

# Package: simmrd (via r-universe)

August 17, 2024

**Title** Simulation of Mendelian Randomization data

**Version** 0.0.0.9000

**Description** This package generates simulation data to use in the evaluation of univariable or multivariable Mendelian Randomization methods. MR scenarios can include uncorrelated horizontal pleiotropy, correlated horizontal pleiotropy, weak instruments, winner's curse, and correlated SNP instruments.

**License** MIT + file LICENSE

**URL** <https://github.com/noahlorinczcomi/simmr>

**BugReports** <https://github.com/noahlorinczcomi/simmr/issues>

**Imports** ggplot2, ggpubr, mvnfast

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.3

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

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

**RemoteUrl** <https://github.com/noahlorinczcomi/simmrd>

**RemoteRef** HEAD

**RemoteSha** 986323e3c913d43f1e27dc2267d9abd48fefbec4

## Contents

adj_overlap . . . . .	2
arl . . . . .	3
biggwas . . . . .	3
classIVs . . . . .	4
generate_individual . . . . .	4
generate_summary . . . . .	5
makeBlocks . . . . .	6

parthcorr . . . . .	7
parthstd . . . . .	7
pfun . . . . .	8
plot_simdata . . . . .	9
plot_simdata_lower . . . . .	9
pruning . . . . .	10
setf . . . . .	11
std . . . . .	11

<b>Index</b>	<b>12</b>
--------------	-----------

---

adj_overlap	<i>Helper function</i>
-------------	------------------------

---

## Description

Helper function

## Usage

```
adj_overlap(
  exposure_overlap_proportions,
  prop_gwas_overlap_Xs_and_Y,
  number_of_exposures
)
```

## Arguments

exposure\_overlap\_proportions  
 scalar or matrix of overlap proportions between exposures GWAS

prop\_gwas\_overlap\_Xs\_and\_Y  
 scalar or vector of overlap proportions between exposures and outcome GWAS

number\_of\_exposures  
 number of exposures

## Examples

```
adj_overlap(
  exposure_overlap_proportions = 0.2,
  prop_gwas_overlap_Xs_and_Y = 0.1,
  number_of_exposures = 3
)
```

---

ar1                      *Helper function*

---

**Description**

Helper function

**Usage**

```
ar1(n, rho = 0.5)
```

**Arguments**

n	The number of rows (and columns) of the matrix
rho	rho

**Examples**

```
ar1(2)
```

---

biggwas                      *Helper function*

---

**Description**

Helper function

**Usage**

```
biggwas(x, G)
```

**Arguments**

x	phenotype vector
G	genotype matrix

**Examples**

```
## Not run:  
biggwas()  
  
## End(Not run)
```

classIVs                      *Helper function*

---

**Description**

Helper function

**Usage**

```
classIVs(ix, uhpix, chpix)
```

**Arguments**

ix	ix
uhpix	uhpix
chpix	chpix

**Examples**

```
## Not run:  
classIVs()  
  
## End(Not run)
```

---

generate\_individual      *An Individual-Level Data-Generating Function*

---

**Description**

This function generates simulated individual-level and data given a list of parameters

**Usage**

```
generate_individual(params)
```

**Arguments**

params	List of parameters used to generate simulated data
--------	--

**Examples**

```
## Not run:
individual_params <-
  list(
    sample_size_Xs = 5e4, # exposure GWAS sample sizes
    sample_size_Y = 5e4, # outcome GWAS sample size
    prop_gwas_overlap_Xs_and_Y = 0.5, # proportion of exposure and outcome GWAS overlap
    number_of_exposures = 2, # 4 number of exposures
    phenotypic_correlation_Xs = 0.2, # phenotypic correlation between exposures
    genetic_correlation_Xs = 0, # genetic correlation between exposures
    Xs_variance_explained_by_U = 1/4 - 0.12, # exposures variance explained by confounder
    Y_variance_explained_by_Xs = c(0, 0.5), # outcome variance explained by exposures
    signs_of_causal_effects = c(1, 1), # signs of causal effects
    Y_variance_explained_by_U = 0.1, # outcome variance explained by confounder
    number_of_causal_SNPs = 200, # number of SNPs causing exposures
    mafs_of_causal_SNPs = stats::runif(100, 0.1, 0.5), # minor allele frequency of causal SNPs
    Xs_variance_explained_by_g = 0.12, # exposures variance explained by SNPs
    number_of_UHP_causal_SNPs = 30, # number of UHP exposure SNPs
    number_of_CHP_causal_SNPs = 10, # number of CHP exposure SNPs
    Y_variance_explained_by_UHP = 0.05, # outcome variance explained by UHP SNPs
    U_variance_explained_by_CHP = 0.05, # outcome variance explained by CHP SNPs
    LD_causal_SNPs = 'I', # independent causal exposure SNPs
    number_of_LD_blocks = 1, # number of independent LD blocks
    MR_standardization = 'Z', # standardization of GWAS summary statistics
    simtype = 'weak', # simulation performed using weak instruments
    MVMR_IV_selection_type = 'joint', # P-values for IV selection based on joint test for exposures
    IV_Pvalue_threshold = 1, # P-value threshold for candidate IVs
    LD_pruning_r2 = 1, # upper boundary of squared LD correlation
    N_of_LD_ref = Inf, # size of the LD reference panel
    fix_Fstatistic_at = 10 # average across exposures using full MVMR IV set
  )
gwas_data <- generate_individual(individual_params)

## End(Not run)
```

---

generate\_summary

*An Summary Statistic Data-Generating Function*


---

**Description**

This function generates simulated summary-level and data given a list of parameters

**Usage**

```
generate_summary(params)
```

**Arguments**

params                    List of parameters used to generate simulated data

## Examples

```
## Not run:
summary_params <-
  list(
    sample_size_Xs = 30000, # exposure GWAS sample sizes
    sample_size_Y = 30000, # outcome GWAS sample size
    prop_gwas_overlap_Xs_and_Y = 1, # proportion of exposures' and outcome GWAS overlap
    number_of_exposures = 3, # number of exposures
    number_of_causal_SNPs = 100, # number of SNPs causing each exposure
    number_of_UHP_causal_SNPs = 0, # number of UHP causal SNPs
    number_of_CHP_causal_SNPs = 20, # number of CHP causal SNPs
    ratio_of_UHP_variance = 0.15, # ratio of UHP variance to valid IV variance
    ratio_of_CHP_variance = 0.25, # ratio of CHP variance to valid IV variance
    CHP_correlation = -0.5, # correlation between CHP and valid IV effect sizes
    simtype = 'winners', # performs IV selection based on P-value
    fix_Fstatistic_at = 10, # ignored because simtype='winners'
    prop_gwas_overlap_Xs = 1, # overlap of exposures' GWAS
    phenotypic_correlation_Xs = 0.3, # phenotypic correlations between exposures
    genetic_correlation_Xs = 0.15, # genetic correlation between exposures
    phenotypic_correlations_Xs_and_Y = 0.3, # phenotypic correlations b/w exposures and outcome
    true_causal_effects = 0.3, # true causal effect sizes
    Xs_variance_explained_by_g = 0.10, # exposure variance explained by SNPs
    LD_causal_SNPs = 'ar1(0.5)', # LD between causal exposure SNPs
    number_of_LD_blocks = 3, # number of independent LD blocks
    MR_standardization = 'none', # does not standardize GWAS estimates
    MVMR_IV_selection_type = 'union', # SNPs associated with >0 exposures are candidate IVs
    IV_Pvalue_threshold = 5e-8, # only SNPs with P<this threshold are candidate IVs
    LD_pruning_r2 = 1, # the upper LD r2 pruning threshold for candidate IVs
    N_of_LD_ref = Inf # the sample size of the LD reference panel
  )
gwas_data <- generate_summary(summary_params)

## End(Not run)
```

---

makeBlocks

*A function to make LD blocks*

---

## Description

Helper function to make LD blocks

## Usage

```
makeBlocks(LD_causal_SNPs, number_of_causal_SNPs, nblocks = 1)
```

## Arguments

LD\_causal\_SNPs the LD structure of the causal SNPs

number\_of\_causal\_SNPs  
the total number of causal SNPs  
nblocks the number of independent LD blocks

**Examples**

```
## Not run:  
makeBlocks()  
  
## End(Not run)
```

---

parthcorr *Helper function*

---

**Description**

Helper function

**Usage**

```
parthcorr(x, n)
```

**Arguments**

x Matrix or numeric value  
n Number of rows and columns of the matrix

**Examples**

```
## Not run:  
parthcorr()  
  
## End(Not run)
```

---

parthstd *Helper function*

---

**Description**

Helper function

**Usage**

```
parthstd(bx, by, bxse, byse, maf, nx, ny, MR_standardization_type)
```

**Arguments**

bx	bx
by	by
bxse	bxse
byse	byse
maf	Minor allele frequency
nx	nx
ny	ny
MR_standardization_type	Standardization type

**Examples**

```
## Not run:  
parthstd()  
  
## End(Not run)
```

---

pfun

*Helper function*

---

**Description**

Helper function

**Usage**

```
pfun(x, y, chpix, uhpix, ...)
```

**Arguments**

x	x
y	y
chpix	chipx
uhpix	uhpix
...	Additional arguments passed to plot()

**Examples**

```
## Not run:  
pfun()  
  
## End(Not run)
```



---

plot_simdata	<i>Plot simulated data</i>
--------------	----------------------------

---

**Description**

Plot simulated data.

**Usage**

```
plot_simdata(  
  data,  
  params = params,  
  exposure_specific_plot = "total",  
  verbose = TRUE  
)
```

**Arguments**

data	direct output from generate()
params	Named list of parameters
exposure_specific_plot	One of 'total', 'joint', 'conditional'
verbose	Logical, default TRUE

**Examples**

```
## Not run:  
# If you used generate_summary(), execute the following  
plot_simdata(gwas_data,summary_params)  
  
# If you used generate_individual(), execute the following  
plot_simdata(gwas_data,individual_params)  
  
## End(Not run)
```

---

plot_simdata_lower	<i>Helper function</i>
--------------------	------------------------

---

**Description**

Helper function

**Usage**

```
plot_simdata_lower(data, params = params, showFstat = TRUE)
```

**Arguments**

data	direct output from generate()
params	Named list of parameters
showFstat	Logical, default TRUE

**Examples**

```
## Not run:  
plot_simdata_lower()  
  
## End(Not run)
```

---

pruning

*Pruning SNPs*

---

**Description**

Pruning SNPs

**Usage**

```
pruning(jointPs, R, r2)
```

**Arguments**

jointPs	joint p-degree of freedom chi-square tests for IVs
R	LD correlation matrix for SNPs
r2	upper squared LD r2 threshold for pruning

**Examples**

```
## Not run:  
pruning()  
  
## End(Not run)
```

---

setf                      *Helper function*

---

**Description**

Helper function

**Usage**

```
setf(bxunstd, nX, fix_Fstatistic_at)
```

**Arguments**

bxunstd	bxunstd
nX	nX
fix_Fstatistic_at	Value to fix the F-statistic at

**Examples**

```
## Not run:  
setf()  
  
## End(Not run)
```

---

std                      *Helper function*

---

**Description**

Helper function

**Usage**

```
std(x)
```

**Arguments**

x	Vector to standardise
---	-----------------------

**Examples**

```
std(0:10)
```

# Index

## \* data

- generate\_individual, 4
- generate\_summary, 5

adj\_overlap, 2

ar1, 3

biggwas, 3

classIVs, 4

generate\_individual, 4

generate\_summary, 5

makeBlocks, 6

parthcorr, 7

parthstd, 7

pfun, 8

plot\_simdata, 9

plot\_simdata\_lower, 9

pruning, 10

setf, 11

std, 11