GSMR

Generalised Summary-data-based Mendelian Randomisaion

GCTA SMR

GSMR OSCA

GCTB

Program in CTG

CTG forum

Overview

The **gsmr** R-package

implements the GSMR (Generalised Summary-data-based Mendelian Randomisation) method to test for

putative

causal

association

between a

risk factor and

a disease

using

summary-level data from genome-wide association studies (GWAS) (Zhu et al. 2018

Nat.
Commun.).
The R
package is
developed by
Zhihong Zhu,
Zhili Zheng,
Futao Zhang

and Jian Yang at Institute for Molecular Bioscience, the University of Queensland. Bug reports or questions: jian.yang@uq.edu

Note: The GSMR method has also been implemented in the GCTA software (GCTA-GSMR)

Citation

Zhu, Z. et al. (2018) Causal

associations between risk factors and common diseases inferred from GWAS summary

data. Nat. Commun. 9, 224 (https://www.natu
017-02317-2).

Source code

gsmr_1.0.9.tar.gz

Note: We included a new HEIDI-outlier method (as part of the GSMR analysis) in gsmr v1.0.7. However, the new HEIDI-outlier method is currently under development and subject to changes during the method development. From the GSMR R package (>=

version 1.0.8), we changed the default back to the original HEIDI-outlier method described in Zhu et al. (2018 Nature Communications) and added a temporary flag (‘gsmr2_beta’) to test the new method. The command to use this flag can be found in the tutorial below. The new HEIDI-outlier method in gsmr (>= version 1.0.8) has been tested by extensive simulations and real data analyses. We will make a formal release in our next GSMR paper.

Sample data are available in

test_data.zip. This document has been integrated in the gsmr R-package, we can check it by the standard command “?function_name” in R.

Installation

The **gsmr**

requires R >= 2.15, you can install it in R by:

```
# gsmr requires the R-package(s)
install.pa
```

```
ckages(c('
survey'));
# install
gsmr
install.pa
ckages("ht
tp://cnsge
nomics.com
/software/
gsmr/stati
c/gsmr_1.0
.0.tar.gz"
.9.tar.gz,repos=NULL,type="source")
```

Update log

V1.0.9 (gmr_1.0.9.tar.gz

PDF, 18 Jun. 2019): Change the flag ‘gsmr_beta’ to ‘gsmr2_beta’.

V1.0.8

(gmr_1.0.8.tar.gz PDF, 21 Jan. 2019): Added a flag ‘gsmr_beta’ to use a testing version of the

HEIDI-outlier method.

V1.0.7 (gmr_1.0.7.tar.gz PDF, 9 Oct. 2018): Added a multi-SNP-based HEIDI-outlier test in the HEIDI-outlier analysis.

V1.0.6 (gmr_1.0.6.tar.gz PDF, 23

Jan. 2018): Added a function to remove SNPs in high LD.

V1.0.5 (gmr_1.0.5.tar.gz PDF, 13

Dec. 2017): Improved the approximation of the sampling covariance matrix.

V1.0.4

(gsmr_1.0.4.tar.gz PDF, 6 Nov. 2017): Added the bi-directional GSMR analysis. The HEIDI-outlier analysis has been integrated in the GSMR analysis by default.

V1.0.3 (gsmr_1.0.3.tar.gz

PDF, 12 Oct. 2017): Added more example data.

Removed the initial versions (8 Nov 2016).

Tutorial

The GSMR analysis only requires summary-level data from GWAS.

Here is an example, where the risk factor (*x*) is LDL cholesterol (LDL-c) and the disease (*y*) is coronary artery disease (CAD). GWAS summary data for both LDL-c and CAD are available in the public domain (Global Lipids Genetics Consortium et al. 2013, Nature Genetics; Nikpay, M. et al. 2015, Nature Genetics).

1. Prepare data for GSMR

1.1 Load the GWAS summary data

```
library("gsmr")
data("gsmr")
head(gsmr_data)
```

```
##
SNP a1 a2
a1_freq
```

```
af_freq
bzx bzx_se
bzx_pval
bzx_n
bzy
## 1 rs109
03129  A
G 0.450019
47 -0.0328
0.0037 3.0
30e-17 169
920.0  0.0
```

```
08038
## 2 rs127
48152  T
C 0.080877
58  0.0499
0.0066 3.2
09e-12 172
987.5  0.0
13671
## 3 rs112
06508  A
G 0.143969
```

```
88  0.0434
0.0055 2.2
56e-14 172
239.0  0.0
30222
## 4 rs112
06510  C
T 0.191289
11 -0.0831
0.0050 2.3
80e-53 172
812.0  0.0
```

```
812.0 -0.0
74519
## 5 rs107
88994  T
C 0.183954
30  0.0687
0.0049 8.8
67e-41 172
941.9  0.0
38267
## 6   rs5
29787  G
```

```
C 0.197130
99 -0.0553
0.0052 8.7
46e-24 161
969.0  0.0
01707
##       bz
y_se      b
zy_pval  b
zy_n
## 1 0.009
2442 0.384
```

```
5651000 18
4305
## 2 0.018
5515 0.461
1690000 18
4305
## 3 0.014
1781 0.033
0400000 18
4305
## 4 0.013
```

```
3438 0.000
0000234 18
4305
## 5 0.011
8752 0.001
2711000 18
4305
## 6 0.013
5491 0.899
7431000 18
4305
```

```
dim(gsmr_data)
```

```
## [1] 188 12
```

This is the input format

for the GSMR analysis. In this data set, there are 188 near-independent SNPs associated with LDL-c at a genome-wide significance level (i.e. $p < 5e-8$).

- SNP: the genetic

instrument

- a1: effect allele
- a2: the other allele
- a1_freq: frequency of a1
- bzx: the

effect size of a1 on risk factor

- bzx_se: standard error of bzx
- bzx_pval: p value for bzx

- bzx_n: per-SNP sample size of GWAS for the risk factor
- bzy: the effect size

of a1 on disease

- bzy_se: standard error of bzy
- bzy_pval: p value for bzy
- bzy_n: per-

SNP sample size of GWAS for the disease

1.2 Estimate the LD correlation matrix

```
# Save the genetic variants and effect alleles in a text file using R
write.table(gsmr_data[,c(1,2)], "gsmr_ex
ample_snps
.allele",
col.names=
F, row.nam
es=F, quot
e=F)
# Extract
the genoty
pe data fr
om a GWAS
dataset us
ing GCTA.
```

```
gcta64 --bfile gsmr_example --extract gsmr_example_snps.allele --update-ref-allele gsmr_example_snps.allele --
```

```
recode --out gsmr_example
```

Note: the two steps above guarantee that the LD correlations are calculated based on the effect alleles for the SNP effects.

```
# Estimate LD correlation matrix using R
```

```
x using R
snp_coeff_
id = scan(
"gsmr_exam
ple.xmat.g
z", what="
", nlines=
1)
snp_coeff
= read.tab
le("gsmr_e
xample.xma
t.gz", header=F, skip=2)
```

```
# Match the SNP genotype data with the summary data
snp_id = R
snp_id = R
educe(inte
rsect, lis
t(gsmr_dat
a$SNP, snp
_coeff_id)
)
gsmr_data
= gsmr_dat
a[match(sn
p_id, gsmr
_data$SNP)
```

```
,]
snp_order
= match(sn
p_id, snp_
coeff_id)
snp_coeff_
id = snp_c
oeff_id[sn
p_order]
snp_coeff
= snp_coef
f[, snp_or
```

```
der]

# Calculate the LD correlation matrix
ldrho = cor(snp_coeff)

# Check the size of
```

```
e size of the correlation matrix and double-check if the order of the SNPs in the LD correlation matrix is consistent wi
```

```
th that in
the GWAS s
ummary dat
a
colnames(l
drho) = ro
wnames(ldr
ho) = snp_
coeff_id
```

```
dim(ldrho)
```

```
## [1] 188
188
```

```
# Show the first 5 rows and columns of the matrix
ldrho[1:5, 1:5]
```

```
##
rs10903129
```

```
rs10903129
rs12748152
rs11206508
rs11206510
rs10788994
## rs10903
129  1.000
000000 -0.
0045378845
0.00806662
1 -0.01372
112 -0.023
```

```
4447102
## rs12748
152 -0.004
537884  1.
0000000000
-0.0066871
81  0.0044
5927  0.00
03629201
## rs11206
508  0.008
06621 -0.
```

```
0066871806
1.00000000
0 -0.21125
757  0.051
2593434
## rs11206
510 -0.013
721120  0.
0044592696
-0.2112575
67  1.0000
0000  0.18
```

```
0000 -0.18
42706205
## rs10788
994 -0.023
444710  0.
0003629201
0.05125934
3 -0.18427
062  1.000
0000000
```

Note: all the analyses implemented in this R-package only require the summary data (e.g. “gsmr_data” and the LD correlation matrix (e.g. “ldrho”) listed above.

2. Standardiza

This is an optional process. If the risk factor was not standardised in GWAS, the effect sizes can be scaled using the method below. Note that this process requires allele frequencies, z-statistics and sample size. After scaling, bzx is interpreted as the per-allele effect of a SNP on the exposure in standard deviation units.

```
snpfreq = gsmr_data$a1_freq # allele frequencies
```

```
of the SNPs
bzx = gsmr_data$bzx # effects of the instruments on risk factor
bzx_se = gsmr_data$bzx_se
```

```
zx_se
# standard errors of bzx
bzx_n = gsmr_data$bzx_n
# GWAS sample size for the risk factor
std_zx = s
```

```
td_effect(
snpfreq, b
zx, bzx_se
, bzx_n)
# perform
standardis
ation
gsmr_data$
std_bzx =
std_zx$b
# standard
ized bzx
```

```
gsmr_data$std_bzx_se = std_zx$se    # standardized bzx_se
head(gsmr_data)
```

```
##
```

```
SNP a1 a2
a1_freq
bzx bzx_se
bzx_pval
bzx_n
bzy
## 1 rs109
03129  A
G 0.450019
47 -0.0328
0.0037 3.0
30e-17 169
```

```
920.0  0.0
08038
## 2 rs127
48152  T
C 0.080877
58  0.0499
0.0066 3.2
09e-12 172
987.5  0.0
13671
## 3 rs112
06508  A
```

```
06508  A
G 0.143969
88  0.0434
0.0055 2.2
56e-14 172
239.0  0.0
30222
## 4 rs112
06510  C
T 0.191289
11 -0.0831
0.0050 2.3
```

```
80e-53 172
812.0 -0.0
74519
## 5 rs107
88994  T
C 0.183954
30  0.0687
0.0049 8.8
67e-41 172
941.9  0.0
38267
## 6   rs5
```

```
29787  G
C 0.197130
99 -0.0553
0.0052 8.7
46e-24 161
969.0  0.0
01707
##      bz
y_se     b
zy_pval  b
zy_n     s
```

```
td_bzx  st
d_bzx_se
## 1 0.009
2442 0.384
5651000 18
4305 -0.03
055942 0.0
03447252
## 2 0.018
5515 0.461
1690000 18
4305  0.04
```

```
713698 0.0
06234550
## 3 0.014
1781 0.033
0400000 18
4305  0.03
829018 0.0
04852442
## 4 0.013
3438 0.000
0000234 18
4305 -0.07
```

```
4305  0.07
181919 0.0
04321251
## 5 0.011
8752 0.001
2711000 18
4305  0.06
149455 0.0
04386074
## 6 0.013
5491 0.899
7431000 18
```

```
4305 -0.04
695042 0.0
0414868
```

3. GSMR analysis

This is the

main analysis of this R-package. It uses SNPs associated with the risk factor (e.g. at $p < 5e-8$) as the instruments to test for putative causal effect of the risk factor on the disease. The analysis involves a step that uses the HEIDI-outlier approach to remove SNPs that have effects on both the risk factor and the disease because of pleiotropy.

```
bzx = gsmr_data$std_
bzx    # SNP effects on the risk factor
bzx_se = gsmr_data$std_bzx_se # standard errors of bzx
bzx_pval = gsmr_data$
```

```
bzx_pval
# p-values
for bzx
bzy = gsmr
_data$bzy
# SNP effe
cts on the
disease
bzy_se = g
smr_data$b
zy_se    #
standard e
standard e
rrors of b
zy
bzy_pval =
gsmr_data$
bzy_pval
# p-values
for bzy
n_ref = 77
03    # Sa
mple size
of the ref
```

```
erence sample
gwas_thresh = 5e-8 # GWAS threshold to select SNPs as the instruments for the GSMR analysis
```

```
single_snp_heidi_thresh = 0.01 # p-value threshold for single-SNP-based HEIDI-outlier analysis
multi_snp_
```

```
heidi_thresh = 0.01 # p-value threshold for multi-SNP-based HEIDI-outlier analysis
nsnps_thresh = 10 # the mini
```

```
mum number of instruments required for the GSMR analysis
heidi_outlier_flag = T    # flag for HEIDI-outlier analysis
```

```
analysis
ld_r2_thresh = 0.05 # LD r2 threshold to remove SNPs in high LD
ld_fdr_thresh = 0.05 # FDR threshold to r
```

```
emove the chance correlations between the SNP instruments
gsmr2_beta = 0     # 0 - the original HEIDI-outlier method; 1
```

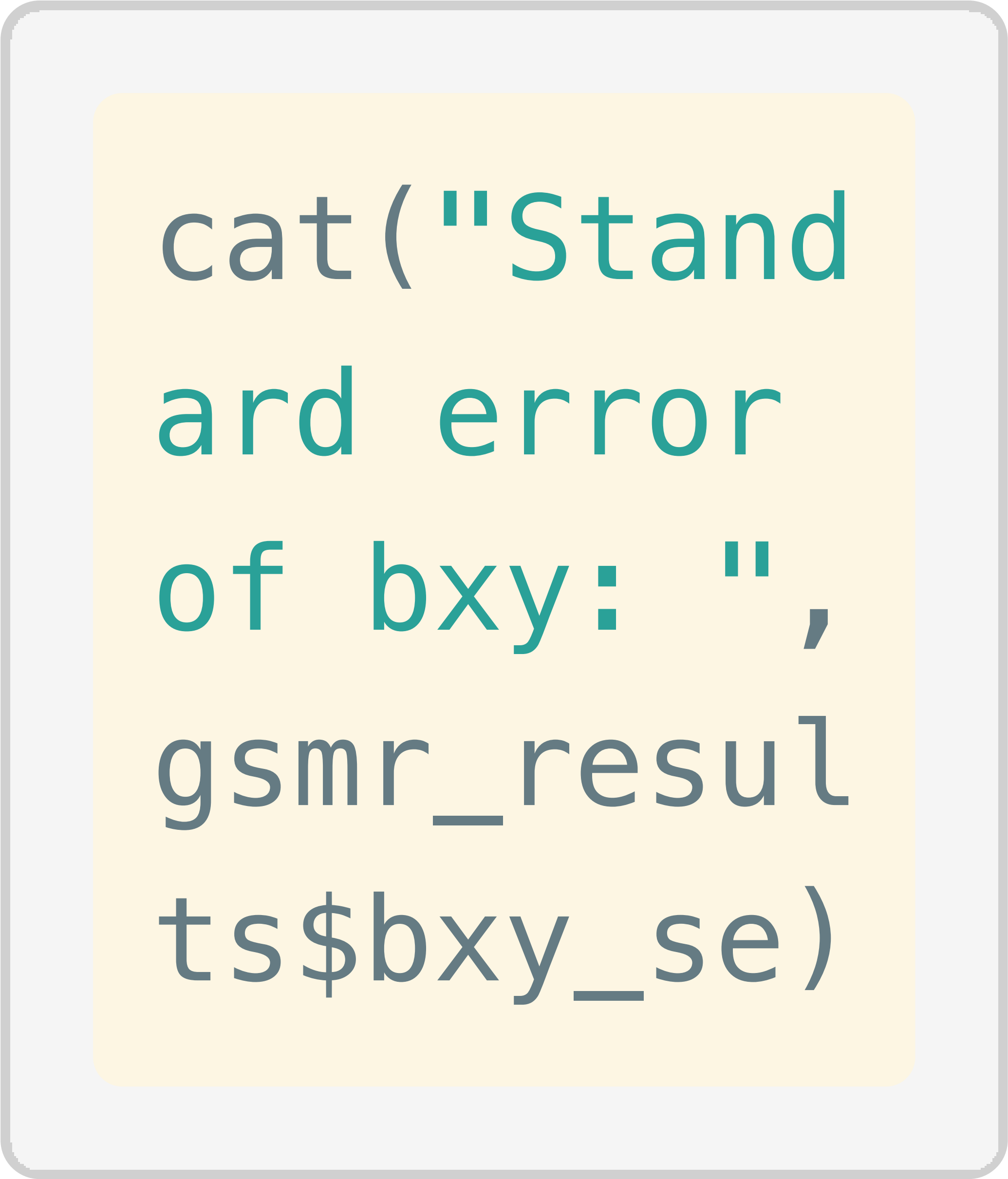
```
- the new
HEIDI-outl
ier method
that is cu
rrently un
der develo
pment
gsmr_resul
ts = gsmr(
bzx, bzx_s
e, bzx_pva
l, bzy, bz
```

```
t, bzy, bz
y_se, bzy_
pval, ldrh
o, snp_coe
ff_id, n_r
ef, heidi_
outlier_fl
ag, gwas_t
hresh, sin
gle_snp_he
idi_thresh
, multi_sn
```

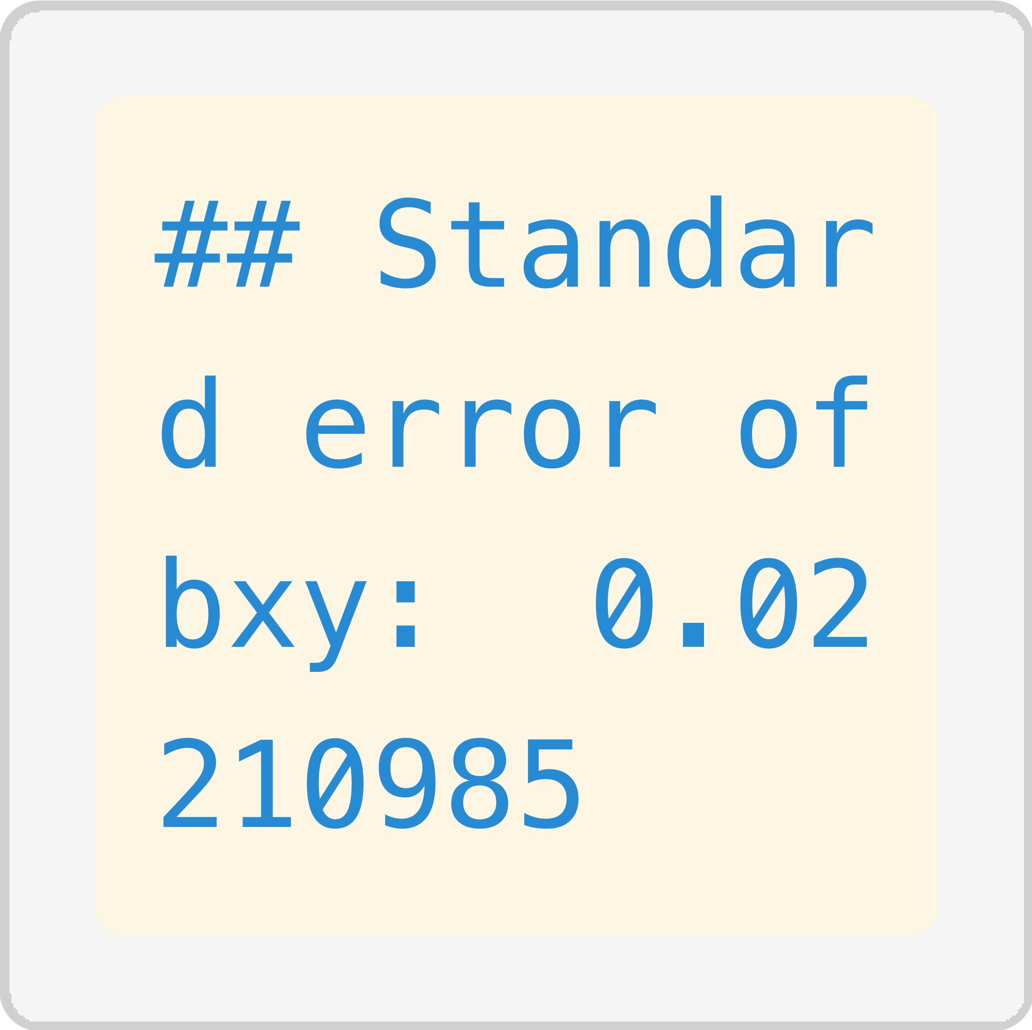
```
p_heidi_th
resh, nsnp
s_thresh,
ld_r2_thre
sh, ld_fdr
_thresh, g
smr2_beta)
# GSMR ana
lysis
filtered_i
ndex=gsmr_
results$us
```

```
ed_index
cat("The estimated effect of the exposure on outcome: ",gsmr_results$bxy)
```

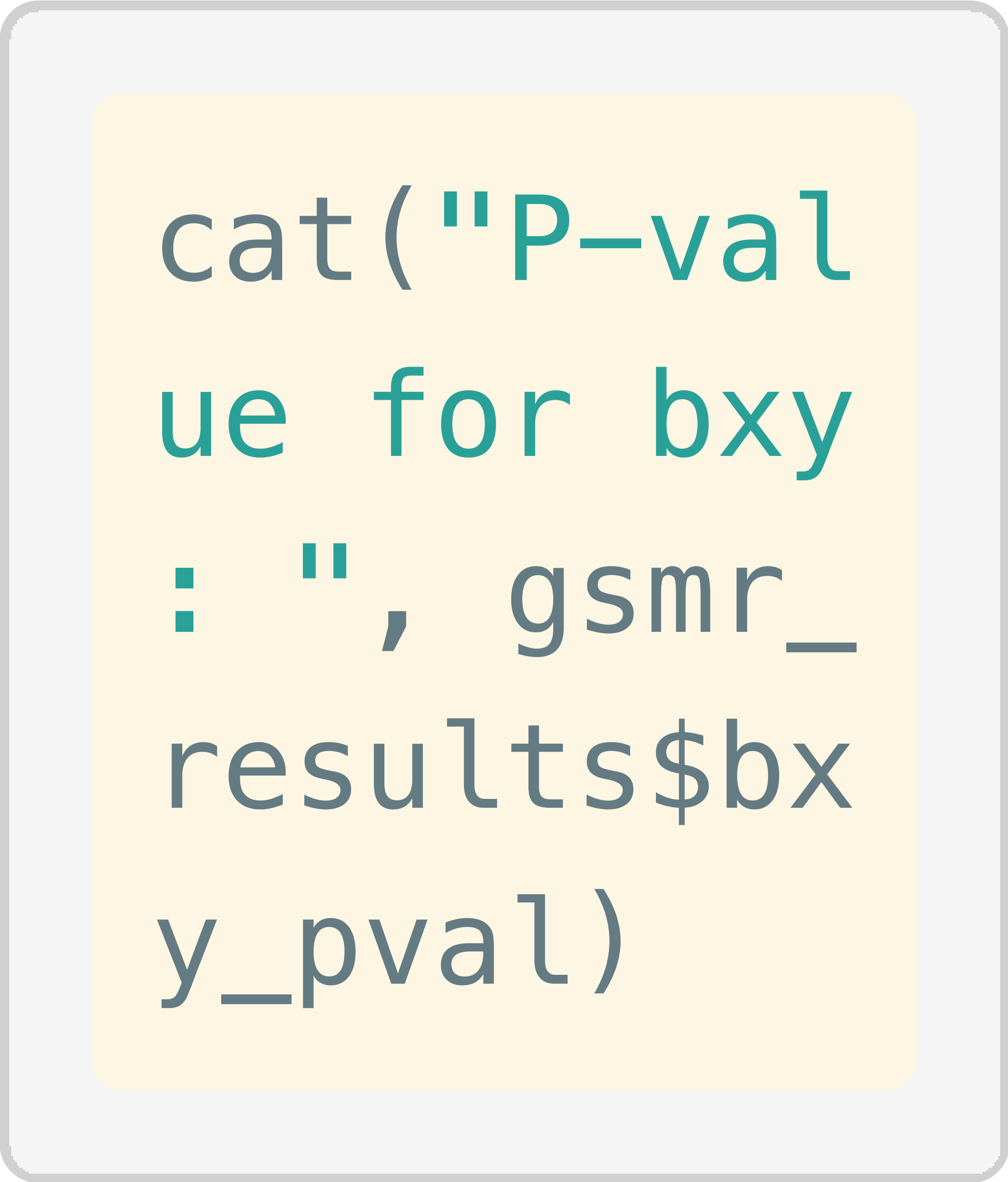
```
## The est
imated eff
ect of the
exposure o
n outcome:
0.4322395
```



```r
cat("Standard error of bxy: ", gsmr_results$bxy_se)
```



```
## Standar
d error of
bxy:  0.02
210985
```



```
cat("P-val
ue for bxy
: ", gsmr_
results$bx
y_pval)
```

```
## P-value
for bxy:
4.15454e-8
5
```

```r
cat("Indexes of the SNPs used in the GSMR analysis: ", gsmr_results$used_index[1:5], "...")
```

```
## Indexes
of the SNP
s used in
the GSMR a
nalysis:
1 2 3 5 6

...
```

```r
cat("Number of SNPs with missing estimates in the summary data: ", length(gsmr_results$na_snps))
```

```
## Number of SNPs with missing estimates in the summary data: 0
```

```r
cat("Number of non-significant SNPs: ", length(gsmr_results$weak_snps))
```

```
## Number of non-significant SNPs:  39
```

```
cat("Numbe
r of SNPs
in high LD
( LD rsq >
", ld_r2_t
hresh, "):
", length(
gsmr_resul
ts$linkage
_snps))
```

```
## Number
of SNPs in
high LD (
LD rsq > 0
.05 ):  5
```

```r
cat("Number of pleiotropic outliers: ", length(gsmr_results$pleio_snps))
```

```
## Number of pleiotropic outliers:  9
```

4. Bi-directional GSMR

analysis

The script below runs bi-directional GSMR analyses, i.e. a forward-GSMR analysis as described above and a reverse-GSMR analysis that uses SNPs associated with the disease (e.g. at $p < 5e-8$) as the instruments to test for putative causal effect of the disease on risk factor.

```
gsmr_results = bi_gsmr(bzx, bzx_se, bzx_pval, bzy, bzy_se, bzy_pval, ld_rho, snp_c
rho, snp_c
oeff_id, n
_ref, heid
i_outlier_
flag, gwas
_thresh, s
ingle_snp_
heidi_thre
sh, multi_
snp_heidi_
thresh, ns
nps_thresh
```

```
, ld_r2_thresh, ld_fdr_thresh, gsmr2_beta)    # GSMR analysis

cat("Effect of risk factor on disease: ",gsmr_resu
lts$forwar
d_bxy)
```

```
## Effect
of risk fa
ctor on di
sease:  0.
4322395
```

```
cat("Standard error of bxy in the forward-GSMR analysis: ",gsmr_results$forward_bxy_se)
```

```
## Standard error of bxy in the forward-GSMR analysis:  0.02210985
```

```
cat("P-value of bxy in the forward-GSMR analysis: ", gsmr_results$forward_bxy_pval)
```

```
## P-value of bxy in the forward-GSMR analysis:  4.15454e-85
```

```
cat("Effect of disease on risk factor: ", gsmr_results$reverse_bxy)
```

```
## Effect of disease on risk factor: -0.02739421
```

```
cat("Standard error of bxy in the reverse-GSMR analysis: ",gsmr_results$reverse_bxy_se)
```

```
## Standar
d error of
bxy in the
 reverse-GS
MR analysi
s:  0.0095
51025
```

```
cat("P-value of bxy in the reverse-GSMR analysis: ", gsmr_results$reverse_bxy_pval)
```

```
## P-value
of bxy in
the revers
e-GSMR ana
lysis:  0.
004128198
```

5. Visualizatio

```
effect_col
= colors()
[75]
vals = c(b
zx[filtere
d_index]-b
zx_se[filt
ered_index
], bzx[fil
tered_inde
x]+bzx_se[
filtered_i
ndex])
xmin = min
(vals); xm
ax = max(v
als)
vals = c(b
zy[filtere
d_index]-b
zy_se[filt
ered_index
]-bzy[fil
```

```
], bzy[fil
tered_inde
x]+bzy_se[
filtered_i
ndex])
ymin = min
(vals); ym
ax = max(v
als)
par(mar=c(
5,5,4,2))
plot(bzx[f
iltered_in
dex], bzy[
filtered_i
ndex], pch
=20, cex=0
.8, bty="n
", cex.axi
s=1.1, cex
.lab=1.2,
        co
l=effect_c
ol, xlim=c
```

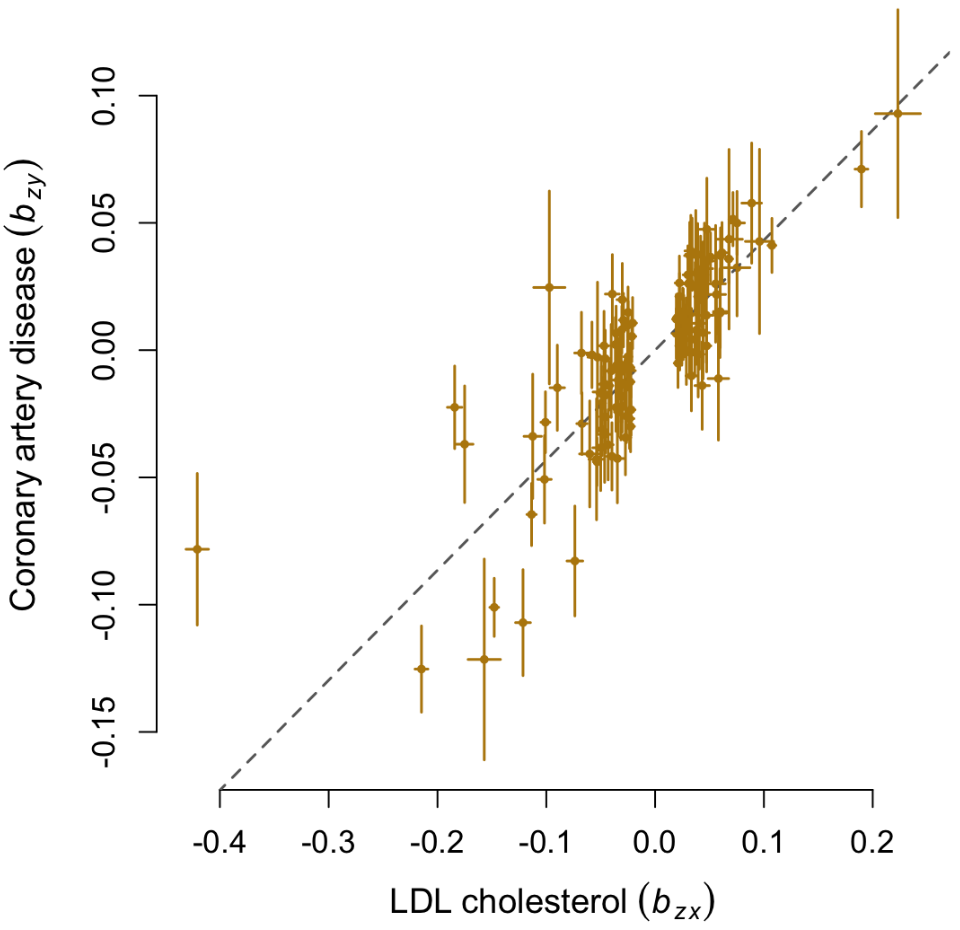
```
et), xlim=c
(xmin, xma
x), ylim=c
(ymin, yma
x),
         xl
ab=express
ion(LDL~ch
olesterol~
(italic(b[
zx]))),
         yl
```

```
ab=expression(Coronary~artery~disease~(italic(b[zy]))))
abline(0, gsmr_results$forward_bxy, lwd=1.5, lty=2, col="dim
grey")

nsnps = length(bzx[filtered_index])
for( i in 1:nsnps )
{
    # x axis
    xstart
    xstart
= bzx[filt
ered_index
[i]] - bzx
_se[filter
ed_index[i
]]; xend =
bzx[filter
ed_index[i
]] + bzx_s
e[filtered
_index[i]]
```

```
    ystart = bzy[filtered_index[i]]; yend = bzy[filtered_index[i]]
    segments(xstart, ystart, xend, yend, lwd=1.5, c
ol=effect_
col)
    # y ax
is
    xstart
= bzx[filt
ered_index
[i]]; xend
= bzx[filt
ered_index
[i]]
    ystart
    ystart
= bzy[filt
ered_index
 [i]] - bzy
_se[filter
ed_index[i
]]; yend =
bzy[filter
ed_index[i
]] + bzy_s
e[filtered
_index[i]]
```

```
    segments(xstart, ystart, xend, yend, lwd=1.5, col=effect_col)
}
```



Package Document

bi_gsmr

> Bi-directional GSMR analysis is composed

> of a forward-GSMR analysis and a reverse-GSMR analysis that uses

> SNPs associated with the disease (e.g. at < 5e-8) as the instruments to test for

> putative causal effect of the disease on the risk factor.

Usage

```
bi_gsmr(bzx
, bzx_se, b
zx_pval, bz
y, bzy_se,
bzy_pval, l
drho, snpid
, heidi_out
lier_flag=T
, gwas_thre
sh=5e-8, si
ngle_snp_he
idi_thresh=0.01, multi_snp_heidi_thresh=0.01, nsnps_thresh=10, ld_r2_thresh=0.05, ld_fdr_thresh=0.05, gsmr2_beta=0)
```

Arguments

```
bzx
```

`bzx_se`

`bzx_pval`

`bzy`

`bzy_se`

`bzy_pval`

`ldrho`

`snpid`

```
n_ref
```

`heidi_outlier_fla`

`gwas_thresh`

`single_snp_heidi_`

`multi_snp_heidi_`

`nsnps_thresh`

`ld_r2_thresh`

```
ld_fdr_thresh
```

```
gsmr2_beta
```

Estimate of

Value

Estimate of

causative effect of risk factor on disease (forward_bxy), the corresponding standard error (forward_bxy_se),

p-value

(forward_bxy_pva

and SNP

index

(forward_index),

and estimate

of causative

effect of disease on risk factor (reverse_bxy), the corresponding standard error (reverse_bxy_se), p-value (reverse_bxy_pva SNP index (reverse_index), SNPs with missing values, with non-significant p-values and those in LD.

Examples

```
data("gsmr")
gsmr_result = bi_gsmr(gsmr_dat
r(gsmr_dat
a$bzx, gsm
r_data$bzx
_se, gsmr_
data$bzx_p
val, gsmr_
data$bzy,
gsmr_data$
bzy_se, gs
mr_data$bz
y_pval, ld
rho, gsmr_
data$SNP,
n_ref, T,
5e-8, 0.01
, 0.01, 10
, 0.05, 0.
05, 0)
```

gsmr

> GSMR (Generalised Summary-data-based Mendelian Randomisation is a flexible and powerful

> approach that utilises multiple genetic instruments to test for causal association between a

> risk factor and disease using summary-level data from independent genome-

> wide association studies.

Usage

```
gsmr(bzx, b
zx_se, bzx_
pval, bzy,
bzy_se, ldr
```

```
ho, snpid,
heidi_outli
er_flag=T,
gwas_thresh
=5e-8, sing
le_heidi_th
resh=0.01,
multi_heidi
_thresh=0.0
1, nsnps_th
resh=10, ld
```

```
_r2_thresh=
0.05, ld_fd
r_thresh=0.
05, gsmr2_b
eta=0)
```

Arguments

`bzx`

`bzx_se`

`bzx_pval`

`bzy`

`bzy_se`

`ldrho`

`snpid`

`n_ref`

```
heidi_outlier_fla
```

`gwas_thresh`

`nsnps_thresh`

`ld_r2_thresh`

`ld_fdr_thresh`

`gsmr2_beta`

```
single_heidi_thr
```

```
multi_heidi_thres
```

Value

Estimate of causative effect of risk factor on disease (bxy), the corresponding standard error (bxy_se), p-value (bxy_pval), SNP index (used_index), SNPs with missing values, with non-significant p-values and those in LD.

Examples

```
data("gsmr")
gsmr_resul
```

```
t = gsmr(g
smr_data$b
zx, gsmr_d
ata$bzx_se
, gsmr_dat
a$bzx_pval
, gsmr_dat
a$bzy, gsm
r_data$bzy
_se, ldrho
, gsmr_dat
a$SNP, n_r
d$SNP, n_r
ef, T, 5e-
8, 0.01, 0
.01, 10, 0
.1, 0.05,
0)
```

std_effect

> Standardizatio of SNP effect and its standard error using z-statistic, allele frequency

> and sample size

Usage

```
std_effect(snp_freq, b, se, n)
```

Arguments

`snp_freq` vector, allele frequen

b

vector,
SNP
effects
risk fac

`se`	vector, standar errors c

`n`	vector, SNP sample sizes fo GWAS the risk factor

Value

Standardised effect (b) and standard error (se)

Examples

```
data("gsmr
")
std_effects = std_effect(gsmr_data$a1_freq, gsmr_data$bzx, gsmr_data$bzx_se, gsmr_data$bzx_n)
```