

# Package: CAMEra (via r-universe)

October 8, 2026

**Title** CAMEra (Cross Ancestral Mendelian Randomisation)

**Version** 0.1.4

**Description** CAMERA estimates joint causal effect in multiple  
ancestries and detects pleiotropy via the zero relevance model.

**License** MIT + file LICENSE

**URL** <https://github.com/MRCIEU/CAMERA>, <https://mrcieu.github.io/CAMERA/>

**BugReports** <https://github.com/MRCIEU/CAMERA/issues/>

**Depends** R (>= 4.1.0)

**Imports** dplyr, ggplot2 (>= 3.3.0), ieugwasr (>= 1.2.0), magrittr,  
tidyr

**Suggests** data.table, GenomicRanges, genetics.binaRies, IRanges, knitr,  
lavaan, parallel, RadialMR, rmarkdown, susieR, testthat (>= 3.0.0), TwoSampleMR, winnerscurse

**VignetteBuilder** knitr

**Remotes** amandaforde/winnerscurse, MRCIEU/genetics.binaRies,  
MRCIEU/ieugwasr, MRCIEU/TwoSampleMR, WSpiller/RadialMR

**Config/roxygen2/markdown** TRUE

**Config/roxygen2/version** 8.1.0

**Config/testthat/edition** 3

**Encoding** UTF-8

**Config/pak/sysreqs** libicu-dev libssl-dev

**Repository** <https://mrcieu.r-universe.dev>

**Date/Publication** 2026-09-25 12:06:11 UTC

**RemoteUrl** <https://github.com/MRCIEU/CAMERA>

**RemoteRef** HEAD

**RemoteSha** 9f314f5c32afb31ace9fed2648074f25780a910a

Contents

|                                            |           |
|--------------------------------------------|-----------|
| CAMeRa-package . . . . .                   | 2         |
| CAMERA . . . . .                           | 3         |
| CAMERA_local . . . . .                     | 18        |
| egger_bootstrap . . . . .                  | 22        |
| fixed_effects_meta_analysis . . . . .      | 23        |
| fixed_effects_meta_analysis_fast . . . . . | 24        |
| identify_blownup_estimates . . . . .       | 24        |
| prop_overlap . . . . .                     | 25        |
| z_meta_analysis . . . . .                  | 26        |
| <b>Index</b>                               | <b>27</b> |

---

|                |                                                                 |
|----------------|-----------------------------------------------------------------|
| CAMeRa-package | <i>CAMeRa: CAMeRa (Cross Ancestral Mendelian Randomisation)</i> |
|----------------|-----------------------------------------------------------------|

---

Description

CAMERA estimates joint causal effect in multiple ancestries and detects pleiotropy via the zero relevance model.

Author(s)

**Maintainer:** Gibran Hemani <g.hemani@bristol.ac.uk> ([ORCID](#))

Authors:

- Gibran Hemani <g.hemani@bristol.ac.uk> ([ORCID](#))
- Yoonsu Cho <yoonsu.cho@bristol.ac.uk>
- Tom Palmer <remlapmot@hotmail.com> ([ORCID](#))

Other contributors:

- Amy Mason <am2609@medschl.cam.ac.uk> ([ORCID](#)) [contributor]
- Mercy Kimani [contributor]

See Also

Useful links:

- <https://github.com/MRCIEU/CAMERA>
- <https://mrcieu.github.io/CAMERA/>
- Report bugs at <https://github.com/MRCIEU/CAMERA/issues/>

CAMERA

*R6 class for CAMERA***Description**

A simple wrapper function. Using a summary set, identify set of instruments for the traits, and perform SEM MR to test the association across the population.

**Public fields**

output A list for the output

source The source of the data.

exposure\_ids Exposures IDs obtained from IEU GWAS database (<https://gwas.mrcieu.ac.uk>) for each population

outcome\_ids Outcome IDs obtained from IEU GWAS database (<https://gwas.mrcieu.ac.uk>) for each population

exposure\_metadata Exposure metadata

outcome\_metadata Outcome metadata

radius Genomic window size to extract SNPs

pops Ancestry information for each population (i.e., AFR, AMR, EUR, EAS, SAS)

bfiles Locations of LD reference files for each population (Download from: <http://files.mrcieu.ac.uk/ld/1kg.v3.tgz>)

plink Location of executable plink (version 1.90 is recommended)

clump\_pop Reference population for clumping

instrument\_raw Instruments for the exposures obtained from `extract_instruments()`

instrument\_regions Genomic regions around each instrument obtained from `extract_instrument_regions()`

ld\_matrices LD matrices for each instrument region obtained from `regional_ld_matrices()`

susie\_results Fine-mapping results from `susie_finemap_regions()`

paintor\_results Fine-mapping results from `paintor_finemap_regions()`

mscaviar\_results Fine-mapping results from `MsCAVIAR_finemap_regions()`

expected\_replications Expected replication of instruments across populations

instrument\_region\_zscores Z scores for the SNPs in each instrument region obtained from `scan_regional_instruments()`

instrument\_fema Instruments selected by meta-analysis across populations obtained from `fema_regional_instruments()`

instrument\_fema\_regions Meta-analysis results for each instrument region obtained from `fema_regional_instruments()`

instrument\_maxz Instruments selected by maximum Z score obtained from `scan_regional_instruments()`

instrument\_susie Instruments selected by susieR fine-mapping obtained from `susie_finemap_regions()`

instrument\_paintor Instruments selected by PAINTOR fine-mapping obtained from `paintor_finemap_regions()`

instrument\_mscaviar Instruments selected by MsCAVIAR fine-mapping obtained from `MsCAVIAR_finemap_regions()`

harmonised\_data\_check Checks of the harmonised data  
 standardised\_instrument\_raw Standardised instrument\_raw obtained from standardise\_data()  
 standardised\_instrument\_maxz Standardised instrument\_maxz obtained from standardise\_data()  
 standardised\_instrument\_susie Standardised instrument\_susie obtained from standardise\_data()  
 standardised\_instrument\_paintor Standardised instrument\_paintor obtained from standardise\_data()  
 standardised\_instrument\_mscaviar Standardised instrument\_mscaviar obtained from standardise\_data()  
 standardised\_outcome Standardised outcome data obtained from standardise\_data()  
 instrument\_specificity Instrument specificity results obtained from estimate\_instrument\_specificity()  
 instrument\_specificity\_summary Summary of the instrument specificity results obtained from  
 estimate\_instrument\_specificity()  
 instrument\_outcome Outcome data for the instruments obtained from make\_outcome\_data()  
 instrument\_outcome\_regions Outcome data for each instrument region imported with import\_from\_local()  
 harmonised\_dat\_sem Harmonised data for the SEM analysis  
 harmonised\_dat Harmonised exposure and outcome data obtained from harmonise()  
 sem\_result SEM results obtained from perform\_basic\_sem()  
 pleiotropy\_agreement Agreement of pleiotropy outlier effects across populations obtained from  
 pleiotropy()  
 pleiotropy\_outliers Pleiotropy outliers obtained from pleiotropy()  
 pleiotropy\_Q\_outliers Heterogeneity for each pleiotropy outlier and population obtained from  
 pleiotropy()  
 summary Summary of the exposure, outcome and population metadata obtained from set\_summary()  
 mrres MR estimates obtained from cross\_estimate()  
 instrument\_heterogeneity\_per\_variant Heterogeneity of the instrument-exposure associations  
 for each variant obtained from estimate\_instrument\_heterogeneity\_per\_variant()  
 mrgxe\_res MR GxE results obtained from mrgxe()

## Methods

### Public methods:

- CAMERA\$import()
- CAMERA\$assign()
- CAMERA\$import\_from\_local()
- CAMERA\$new()
- CAMERA\$instrument\_heterogeneity()
- CAMERA\$estimate\_instrument\_specificity()
- CAMERA\$replication\_evaluation()
- CAMERA\$check\_phenotypes()
- CAMERA\$cross\_estimate()
- CAMERA\$plot\_cross\_estimate()
- CAMERA\$extract\_instruments()
- CAMERA\$extract\_instrument\_regions()

- CAMERA\$scan\_regional\_instruments()
- CAMERA\$plot\_regional\_instruments\_maxz()
- CAMERA\$regional\_ld\_matrices()
- CAMERA\$susie\_finemap\_regions()
- CAMERA\$paintor\_finemap\_regions()
- CAMERA\$MsCAVIAR\_finemap\_regions()
- CAMERA\$fema\_regional\_instruments()
- CAMERA\$plot\_regional\_instruments()
- CAMERA\$get\_metadata()
- CAMERA\$estimate\_instrument\_heterogeneity\_per\_variant()
- CAMERA\$mrgxe()
- CAMERA\$mrgxe\_plot()
- CAMERA\$mrgxe\_plot\_variant()
- CAMERA\$make\_outcome\_data()
- CAMERA\$make\_outcome\_local()
- CAMERA\$harmonise()
- CAMERA\$set\_summary()
- CAMERA\$pleiotropy()
- CAMERA\$plot\_pleiotropy()
- CAMERA\$plot\_pleiotropy\_heterogeneity()
- CAMERA\$perform\_basic\_sem()
- CAMERA\$runsem()
- CAMERA\$standardise\_data()
- CAMERA\$clone()

CAMERA\$import(): Migrate the results from a previous CAMERA

*Usage:*

CAMERA\$import(x)

*Arguments:*

x A CAMERA object to import results from

CAMERA\$assign(): Assign values to fields of the CAMERA object

*Usage:*

CAMERA\$assign(...)

*Arguments:*

... Named arguments, where each name is a field and each value is assigned to it

CAMERA\$import\_from\_local(): Import instrument and outcome data from local summary statistics, e.g. as organised by CAMERA\_local

*Usage:*

```

CAMERA$import_from_local(
  instrument_raw,
  instrument_outcome,
  instrument_regions,
  instrument_outcome_regions,
  exposure_ids,
  outcome_ids,
  pops,
  ...
)

```

*Arguments:*

`instrument_raw` Data frame of instruments for the exposures

`instrument_outcome` Data frame of instrument associations with the outcomes

`instrument_regions` List of genomic regions around each instrument for the exposures

`instrument_outcome_regions` List of genomic regions around each instrument for the outcomes

`exposure_ids` Exposures IDs obtained from IEU GWAS database (<https://gwas.mrcieu.ac.uk>) for each population

`exposure_ids` ID for the exposure. Default is `x$exposure_ids`.

`exposure_ids` IDs for the exposures. Default is `x$exposure_ids`.

`outcome_ids` Outcome IDs obtained from IEU GWAS database (<https://gwas.mrcieu.ac.uk>) for each population

`outcome_ids` IDs for the outcomes. Default is `x$outcome_ids`.

`pops` Ancestry information for each population (i.e., AFR, AMR, EUR, EAS, SAS)

`...` Further named arguments passed to `assign()`

`CAMERA$new()`: Create a new dataset and initialise an R interface

*Usage:*

```

CAMERA$new(
  exposure_ids = NULL,
  outcome_ids = NULL,
  pops = NULL,
  bfiles = NULL,
  plink = NULL,
  radius = NULL,
  clump_pop = NULL,
  x = NULL
)

```

*Arguments:*

`exposure_ids` Exposures IDs obtained from IEU GWAS database (<https://gwas.mrcieu.ac.uk>) for each population

`exposure_ids` ID for the exposure. Default is `x$exposure_ids`.

`exposure_ids` IDs for the exposures. Default is `x$exposure_ids`.

`outcome_ids` Outcome IDs obtained from IEU GWAS database (<https://gwas.mrcieu.ac.uk>) for each population

outcome\_ids IDs for the outcomes. Default is x\$outcome\_ids.  
 pops Ancestry information for each population (i.e., AFR, AMR, EUR, EAS, SAS)  
 bfiles Locations of LD reference files for each population (Download from: <http://fileserv.mrcieu.ac.uk/ld/1kg.v3.tgz>)  
 plink Location of executable plink (version 1.90 is recommended)  
 radius Genomic window size to extract SNPs  
 clump\_pop Reference population for clumping  
 x Import data where available

**CAMERA\$instrument\_heterogeneity():** The function evaluates heterogeneity in the association of selected instruments and the exposure/outcome between the populations. Heterogeneity (Q statistics) is calculated based on an IVW or simple MODE MR estimator. The instruments can be identified using "Raw", "MaxZ", or fine-mapping (Susie, PAINTOR) methods.

*Usage:*

```

CAMERA$instrument_heterogeneity(
  instrument = self$instrument_raw,
  alpha = "bonferroni",
  method = "ivw",
  outlier_removal = FALSE
)

```

*Arguments:*

**instrument** Instruments for the exposure that are selected by using the provided methods in CAMERA (x\$instrument\_raw, x\$instrument\_maxz, x\$instrument\_susie, x\$instrument\_paintor). Default is x\$instrument\_raw.  
**alpha** Statistical threshold to determine significance. Default is "bonferroni", which is equal to 0.05/number of the instruments. For each pair of populations, the instruments that are significant ( $p < \alpha$ ) in the reference population are taken and their associations in the replication population are regressed on their associations in the reference population. If the instruments have the same effects in both populations the slope (agreement) is expected to be close to 1.  
**method** Method used to estimate the agreement slope, either "ivw" (inverse variance weighted, using `TwoSampleMR::mr_ivw()`) or "simple\_mode" (using `TwoSampleMR::mr_simple_mode()`). Default is "ivw".  
**outlier\_removal** Remove outliers identified by radial IVW MR before estimating heterogeneity. Only the first two exposures are used to identify outliers. Default is FALSE.

**Returns:** Data frame with one row for each ordered pair of populations, with columns:

- Reference, Replication: IDs of the reference and replication datasets
- nsnp: number of instruments significant in the reference dataset that are also present in the replication dataset
- agreement: slope of the regression of the replication associations on the reference associations
- se, pval: standard error of agreement and p-value for the test of the null hypothesis that the slope is 0 (i.e. no association between the instrument associations in the two populations)
- Q, Q\_pval ("ivw" only): Cochran's Q statistic and its p-value for the null hypothesis that all instruments share the same slope, i.e. no heterogeneity in the agreement between populations

- I2 ("ivw" only): the proportion of the variation in the ratio of the associations that is due to heterogeneity rather than chance, calculated as  $(Q - \text{nsnp}) / Q$

`CAMERA$estimate_instrument_specificity()`: The function estimates what fraction of the instruments is expected to be replicated across the populations under the hypothesis that the effect estimates are the same.

*Usage:*

```
CAMERA$estimate_instrument_specificity(
  instrument,
  alpha = "bonferroni",
  winnerscurse = FALSE
)
```

*Arguments:*

`instrument` Instruments for the exposure that are selected by using the provided methods in CAMERA (`x$instrument_raw`, `x$instrument_maxz`, `x$instrument_susie`, `x$instrument_paintor`). Default is `x$instrument_raw`.

`alpha` Statistical threshold to determine significance. Default is "bonferroni", which is equal to  $0.05/\text{number of the instruments}$ .

`winnerscurse` Use this option to correct winners' curse bias.

*Details:* For each pair of populations, the instruments that are significant ( $p < \alpha$ ) in the discovery population are compared with their associations in the replication population. Following Okbay et al. (2016), under the hypothesis that the instruments have the same effects in both populations, the expected number of instruments with the same sign, and the expected number with  $p < \alpha$ , in the replication population are calculated taking account of the precision of the replication associations. These are compared with the observed numbers.

*Returns:* Data frame, also stored in `x$instrument_specificity_summary`, with columns:

- `discovery, replication`: IDs of the discovery and replication datasets
- `metric`: "Sign" (the instrument has the same sign in both populations) or "P-value" (the instrument has  $p < \alpha$  in the replication population)
- `nsnp`: number of instruments compared
- `datum, value`: the "Expected" and "Observed" number of instruments meeting metric
- `pdiff`: p-value from a binomial test of the null hypothesis that the observed number equals the expected number. A small p-value suggests the instruments replicate less (or more) than expected if their effects were the same in both populations.

Per variant results are stored in `x$instrument_specificity`, including the expected probabilities of sign agreement (`sign`) and replication (`sig`), and `distinct`, which flags instruments that were expected to replicate but did not (`sig > 0.8` and replication  $p > 0.1$ ), or were expected to have the same sign but did not (`sign > 0.8`).

`CAMERA$replication_evaluation()`: The function explains what contributes to the replication of gene-trait association between the populations, considering LD structure.

*Usage:*

```
CAMERA$replication_evaluation(
  instrument = self$instrument_raw,
  ld = self$ld_matrices
)
```

*Arguments:*

*instrument* Instruments for the exposure that are selected by using the provided methods in CAMERA (*x\$instrument\_raw*, *x\$instrument\_maxz*, *x\$instrument\_susie*, *x\$instrument\_paintor*). Default is *x\$instrument\_raw*.

*ld* LD matrix obtained by using *x\$regional\_ld\_matrices()*

*Returns:* Table of the regression result

**CAMERA\$check\_phenotypes():** This function evaluates how comparable genetic associations in population 1 are with those in population 2. The function checks whether 1) the chosen IDs for the exposure or outcome can be used for the further steps and 2) the units for the genetic associations are comparable between populations.

*Usage:*

```
CAMERA$check_phenotypes(ids = self$exposure_ids)
```

*Arguments:*

*ids* ID for the exposure or the outcome. Default is *x\$exposure\_ids*.

*Returns:* Table of the result.

**CAMERA\$cross\_estimate():** This method performs a IVW analysis on the harmonised data. It fits two linear models, one considering all populations and another considering each population separately. It then performs a fixed effects meta-analysis to estimate the heterogeneity across different populations. The results are stored in the *mrres* attribute of the CAMERA object.

*Usage:*

```
CAMERA$cross_estimate(dat = self$harmonised_dat)
```

*Arguments:*

*dat* A data frame containing the harmonised data. It should have the columns *beta.y*, *beta.x*, *se.y*, and *pops*. If not provided, the method uses the *harmonised\_dat* attribute of the CAMERA object.

*Returns:* Data frame, also stored in *x\$mrres*, with one row for the estimate across all populations (*pops == "All"*) and one row for each population, with columns:

- *pops*: population
- Estimate, Std. Error, t value,  $\Pr(>|t|)$ : the IVW MR estimate, i.e. the weighted regression of the SNP-outcome associations on the SNP-exposure associations through the origin with weights  $1 / se.y^2$ , its standard error, t statistic and p-value
- Qj, Qjpval, Qdf: for each population, the contribution of its estimate to the heterogeneity between populations and its p-value (1 degree of freedom), from a fixed effects meta analysis of the population estimates; a small p-value suggests the estimate in that population differs from the others. For the "All" row these are Cochran's Q statistic across all populations, its p-value (null hypothesis: the same causal effect in all populations) and its degrees of freedom.

**CAMERA\$plot\_cross\_estimate():** Plot the results from *cross\_estimate*

*Usage:*

```
CAMERA$plot_cross_estimate(est = self$mrres, qj_alpha = 0.05)
```

*Arguments:*

`est` Results from `cross_estimate()`. Default is `x$mrres`.

`qj_alpha` Significance threshold for highlighting heterogeneous estimates. Default is 0.05.

*Returns:* Plot

`CAMERA$extract_instruments()`: This function searches for GWAS significant SNPs ( $P < 5E-8$ ) for a specified set of the exposures. This method is equivalent to the `instrumnet` extraction method for Multivariable MR. Reference here: [https://mrcieu.github.io/TwoSampleMR/reference/mv\\_extract\\_exposures.html](https://mrcieu.github.io/TwoSampleMR/reference/mv_extract_exposures.html)

*Usage:*

```
CAMERA$extract_instruments(
  exposure_ids = self$exposure_ids,
  harmonise_strictness = 1,
  ...
)
```

*Arguments:*

`exposure_ids` ID for the exposure. Default is `x$exposure_ids`.

`harmonise_strictness` Passed to `TwoSampleMR::mv_extract_exposures()`. Default is 1, which assumes all alleles are on the forward strand, as they are in OpenGWAS. The `TwoSampleMR` default of 2 infers the strand of palindromic SNPs from their allele frequencies, which wrongly flips palindromic SNPs whose allele frequencies are on different sides of 0.5 in different populations, and drops palindromic SNPs with intermediate allele frequencies.

... Further arguments passed to `TwoSampleMR::mv_extract_exposures()`

*Returns:* Data frame in `x$instrument_raw`

`CAMERA$extract_instrument_regions()`: This function extract genomic regions around each instrument (e.g. 50kb) obtained from `x$extract_instruments()`.

*Usage:*

```
CAMERA$extract_instrument_regions(
  radius = self$radius,
  instrument_raw = self$instrument_raw,
  exposure_ids = self$exposure_ids
)
```

*Arguments:*

`radius` Set a range of the region to search

`instrument_raw` A set of instruments obtained from `x$extract_instruments()`

`exposure_ids` ID for the exposure. Default is `x$exposure_ids`.

*Returns:* Data frame in `x$instrument_regions`

`CAMERA$scan_regional_instruments()`: The function searches for a SNP that is best associated across the populations within the identified regions by using `x$extract_instrument_regions()`.

*Usage:*

```
CAMERA$scan_regional_instruments(
  instrument_raw = self$instrument_raw,
  instrument_regions = self$instrument_regions
)
```

*Arguments:*

`instrument_raw` Instruments for the exposures by using `x$extract_instrument()`  
`instrument_regions` Genomic regions identified by using `x$extract_instrument_regions()`

*Returns:* List of z scores for the chosen SNPs in `x$instrument_region_zscores`. Data frame in `x$instrument_maxz`

`CAMERA$plot_regional_instruments_maxz()`: The function draws a plot shows how CAMERA selected the instruments for the exposure from the selected genomic regions.

*Usage:*

```
CAMERA$plot_regional_instruments_maxz(
  instrument_region_zscores = self$instrument_region_zscores,
  instruments = self$instrument_raw,
  region = 1:min(10, nrow(instruments)),
  comparison = FALSE
)
```

*Arguments:*

`instrument_region_zscores` Y axis. Z scores based on the SNP-exposure associations.  
`instruments` Use this option to draw a separate plot for the selected instruments.  
`region` X axis. Number of genomic regions to be shown in the plot. Default is 10.  
`comparison` Use this option to compare the selected instruments by different instrument selection methods in one plot.

*Returns:* Plot

`CAMERA$regional_ld_matrices()`: Generate LD matrices for instrument regions. If we want to do fine mapping we need to get an LD matrix for the whole region (for each population) We then need to harmonise the LD matrix to the summary data, and the summary datasets to each other. The function obtains an LD matrix for the selected genomic regions.

*Usage:*

```
CAMERA$regional_ld_matrices(
  instrument_regions = self$instrument_regions,
  bfiles = self$bfiles,
  pops = self$pops,
  plink = self$plink
)
```

*Arguments:*

`instrument_regions` Genomic regions identified by using `x$extract_instrument_regions()`  
`bfiles` Location of LD reference files for each population (Download from: <http://files.mrcieu.ac.uk/ld/1kg.v3.tgz>)  
`pops` Ancestry information for each population (i.e. AFR, AMR, EUR, EAS, SAS)  
`plink` Location of executable plink (version 1.90 is recommended)

*Returns:* Data frame of LD matrix (`x$ld_matrices`)

`CAMERA$susie_finemap_regions()`: Fine-mapping using `susieR` (<https://github.com/stephenslab/susieR>) to extract instruments for the exposure for multiple populations. The function identifies credible sets that overlap between the populations and determine the best SNP in a genomic region to be used as instrument.

*Usage:*

```
CAMERA$susie_finemap_regions(
  dat = self$instrument_regions,
  ld = self$ld_matrices
)
```

*Arguments:*

**dat** Genomic regions identified by using `x$extract_instrument_regions()`  
**ld** LD matrix obtained by using `x$regional_ld_matrices()`

**Returns:** Result from `susieR` in `x$susie_results`. Data frame in `x$instrument_susie`

**CAMERA\$paintor\_finemap\_regions():** Fine-mapping using PAINTOR ([https://github.com/gkichaev/PAINTOR\\_V3.0](https://github.com/gkichaev/PAINTOR_V3.0)) to extract instruments for the exposure for multiple populations. The function chooses the SNP with highest posterior probability and associations in each exposure.

*Usage:*

```
CAMERA$paintor_finemap_regions(
  region = self$instrument_regions,
  ld = self$ld_matrices,
  PAINTOR = "PAINTOR",
  workdir = tempdir()
)
```

*Arguments:*

**region** Genomic regions identified by using `x$extract_instrument_regions()`  
**ld** LD matrix obtained by using `x$regional_ld_matrices()`  
**PAINTOR** Path to executable PAINTOR. Default="PAINTOR"  
**workdir** Working directory to save the output. Default=tempdir()

**Returns:** Result from PAINTOR in `x$paintor_results`. Data frame in `x$instrument_paintor`

**CAMERA\$MsCAVIAR\_finemap\_regions():** Fine-mapping using MsCAVIAR (<https://github.com/nlapier2/MsCAVIAR>) to extract instruments for the exposure for multiple populations.

*Usage:*

```
CAMERA$MsCAVIAR_finemap_regions(
  region = self$instrument_regions,
  ld = self$ld_matrices,
  MsCAVIAR = "MsCAVIAR",
  workdir = tempdir()
)
```

*Arguments:*

**region** Genomic regions identified by using `x$extract_instrument_regions()`  
**ld** LD matrix obtained by using `x$regional_ld_matrices()`  
**MsCAVIAR** Path to executable MsCAVIAR. Default="MsCAVIAR"  
**workdir** Working directory to save the output. Default=tempdir()

**Returns:** Result from MsCAVIAR in `x$mscaviar_results`. Data frame in `x$instrument_mscaviar`

**CAMERA\$fema\_regional\_instruments():** Identify the best variant for each region by meta-analysing the associations across the populations

*Usage:*

```
CAMERA$fema_regional_instruments(
  method = "fema",
  instrument_regions = self$instrument_regions,
  instrument_raw = self$instrument_raw,
  n = self$exposure_metadata$sample_size
)
```

*Arguments:*

method Meta-analysis method, either "fema" (fixed effects meta-analysis) or "zma" (Z score meta-analysis). Default is "fema".

instrument\_regions Genomic regions identified by using x\$extract\_instrument\_regions()

instrument\_raw Instruments for the exposures obtained from x\$extract\_instruments()

n Sample sizes for each exposure, required for method = "zma". Default is the sample sizes from x\$exposure\_metadata.

*Returns:* Data frame of the selected instruments, also stored in x\$instrument\_fema

CAMERA\$plot\_regional\_instruments(): Plot the associations in a region for each population and for the meta-analysis

*Usage:*

```
CAMERA$plot_regional_instruments(
  region,
  instrument_regions = self$instrument_regions,
  meta_analysis_regions = self$instrument_fema_regions
)
```

*Arguments:*

region Name of the region to plot, i.e. one of names(x\$instrument\_regions)

instrument\_regions Genomic regions identified by using x\$extract\_instrument\_regions()

meta\_analysis\_regions Meta-analysis results for each region obtained from x\$fema\_regional\_instruments()

*Returns:* Plot

CAMERA\$get\_metadata(): Get metadata for the exposures and outcomes from the OpenGWAS API

*Usage:*

```
CAMERA$get_metadata(
  exposure_ids = self$exposure_ids,
  outcome_ids = self$outcome_ids
)
```

*Arguments:*

exposure\_ids IDs for the exposures. Default is x\$exposure\_ids.

outcome\_ids IDs for the outcomes. Default is x\$outcome\_ids.

*Returns:* List of the exposure and outcome metadata, also stored in x\$exposure\_metadata and x\$outcome\_metadata

`CAMERA$estimate_instrument_heterogeneity_per_variant()`: Test for heterogeneity of effect estimates between populations. For each SNP this function will provide a Cochran's Q test statistic - a measure of heterogeneity of effect sizes between populations. A low p-value means high heterogeneity. In addition, for every SNP it gives a per population p-value - this can be interpreted as asking for each SNP is a particular giving an outlier estimate.

*Usage:*

```
CAMERA$estimate_instrument_heterogeneity_per_variant(dat = self$harmonised_dat)
```

*Arguments:*

`dat` Harmonised data. Default is `x$harmonised_dat`.

*Returns:* Data frame of the Q statistic, its p-value and FDR adjusted p-value for each SNP, also stored in `x$instrument_heterogeneity_per_variant`

`CAMERA$mrgxe()`: Estimate the pleiotropic effect of each variant using MR GxE across populations, see `egger_bootstrap()`

*Usage:*

```
CAMERA$mrgxe(
  dat = self$harmonised_dat,
  variant_list = subset(self$instrument_heterogeneity_per_variant, Qfdr < 0.05)$SNP,
  nboot = 100
)
```

*Arguments:*

`dat` Harmonised data. Default is `x$harmonised_dat`.

`variant_list` SNPs to analyse. Default is the SNPs with FDR < 0.05 in `x$instrument_heterogeneity_per_variant`

`nboot` Number of bootstraps. Default is 100.

*Returns:* Data frame of the MR GxE results, also stored in `x$mrgxe_res`

`CAMERA$mrgxe_plot()`: Plot the MR GxE pleiotropy estimates for each variant

*Usage:*

```
CAMERA$mrgxe_plot(mrgxe_res = self$mrgxe_res)
```

*Arguments:*

`mrgxe_res` Results from `mrgxe()`. Default is `x$mrgxe_res`.

*Returns:* Plot

`CAMERA$mrgxe_plot_variant()`: Plot the instrument-exposure against instrument-outcome associations across populations for selected variants

*Usage:*

```
CAMERA$mrgxe_plot_variant(
  variant = self$mrgxe_res %>% dplyr::ungroup() %>% dplyr::filter(p.adjust(a_pval,
    "fdr") < 0.05) %>% {
    . $SNP
  },
  dat = self$harmonised_dat
)
```

*Arguments:*

variant SNPs to plot. Default is the SNPs with FDR < 0.05 for the pleiotropy estimate in `x$mrgxe_res`.

dat Harmonised data. Default is `x$harmonised_dat`.

*Returns:* Plot

`CAMERA$make_outcome_data()`: The function extracts summary statistics of given a list of instruments and the outcomes

*Usage:*

```
CAMERA$make_outcome_data(exp = self$instrument_raw, p_exp = 0.05/nrow(exp))
```

*Arguments:*

exp Instruments for the exposure that are selected by using the provided methods in CAMERA (`x$instrument_raw`, `x$instrument_maxz`, `x$instrument_susie`, `x$instrument_paintor`). Default is `x$instrument_raw`.

p\_exp Statistical threshold to determine significance. Default is "bonferroni", which is equal to 0.05/number of the instruments.

*Returns:* Data frame in `x$instrument_outcome`

`CAMERA$make_outcome_local()`: The function extracts summary statistics of given a list of instruments and the outcomes from local data

*Usage:*

```
CAMERA$make_outcome_local(
  exp = self$instrument_raw,
  out = self$instrument_outcome_regions,
  p_exp = 0.05/nrow(exp)
)
```

*Arguments:*

exp Instruments for the exposure that are selected by using the provided methods in CAMERA (`x$instrument_raw`, `x$instrument_fema`, `x$instrument_susie`, `x$instrument_paintor`). Default is `x$instrument_raw`.

out Outcome data for each instrument region. Default is `x$instrument_outcome_regions`.

p\_exp Statistical threshold to determine significance. Default is "bonferroni", which is equal to 0.05/number of the instruments.

*Returns:* Data frame in `x$instrument_outcome`

`CAMERA$harmonise()`: This function harmonises the alleles and effects between the exposure and outcome.

*Usage:*

```
CAMERA$harmonise(exp = self$instrument_raw, out = self$instrument_outcome)
```

*Arguments:*

exp Instruments for the exposure that are selected by using the provided methods in CAMERA (`x$instrument_raw`, `x$instrument_fema`, `x$instrument_susie`, `x$instrument_paintor`). Default is `x$instrument_raw`.

out Instruments for the outcome by using `make_outcome_data()`. Default is `x$instrument_outcome`.

*Returns:* Data frame in x\$harmonised\_dat\_sem

CAMERA\$set\_summary(): Generate summary of exposure, outcome and population metadata

*Usage:*

```
CAMERA$set_summary()
```

*Returns:* data frame

CAMERA\$pleiotropy(): Estimate similarity of horizontal pleiotropy across ancestries. For each ancestry, identify outliers in the MR analysis based on per-variant Q statistics. Then estimate the deviation from the main estimates for all outliers across all ancestries. Finally, determine if the pleiotropy deviation is consistent across all ancestries

*Usage:*

```
CAMERA$pleiotropy(harmonised_dat = self$harmonised_dat, mrres = self$mrres)
```

*Arguments:*

harmonised\_dat Outcome from harmonise function

mrres Outcome from cross\_estimate

*Returns:*

- data frame of outliers pleiotropy\_outliers
- data frame of heterogeneity for each outlier / ancestry combination pleiotropy\_Q\_outliers
- data frame of agreement of outlier effects pleiotropy\_agreement

CAMERA\$plot\_pleiotropy(): Plot pleiotropy results

*Usage:*

```
CAMERA$plot_pleiotropy(dat = self$pleiotropy_outliers)
```

*Arguments:*

dat Output from pleiotropy - pleiotropy\_outliers

*Returns:* plot

CAMERA\$plot\_pleiotropy\_heterogeneity(): Plot pleiotropy results per variant

*Usage:*

```
CAMERA$plot_pleiotropy_heterogeneity(
  dat = self$pleiotropy_Q_outliers,
  pthresh = 0.05
)
```

*Arguments:*

dat Output from pleiotropy - pleiotropy\_Q\_outliers

pthresh p-value from Q statistic for inclusion in plots

*Returns:* plot

CAMERA\$perform\_basic\_sem(): Perform basic SEM analysis of the joint estimates in multiple ancestries.

*Usage:*

```
CAMERA$perform_basic_sem(harmonised_dat = self$harmonised_dat_sem)
```

*Arguments:*

harmonised\_dat Harmonised dataset obtained by using harmonised\_dat()

*Returns:* Table of the SEM results. Summary of the results available in x\$sem\_result.

CAMERA\$runsem(): Fit an SEM model using lavaan

*Usage:*

```
CAMERA$runsem(model, data, modname)
```

*Arguments:*

model lavaan model syntax

data Data frame of the data for the model

modname Name of the model used to label the results

*Returns:* Data frame of the SEM estimates for each population

CAMERA\$standardise\_data(): This function standardises the betas and SEs for the instruments-exposure/outcome associations when unit information is not matched across the populations. In case of large differences in genetic effects observed between the populations (e.g. due to sample size difference), the function scales the betas and SEs.

*Usage:*

```
CAMERA$standardise_data(
  dat = self$instrument_raw,
  standardise_unit = FALSE,
  standardise_scale = FALSE,
  scaling_method = "simple_mode"
)
```

*Arguments:*

dat Instruments for the exposure that are selected by using the provided methods in CAMERA (x\$instrument\_raw, x\$instrument\_maxz, x\$instrument\_susie, x\$instrument\_paintor). Default is x\$instrument\_raw.

dat A data frame containing the harmonised data. It should have the columns beta.y, beta.x, se.y, and pops. If not provided, the method uses the harmonised\_dat attribute of the CAMERA object.

dat Genomic regions identified by using x\$extract\_instrument\_regions()

dat Harmonised data. Default is x\$harmonised\_dat.

standardise\_unit Use this option if unit information is not matched.

standardise\_scale Use this option if genetic effects are substantially different due to study power.

scaling\_method Choose the methods to obtain scaling units (MR estimates of exposure 1 and exposure 2 or outcome 1 and outcome 2). Default is "simple\_mode".

CAMERA\$clone(): The objects of this class are cloneable with this method.

*Usage:*

```
CAMERA$clone(deep = FALSE)
```

*Arguments:*

deep Whether to make a deep clone.

---

|              |                           |
|--------------|---------------------------|
| CAMERA_local | <i>CAMERA_local class</i> |
|--------------|---------------------------|

---

## Description

A simple wrapper function for importing data from local files for use with the CAMERA class.

Create the object with `CAMERA_local$new()` and then call its `organise()` method. This reads the summary statistics files listed in `metadata`, finds and clumps the top hits for each dataset, pools them across populations, and extracts the region around each instrument from every dataset. The results are stored in the `instrument_raw`, `instrument_outcome`, `instrument_regions` and `instrument_outcome_regions` fields, which can be passed to the `import_from_local()` method of a [CAMERA](#) object.

See `vignette("import-local", package = "CAMeRa")` for a worked example.

## Details

`metadata` must be a data frame with one row per summary statistics file (i.e. per trait and population) and the following columns:

- `what`: either "exposure" or "outcome"
- `trait`: name of the trait, the same for all populations of a trait
- `pop`: population, e.g. "EUR"; must match the `pop` column of `ld_ref`
- `id`: a unique identifier for the dataset
- `fn`: path to the summary statistics file, which is read with `data.table::fread()`
- `chr_col`, `pos_col`, `ea_col`, `oa_col`, `eaf_col`, `beta_col`, `se_col`, `pval_col`: the name or number of the column in the file containing the chromosome, base pair position, effect allele, other allele, effect allele frequency, effect estimate, standard error, and p-value respectively

Other columns are ignored. Variants are identified by chromosome, position and alleles (as `chr:pos_a1_a2`, with the alleles in alphabetical order), so all files must use the same genome build.

`ld_ref` must be a data frame with columns `pop` and `bfile`, giving for each population the path to a PLINK binary fileset (without the `.bed/.bim/.fam` extension) used as the LD reference for clumping.

Top hits are only taken from datasets with at least two variants with p-value below `pthresh`. Currently `organise()` uses the first exposure trait and the first outcome trait in `metadata`.

## Public fields

`metadata` Data frame with information about the data. One row per dataset. See Details for the required columns

`ld_ref` Data frame with two columns - `pop` = population (referencing the `pop` values in `metadata`), `bfile` = path to plink file for that reference

`mc.cores` The number of processor cores to use

`plink_bin` Location of executable plink (version 1.90 is recommended)

radius Genomic window size to extract SNPs

minmaf Minimum allele frequency per dataset

pthresh P-value threshold for instrument inclusion

instrument\_raw A data frame of pooled instruments across all ancestries, that has been extracted from each ancestry for the exposure traits

instrument\_outcome Instruments in instrument\_raw extracted from the outcome datasets

instrument\_regions Named list of data frames of length number of unique instruments in instrument\_raw. Names of each item are the instruments. Each item is a list of regional extracts around the instrument from each population exposure study.

instrument\_outcome\_regions As per instrument\_regions but for the outcome datasets.

## Methods

### Public methods:

- CAMERA\_local\$new()
- CAMERA\_local\$standardise()
- CAMERA\_local\$read\_file()
- CAMERA\_local\$pool\_tophits()
- CAMERA\_local\$organise\_data()
- CAMERA\_local\$fixed\_effects\_meta\_analysis\_fast()
- CAMERA\_local\$organise()
- CAMERA\_local\$clone()

CAMERA\_local\$new(): Create a new CAMERA\_local object

#### Usage:

```
CAMERA_local$new(
  metadata,
  ld_ref,
  plink_bin,
  mc.cores = 1,
  radius = 25000,
  pthresh = 5e-08,
  minmaf = 0.01
)
```

#### Arguments:

metadata Data frame with information about the data. One row per dataset. See Details for the required columns

ld\_ref Data frame with two columns - pop = population (referencing the pop values in metadata), bfile = path to plink file for that reference. See Details

plink\_bin Location of executable plink (version 1.90 is recommended)

mc.cores Number of cores to use

radius Genomic window size to extract SNPs

radius Default 250000

pthresh P-value threshold for instrument inclusion

minmaf Minimum allele frequency per dataset

`CAMERA_local$standardise()`: Standardise the allele coding so that the effect allele is the alphabetically first allele, flipping the effect estimate and allele frequency where needed, and create a variant ID of the form `chr:pos_ea_oa`

*Usage:*

```
CAMERA_local$standardise(
  d,
  ea_col = "ea",
  oa_col = "oa",
  beta_col = "beta",
  eaf_col = "eaf",
  chr_col = "chr",
  pos_col = "pos",
  vid_col = "vid"
)
```

*Arguments:*

`d` data.frame  
`ea_col` Column name for effect allele  
`oa_col` Column name for other allele  
`beta_col` Column name containing beta coefficients  
`eaf_col` Column name containing allele frequency for effect allele  
`chr_col` Column name containing chromosome  
`pos_col` Column name containing position  
`vid_col` Column name containing variant ID

*Returns:* `d` with standardised alleles and a variant ID column

`CAMERA_local$read_file()`: Read in the summary statistics file for one dataset, drop variants with minor allele frequency below `minmaf`, and standardise the alleles

*Usage:*

```
CAMERA_local$read_file(m, minmaf = self$minmaf)
```

*Arguments:*

`m` File object  
`minmaf` Minimum allele frequency per dataset

*Returns:* A tibble with columns `chr`, `pos`, `eaf`, `beta`, `se`, `pval`, `ea`, `oa` and `vid`

`CAMERA_local$pool_tophits()`: Pool the top hits across datasets. For each trait, the regions within radius of its top hits are merged, and the variants in each region are extracted from every dataset. Within each region the variant with the largest absolute z-score for the trait is selected, considering only variants present in a number of datasets for which at least one such variant has p-value below `pthresh`

*Usage:*

```

CAMERA_local$pool_tophits(
  rawdat,
  tophits,
  metadata,
  radius = 250000,
  pthresh = 5e-08,
  mc.cores = 1
)

```

*Arguments:*

`rawdat` The raw data

`tophits` The top hits

`metadata` Data frame with information about the data. One row per dataset. See Details for the required columns

`radius` Genomic window size to extract SNPs

`radius` Default 250000

`pthresh` P-value threshold for instrument inclusion

`mc.cores` Number of cores to use

*Returns:* A list with elements `region_list` (the merged regions for each trait), `region_extract` (the extracted variants in each region) and `tophit_pool` (the selected variant from each region in every dataset)

`CAMERA_local$organise_data()`: Read in the data (unless `rawdat` is supplied), find the top hits in each dataset by clumping the variants with p-value below `pthresh` using [ieugwasr::ld\\_clump\(\)](#), and pool them with `pool_tophits()`

*Usage:*

```

CAMERA_local$organise_data(
  metadata = self$metadata,
  plink_bin = self$plink_bin,
  ld_ref = self$ld_ref,
  pthresh = self$pthresh,
  minmaf = self$minmaf,
  radius = self$radius,
  mc.cores = self$mc.cores,
  rawdat = NULL
)

```

*Arguments:*

`metadata` Data frame with information about the data. One row per dataset. See Details for the required columns

`plink_bin` Location of executable plink (version 1.90 is recommended)

`ld_ref` Data frame with two columns - `pop` = population (referencing the `pop` values in `metadata`), `bfile` = path to plink file for that reference. See Details

`pthresh` P-value threshold for instrument inclusion

`minmaf` Minimum allele frequency per dataset

`radius` Genomic window size to extract SNPs

`radius` Default 250000

`mc.cores` Number of cores to use

`rawdat` The raw data

*Returns:* As for `pool_tophits()`

`CAMERA_local$fixed_effects_meta_analysis_fast()`: Fixed effect inverse variance weighted meta-analysis of each row of `beta_mat` and `se_mat`

*Usage:*

`CAMERA_local$fixed_effects_meta_analysis_fast(beta_mat, se_mat)`

*Arguments:*

`beta_mat` Matrix of beta coefficients

`se_mat` Matrix of SEs

*Returns:* A vector of p-values, one per row

`CAMERA_local$organise()`: Organise the data for the exposure and outcome, populating the `instrument_raw`, `instrument_outcome`, `instrument_regions` and `instrument_outcome_regions` fields

*Usage:*

`CAMERA_local$organise()`

`CAMERA_local$clone()`: The objects of this class are cloneable with this method.

*Usage:*

`CAMERA_local$clone(deep = FALSE)`

*Arguments:*

`deep` Whether to make a deep clone.

## See Also

[CAMERA](#), `vignette("import-local", package = "CAMErA")`

---

egger\_bootstrap

*Perform MR GxE*

---

## Description

For a single variant estimated in different sub groups.

## Usage

`egger_bootstrap(b_gx, se_gx, b_gy, se_gy, nboot = 1000)`

**Arguments**

|       |                                                                    |
|-------|--------------------------------------------------------------------|
| b_gx  | Vector of instrument-exposure associations, one for each sub group |
| se_gx | Vector of standard errors to b_gx                                  |
| b_gy  | Vector of instrument-outcome associations, one for each sub group  |
| se_gy | Vector of standard errors for b_gy                                 |
| nboot | Number of bootstraps. Default=1000                                 |

**Details**

Estimate the degree of pleiotropy using MR GxE. This method uses a negative control type approach based on an assumption that the instrument-exposure association is uncorrelated with the pleiotropic effect. Therefore, as the instrument-exposure association reduces in magnitude, the effect on the outcome will reduce towards an intercept term which represents the pleiotropic effect.

Standard errors are obtained from parametric bootstrap

**Value**

List

- a = intercept estimate (pleiotropy)
- b = slope estimate (b\_iv effect)
- a\_se = standard error of intercept
- b\_se = standard error of slope
- a\_pval = p-value of intercept estimate
- b\_pval = p-value of slope estimate
- a\_mean = mean value of intercept from bootstraps
- b\_mean = mean value of slope estimates from bootstraps

---

fixed\_effects\_meta\_analysis

*Perform fixed effects meta analysis for one association*

---

**Description**

Perform fixed effects meta analysis for one association

**Usage**

```
fixed_effects_meta_analysis(beta_vec, se_vec, infl = 10000)
```

**Arguments**

|          |                                                                                                                                    |
|----------|------------------------------------------------------------------------------------------------------------------------------------|
| beta_vec | Vector of betas                                                                                                                    |
| se_vec   | Vector of ses                                                                                                                      |
| infl     | Inflation factor - how much larger is the estimate than the estimate of the tightest SE - for use in removing unreliable estimates |

**Value**

list of results

---

fixed\_effects\_meta\_analysis\_fast

*Fixed effects meta analysis vectorised across multiple SNPs*

---

**Description**

Assumes effects across studies are all on the same scale

**Usage**

```
fixed_effects_meta_analysis_fast(beta_mat, se_mat)
```

**Arguments**

|          |                                                      |
|----------|------------------------------------------------------|
| beta_mat | Matrix of betas - rows are SNPs, columns are studies |
| se_mat   | Matrix of SEs - rows are SNPs, columns are studies   |

**Value**

list of meta analysis betas and SEs

---

identify\_blowup\_estimates

*Identify blown up estimates*

---

**Description**

Sometimes estimates appear unstable. They are likely unreliable and best to not use for heterogeneity analyses etc.

**Usage**

```
identify_blowup_estimates(b, se, infl)
```

**Arguments**

|      |                                                                                         |
|------|-----------------------------------------------------------------------------------------|
| b    | Vector of betas                                                                         |
| se   | Vector of SEs                                                                           |
| infl | Inflation factor - how much larger is the estimate than the estimate of the tightest SE |

**Value**

index of betas to remove

---

|              |                                                                                                        |
|--------------|--------------------------------------------------------------------------------------------------------|
| prop_overlap | <i>Estimate expected vs observed replication of effects between discovery and replication datasets</i> |
|--------------|--------------------------------------------------------------------------------------------------------|

---

**Description**

Taken from Okbay et al 2016. Under the assumption that all discovery effects are unbiased, what fraction of associations would replicate in the replication dataset, given the differential power of the discovery and replication datasets. Uses standard error of the replication dataset to account for differences in sample size and distribution of independent variable

**Usage**

```
prop_overlap(b_disc, b_rep, se_disc, se_rep, alpha)
```

**Arguments**

|         |                                            |
|---------|--------------------------------------------|
| b_disc  | Vector of discovery betas                  |
| b_rep   | Vector of replication betas                |
| se_disc | Vector of discovery standard errors        |
| se_rep  | Vector of replication standard errors      |
| alpha   | Nominal replication significance threshold |

**Value**

List of results

- res: aggregate expected replication rate vs observed replication rate
- variants: per variant expected replication rates

---

|                 |                                    |
|-----------------|------------------------------------|
| z_meta_analysis | <i>P-value based meta analysis</i> |
|-----------------|------------------------------------|

---

**Description**

Uses weighted Z scores following advice from <https://onlinelibrary.wiley.com/doi/full/10.1111/j.1420-9101.2005.00917.x> Suggested weights are  $1/se^2$  However se is scale dependent and it would be ideal to avoid scale issues at this stage So using instead calculate expected se based on n and af. Assumes continuous traits in the way it uses n (i.e. not case control aware at the moment)

**Usage**

```
z_meta_analysis(beta_mat, se_mat, n, eaf_mat)
```

**Arguments**

|          |                                                                   |
|----------|-------------------------------------------------------------------|
| beta_mat | Matrix of betas - rows are SNPs, columns are studies              |
| se_mat   | Matrix of SEs - rows are SNPs, columns are studies                |
| n        | Vector of sample sizes for each                                   |
| eaf_mat  | Matrix of allele frequencies - rows are SNPs, columns are studies |

# Index

CAMERA, [3](#), [18](#), [22](#)  
CAMEra (CAMEra-package), [2](#)  
camera (CAMEra-package), [2](#)  
CAMEra-package, [2](#)  
CAMERA\_local, [18](#)  
  
data.table::fread(), [18](#)  
  
egger\_bootstrap, [22](#)  
  
fixed\_effects\_meta\_analysis, [23](#)  
fixed\_effects\_meta\_analysis\_fast, [24](#)  
  
identify\_blowup\_estimates, [24](#)  
ieugwasr::ld\_clump(), [21](#)  
  
prop\_overlap, [25](#)  
  
TwoSampleMR::mr\_ivw(), [7](#)  
TwoSampleMR::mr\_simple\_mode(), [7](#)  
  
z\_meta\_analysis, [26](#)