Package: SlopeHunter (via r-universe)

September 27, 2024

Type Package

Title Slope-Hunter: A ROBUST METHOD FOR COLLIDER BIAS CORRECTION IN CONDITIONAL GENOME-WIDE ASSOCIATION STUDIES

Version 1.1.0

Date 2022-09-27

Author@R person(``Osama'', ``Mahmoud'', email = ``o.mahmoud@essex.ac.uk'', role = c(``aut'', ``cre'', ``cph''), comment = c(ORCID = ``0000-0003-0342-6704''))

Description Studying genetic associations with prognosis (e.g. survival, disability, subsequent disease events) is problematic due to selection bias - also termed index event bias or collider bias - whereby selection on disease status can induce associations between causes of incidence with prognosis. The Slope-Hunter approach adjusts genetic associations for this bias assuming that the contribution of the set of genetic variants affecting incidence only to the heritability of incidence is at least as large as the contribution of those affecting both incidence and prognosis.

License GPL-2 | GPL-3

URL http://osmahmoud.com/SlopeHunter/

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0)

Imports ggplot2 (>= 2.1.0), mclust, plotly, stats, ieugwasr, data.table (>= 1.14.2), dplyr, tools

Suggests knitr (>= 1.12), rmarkdown (>= 0.9)

VignetteBuilder knitr

RoxygenNote 7.1.2

Author Osama Mahmoud [aut, cre, cph](<https://orcid.org/0000-0003-0342-6704>)

Maintainer Osama Mahmoud <o.mahmoud@essex.ac.uk>

Remotes mrcieu/ieugwasr

Roxygen list(markdown = TRUE)

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

RemoteUrl https://github.com/Osmahmoud/SlopeHunter

RemoteRef HEAD

RemoteSha b0686fe0c7a4e6d056d902765ba9cd0a0b35ad5c

Contents

data_example *Simulated effects on quantitative incidence and prognosis traits*

Description

A simulated dataset for 10,000 independent variables (e.g. SNPs) consisting of regression coefficients on incidence and prognosis, with their standard errors. Among all the SNPs, 5% (500 variables) have effects on incidence only, 5% (500 variables) on prognosis only, and 5% have correlated effects on both with a correlation coeficient of '-0.5'. The estimates are obtained from linear regression in a simulated dataset of 20,000 individuals.

Usage

```
data_example
```

Format

A data frame with 10,000 rows and 5 variables:

xbeta Regression coefficient on incidence

xse Standard error of `xbeta`

ybeta Regression coefficient on prognosis

yse Standard error of `ybeta`

yp P-value of the association with prognosis

Examples

```
# Load the \code{SlopeHunter} package
require(SlopeHunter)

# Load the input data set
data(data_example, package = ”SlopeHunter”)
head(data_example)

# Implement the Slope-Hunter method
Sh.Model <- hunt(dat = data_example, xbeta_col=”xbeta”, xse_col=”xse”,
                        ybeta_col=”ybeta”, yse_col=”yse”, yp_col=”yp”,
                xp_thresh = 0.001, Bootstrapping = TRUE, show_adjustments = TRUE, seed=2021)

# [1] ”Estimated slope: -0.274120383700514”
# [1] ”SE of the slope: 0.0229566376478153”
# [1] ”95% CI: -0.319115393490232, -0.229125373910796”

# Display the estimated slope (adjustment factor)
Sh.Model$b
# [1] -0.2741204

# Extract information about cluster memberships of SNPs included in the analysis
Adj <- Sh.Model$Fit

# Show the first 6 values of the unadjusted estimated effects on prognosis
head(data_example$ybeta)
# [1] -0.0092889266  0.0005575032  0.0112203795 -0.0095533069  0.0082635203  0.0026550045


# Show results of the first 6 corrected variants:
head(Sh.Model$est)

#   xbeta  xse   ybeta  yse   yp    xp    SNP  ybeta_adj   yse_adj yp_adj
# 1 -0.007 0.007 -0.009 0.006 0.136 0.300 snp1 -0.011      0.006   0.083
# 2  0.014 0.007  0.000 0.006 0.928 0.042 snp2  0.004      0.006   0.492
# 3 -0.011 0.007  0.011 0.006 0.072 0.097 snp3  0.008      0.006   0.220
# 4  0.004 0.007 -0.009 0.006 0.125 0.493 snp4 -0.008      0.006   0.208
# 5 -0.025 0.007  0.008 0.006 0.185 0.000 snp5  0.001      0.006   0.851
# 6  0.013 0.007  0.002 0.006 0.670 0.054 snp6  0.006      0.006   0.329

# Generate an interactive plot for the estimated clusters (hover on the data points to view info)
require(ggplot2)
require(plotly)
ggplotly(Sh.Model$plot)
```

download_plink *Check and download PLINK 1.90 executable suitable for the operating system, and return its path Inspired by https://github.com/MRCIEU/genetics.binaRies*

Description

Check and download PLINK 1.90 executable suitable for the operating system, and return its path Inspired by https://github.com/MRCIEU/genetics.binaRies

Usage

```
download_plink()
```

format_data *Format input data*

Description

Reads in and format input data. It checks and organises columns for Slope-Hunter analyses. Infers p-values when possible from beta and se.

Usage

```
format_data(
  dat,
  type = "incidence",
  snps = NULL,
  snp_col = "SNP",
  beta_col = "BETA",
  se_col = "SE",
  pval_col = "PVAL",
  eaf_col = "EAF",
  effect_allele_col = "EA",
  other_allele_col = "OA",
  gene_col = "GENE",
  chr_col = "CHR",
  pos_col = "POS",
  min_pval = 1e-200,
  log_pval = FALSE
)
```

Arguments

`dat` Data frame. Must have header with at least the `SNP`, `beta`, `se` and `EA` columns present.

`type` Is this the incidence or the prognosis data that is being read in? The default is `"incidence"`.

`snps` SNPs to extract. If NULL, then it keeps all. The default is `NULL`.

`snp_col` Required name of column with SNP rs IDs. The default is `"SNP"`.

`beta_col` Required name of column with effect sizes. The default is `"BETA"`.

`se_col` Required name of column with standard errors. The default is `"SE"`.

`pval_col` Name of column with p-value (optional). The default is `"PVAL"`. It will be Inferred when possible from beta and se.

`eaf_col` Name of column with effect allele frequency (optional). The default is `"EAF"`.

`effect_allele_col`
Required for harmonisation. Name of column with effect allele. Must be "A", "C", "T" or "G". The default is `"EA"`.

`other_allele_col`
Required for harmonisation. Name of column with non-effect allele. Must be "A", "C", "T" or "G". The default is `"OA"`.

`gene_col` Optional column for gene name. The default is `"GENE"`.

`chr_col` Optional column for chromosome number. The default is `"CHR"`.

`pos_col` Optional column for SNP position. The default is `"POS"`.

`min_pval` Minimum allowed p-value. The default is `1e-200`.

`log_pval` The p-value is -log10(P). The default is `FALSE`.

Value

data frame

`harmonise_effects` *Harmonise and format data for Slope-Hunter*

Description

Harmonise the alleles and effects between the incidence and prognosis (inspired by https://github.com/MRCIEU/TwoSampleM

Usage

```
harmonise_effects(
  incidence_dat,
  prognosis_dat,
  incidence_formatted = TRUE,
  prognosis_formatted = TRUE,
  by.pos = FALSE,
  pos_cols = c("POS.incidence", "POS.prognosis"),
  snp_cols = c("SNP", "SNP"),
  beta_cols = c("BETA.incidence", "BETA.prognosis"),
  se_cols = c("SE.incidence", "SE.prognosis"),
  EA_cols = c("EA.incidence", "EA.prognosis"),
  OA_cols = c("OA.incidence", "OA.prognosis"),
  chr_cols = c("CHR.incidence", "CHR.prognosis"),
  gene_col = c("GENE.incidence", "GENE.prognosis")
)
```

Arguments

`incidence_dat` data.table for incidence data. It is recommended to be an output from `read_incidence`. If not, it tries to format it before harmonisation.

`prognosis_dat` data.table for prognosis data. It is recommended to be an output from `read_prognosis`. If not, it tries to format it before harmonisation.

`incidence_formatted` Logical indicationg whether `incidence_dat` is formatted using `read_incidence`.

`prognosis_formatted` Logical indicationg whether `prognosis_dat` is formatted using `read_prognosis`.

`by.pos` Logical, if `TRUE` the harmonisation will be performed by matching the exact SNP positions between the incidence and prognosis datasets.

`pos_cols` A vector of length 2 specifying the name of the genetic position columns in the incidence and prognosis datasets respectively.

`snp_cols` A vector of length 2 specifying the name of the snp columns in the incidence and prognosis datasets respectively. This is the column on which the data will be merged if `by.pos` is `FASLE`.

`beta_cols` A vector of length 2 specifying the name of the beta columns in the incidence and prognosis datasets respectively.

`se_cols` A vector of length 2 specifying the name of the se columns in the incidence and prognosis datasets respectively.

`EA_cols` A vector of length 2 specifying the name of the effect allele columns in the incidence and prognosis datasets respectively.

`OA_cols` A vector of length 2 specifying the name of the non-effect allele columns in the incidence and prognosis datasets respectively.

`chr_cols` A vector of length 2 specifying the name of the chromosome columns in the incidence and prognosis datasets respectively.

`gene_col` A vector of length 2 specifying the name of the gene columns in the incidence and prognosis datasets respectively.

Details

In order to perform Slope-Hunter analysis the effect of a SNP on an incidence and prognosis traits must be harmonised to be relative to the same allele.

This function will try to harmonise the incidence and prognosis data sets on the specified columns. Where necessary, correct strand for non-palindromic SNPs (i.e. flip the sign of effects so that the effect allele is the same in both datasets), and drop all palindromic SNPs from the analysis (i.e. with the allele A/T or G/C). The alleles that do not match between data sets (e.g T/C in one data set and A/C in the other) will also be dropped.

Value

A data.frame with harmonised effects and alleles

`hunt` *Estimate collider bias*

Description

Estimate collider bias

Usage

```
hunt(
  dat,
  snp_col = "SNP",
  xbeta_col = "BETA.incidence",
  xse_col = "SE.incidence",
  xp_col = "Pval.incidence",
  ybeta_col = "BETA.prognosis",
  yse_col = "SE.prognosis",
  yp_col = "Pval.prognosis",
  xp_thresh = 0.001,
  init_pi = 0.6,
  init_sigmaIP = 1e-05,
  Bootstrapping = TRUE,
  M = 100,
  seed = 777,
  Plot = TRUE,
  show_adjustments = FALSE
)
```

Arguments

`dat` Data frame. Must have header with at least the `xbeta`, `xse`, `ybeta` and `yse` columns present.

`snp_col` Name of column with SNP IDs.

`xbeta_col` Required name of column with effects on the incidence trait.

`xse_col` Required name of column with standard errors of `xbeta`.

`xp_col` Name of column with p-value for `xbeta` (optional). If not given, It will be inferred from `xbeta` and `xse`.

`ybeta_col` Required name of column with unadjusted effects on the prognosis trait.

`yse_col` Required name of column with standard errors of `ybeta`.

`yp_col` Name of column with p-value for `ybeta` (optional). If not given, It will be inferred from `ybeta` and `yse`.

`xp_thresh` p-value threshold for SNP-incidence associations. Effects with p-values larger than `xp.thresh` will be excluded prior to fitting the main model-based clustering.

`init_pi` initial value for the weight of the mixture component that represents the cluster of SNPs affecting x only.

`init_sigmaIP` initial value for the covariance between x and y.

`Bootstrapping` Logical, if TRUE estimate the standard error of the adjustment factor using the Bootstrap method.

`M` Number of bootstrap samples drawn to estimate the standard error of the adjustment factor.

`seed` Random number seed used for drawing the bootstrap samples.

`Plot` Logical, if TRUE (the default), calling the function should plot the final clusters.

`show_adjustments`
Logical indicating whether to show adjusted effects of the given SNPs in the outputs.

Value

List of the following:

- est: estimated adjusted associations, their standard errors and p-values (only if `show_adjustments` is TRUE).
- b: The estimated slope (adjustment factor).
- bse: Standard error of the estimated slope.
- b_CI: 95\
- pi: Estimated probability of the mixture component of SNPs affecting only incidence.
- entropy: The entropy of the estimated clusters.
- plot: Generated plot of the SlopeHunter fitted model.
- Fit: a Data frame summarising the fitted model-based clustering with the following columns:
    - cluster: cluster of the variants defined as follows:
        - `Hunted` = assigned to the cluster of SNPs affecting only incidence.
        - `Pleiotropic` = assigned to the cluster affecting both incidence and prognosis - i.e. variants that affect incidence and have direct effect on prognosis.
    - pt and p0: membership probabilities of the variants for the hunted and pleiotropic clusters respectively.
    - associations of variants with x and y, their standard errors and p-values.
- iter: Number of the EM algorithm's iterations.
- Bts.est: Details on the bootstrap estimate of the standard error of the adjustment factor, if `Bootstrapping` is TRUE.

`ld_local` *clump function using local plink binary and ld reference dataset This function is modified from: https://github.com/MRCIEU/ieugwasr/blob/master/R/ld_clump.R*

Description

clump function using local plink binary and ld reference dataset This function is modified from: https://github.com/MRCIEU/ieugwasr/blob/master/R/ld_clump.R

Usage

```
ld_local(dat, clump_kb = 250, clump_r2 = 0.1, clump_p1 = 1, bfile)
```

Arguments

`dat`	Dataframe. Must have a variant name column (`rsid`) and pval column called (`pval`).
`clump_kb`	Clumping window, default is `250`.
`clump_r2`	Clumping r-squared threshold, default is `0.1`.
`clump_p1`	Clumping sig level for index SNPs, default is `1`.
`bfile`	Path to the bed/bim/fam LD reference (e.g. "1kg.v3/EUR" for local 1000 EUR ref. population file).

`LD_prune` *Perform LD pruning on SNP data*

Description

Uses PLINK clumping method ('–clump' command), where a greedy search algorithm is implemented to randomly select a variant (or the variant with the lowest p-value, if a user wish to), referred to as the index SNP, and remove all variants within a certain kb distance in linkage disequilibrium with the index SNP, based on an r-squared threshold from the 1000 Genomes reference panel phase 3 data. Then repeats until no variants are left.

Usage

```
LD_prune(
  dat,
  clump_kb = 250,
  clump_r2 = 0.1,
  Random = TRUE,
  clump_p1 = 1,
  local = FALSE,
  ref_pop = "EUR",
  ref_bfile,
  seed = 77777
)
```

Arguments

dat	Output from `harmonise_effects`. Must have a SNP name column (SNP).
clump_kb	Clumping window, default is `250`.
clump_r2	Clumping r-squared threshold, default is `0.1`.
Random	Logical, if `TRUE` (the default), SNPs will be randomly pruned. Otherwise, based on p-values.
clump_p1	Clumping sig level for index SNPs, default is `1`.
local	Logical, if `FALSE` (the default), the MRC-IEU API 'http://gwas-api.mrcieu.ac.uk/' will be used for clumping. Otherwise, your local machine will be used for clumping given that you provide a bed/bim/fam LD reference dataset.
ref_pop	Super-population to use as reference panel at the API (when `local` is FALSE). Default = `"EUR"`.
ref_bfile	Path to the bed/bim/fam LD reference (e.g. "1kg.v3/EUR" for local 1000 EUR ref. population file). If `local`=TRUE, then this should be provided.
seed	Random number seed for random pruning

Value

Data frame

---

plot.SH *Plotting model for Slope-Hunter clustering*

Description

Plotting model for Slope-Hunter clustering

Usage

```
## S3 method for class 'SH'
plot(
  x,
  what = c("clusters", "classification", "uncertainty", "density"),
  xlab = NULL,
  ylab = NULL,
  addEllipses = TRUE,
  main = FALSE,
  ...
)
```

Arguments

`x` Output from `slopehunter`.

`what` A string specifying the type of graph requested. Available choices are: "clusters": showing clusters. The plot can display membership probabilities of each variable (e.g. SNP) to the target cluster (G1) by hovering over the points. "classification": A plot showing point assigned to each cluster (class). "uncertainty": A plot of classification uncertainty. "density": A plot of estimated density.

`xlab` Optional label for the x-axis in case of "classification", "uncertainty", or "density" plots.

`ylab` Optional label for the y-axis in case of "classification", "uncertainty", or "density" plots.

`addEllipses` A logical indicating whether or not to add ellipses with axes corresponding to the within-cluster covariances in case of "classification" or "uncertainty" plots.

`main` A logical or NULL indicating whether or not to add a title to the plot identifying the type of plot drawn in case of "classification", "uncertainty", or "density" plots.

`...` Other graphics parameters.

`read_incidence` *Read incidence data*

Description

Reads in incidence data. Checks and organises columns for use with the Slope-Hunter analyses. Infers p-values when possible from beta and se.

Usage

```
read_incidence(
  filename,
  snp_col = "SNP",
  beta_col = "BETA",
  se_col = "SE",
  pval_col = "PVAL",
  eaf_col = "EAF",
  effect_allele_col = "EA",
  other_allele_col = "OA",
  gene_col = "GENE",
  chr_col = "CHR",
  pos_col = "POS",
  min_pval = 1e-200,
  log_pval = FALSE
)
```

Arguments

`filename` Filename (formatted as .gz, .csv or .txt). Must have header with at least the `SNP`, `beta`, `se` and `EA`columns present.

`snp_col` Required name of column with SNP rs IDs. The default is `"SNP"`.

`beta_col` Required name of column with effect sizes. The default is `"BETA"`.

`se_col` Required name of column with standard errors. The default is `"SE"`.

`pval_col` Name of column with p-value (optional). The default is `"PVAL"`. It will be Inferred when possible from beta and se.

`eaf_col` Name of column with effect allele frequency (optional). The default is `"EAF"`.

`effect_allele_col`
Required for harmonisation. Name of column with effect allele. Must be "A", "C", "T" or "G". The default is `"EA"`.

`other_allele_col`
Required for harmonisation. Name of column with non-effect allele. Must be "A", "C", "T" or "G". The default is `"OA"`.

`gene_col` Optional column for gene name. The default is `"GENE"`.

`chr_col` Optional column for chromosome number. The default is `"CHR"`.

`pos_col` Optional column for SNP position. The default is `"POS"`.

`min_pval` Minimum allowed p-value. The default is `1e-200`.

`log_pval` The p-value is -log10(P). The default is `FALSE`.

Value

data frame

---

read_prognosis *Read prognosis data*

Description

Reads in prognosis data. Checks and organises columns for use with the Slope-Hunter analyses. Infers p-values when possible from beta and se.

Usage

```
read_prognosis(
  filename,
  snp_col = "SNP",
  beta_col = "BETA",
  se_col = "SE",
  pval_col = "PVAL",
  eaf_col = "EAF",
  effect_allele_col = "EA",
  other_allele_col = "OA",
  gene_col = "GENE",
  chr_col = "CHR",
  pos_col = "POS",
  min_pval = 1e-200,
  log_pval = FALSE
)
```

Arguments

`filename` Filename. Must have header with at least the `SNP`, `beta`, `se` and `EA`columns present.

`snp_col` Required name of column with SNP rs IDs. The default is `"SNP"`.

`beta_col` Required name of column with effect sizes. The default is `"BETA"`.

`se_col` Required name of column with standard errors. The default is `"SE"`.

`pval_col` Name of column with p-value (optional). The default is `"PVAL"`. It will be Inferred when possible from beta and se.

`eaf_col` Name of column with effect allele frequency (optional). The default is `"EAF"`.

`effect_allele_col` Required for harmonisation. Name of column with effect allele. Must be "A", "C", "T" or "G". The default is `"EA"`.

`other_allele_col` Required for harmonisation. Name of column with non-effect allele. Must be "A", "C", "T" or "G". The default is `"OA"`.

`gene_col` Optional column for gene name. The default is `"GENE"`.

`chr_col` Optional column for chromosome number. The default is `"CHR"`.

`pos_col` Optional column for SNP position. The default is `"POS"`.

`min_pval` Minimum allowed p-value. The default is `1e-200`.

`log_pval` The p-value is -log10(P). The default is `FALSE`.

Value

data frame

`SHadj` *Correct index event bias for new data*

Description

Correct index event bias for new data

Usage

```
SHadj(
  x,
  dat,
  snp_col = "SNP",
  xbeta_col = "BETA.incidence",
  xse_col = "SE.incidence",
  ybeta_col = "BETA.prognosis",
  yse_col = "SE.prognosis"
)
```

Arguments

`x` an pbject of the class SH obtained from the `slopehunter` function.

`dat` A data.frame with harmonised effects and alleles, formatted using the `harmonise_effects` function.

`snp_col` Name of column with SNP IDs.

`xbeta_col` Required name of column with effects on the incidence trait.

`xse_col` Required name of column with standard errors of `xbeta`.

`ybeta_col` Required name of column with unadjusted effects on the prognosis trait.

`yse_col` Required name of column with standard errors of `ybeta`.

Value

data.frame with adjusted estimates

`shclust` *Implement the EM algorithm for the SlopeHunter model-based clustering*

Description

Implement the EM algorithm for the SlopeHunter model-based clustering

Usage

```
shclust(gwas, pi0, sxy1)
```

Arguments

`gwas` a data frame with columns: xbeta; xse; ybeta; yse.

`pi0` initial value for the weight of the mixture component that represents the cluster of SNPs affecting x only.

`sxy1` initial value for the covariance between x and y.

Value

EM fit for SlopeHunter estimator

Index