.rdata")
# Update the results with mixture of experts
r <- mr_moe(r, rf)

# Now you can view the estimates, and see that they have
# been sorted in order from most likely to least likely to
# be accurate, based on MOE prediction
r[[1]]$estimates

## End(Not run)
```

`mr_penalised_weighted_median` *Penalised weighted median MR*

Description

Modification to standard weighted median MR Updated based on Burgess 2016 "Robust instrumental variable methods using multiple candidate instruments with application to Mendelian randomization"

Usage

```
mr_penalised_weighted_median(
  b_exp,
  b_out,
  se_exp,
  se_out,
  parameters = default_parameters()
)
```

Arguments

`b_exp`	Vector of genetic effects on exposure
`b_out`	Vector of genetic effects on outcome
`se_exp`	Standard errors of genetic effects on exposure
`se_out`	Standard errors of genetic effects on outcome
`parameters`	List containing `penk` - Constant term in penalisation, and `nboot` - number of bootstrap replications to calculate SE. `default_parameters()` sets `parameters=list(penk=20, nboot=1000)`.

Value

List with the following elements:

b MR estimate
se Standard error
pval p-value

---

mr_pleiotropy_test *Test for horizontal pleiotropy in MR analysis*

Description

Performs MR Egger and returns intercept values.

Usage

```
mr_pleiotropy_test(dat)
```

Arguments

dat	Harmonised exposure and outcome data. Output from harmonise_data().

Value

data frame

---

mr_raps *Robust adjusted profile score*

Description

Robust adjusted profile score

Usage

```
mr_raps(b_exp, b_out, se_exp, se_out, parameters = default_parameters())
```

Arguments

b_exp	Vector of genetic effects on exposure.
b_out	Vector of genetic effects on outcome.
se_exp	Standard errors of genetic effects on exposure.
se_out	Standard errors of genetic effects on outcome.
parameters	A list of parameters. Specifically, over.dispersion and loss.function. over.dispersion is a logical concerning should the model consider overdispersion (systematic pleiotropy). And loss.function allows using either the squared error loss ("l2") or robust loss functions/scores ("huber" or "tukey"). The default is parameters=list(overdispersion = TRUE, loss.function = "tukey").

Details

This function calls the mr.raps package. Please refer to the documentation of that package for more detail.

Value

List with the following elements:

b MR estimate

se Standard error

pval p-value

nsnp Number of SNPs

References

Qingyuan Zhao, Jingshu Wang, Jack Bowden, Dylan S. Small. Statistical inference in two-sample summary-data Mendelian randomization using robust adjusted profile score. Forthcoming.

---

mr_report *Generate MR report*

Description

Using the output from the mr() function this report will generate a report containing tables and graphs summarising the results. A separate report is produced for each exposure - outcome pair that was analysed.

Usage

```
mr_report(
  dat,
  output_path = ".",
  output_type = "html",
  author = "Analyst",
  study = "Two Sample MR",
  path = system.file("reports", package = "TwoSampleMR"),
  ...
)
```

Arguments

dat	Output from harmonise_data()
output_path	Directory in which reports should be saved.
output_type	Choose "html" or "md". Default is "html". All output files including cache and figures will appear in the folder specified in output_path.
author	Author name.

study	Study title.
path	The filepath to the report template.
...	Extra options to be passed to knitr::knit().

`mr_rucker` *MR Rucker framework*

Description

MR Rucker framework.

Usage

```
mr_rucker(dat, parameters = default_parameters())
```

Arguments

dat	Output from harmonise_data().
parameters	List of Qthresh for determining transition between models, and alpha values for calculating confidence intervals. Defaults to 0.05 for both in default_parameters().

Value

list

`mr_rucker_bootstrap` *Run rucker with bootstrap estimates*

Description

Run Rucker with bootstrap estimates.

Usage

```
mr_rucker_bootstrap(dat, parameters = default_parameters())
```

Arguments

dat	Output from harmonise_data().
parameters	List of parameters. The default is default_parameters().

Value

List

mr_rucker_cooksdistance *MR Rucker with outliers automatically detected and removed*

Description

Uses Cook's distance D > 4/nsnp to iteratively remove outliers.

Usage

```
mr_rucker_cooksdistance(dat, parameters = default_parameters())
```

Arguments

dat	Output from harmonise_data().
parameters	List of parameters. The default is default_parameters().

Value

List

mr_rucker_jackknife *Run rucker with jackknife estimates*

Description

Run rucker with jackknife estimates.

Usage

```
mr_rucker_jackknife(dat, parameters = default_parameters())
```

Arguments

dat	Output from harmonise_data.
parameters	List of parameters. The default is default_parameters().

Value

List

mr_scatter_plot — *Create scatter plot with lines showing the causal estimate for different MR tests*

Description

Requires dev version of ggplot2

Usage

```
mr_scatter_plot(mr_results, dat)
```

Arguments

mr_results	Output from mr().
dat	Output from harmonise_data().

Value

List of plots

mr_sign — *MR sign test*

Description

Tests how often the SNP-exposure and SNP-outcome signs are concordant. This is to avoid the problem of averaging over all SNPs, which can suffer bias due to outliers with strong effects; and to avoid excluding SNPs which is implicit in median and mode based estimators. The effect estimate here is not to be interpreted as the effect size - it is the proportion of SNP-exposure and SNP-outcome effects that have concordant signs. e.g. +1 means all have the same sign, -1 means all have opposite signs, and 0 means that there is an equal number of concordant and discordant signs. Restricted to only work if there are 6 or more valid SNPs.

Usage

```
mr_sign(b_exp, b_out, se_exp = NULL, se_out = NULL, parameters = NULL)
```

Arguments

b_exp	Vector of genetic effects on exposure
b_out	Vector of genetic effects on outcome
se_exp	Not required
se_out	Not required
parameters	Not required

Value

List with the following elements:

b Concordance (see description)

se NA

pval p-value

nsnp Number of SNPs (excludes NAs and effect estimates that are 0)

`mr_simple_median` *Simple median method*

Description

Perform MR using summary statistics. Bootstraps used to calculate standard error.

Usage

```
mr_simple_median(
  b_exp,
  b_out,
  se_exp,
  se_out,
  parameters = default_parameters()
)
```

Arguments

`b_exp`	Vector of genetic effects on exposure.
`b_out`	Vector of genetic effects on outcome.
`se_exp`	Standard errors of genetic effects on exposure.
`se_out`	Standard errors of genetic effects on outcome.
`parameters`	The number of bootstrap replications used to calculate the SE can be set through `parameters=list(nboot = 1000)`. The default is `list(nboot=1000)`.

Value

List with the following elements:

b MR estimate

se Standard error

pval p-value

nsnp The number of SNPs

mr_simple_mode *MR simple mode estimator*

Description

MR simple mode estimator

Usage

```
mr_simple_mode(b_exp, b_out, se_exp, se_out, parameters = default_parameters())
```

Arguments

b_exp	Vector of genetic effects on exposure
b_out	Vector of genetic effects on outcome
se_exp	Standard errors of genetic effects on exposure
se_out	Standard errors of genetic effects on outcome
parameters	List containing phi - Bandwidth parameter, and nboot - number of bootstraps to calculate SE. default_parameters() sets list(phi=1, nboot=1000).

Value

List with the following elements:

b MR estimate

se Standard error

pval p-value

mr_simple_mode_nome *MR simple mode estimator (NOME)*

Description

MR simple mode estimator (NOME).

Usage

```
mr_simple_mode_nome(
  b_exp,
  b_out,
  se_exp,
  se_out,
  parameters = default_parameters()
)
```

Arguments

b_exp	Vector of genetic effects on exposure
b_out	Vector of genetic effects on outcome
se_exp	Standard errors of genetic effects on exposure
se_out	Standard errors of genetic effects on outcome
parameters	List containing phi - Bandwidth parameter, and nboot - number of bootstraps to calculate SE. default_parameters() sets list(phi=1, nboot=1000).

Value

List with the following elements:

b MR estimate

se Standard error

pval p-value

---

mr_singlesnp *Perform 2 sample MR on each SNP individually*

Description

Perform 2 sample MR on each SNP individually

Usage

```
mr_singlesnp(
  dat,
  parameters = default_parameters(),
  single_method = "mr_wald_ratio",
  all_method = c("mr_ivw", "mr_egger_regression")
)
```

Arguments

dat	Output from harmonise_data().
parameters	List of parameters. The default is default_parameters().
single_method	Function to use for MR analysis. The default is "mr_wald_ratio".
all_method	Functions to use for MR analysis. The default is c("mr_ivw", "mr_egger_regression").

Value

List of data frames

`mr_steiger` *MR Steiger test of directionality*

Description

A statistical test for whether the assumption that exposure causes outcome is valid

Usage

```
mr_steiger(p_exp, p_out, n_exp, n_out, r_exp, r_out, r_xxo = 1, r_yyo = 1, ...)
```

Arguments

`p_exp`	Vector of p-values of SNP-exposure
`p_out`	Vector of p-values of SNP-outcome
`n_exp`	Sample sizes for p_exp
`n_out`	Sample sizes for p_out
`r_exp`	Vector of absolute correlations for SNP-exposure
`r_out`	Vector of absolute correlations for SNP-outcome
`r_xxo`	Measurememt precision of exposure
`r_yyo`	Measurement precision of outcome
`...`	Further arguments to be passed to `lattice::wireframe()`

Value

List with the following elements:

r2_exp Estimated variance explained in x

r2_out Estimated variance explained in y

r2_exp_adj Predicted variance explained in x accounting for estimated measurement error

r2_out_adj Predicted variance explained in y accounting for estimated measurement error

correct_causal_direction `TRUE/FALSE`

steiger_test p-value for inference of direction

correct_causal_direction_adj `TRUE/FALSE`, direction of causality for given measurement error parameters

steiger_test_adj p-value for inference of direction of causality for given measurement error parameters

vz Total volume of the error parameter space

vz0 Volume of the parameter space that gives the incorrect answer

vz1 Volume of the paramtere space that gives the correct answer

sensitivity_ratio Ratio of vz1/vz0. Higher means inferred direction is less susceptible to measurement error

sensitivity_plot Plot of parameter space of causal directions and measurement error

`mr_steiger2` *MR Steiger test of directionality*

Description

A statistical test for whether the assumption that exposure causes outcome is valid

Usage

```
mr_steiger2(r_exp, r_out, n_exp, n_out, r_xxo = 1, r_yyo = 1, ...)
```

Arguments

`r_exp`	Vector of correlations of SNP-exposure
`r_out`	Vector of correlations of SNP-outcome
`n_exp`	Sample sizes for p_exp
`n_out`	Sample sizes for p_out
`r_xxo`	Measurememt precision of exposure
`r_yyo`	Measurement precision of outcome
`...`	Further arguments to be passed to `lattice::wireframe()`

Value

List with the following elements:

r2_exp Estimated variance explained in x

r2_out Estimated variance explained in y

r2_exp_adj Predicted variance explained in x accounting for estimated measurement error

r2_out_adj Predicted variance explained in y accounting for estimated measurement error

correct_causal_direction `TRUE/FALSE`

steiger_test p-value for inference of direction

correct_causal_direction_adj `TRUE/FALSE`, direction of causality for given measurement error parameters

steiger_test_adj p-value for inference of direction of causality for given measurement error parameters

vz Total volume of the error parameter space

vz0 Volume of the parameter space that gives the incorrect answer

vz1 Volume of the paramtere space that gives the correct answer

sensitivity_ratio Ratio of vz1/vz0. Higher means inferred direction is less susceptible to measurement error

sensitivity_plot Plot of parameter space of causal directions and measurement error

mr_two_sample_ml *Maximum likelihood MR method*

Description

Maximum likelihood MR method

Usage

```
mr_two_sample_ml(b_exp, b_out, se_exp, se_out, parameters)
```

Arguments

b_exp	Vector of genetic effects on exposure.
b_out	Vector of genetic effects on outcome.
se_exp	Standard errors of genetic effects on exposure.
se_out	Standard errors of genetic effects on outcome.
parameters	List of parameters.

Value

List with the following elements:

b causal effect estimate

se standard error

pval p-value

Q, Q_df, Q_pval Heterogeneity stats

mr_uwr *Unweighted regression*

Description

The default multiplicative random effects IVW estimate. The standard error is corrected for under dispersion Use the `mr_ivw_mre()` function for estimates that don't correct for under dispersion.

Usage

```
mr_uwr(b_exp, b_out, se_exp, se_out, parameters = default_parameters())
```

Arguments

b_exp	Vector of genetic effects on exposure.
b_out	Vector of genetic effects on outcome.
se_exp	Standard errors of genetic effects on exposure.
se_out	Standard errors of genetic effects on outcome.
parameters	List of parameters. The default is default_parameters().

Value

List with the following elements:

b MR estimate

se Standard error

pval p-value

Q, Q_df, Q_pval Heterogeneity stats

---

mr_wald_ratio *Perform 2 sample IV using Wald ratio.*

Description

Perform 2 sample IV using Wald ratio.

Usage

```
mr_wald_ratio(b_exp, b_out, se_exp, se_out, parameters)
```

Arguments

b_exp	Vector of genetic effects on exposure.
b_out	Vector of genetic effects on outcome.
se_exp	Standard errors of genetic effects on exposure.
se_out	Standard errors of genetic effects on outcome.
parameters	List of parameters.

Value

List with the following elements:

b causal effect estimate

se standard error

pval p-value

nsnp 1

mr_weighted_median *Weighted median method*

Description

Perform MR using summary statistics. Bootstraps used to calculate standard error.

Usage

```
mr_weighted_median(
  b_exp,
  b_out,
  se_exp,
  se_out,
  parameters = default_parameters()
)
```

Arguments

b_exp	Vector of genetic effects on exposure.
b_out	Vector of genetic effects on outcome.
se_exp	Standard errors of genetic effects on exposure.
se_out	Standard errors of genetic effects on outcome.
parameters	The default is `default_parameters()`. Specify the number of bootstrap replications to calculate the SE with `nboot`. The default is `list(nboot=1000)`.

Value

List with the following elements:

b MR estimate

se Standard error

pval p-value

mr_weighted_mode *MR weighted mode estimator*

Description

MR weighted mode estimator

Usage

```
mr_weighted_mode(
  b_exp,
  b_out,
  se_exp,
  se_out,
  parameters = default_parameters()
)
```

Arguments

b_exp	Vector of genetic effects on exposure
b_out	Vector of genetic effects on outcome
se_exp	Standard errors of genetic effects on exposure
se_out	Standard errors of genetic effects on outcome
parameters	List containing phi - Bandwidth parameter, and nboot - number of bootstraps to calculate SE. default_parameters() sets list(phi=1, nboot=1000).

Value

List with the following elements:

b MR estimate

se Standard error

pval p-value

---

mr_weighted_mode_nome *MR weighted mode estimator (NOME)*

Description

Weighted mode estimator

Usage

```
mr_weighted_mode_nome(
  b_exp,
  b_out,
  se_exp,
  se_out,
  parameters = default_parameters()
)
```

Arguments

b_exp	Vector of genetic effects on exposure
b_out	Vector of genetic effects on outcome
se_exp	Standard errors of genetic effects on exposure
se_out	Standard errors of genetic effects on outcome
parameters	List containing phi - Bandwidth parameter, and nboot - number of bootstraps to calculate SE. default_parameters() sets list(phi=1, nboot=1000).

Value

List with the following elements:

b MR estimate

se Standard error

pval p-value

---

mr_wrapper *Perform full set of MR analyses*

Description

Perform full set of MR analyses

Usage

```
mr_wrapper(dat, parameters = default_parameters())
```

Arguments

dat	Output from harmonise_data().
parameters	Parameters to pass to MR functions. Output from default_parameters() used as default.

Value

list

mv_basic *Perform basic multivariable MR*

Description

Performs initial multivariable MR analysis from Burgess et al 2015. For each exposure the outcome is residualised for all the other exposures, then unweighted regression is applied.

Usage

```
mv_basic(mvdat, pval_threshold = 5e-08)
```

Arguments

mvdat	Output from mv_harmonise_data().
pval_threshold	P-value threshold to include instruments. The default is 5e-8.

Value

List of results

mv_extract_exposures *Extract exposure variables for multivariable MR*

Description

Requires a list of IDs from available_outcomes. For each ID, it extracts instruments. Then, it gets the full list of all instruments and extracts those SNPs for every exposure. Finally, it keeps only the SNPs that are a) independent and b) present in all exposures, and harmonises them to be all on the same strand.

Usage

```
mv_extract_exposures(
  id_exposure,
  clump_r2 = 0.001,
  clump_kb = 10000,
  harmonise_strictness = 2,
  opengwas_jwt = ieugwasr::get_opengwas_jwt(),
  find_proxies = TRUE,
  force_server = FALSE,
  pval_threshold = 5e-08,
  pop = "EUR",
  plink_bin = NULL,
  bfile = NULL
)
```

Arguments

id_exposure	Array of IDs (e.g. c(299, 300, 302) for HDL, LDL, trigs)
clump_r2	The default is 0.01.
clump_kb	The default is 10000.
harmonise_strictness	See the action option of harmonise_data(). The default is 2.
opengwas_jwt	Used to authenticate protected endpoints. Login to https://api.opengwas.io to obtain a jwt. Provide the jwt string here, or store in .Renviron under the keyname OPENGWAS_JWT.
find_proxies	Look for proxies? This slows everything down but is more accurate. The default is TRUE.
force_server	Whether to search through pre-clumped dataset or to re-extract and clump directly from the server. The default is FALSE.
pval_threshold	Instrument detection p-value threshold. Default = 5e-8
pop	Which 1000 genomes super population to use for clumping when using the server
plink_bin	If NULL and bfile is not NULL then will detect packaged plink binary for specific OS. Otherwise specify path to plink binary. Default = NULL
bfile	If this is provided then will use the API. Default = NULL

Value

data frame in exposure_dat format

---

mv_extract_exposures_local

Attempt to perform MVMR using local data

Description

Allows you to read in summary data from text files to format the multivariable exposure dataset.

Usage

```
mv_extract_exposures_local(
  filenames_exposure,
  sep = " ",
  phenotype_col = "Phenotype",
  snp_col = "SNP",
  beta_col = "beta",
  se_col = "se",
  eaf_col = "eaf",
  effect_allele_col = "effect_allele",
  other_allele_col = "other_allele",
```

```
  pval_col = "pval",
  units_col = "units",
  ncase_col = "ncase",
  ncontrol_col = "ncontrol",
  samplesize_col = "samplesize",
  gene_col = "gene",
  id_col = "id",
  min_pval = 1e-200,
  log_pval = FALSE,
  pval_threshold = 5e-08,
  plink_bin = NULL,
  bfile = NULL,
  clump_r2 = 0.001,
  clump_kb = 10000,
  pop = "EUR",
  harmonise_strictness = 2
)
```

Arguments

filenames_exposure	Filenames for each exposure dataset. Must have header with at least SNP column present. Following arguments are used for determining how to read the filename and clumping etc.
sep	Specify delimeter in file. The default is space, i.e. `sep=" "`. If length is 1 it will use the same `sep` value for each exposure dataset. You can provide a vector of values, one for each exposure dataset, if the values are different across datasets. The same applies to all dataset-formatting options listed below.
phenotype_col	Optional column name for the column with phenotype name corresponding the the SNP. If not present then will be created with the value `"Outcome"`. Default is `"Phenotype"`.
snp_col	Required name of column with SNP rs IDs. The default is `"SNP"`.
beta_col	Required for MR. Name of column with effect sizes. THe default is `"beta"`.
se_col	Required for MR. Name of column with standard errors. The default is `"se"`.
eaf_col	Required for MR. Name of column with effect allele frequency. The default is `"eaf"`.
effect_allele_col	Required for MR. Name of column with effect allele. Must be "A", "C", "T" or "G". The default is `"effect_allele"`.
other_allele_col	Required for MR. Name of column with non effect allele. Must be "A", "C", "T" or "G". The default is `"other_allele"`.
pval_col	Required for enrichment tests. Name of column with p-value. The default is `"pval"`.
units_col	Optional column name for units. The default is `"units"`.
ncase_col	Optional column name for number of cases. The default is `"ncase"`.

ncontrol_col	Optional column name for number of controls. The default is "ncontrol".
samplesize_col	Optional column name for sample size. The default is "samplesize".
gene_col	Optional column name for gene name. The default is "gene".
id_col	Optional column name to give the dataset an ID. Will be generated automatically if not provided for every trait / unit combination. The default is "id".
min_pval	Minimum allowed p-value. The default is 1e-200.
log_pval	The pval is -log10(P). The default is FALSE.
pval_threshold	Default=5e-8 for clumping
plink_bin	If NULL and bfile is not NULL then will detect packaged plink binary for specific OS. Otherwise specify path to plink binary. Default = NULL
bfile	If this is provided then will use the API. Default = NULL
clump_r2	Default=0.001 for clumping
clump_kb	Default=10000 for clumping
pop	Which 1000 genomes super population to use for clumping when using the server
harmonise_strictness	See action argument in harmonise_data(). Default=2

Details

Note that you can provide an array of column names for each column, which is of length filenames_exposure

Value

List

---

mv_harmonise_data *Harmonise exposure and outcome for multivariable MR*

Description

Harmonise exposure and outcome for multivariable MR

Usage

```
mv_harmonise_data(exposure_dat, outcome_dat, harmonise_strictness = 2)
```

Arguments

exposure_dat	Output from mv_extract_exposures().
outcome_dat	Output from extract_outcome_data(exposure_dat$SNP, id_output).
harmonise_strictness	See the action option of harmonise_data(). The default is 2.

Value

List of vectors and matrices required for mv analysis.

exposure_beta a matrix of beta coefficients, in which rows correspond to SNPs and columns correspond to exposures.

exposure_se is the same as `exposure_beta`, but for standard errors.

exposure_pval the same as `exposure_beta`, but for p-values.

expname A data frame with two variables, `id.exposure` and `exposure` which are character strings.

outcome_beta an array of effects for the outcome, corresponding to the SNPs in `exposure_beta`.

outcome_se an array of standard errors for the outcome.

outcome_pval an array of p-values for the outcome.

outname A data frame with two variables, `id.outcome` and `outcome` which are character strings.

`mv_ivw` *Perform IVW multivariable MR*

Description

Performs modified multivariable MR analysis. For each exposure the instruments are selected then all exposures for those SNPs are regressed against the outcome together, weighting for the inverse variance of the outcome.

Usage

```
mv_ivw(mvdat, pval_threshold = 5e-08)
```

Arguments

`mvdat`	Output from `mv_harmonise_data()`.
`pval_threshold`	P-value threshold to include instruments. The default is `5e-8`.

Value

List of results

mv_lasso_feature_selection *Apply LASSO feature selection to mvdat object*

Description

Apply LASSO feature selection to mvdat object

Usage

```
mv_lasso_feature_selection(mvdat)
```

Arguments

mvdat	Output from mv_harmonise_data().

Value

data frame of retained features

mv_multiple *Perform IVW multivariable MR*

Description

Performs modified multivariable MR analysis. For each exposure the instruments are selected then all exposures for those SNPs are regressed against the outcome together, weighting for the inverse variance of the outcome.

Usage

```
mv_multiple(
  mvdat,
  intercept = FALSE,
  instrument_specific = FALSE,
  pval_threshold = 5e-08,
  plots = FALSE
)
```

Arguments

mvdat	Output from mv_harmonise_data().
intercept	Should the intercept by estimated (TRUE) or force line through the origin (FALSE, default).
instrument_specific	Should the estimate for each exposure be obtained by using all instruments from all exposures (FALSE, default) or by using only the instruments specific to each exposure (TRUE).
pval_threshold	P-value threshold to include instruments. The default is 5e-8.
plots	Create plots? The default is FALSE.

Value

List of results

---

mv_residual *Perform basic multivariable MR*

Description

Performs initial multivariable MR analysis from Burgess et al 2015. For each exposure the outcome is residualised for all the other exposures, then unweighted regression is applied.

Usage

```
mv_residual(
  mvdat,
  intercept = FALSE,
  instrument_specific = FALSE,
  pval_threshold = 5e-08,
  plots = FALSE
)
```

Arguments

mvdat	Output from mv_harmonise_data().
intercept	Should the intercept by estimated (TRUE) or force line through the origin (FALSE, default).
instrument_specific	Should the estimate for each exposure be obtained by using all instruments from all exposures (FALSE, default) or by using only the instruments specific to each exposure (TRUE).
pval_threshold	P-value threshold to include instruments. The default is 5e-8.
plots	Create plots? The default is FALSE.

Value

List of results

`mv_subset` *Perform multivariable MR on subset of features*

Description

The function proceeds as follows:

1. Select features (by default this is done using LASSO feature selection).
2. Subset the mvdat to only retain relevant features and instruments.
3. Perform MVMR on remaining data.

Usage

```
mv_subset(
  mvdat,
  features = mv_lasso_feature_selection(mvdat),
  intercept = FALSE,
  instrument_specific = FALSE,
  pval_threshold = 5e-08,
  plots = FALSE
)
```

Arguments

`mvdat`	Output from `mv_harmonise_data()`.
`features`	Dataframe of features to retain, must have column with name 'exposure' that has list of exposures to retain from mvdat. The default is `mvdat_lasso_feature_selection(mvdat)`.
`intercept`	Should the intercept by estimated (`TRUE`) or force line through the origin (`FALSE`, the default).
`instrument_specific`	Should the estimate for each exposure be obtained by using all instruments from all exposures (`FALSE`, default) or by using only the instruments specific to each exposure (`TRUE`).
`pval_threshold`	P-value threshold to include instruments. The default is `5e-8`.
`plots`	Create plots? The default is `FALSE`.

Value

List of results

power_prune *Power prune*

Description

When there are duplicate summary sets for a particular exposure-outcome combination, this function keeps the exposure-outcome summary set with the highest expected statistical power. This can be done by dropping the duplicate summary sets with the smaller sample sizes. Alternatively, the pruning procedure can take into account instrument strength and outcome sample size. The latter is useful, for example, when there is considerable variation in SNP coverage between duplicate summary sets (e.g. because some studies have used targeted or fine mapping arrays). If there are a large number of SNPs available to instrument an exposure, the outcome GWAS with the better SNP coverage may provide better power than the outcome GWAS with the larger sample size.

Usage

```
power_prune(dat, method = 1, dist.outcome = "binary")
```

Arguments

dat	Results from harmonise_data().
method	Should the duplicate summary sets be pruned on the basis of sample size alone (method = 1) or a combination of instrument strength and sample size (method = 2)? Default set to 1. When set to 1, the duplicate summary sets are first dropped on the basis of the outcome sample size (smaller duplicates dropped). If duplicates are still present, remaining duplicates are dropped on the basis of the exposure sample size (smaller duplicates dropped). When method is set to 2, duplicates are dropped on the basis of instrument strength (amount of variation explained in the exposure by the instrumental SNPs) and sample size, and assumes that the SNP-exposure effects correspond to a continuous trait with a normal distribution (i.e. exposure cannot be binary). The SNP-outcome effects can correspond to either a binary or continuous trait. If the exposure is binary then method=1 should be used.
dist.outcome	The distribution of the outcome. Can either be "binary" or "continuous". Default set to "binary".

Value

data.frame with duplicate summary sets removed

`read_exposure_data` *Read exposure data*

Description

Reads in exposure data. Checks and organises columns for use with MR or enrichment tests. Infers p-values when possible from beta and se.

Usage

```
read_exposure_data(
  filename,
  clump = FALSE,
  sep = " ",
  phenotype_col = "Phenotype",
  snp_col = "SNP",
  beta_col = "beta",
  se_col = "se",
  eaf_col = "eaf",
  effect_allele_col = "effect_allele",
  other_allele_col = "other_allele",
  pval_col = "pval",
  units_col = "units",
  ncase_col = "ncase",
  ncontrol_col = "ncontrol",
  samplesize_col = "samplesize",
  gene_col = "gene",
  id_col = "id",
  min_pval = 1e-200,
  log_pval = FALSE,
  chr_col = "chr",
  pos_col = "pos"
)
```

Arguments

`filename`	Filename. Must have header with at least SNP column present.
`clump`	Whether to perform LD clumping with `clump_data()` on the exposure data. The default is `FALSE`.
`sep`	Specify delimeter in file. The default is a space, i.e. `" "`.
`phenotype_col`	Optional column name for the column with phenotype name corresponding the the SNP. If not present then will be created with the value "Outcome". The default is `"Phenotype"`.
`snp_col`	Required name of column with SNP rs IDs. The default is `"SNP"`.
`beta_col`	Required for MR. Name of column with effect sizes. The default is `"beta"`.

se_col	Required for MR. Name of column with standard errors. The default is "se".
eaf_col	Required for MR. Name of column with effect allele frequency. The default is "eaf".
effect_allele_col	Required for MR. Name of column with effect allele. Must be "A", "C", "T" or "G". The default is "effect_allele".
other_allele_col	Required for MR. Name of column with non effect allele. Must be "A", "C", "T" or "G". The default is "other_allele".
pval_col	Required for enrichment tests. Name of column with p-value. The default is "pval".
units_col	Optional column name for units. The default is "units".
ncase_col	Optional column name for number of cases. The default is "ncase".
ncontrol_col	Optional column name for number of controls. The default is "ncontrol".
samplesize_col	Optional column name for sample size. The default is "samplesize".
gene_col	Optional column name for gene name. The default is "gene".
id_col	Optional column name to give the dataset an ID. Will be generated automatically if not provided for every trait / unit combination. The default is "id".
min_pval	Minimum allowed p-value. The default is 1e-200.
log_pval	The p-value is -log10(P). The default is FALSE.
chr_col	Optional column name for chromosome. Default is "chr".
pos_col	Optional column name for genetic position Default is "pos".

Value

data frame

`read_outcome_data` *Read outcome data*

Description

Reads in outcome data. Checks and organises columns for use with MR or enrichment tests. Infers p-values when possible from beta and se.

Usage

```
read_outcome_data(
  filename,
  snps = NULL,
  sep = " ",
  phenotype_col = "Phenotype",
  snp_col = "SNP",
```

```
  beta_col = "beta",
  se_col = "se",
  eaf_col = "eaf",
  effect_allele_col = "effect_allele",
  other_allele_col = "other_allele",
  pval_col = "pval",
  units_col = "units",
  ncase_col = "ncase",
  ncontrol_col = "ncontrol",
  samplesize_col = "samplesize",
  gene_col = "gene",
  id_col = "id",
  min_pval = 1e-200,
  log_pval = FALSE,
  chr_col = "chr",
  pos_col = "pos"
)
```

Arguments

filename	Filename. Must have header with at least SNP column present.
snps	SNPs to extract. If NULL, which the default, then doesn't extract any and keeps all.
sep	Specify delimeter in file. The default is space, i.e. sep=" ".
phenotype_col	Optional column name for the column with phenotype name corresponding the the SNP. If not present then will be created with the value "Outcome". Default is "Phenotype".
snp_col	Required name of column with SNP rs IDs. The default is "SNP".
beta_col	Required for MR. Name of column with effect sizes. THe default is "beta".
se_col	Required for MR. Name of column with standard errors. The default is "se".
eaf_col	Required for MR. Name of column with effect allele frequency. The default is "eaf".
effect_allele_col	Required for MR. Name of column with effect allele. Must be "A", "C", "T" or "G". The default is "effect_allele".
other_allele_col	Required for MR. Name of column with non effect allele. Must be "A", "C", "T" or "G". The default is "other_allele".
pval_col	Required for enrichment tests. Name of column with p-value. The default is "pval".
units_col	Optional column name for units. The default is "units".
ncase_col	Optional column name for number of cases. The default is "ncase".
ncontrol_col	Optional column name for number of controls. The default is "ncontrol".
samplesize_col	Optional column name for sample size. The default is "samplesize".
gene_col	Optional column name for gene name. The default is "gene".

id_col	Optional column name to give the dataset an ID. Will be generated automatically if not provided for every trait / unit combination. The default is "id".
min_pval	Minimum allowed p-value. The default is 1e-200.
log_pval	The pval is -log10(P). The default is FALSE.
chr_col	Optional column name for chromosome. Default is "chr".
pos_col	Optional column name for genetic position Default is "pos".

Value

data frame

`run_mrmix` *Perform MRMix analysis on harmonised dat object*

Description

See https://github.com/gqi/MRMix for more details.

Usage

```
run_mrmix(dat)
```

Arguments

dat	Output from harmonise_data(). Ensures that no eaf.exposure values are missing.

Value

List of results, with one list item for every exposure/outcome pair in dat object

---

`run_mr_presso` *Wrapper for MR-PRESSO*

Description

See https://github.com/rondolab/MR-PRESSO for more details.

Usage

```
run_mr_presso(dat, NbDistribution = 1000, SignifThreshold = 0.05)
```

Arguments

dat	Output from harmonise_data().
NbDistribution	Number of bootstrap replications. The default is 1000.
SignifThreshold	Outlier significance threshold. The default is 0.05.

Value

List of results for every exposure/outcome combination

---

size.prune *Size prune*

Description

Whens there are duplicate summary sets for a particular exposure-outcome combination, this function drops the duplicates with the smaller total sample size (for binary outcomes, the number of cases is used instead of total sample size).

Usage

```
size.prune(dat)
```

Arguments

dat	Results from harmonise_data().

Value

data frame

---

sort_1_to_many *Sort results for 1-to-many forest plot*

Description

This function sorts user-supplied results for the forest_plot_1_to_many() function. The user supplies their results in the form of a data frame.

Usage

```
sort_1_to_many(
  mr_res,
  b = "b",
  trait_m = "outcome",
  sort_action = 4,
  group = NULL,
  priority = NULL
)
```

Arguments

mr_res	Data frame of results supplied by the user.
b	Name of the column specifying the effect of the exposure on the outcome. The default is "b".
trait_m	The column specifying the names of the traits. Corresponds to 'many' in the 1-to-many forest plot. The default is "outcome".
sort_action	Choose how to sort results. • sort_action = 1: sort results by effect size within groups. Use the group order supplied by the user. • sort_action = 2: sort results by effect size and group. Overides the group ordering supplied by the user. • sort_action = 3: group results for the same trait together (e.g. multiple results for the same trait from different MR methods). • sort_action = 4: sort by decreasing effect size (largest effect size at top and smallest at bottom). • sort_action = 5: sort by increasing effect size (smallest effect size at top and largest at bottom).
group	Name of grouping variable in mr_res.
priority	If sort_action = 3, choose which value of the trait_m variable should be given priority and go above the other trait_m values. The trait with the largest effect size for the prioritised group will go to the top of the plot.

Value

data frame.

---

split_exposure *Split exposure column*

Description

This function takes the exposure column from the results generated by mr() and splits it into separate columns for 'exposure name' and 'id'.

Usage

```
split_exposure(mr_res)
```

Arguments

mr_res	Results from mr().

Value

data frame

---

split_outcome *Split outcome column*

Description

This function takes the outcome column from the results generated by mr() and splits it into separate columns for 'outcome name' and 'id'.

Usage

```
split_outcome(mr_res)
```

Arguments

mr_res	Results from mr().

Value

data frame

---

standardise_units *Try to standardise continuous traits to be in standard deviation units*

Description

Uses estimate_trait_sd().

Usage

```
standardise_units(dat)
```

Arguments

dat	Output from harmonise_data().

Value

Data frame

steiger_filtering *Steiger filtering function*

Description

This function takes an object from harmonise_data() and does the following: If there is no rsq.exposure or rsq.outcome column it will try to estimate it. This is done differently for traits that have "log odds" units. To estimate rsq for quantitative traits there must be either p-values and sample sizes for each SNP, or effect sizes and standard errors AND the units are in SD units (the column must contain "SD"). To estimate rsq for binary traits the units must be called "log odds" and there must be beta.exposure, eaf.exposure, ncase.exposure, ncontrol.exposure, prevalence.exposure. The same principles apply for calculating the rsq for the outcome trait, except column names are beta.outcome etc. If prevalence isn't supplied then it uses 0.1 by default.

Usage

```
steiger_filtering(dat)
```

Arguments

dat	Output from harmonise_data().

Details

Once rsq is calculated for the exposure and outcome, it will then perform the Steiger test for each SNP to see if the rsq of the exposure is larger than the rsq of the outcome.

Note that Steiger filtering, while useful, does have its own pitfalls. Try to use replication effect estimates for the exposure (which are not biased by winner's curse), and note that if there is strong antagonistic confounding or differential measurement error between the exposure and outcome then the causal directions could be inferred incorrectly.

Value

harmonise_data() style data frame with additional columns rsq.exposure, rsq.outcome, steiger_dir (which is TRUE if the rsq.exposure is larger than rsq.outcome) and steiger_pval which is a test to see if rsq.exposure is significantly larger than rsq.outcome.

steiger_sensitivity *Evaluate the Steiger test's sensitivity to measurement error*

Description

Evaluate the Steiger test's sensitivity to measurement error

Usage

```
steiger_sensitivity(rgx_o, rgy_o, ...)
```

Arguments

rgx_o	Observed variance of exposure explained by SNPs
rgy_o	Observed variance of outcome explained by SNPs
...	Further arguments to be passed to lattice::wireframe()

Value

List with the following elements:

vz Total volume of the error parameter space

vz0 Volume of the parameter space that gives the incorrect answer

vz1 Volume of the paramtere space that gives the correct answer

sensitivity_ratio Ratio of vz1/vz0. Higher means inferred direction is less susceptible to measurement error

pl plot of parameter space

---

subset_on_method *Subset MR-results on method*

Description

This function takes MR results from mr() and restricts to a single method per exposure x disease combination.

Usage

```
subset_on_method(
  mr_res,
  single_snp_method = "Wald ratio",
  multi_snp_method = "Inverse variance weighted"
)
```

Arguments

mr_res	Results from mr().
single_snp_method	Which of the single SNP methods to use when only 1 SNP was used to estimate the causal effect? The default is "Wald ratio".
multi_snp_method	Which of the multi-SNP methods to use when there was more than 1 SNPs used to estimate the causal effect? The default is "Inverse variance weighted".

Value

data frame.

`trim` *Trim function to remove leading and trailing blank spaces*

Description

Trim function to remove leading and trailing blank spaces

Usage

```
trim(x)
```

Arguments

`x`	Character or array of character

Value

Character or array of character

`weighted_median` *Weighted median method*

Description

New method from Jack

Usage

```
weighted_median(b_iv, weights)
```

Arguments

`b_iv`	Wald ratios
`weights`	Weights of each SNP

Value

MR estimate

`weighted_median_bootstrap` *Calculate standard errors for weighted median method using bootstrap*

Description

Based on new script for weighted median confidence interval, update 31 July 2015.

Usage

```
weighted_median_bootstrap(b_exp, b_out, se_exp, se_out, weights, nboot)
```

Arguments

`b_exp`	Vector of genetic effects on exposure.
`b_out`	Vector of genetic effects on outcome.
`se_exp`	Standard errors of genetic effects on exposure.
`se_out`	Standard errors of genetic effects on outcome.
`weights`	Weights to apply to each SNP.
`nboot`	Number of bootstrap replications. The default is `1000`.

Value

Empirical standard error

Index