Package: mrlocus (via r-universe)

January 24, 2025

Title MRLocus - Mendelian Randomization per locus

Version 0.0.26

Maintainer Michael Love <michaelisaiahlove@gmail.com>

Description Mendelian Randomization per locus, leveraging eQTL and GWAS summary statistics, for estimation of gene-to-trait effect size and dispersion.

URL https://github.com/thelovelab/mrlocus

License GPL (>=2)

Encoding UTF-8

LazyData true

Biarch true

Depends R (>= 3.4.0)

Imports methods, Rcpp (>= 0.12.0), rstan (>= 2.18.1), rstantools (>= 2.0.0), matrixStats, MASS

Suggests mclust, pheatmap, gridExtra, knitr, rmarkdown, testthat

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

SystemRequirements GNU make

RoxygenNote 7.2.3

VignetteBuilder knitr, rmarkdown

Config/pak/sysreqs make

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

RemoteUrl https://github.com/thelovelab/mrlocus

RemoteRef HEAD

RemoteSha d3dd5068a107fc9891f100ada4017207453af08f

Contents

`mrlocus-package` *MRLocus - see this page for typical order of functions*

Description

Mendelian Randomization per locus, leveraging eQTL and GWAS summary statistics, for estimation of gene-to-trait effect size and dispersion.

The main functions (in order of typical usage) are:

- `collapseHighCorSNPs` - collapse high correlation SNPs
- `flipAllelesAndGather` - flip alleles and gather for colocalization
- `fitBetaColoc` - perform colocalization across signal clusters
- `extractForSlope` - extract one SNP per signal cluster for MR analysis
- `fitSlope` - perform MR analysis to estimate gene-to-trait effect
- `plotMrlocus` - plot estimates from MR analysis
- `priorCheck` - perform prior predictive checks

References

Anqi Zhu*, Nana Matoba*, Emmaleigh Wilson, Amanda L. Tapia, Yun Li, Joseph G. Ibrahim, Jason L. Stein, Michael I. Love. MRLocus: identifying causal genes mediating a trait through Bayesian estimation of allelic heterogeneity. (2020) bioRxiv https://doi.org/10.1101/2020.08.14.250720

Stan Development Team (2019). RStan: the R interface to Stan. R package version 2.19.2. https://mc-stan.org

`collapseHighCorSNPs` *Collapse high correlation SNPs*

Description

A helper function to collapse sets of highly correlated SNPs within signal clusters. This is recommended to run before `flipAllelesAndGather`, and before `fitBetaColoc`.

Usage

```
collapseHighCorSNPs(
  sum_stat,
  ld_mat,
  ld_mat2 = NULL,
  threshold = 0.95,
  score = NULL,
  plot = TRUE,
  snp_id = NULL
)
```

Arguments

`sum_stat`	list of summary statistic tables, which is a list over signal clusters. Each element of the list should be a data.frame describing the eQTL and GWAS summary statistics. The only column in `sum_stat` that is used by the function is `score` (optional)
`ld_mat`	list of LD matrices across signal clusters
`ld_mat2`	optional second list of LD matrices (for different populations). it will be returned alongside the first `ld_mat`, which is used for the collapsing. The second list of LD matrices is just subset to the same set of SNPs as the first
`threshold`	threshold on absolute value of correlation for collapsing, e.g. will collapse if SNPs are more correlated (or anti-correlated) than this amount
`score`	name of a column of sum_stat data.frames with a score, such that collapsing will choose the highest score SNP per collapsed cluster. Otherwise, if set to NULL, the first SNP will be used
`plot`	logical, draw a before/after grid of plots
`snp_id`	name of SNP id column in `sum_stat`, if specified will output a column `collapsed` that lists which SNP ids are represented in the output (i.e. which highly correlated SNPs were collapsed).

Value

list with subset `ld_mat` and `sum_stat` lists (and `ld_mat2` if provided)

`extractForSlope` *Extract SNPs from colocalization for slope fitting*

Description

Extracts one or more SNPs from each signal cluster based on the posterior estimate of the effect size for A (largest effect size in the positive direction). After running this function, it is recommended to use `trimClusters` to remove signal clusters that are too highly correlated.

Usage

```
extractForSlope(
  res,
  niter = 0,
  plot = TRUE,
  label = "Effect size of",
  a = "eQTL",
  b = "GWAS"
)
```

Arguments

`res`	list with the following named elements: • `beta_hat_a` - list of point estimates of coefficients for A from colocalization • `beta_hat_b` - " " for B • `sd_a` - list of sampling SD for `beta_hat_a` (in practice original SE are provided here) • `sd_b` - " " for `beta_hat_b` " " • alleles (optional) list of data.frame with allele information
`niter`	number of iterations of EM to run for mclust, if set to 0, only the maximum variant (in terms of A effect size) per signal cluster is output. Default is to not run clustering, but to take the SNP with the largest effect size in A (in the positive direction)
`plot`	logical, draw a before after of which variants will be included for slope estimation
`label`	what preceeds `a` and `b` in the x- and y-axis labels
`a`	name of A experiment
`b`	name of B experiment

Value

list of vectors of the first four arguments, collapsed now across signal clusters, representing variants with positive effect on A. So the null variants have been removed (and any variants per cluster that indicated a negative effect on A). If `alleles` data.frames were included in the input, they will also be passed through as a single data.frame with the selected SNPs per signal cluster

`fitBetaColoc` *Fit the colocalization model per signal cluster*

Description

Implements the MRLocus colocalization step, in which summary statistics from A and B studies are used in a generative model, with a horseshoe prior on the latent true effect sizes. Posterior effect sizes and posterior SD are returned, although in pratice we use the original SE in the subsequent `fitSlope` model. See Supplementary Methods of the MRLocus manuscript.

Usage

```
fitBetaColoc(beta_hat_a, beta_hat_b, se_a, se_b, Sigma_a, Sigma_b, ...)
```

Arguments

`beta_hat_a`	vector of estimated coefficients for A
`beta_hat_b`	" " for B
`se_a`	vector of standard errors for beta_hat_a
`se_b`	" " for beta_hat_b
`Sigma_a`	correlation matrix of SNPs for A
`Sigma_b`	" " for B (this could be different for different LD matrix)
`...`	further arguments passed to rstan::sampling

Value

a list with the following elements:

- `stanfit` object
- posterior means for estimated coefficients for A and B
- posterior standard deviations for A and B
- scaling factors for A and B

Two important notes: (1) in MRLocus manuscript, original SE are used instead of posterior SD in the slope fitting step, (2) the posterior means and SD for estimated coefficients are appropriately scaled, while the results from the `stanfit` object are not scaled. In order to scale the results from the stanfit object, `scale_a` and `scale_b` should be divided out from both coefficients and SDs (see Supplementary Methods).

References

Anqi Zhu*, Nana Matoba*, Emma P. Wilson, Amanda L. Tapia, Yun Li, Joseph G. Ibrahim, Jason L. Stein, Michael I. Love. MRLocus: identifying causal genes mediating a trait through Bayesian estimation of allelic heterogeneity. (2021) PLOS Genetics https://doi.org/10.1371/journal.pgen.1009455

`fitSlope` *Fit the mediation slope model: effect of A on B*

Description

Implements the MRLocus slope fitting step, in which the estimated coefficients and their original SE are used to determine the mediation slope (`alpha`), and the dispersion of individual signal clusters around the slope (`sigma`). This function follows the colocalization step `fitBetaColoc` and `extractForSlope`. The output `fitSlope` can be visualized with `plotMrlocus`. For details on the model, see Supplementary Methods of the MRLocus manuscript. See vignette for example of model interpretation.

Usage

```
fitSlope(
  res,
  sd_beta = NULL,
  mu_alpha = NULL,
  sd_alpha = NULL,
  sd_sigma = NULL,
  ...
)
```

Arguments

`res`	list with the following named elements: • `beta_hat_a` - point estimates of coefficients for A from colocalization • `beta_hat_b` - " " for B • `sd_a` - sampling SD for `beta_hat_a` (in practice original SE are provided here) • `sd_b` - " " for `beta_hat_b` " " • alleles (optional) data.frame with allele information
`sd_beta`	prior SD for beta A (default value will be derived from data)
`mu_alpha`	prior mean for `alpha` (default value of 0)
`sd_alpha`	prior SD for `alpha` (default value will be derived from data)
`sd_sigma`	prior SD for `sigma` (default value of 1)
`...`	further arguments passed to `rstan::sampling`

Details

Note that since v0.0.26, the default prior mean for alpha is 0, whereas in the paper it was derived from the data.

Note that if summary statistics for only one SNP are provided a warning will be printed (this is not a recommended use of MRLocus) and a parametric simulation is used to estimate the slope, instead of the Bayesian model.

Value

list with the following elements: `stanfit` object, original estimated coefficients and standard deviations, as well as the `alleles` data.frame (if it was provided)

References

Anqi Zhu*, Nana Matoba*, Emma P. Wilson, Amanda L. Tapia, Yun Li, Joseph G. Ibrahim, Jason L. Stein, Michael I. Love. MRLocus: identifying causal genes mediating a trait through Bayesian estimation of allelic heterogeneity. (2021) PLOS Genetics https://doi.org/10.1371/journal.pgen.1009455

`flipAllelesAndGather` *Flip alleles and gather results into lists*

Description

A helper function to flip alleles from eQTL and GWAS datasets, such that they agree on the effect allele, that the SNPs in a signal cluster are in postive correlation with the index SNP (eQTL), and that the effect allele is coded such that it is the expression increasing allele. This is recommended to run after `collapseHighCorSNPs`, and before `fitBetaColoc`.

Usage

```
flipAllelesAndGather(
  sum_stat,
  ld_mat,
  ld_mat2 = NULL,
  a,
  b,
  ref,
  eff,
  beta,
  se,
  a2_plink,
  a2_plink_mat2 = NULL,
  snp_id,
  sep,
  ab_last = TRUE,
  alleles_same = FALSE,
  plot = TRUE
)
```

Arguments

`sum_stat`	list of summary statistic tables. A list over signal clusters, where each element is a data.frame with summary statistics from eQTL and GWAS datasets. The names of the columns are specified by arguments below (e.g. `a`, `b`, `ref`, `eff`, etc.)

ld_mat	list of LD matrices
ld_mat2	optional second list of LD matrices (for different populations). it will be returned alongside the first ld_mat, which is used for the allele flipping. the second list of LD matrices is just flipped in the same way
a	name of A in columns of sum_stat ("eQTL")
b	name of B ("GWAS")
ref	name of reference allele
eff	name of effect allele
beta	name of estimated coefficient
se	name of standard error
a2_plink	name of the column representing the a2 allele (reference allele) according to plink. the default for plink v1.9 and earlier was to reset a2 to the major allele, unless an optional flag was used. in plink v2.0 and onward, one should check to see which allele is used as reference for calculating the LD matrix. If plink was not used, this argument should just point to the reference allele that was used for LD calculation
a2_plink_mat2	name of the column representing the a2 allele for the second LD matrix, ld_mat2 (needed only if ld_mat2 was specified)
snp_id	name of SNP id
sep	character separator in column names that involve A/B
ab_last	logical, A/B descriptor is last in column names (e.g. "beta_eqtl", "se_eqtl"))
alleles_same	logical, A/B/LD matrix alleles are identical
plot	logical, draw a scatterplot of the flipped betas

Value

list with estimated coefficients, standard errors, LD matrix, and alleles data.frame

---

makeSimDataForMrlocus *Make simple simulated summary data*

Description

Make simple simulated summary data

Usage

```
makeSimDataForMrlocus(
  nsnp = c(7:10),
  idx = 5,
  alpha = 0.5,
  sigma = 0.05,
  betas = 1:4,
  se = 0.25,
  n_mult = 1
)
```

Arguments

nsnp	number of SNPs per signal cluster
idx	the causal SNP (same per cluster for simplicity)
alpha	the true slope of B coefficients over A coefficients
sigma	the SD of true B coefficients around the conditional values given true A coefficients
betas	the true A coefficients
se	the standard errors for betas
n_mult	how many more samples the B study has

Value

a list of beta_hat_a, beta_hat_b, se_a, se_b, Sigma_a, Sigma_b (themselves lists), and alleles (a list of data.frames each with id, ref, eff for the SNP id, reference allele, and effect allele).

---

normalizedAllelicSpread

Estimate the normalized allelic spread

Description

A useful statistic is to normalize the estimate of sigma from estimateSlope, the dispersion, with respect to typical mediated effect sizes. This helps in particular to compare sigma across different GWAS traits, which may have different scale.

Usage

```
normalizedAllelicSpread(fit)
```

Arguments

fit	a list output by estimateSlope

Details

We define the normalized allelic spread as an estimate of sigma divided by something called the "mean mediated effect". The mean mediated effect is defined as the mean QTL effect size multiplied by the estimate of the slope. So in the plotMrlocus plot, the mean mediated effect is found by taking the mean among the spread of points on the x-axis, then going up to the linear trend. This function outputs the normalized allelic spread which is again: sigma / mean mediated effect.

Value

the normalized allelic spread, a numeric

plotInitEstimates *Plot initial estimates over signal clusters*

Description

Plot initial estimates over signal clusters

Usage

```
plotInitEstimates(x, label = "Effect size of", a = "eQTL", b = "GWAS")
```

Arguments

x	list of signal clusters data with beta_hat_a and beta_hat_b lists
label	what preceeds a and b in the x- and y-axis labels
a	name of A experiment
b	name of B experiment

plotMrlocus *Plot estimates from MRLocus slope fitting step*

Description

Plot estimates from MRLocus slope fitting step

Usage

```
plotMrlocus(
  res,
  q = c(0.1, 0.9),
  sigma_mult = 1.28,
  label = "Effect size of",
  a = "eQTL",
  b = "GWAS",
  xlim = NULL,
  ylim = NULL,
  legend = TRUE,
  digits = 3,
  col_slope = "blue",
  col_band = rgb(0, 0, 1, 0.1),
  col_dashed = rgb(0, 0, 1, 0.5),
  ...
)
```

Arguments

res	the output from fitSlope
q	the quantiles of the posterior to use for drawing the uncertainty on the slope. The default is an 80 percent interval
sigma_mult	multiplier on estimate of sigma for drawing the dispersion band (e.g. qnorm(1 - .2/2) ~= 1.28 should include 80 percent of coefficient pairs)
label	what preceeds a and b in the x- and y-axis labels
a	name of A experiment
b	name of B experiment
xlim	xlim (if NULL will be set automatically)
ylim	ylim (if NULL will be set automatically)
legend	logical, whether to show a legend
digits	number of digits to show in legend
col_slope	the color of the slope (alpha)
col_band	the color of the band
col_dashed	the color of the dashed lines
...	arguments passed to plot

---

priorCheck *Basic prior checks on MRLocus slope fit*

Description

This function provides some basic checks on the strength of the prior in the MRLocus slope fitting Bayesian model. It is not desired that the prior overly influences the posterior inference.

Usage

```
priorCheck(res, n = 200, plot = TRUE, type = 1)
```

Arguments

res	output of fitSlope
n	integer, for the plot how many data points to simulate
plot	logical, whether to draw the plots
type	integer, return type. by default (type=1) the function returns a table. By setting type=2, the prior predictive draws for alpha, sigma, beta_a, and beta_b are returned. See Details regarding the simulated draws for beta_a

Details

The posterior-over-prior SD ratio is calculated and returned in a table, and two plots are made that show parameters drawn from the estimated priors (in MRLocus, two priors are estimated from the data - the SD of beta, the instrument effects, and the SD of alpha, the slope). Alternatively, the prior predictive draws themselves can be returned instead of the table (by setting `type=2`).

If the posterior-over-prior SD ratio is close to 1 for either alpha or sigma, this indicates undesirable influence of the prior on the posterior inference. For comparison, some consider a posterior-prior SD ratio of 0.1 or higher to be described as an 'informative prior' (from Stan wiki on prior choice recommendations). We note that an 'informative prior' alone is not problematic for MRLocus, and the prior estimation steps have been designed to be informative as to reasonable values for some of the prior parameters of alpha and sigma.

The plots show parameters generated from the prior and the model. The simulated true values of `beta_a` and `beta_b` are drawn as black circles (summary statistics would then be drawn from these according to the reported SEs, but this step of the model is omitted in this plot). The two plots differ in that the second plot uses fixed alpha (fixed to the prior mean) instead of drawing it from the model (so that the prior for sigma can better be visualized). The fitted estimates of `beta_a` and `beta_b` from the colocalization step are shown as blue X's. One exception where parameters are not drawn from the prior is: `beta_a` values are instead drawn as uniform between 0 and 1.1x the maximum value of `beta_hat_a` from the colocalization step (for ease of visualization).

Value

a data.frame with information about prior and posterior SD for alpha and sigma, and two plots are generated (see Details)

`trimClusters` *Trim signal clusters based on pairwise r2*

Description

This function identifies the clusters to remove such that all remaining clusters have pairwise r2 below a given threshold. It is assumed the r2 matrix is ordered such that the first row/column corresponds to the most significant signal cluster, and so on. trimClusters either removes clusters correlated with the most significant cluster (first) and proceeds downwards, or removes clusters with correlation to more significant clusters and proceeds upwards. Moving downward tends to preserve more clusters. The function stops once the pairwise r2 are all below the threshold. It is recommended to run this function after `extractForSlope`.

Usage

```
trimClusters(r2, r2_threshold, direction = "down")
```

Arguments

`r2`	the matrix of r2 values
`r2_threshold`	the threshold on r2
`direction`	to proceed from the first cluster down (`"down"`) or from the last cluster up (`"up"`)

Value

a numeric vector (possibly length 0) of the signal clusters that should be trimmed/removed

Index