Package ‘colocboost’

May 2, 2025

Type Package

Date 2025-04-22

Title Multi-Context Colocalization Analysis for QTL and GWAS Studies

Version 1.0.4

Maintainer Xuewei Cao <xc2270@cumc.columbia.edu>

Description A multi-task learning approach to variable selection regression with highly correlated predictors and sparse effects,
based on frequentist statistical inference. It provides statistical evidence to identify which subsets of predictors have non-zero
effects on which subsets of response variables, motivated and designed for colocalization analysis across genome-wide association studies (GWAS)
and quantitative trait loci (QTL) studies.
The ColocBoost model is described in Cao et. al. (2025) <doi:10.1101/2025.04.17.25326042>.

Encoding UTF-8

LazyDataCompression xz

LazyData true

RoxygenNote 7.3.2

URL https://github.com/StatFunGen/colocboost

BugReports https://github.com/StatFunGen/colocboost/issues

Depends R (>= 4.0.0)

Imports Rfast, matrixStats

Suggests testthat (>= 3.0.0), knitr, rmarkdown, ashr, MASS, susieR

VignetteBuilder knitr

Config/testthat/edition 3

License MIT + file LICENSE

NeedsCompilation no

Author Xuewei Cao [cre, aut, cph],
Haochen Sun [aut, cph],
Ru Feng [aut, cph],

Daniel Nachun [aut, cph],
Kushal Dey [aut, cph],
Gao Wang [aut, cph]

Repository CRAN

Date/Publication 2025-05-02 09:20:07 UTC

Contents

`Ambiguous_Colocalization` *A real data example includes an ambiguous colocalization between eQTL and GWAS*

Description

An example result from one of our real data applications, which shows an ambiguous colocalization between eQTL and GWAS.

Usage

```
Ambiguous_Colocalization
```

Format

`Ambiguous_Colocalization`:

A list with 2 elements

ColocBoost_Results A `colocboost` output object

SuSiE_Results Two `susie` output object for eQTL and GWAS

COLOC_V5_Results A `coloc` output object

Source

The Ambiguous_Colocalization dataset contains a real data example from one of our real data applications, which shows an ambiguous colocalization between eQTL and GWAS. The dataset is specifically designed for evaluating and demonstrating the capabilities of ColocBoost in real data applications. See details in tutorial vignette `https://statfungen.github.io/colocboost/articles/index.html`.

See Also

Other colocboost_data: `Heterogeneous_Effect`, `Ind_5traits`, `Non_Causal_Strongest_Marginal`, `Sumstat_5traits`, `Weaker_GWAS_Effect`

`colocboost` *ColocBoost: A gradient boosting informed multi-omics xQTL colocalization method*

Description

`colocboost` implements a proximity adaptive smoothing gradient boosting approach for multi-trait colocalization at gene loci, accommodating multiple causal variants. This method, introduced by Cao etc. (2025), is particularly suited for scaling to large datasets involving numerous molecular quantitative traits and disease traits. In brief, this function fits a multiple linear regression model $Y = XB + E$ in matrix form. ColocBoost can be generally used in multi-task variable selection regression problem.

Usage

```
colocboost(
  X = NULL,
  Y = NULL,
  sumstat = NULL,
  LD = NULL,
  dict_YX = NULL,
  dict_sumstatLD = NULL,
  outcome_names = NULL,
  focal_outcome_idx = NULL,
  focal_outcome_variables = TRUE,
  overlap_variables = FALSE,
  intercept = TRUE,
  standardize = TRUE,
  effect_est = NULL,
  effect_se = NULL,
  effect_n = NULL,
  M = 500,
  stop_thresh = 1e-06,
  tau = 0.01,
  learning_rate_init = 0.01,
  learning_rate_decay = 1,
  dynamic_learning_rate = TRUE,
  prioritize_jkstar = TRUE,
  func_compare = "min_max",
  jk_equiv_corr = 0.8,
  jk_equiv_loglik = 1,
  coloc_thresh = 0.1,
  lambda = 0.5,
  lambda_focal_outcome = 1,
  func_simplex = "LD_z2z",
  func_multi_test = "lfdr",
  stop_null = 1,
  multi_test_max = 1,
  multi_test_thresh = 1,
  ash_prior = "normal",
  p.adjust.methods = "fdr",
  residual_correlation = NULL,
  coverage = 0.95,
  min_cluster_corr = 0.8,
  dedup = TRUE,
  overlap = TRUE,
  n_purity = 100,
  min_abs_corr = 0.5,
  median_abs_corr = NULL,
  median_cos_abs_corr = 0.8,
  tol = 1e-09,
  merge_cos = TRUE,
  sec_coverage_thresh = 0.8,
  weight_fudge_factor = 1.5,
  check_null = 0.1,
  check_null_method = "profile",
  check_null_max = 0.025,
  weaker_effect = TRUE,
  LD_free = FALSE,
  output_level = 1
)
```

Arguments

`X` A list of genotype matrices for different outcomes, or a single matrix if all outcomes share the same genotypes. Each matrix should have column names, if sample sizes and variables possibly differing across matrices.

`Y` A list of vectors of outcomes or an N by L matrix if it is considered for the same X and multiple outcomes.

`sumstat` A list of data.frames of summary statistics. The columns of data.frame should include either `z` or `beta`/`sebeta`. `n` is the sample size for the summary statistics, it is highly recommendation to provide. `variant` is required if sumstat for different outcomes do not have the same number of variables. `var_y` is the variance of phenotype (default is 1 meaning that the Y is in the “standardized” scale).

`LD` A list of correlation matrix indicating the LD matrix for each genotype. It also could be a single matrix if all sumstats were obtained from the same genotypes.

`dict_YX` A L by 2 matrix of dictionary for `X` and `Y` if there exist subsets of outcomes corresponding to the same X matrix. The first column should be 1:L for L outcomes. The second column should be the index of `X` corresponding to the outcome. The innovation: do not provide the same matrix in `X` to reduce the computational burden.

`dict_sumstatLD` A L by 2 matrix of dictionary for `sumstat` and `LD` if there exist subsets of outcomes corresponding to the same sumstat. The first column should be 1:L for L sumstat The second column should be the index of `LD` corresponding to the sumstat. The innovation: do not provide the same matrix in `LD` to reduce the computational burden.

`outcome_names` The names of outcomes, which has the same order for Y.

`focal_outcome_idx` The index of the focal outcome if perform GWAS-xQTL ColocBoost

`focal_outcome_variables` If `focal_outcome_variables = TRUE`, only consider the variables exist in the focal outcome.

`overlap_variables` If `overlap_variables = TRUE`, only perform colocalization in the overlapped region.

`intercept` If `intercept = TRUE`, the intercept is fitted. Setting `intercept = FALSE` is generally not recommended.

`standardize` If `standardize = TRUE`, standardize the columns of genotype and outcomes to unit variance.

`effect_est` Matrix of variable regression coefficients (i.e. regression beta values) in the genomic region

`effect_se` Matrix of standard errors associated with the beta values

`effect_n` A scalar or a vector of sample sizes for estimating regression coefficients. Highly recommended!

`M` The maximum number of gradient boosting rounds for each outcome (default is 500).

`stop_thresh` The stop criterion for overall profile loglikelihood function.

`tau` The smooth parameter for proximity adaptive smoothing weights for the best update jk-star.

`learning_rate_init` The minimum learning rate for updating in each iteration.

`learning_rate_decay` The decayrate for learning rate. If the objective function is large at the early iterations, we need to have the higher learning rate to improve the computational efficiency.

`dynamic_learning_rate` If `dynamic_learning_rate = TRUE`, the dynamic learning rate based on `learning_rate_init` and `learning_rate_decay` will be used in SEC.

`prioritize_jkstar` When `prioritize_jkstar = TRUE`, the selected outcomes will prioritize best update j_k^star in SEC.

`func_compare` The criterion when we update jk-star in SEC (default is "min_max").

`jk_equiv_corr` The LD cutoff between overall best update jk-star and marginal best update jk-l for lth outcome

`jk_equiv_loglik` The change of loglikelihood cutoff between overall best update jk-star and marginal best update jk-l for lth outcome

`coloc_thresh` The cutoff of checking if the best update jk-star is the potential causal variable for outcome l if jk-l is not similar to jk-star (used in Delayed SEC).

`lambda` The ratio [0,1] for z^2 and z in fun_prior simplex, default is 0.5

`lambda_focal_outcome` The ratio for z^2 and z in fun_prior simplex for the focal outcome, default is 1

`func_simplex` The data-driven local association simplex δ for smoothing the weights. Default is "LD_z2z" is the elastic net for z-score and also weighted by LD.

`func_multi_test` The alternative method to check the stop criteria. When `func_multi_test = "lfdr"`, boosting iterations will be stopped if the local FDR for all variables are greater than `lfsr_max`.

`stop_null` The cutoff of nominal p-value when `func_multi_test = "Z"`.

`multi_test_max` The cutoff of the smallest FDR for stop criteria when `func_multi_test = "lfdr"` or `func_multi_test = "lfsr"`.

`multi_test_thresh` The cutoff of the smallest FDR for pre-filtering the outcomes when `func_multi_test = "lfdr"` or `func_multi_test = "lfsr"`.

`ash_prior` The prior distribution for calculating lfsr when `func_multi_test = "lfsr"`.

`p.adjust.methods` The adjusted pvalue method in stats:p.adj when `func_multi_test = "fdr"`

`residual_correlation` The residual correlation based on the sample overlap, it is diagonal if it is NULL.

`coverage` A number between 0 and 1 specifying the “coverage” of the estimated colocalization confidence sets (CoS) (default is 0.95).

`min_cluster_corr`
The small correlation for the weights distributions across different iterations to be decided having only one cluster.

`dedup` If `dedup = TRUE`, the duplicate confidence sets will be removed in the post-processing.

`overlap` If `overlap = TRUE`, the overlapped confidence sets will be removed in the post-processing.

`n_purity` The maximum number of confidence set (CS) variables used in calculating the correlation ("purity") statistics. When the number of variables included in the CS is greater than this number, the CS variables are randomly subsampled.

`min_abs_corr` Minimum absolute correlation allowed in a confidence set. The default is 0.5 corresponding to a squared correlation of 0.25, which is a commonly used threshold for genotype data in genetic studies.

`median_abs_corr`
An alternative "purity" threshold for the CS. Median correlation between pairs of variables in a CS less than this threshold will be filtered out and not reported. When both min_abs_corr and median_abs_corr are set, a CS will only be removed if it fails both filters. Default set to NULL but it is recommended to set it to 0.8 in practice.

`median_cos_abs_corr`
Median absolute correlation between variants allowed to merge multiple colocalized sets. The default is 0.8 corresponding to a stringent threshold to merge colocalized sets, which may resulting in a huge set.

`tol` A small, non-negative number specifying the convergence tolerance for checking the overlap of the variables in different sets.

`merge_cos` When `merge_cos = TRUE`, the sets for only one outcome will be merged if passed the `median_cos_abs_corr`.

`sec_coverage_thresh`
A number between 0 and 1 specifying the weight in each SEC (default is 0.8).

`weight_fudge_factor`
The strength to integrate weight from different outcomes, default is 1.5

`check_null` The cut off value for change conditional objective function. Default is 0.1.

`check_null_method`
The metric to check the null sets. Default is "profile"

`check_null_max` The smallest value of change of profile loglikelihood for each outcome.

`weaker_effect` If `weaker_effect = TRUE`, consider the weaker single effect due to coupling effects

`LD_free` When `LD_free = FALSE`, objective function doesn't include LD information.

`output_level` When `output_level = 1`, return basic cos details for colocalization results When `output_level = 2`, return the ucos details for the single specific effects. When `output_level = 3`, return the entire Colocboost model to diagnostic results (more space).

Details

The function colocboost implements the proximity smoothed gradient boosting method from Cao etc (2025). There is an additional step to help merge the confidence sets with small between_putiry (default is 0.8) but within the same locus. This step addresses potential instabilities in linkage disequilibrium (LD) estimation that may arise from small sample sizes or discrepancies in minor allele frequencies (MAF) across different confidence sets.

Value

A "colocboost" object with some or all of the following elements:

cos_summary	A summary table for colocalization events.
vcp	The variable colocalized probability for each variable.
cos_details	A object with all information for colocalization results.
data_info	A object with detailed information from input data
model_info	A object with detailed information for colocboost model
ucos_details	A object with all information for trait-specific effects when output_level = 2.
diagnositci_details	A object with diagnostic details for ColocBoost model when output_level = 3.

Source

See detailed instructions in our tutorial portal: https://statfungen.github.io/colocboost/index.html

Examples

```
# colocboost example
set.seed(1)
N <- 1000
P <- 100
# Generate X with LD structure
sigma <- 0.9^abs(outer(1:P, 1:P, "-"))
X <- MASS::mvrnorm(N, rep(0, P), sigma)
colnames(X) <- paste0("SNP", 1:P)
L <- 3
true_beta <- matrix(0, P, L)
true_beta[10, 1] <- 0.5 # SNP10 affects trait 1
true_beta[10, 2] <- 0.4 # SNP10 also affects trait 2 (colocalized)
true_beta[50, 2] <- 0.3 # SNP50 only affects trait 2
true_beta[80, 3] <- 0.6 # SNP80 only affects trait 3
Y <- matrix(0, N, L)
for (l in 1:L) {
  Y[, l] <- X %*% true_beta[, l] + rnorm(N, 0, 1)
}
res <- colocboost(X = X, Y = Y)
res$cos_details$cos$cos_index
```

colocboost_plot *Plot visualization plot from a ColocBoost output.*

Description

colocboost_plot generates visualization plots for colocalization events from a ColocBoost analysis.

Usage

```
colocboost_plot(
  cb_output,
  y = "log10p",
  grange = NULL,
  plot_cos_idx = NULL,
  outcome_idx = NULL,
  plot_all_outcome = FALSE,
  plot_focal_only = FALSE,
  plot_focal_cos_outcome_only = FALSE,
  points_color = "grey80",
  cos_color = NULL,
  add_vertical = FALSE,
  add_vertical_idx = NULL,
  outcome_names = NULL,
  plot_cols = 2,
  variant_coord = FALSE,
  show_top_variables = FALSE,
  show_cos_to_uncoloc = FALSE,
  show_cos_to_uncoloc_idx = NULL,
  show_cos_to_uncoloc_outcome = NULL,
  plot_ucos = FALSE,
  plot_ucos_idx = NULL,
  title_specific = NULL,
  ylim_each = TRUE,
  outcome_legend_pos = "top",
  outcome_legend_size = 1.8,
  cos_legend_pos = c(0.05, 0.4),
  show_variable = FALSE,
  lab_style = c(2, 1),
  axis_style = c(2, 1),
  title_style = c(2.5, 2),
  ...
)
```

Arguments

cb_output	Output object from colocboost analysis

`y` Specifies the y-axis values, default is "log10p" for -log10 transformed marginal association p-values.

`grange` Optional plotting range of x-axis to zoom in to a specific region.

`plot_cos_idx` Optional indices of CoS to plot

`outcome_idx` Optional indices of outcomes to include in the plot. `outcome_idx=NULL` to plot only the outcomes having colocalization.

`plot_all_outcome`
Optional to plot all outcome in the same figure.

`plot_focal_only`
Logical, if TRUE only plots colocalization with focal outcome, default is FALSE.

`plot_focal_cos_outcome_only`
Logical, if TRUE only plots colocalization including at least on colocalized outcome with focal outcome, default is FALSE.

`points_color` Background color for non-colocalized variables, default is "grey80".

`cos_color` Optional custom colors for CoS.

`add_vertical` Logical, if TRUE adds vertical lines at specified positions, default is FALSE

`add_vertical_idx`
Optional indices for vertical lines.

`outcome_names` Optional vector of outcomes names for the subtitle of each figure. `outcome_names=NULL` for the outcome name shown in `data_info`.

`plot_cols` Number of columns in the plot grid, default is 2. If you have many colocalization. please consider increasing this.

`variant_coord` Logical, if TRUE uses variant coordinates on x-axis, default is FALSE. This is required the variable names including position information.

`show_top_variables`
Logical, if TRUE shows top variables for each CoS, default is FALSE

`show_cos_to_uncoloc`
Logical, if TRUE shows colocalization to uncolocalized outcomes to diagnose, default is FALSE

`show_cos_to_uncoloc_idx`
Optional indices for showing CoS to all uncolocalized outcomes

`show_cos_to_uncoloc_outcome`
Optional outcomes for showing CoS to uncolocalized outcomes

`plot_ucos` Logical, if TRUE plots also trait-specific (uncolocalized) sets , default is FALSE

`plot_ucos_idx` Optional indices of trait-specific (uncolocalized) sets to plot when included

`title_specific` Optional specific title to display in plot title

`ylim_each` Logical, if TRUE uses separate y-axis limits for each plot, default is TRUE

`outcome_legend_pos`
Position for outcome legend, default is "top"

`outcome_legend_size`
Size for outcome legend text, default is 1.2

`cos_legend_pos` Proportion of the legend from (left edge, bottom edge), default as (0.05, 0.4) at the left - median position `show_variable` Logical, if TRUE displays variant IDs, default is FALSE

`lab_style` Vector of two numbers for label style (size, boldness), default is c(2, 1)

`axis_style` Vector of two numbers for axis style (size, boldness), default is c(2, 1)

`title_style` Vector of two numbers for title style (size, boldness), default is c(2.5, 2)

`...` Additional parameters passed to `plot` functions

Value

Visualization plot for each colocalization event.

Source

See detailed instructions in our tutorial portal: `https://statfungen.github.io/colocboost/articles/Visualization_ColocBoost_Output.html`

Examples

```
# colocboost example
set.seed(1)
N <- 1000
P <- 100
# Generate X with LD structure
sigma <- 0.9^abs(outer(1:P, 1:P, "-"))
X <- MASS::mvrnorm(N, rep(0, P), sigma)
colnames(X) <- paste0("SNP", 1:P)
L <- 3
true_beta <- matrix(0, P, L)
true_beta[10, 1] <- 0.5 # SNP10 affects trait 1
true_beta[10, 2] <- 0.4 # SNP10 also affects trait 2 (colocalized)
true_beta[50, 2] <- 0.3 # SNP50 only affects trait 2
true_beta[80, 3] <- 0.6 # SNP80 only affects trait 3
Y <- matrix(0, N, L)
for (l in 1:L) {
  Y[, l] <- X %*% true_beta[, l] + rnorm(N, 0, 1)
}
res <- colocboost(X = X, Y = Y)
colocboost_plot(res, plot_cols = 1)
colocboost_plot(res, plot_cols = 1, outcome_idx = 1:3)
```

`get_ambiguous_colocalization`

Get ambiguous colocalization events from trait-specific (uncolocalized) effects.

Description

`get_ambiguous_colocalization` get the colocalization by discarding the weaker colocalization events or colocalized outcomes

Usage

```
get_ambiguous_colocalization(
  cb_output,
  min_abs_corr_between_ucos = 0.5,
  median_abs_corr_between_ucos = 0.8,
  tol = 1e-09
)
```

Arguments

`cb_output` Output object from `colocboost` analysis

`min_abs_corr_between_ucos`
Minimum absolute correlation for variants across two trait-specific (uncolocalized) effects to be considered colocalized. The default is 0.5.

`median_abs_corr_between_ucos`
Median absolute correlation for variants across two trait-specific (uncolocalized) effects to be considered colocalized. The default is 0.8.

`tol` A small, non-negative number specifying the convergence tolerance for checking the overlap of the variables in different sets.

Value

A `"colocboost"` object of colocboost output with additional elements:

`ambiguous_cos` If exists, a list of ambiguous trait-specific (uncolocalized) effects.

Source

See detailed instructions in our tutorial portal: `https://statfungen.github.io/colocboost/articles/Interpret_ColocBoost_Output.html`

See Also

Other colocboost_inference: `get_colocboost_summary()`, `get_robust_colocalization()`

Examples

```
data(Ambiguous_Colocalization)
test_colocboost_results <- Ambiguous_Colocalization$ColocBoost_Results
res <- get_ambiguous_colocalization(test_colocboost_results)
names(res$ambiguous_cos)
```

get_colocboost_summary

Get summary tables from a ColocBoost output.

Description

get_colocboost_summary get colocalization and trait-specific summary table with or without the outcomes of interest.

Usage

```
get_colocboost_summary(
  cb_output,
  summary_level = 1,
  outcome_names = NULL,
  interest_outcome = NULL,
  region_name = NULL,
  min_abs_corr_between_ucos = 0.5,
  median_abs_corr_between_ucos = 0.8
)
```

Arguments

cb_output
: Output object from colocboost analysis

summary_level
: When summary_level = 1, return basic summary table for colocalization results. See details in get_ucos_summary function when summary_level = 2.

outcome_names
: Optional vector of names of outcomes, which has the same order as Y in the original analysis.

interest_outcome
: Optional vector specifying a subset of outcomes from outcome_names to focus on. When provided, only colocalization events that include at least one of these outcomes will be returned.

region_name
: Optional character string. When provided, adds a column with this gene name to the output table for easier filtering in downstream analyses.

min_abs_corr_between_ucos
: Minimum absolute correlation for variants across two trait-specific (uncolocalized) effects to be considered colocalized. The default is 0.5.

median_abs_corr_between_ucos
: Median absolute correlation for variants across two trait-specific (uncolocalized) effects to be considered colocalized. The default is 0.8.

Details

When summary_level = 1, additional details and examples are introduced in get_cos_summary. When summary_level = 2 or summary_level = 3, additional details for trait-specific effects and ambiguous colocalization events are included. See get_ucos_summary for details on these tables.

Value

A list containing results from the ColocBoost analysis:

- When summary_level = 1 (default):
  - cos_summary: A summary table for colocalization events with the following columns:
    - focal_outcome: The focal outcome being analyzed if exists. Otherwise, it is FALSE.
    - colocalized_outcomes: Colocalized outcomes for colocalization confidence set (CoS)
    - cos_id: Unique identifier for colocalization confidence set (CoS)
    - purity: Minimum absolute correlation of variables within colocalization confidence set (CoS)
    - top_variable: The variable with highest variant colocalization probability (VCP)
    - top_variable_vcp: Variant colocalization probability for the top variable
    - cos_npc: Normalized probability of colocalization
    - min_npc_outcome: Minimum normalized probability of colocalized traits
    - n_variables: Number of variables in colocalization confidence set (CoS)
    - colocalized_index: Indices of colocalized variables
    - colocalized_variables: List of colocalized variables
    - colocalized_variables_vcp: Variant colocalization probabilities for all colocalized variables
- When summary_level = 2:
  - cos_summary: As described above
  - ucos_summary: A summary table for trait-specific (uncolocalized) effects
- When summary_level = 3:
  - cos_summary: As described above
  - ucos_summary: A summary table for trait-specific (uncolocalized) effects
  - ambiguous_cos_summary: A summary table for ambiguous colocalization events from trait-specific effects

Source

See detailed instructions in our tutorial portal: https://statfungen.github.io/colocboost/articles/Interpret_ColocBoost_Output.html

See Also

Other colocboost_inference: get_ambiguous_colocalization(), get_robust_colocalization()

Examples

```
# colocboost example
set.seed(1)
N <- 1000
P <- 100
# Generate X with LD structure
sigma <- 0.9^abs(outer(1:P, 1:P, "-"))
```

```
X <- MASS::mvrnorm(N, rep(0, P), sigma)
colnames(X) <- paste0("SNP", 1:P)
L <- 3
true_beta <- matrix(0, P, L)
true_beta[10, 1] <- 0.5 # SNP10 affects trait 1
true_beta[10, 2] <- 0.4 # SNP10 also affects trait 2 (colocalized)
true_beta[50, 2] <- 0.3 # SNP50 only affects trait 2
true_beta[80, 3] <- 0.6 # SNP80 only affects trait 3
Y <- matrix(0, N, L)
for (l in 1:L) {
  Y[, l] <- X %*% true_beta[, l] + rnorm(N, 0, 1)
}
res <- colocboost(X = X, Y = Y)
get_colocboost_summary(res)
```

get_cormat *A fast function to calculate correlation matrix (LD matrix) from individual level data*

Description

This function calculates the correlation matrix (LD matrix) from individual level data.

Usage

```
get_cormat(X, intercepte = TRUE)
```

Arguments

X	A matrix of individual level data.
intercepte	A logical value indicating whether to include an intercept in the model. Default is FALSE.

Value

A correlation matrix (LD matrix).

See Also

Other colocboost_utilities: get_cos(), get_cos_purity(), get_cos_summary(), get_hierarchical_clusters(), get_ucos_summary()

Examples

```
# colocboost example
set.seed(1)
N <- 1000
P <- 100
# Generate X with LD structure
sigma <- 0.9^abs(outer(1:P, 1:P, "-"))
X <- MASS::mvrnorm(N, rep(0, P), sigma)
cormat <- get_cormat(X)
```

get_cos *Extract CoS at different coverage*

Description

get_cos extracts colocalization confidence sets (CoS) at different coverage levels from ColocBoost results. When genotype data (X) or correlation matrix (Xcorr) is provided, it can also calculate and filter CoS based on purity statistics, ensuring that variants within each CoS are sufficiently correlated.

Usage

```
get_cos(
  cb_output,
  coverage = 0.95,
  X = NULL,
  Xcorr = NULL,
  n_purity = 100,
  min_abs_corr = 0.5,
  median_abs_corr = NULL
)
```

Arguments

cb_output	Output object from colocboost analysis
coverage	A number between 0 and 1 specifying the “coverage” of the estimated colocalization confidence sets (CoS) (default is 0.95).
X	Genotype matrix of values of the p variables. Used to compute correlations if Xcorr is not provided.
Xcorr	Correlation matrix of correlations between variables. Alternative to X.
n_purity	The maximum number of CoS variables used in calculating the correlation (“purity”) statistics.
min_abs_corr	The minimum absolute correlation value of variants in a CoS to be considered pass (“purity”) statistics.

median_abs_corr

The median absolute correlation value of variants in a CoS to be considered pass (“purity”) statistics. When the number of variables included in the CoS is greater than this number, the CoS variables are randomly subsampled.

Value

A list of indices of variables in each CoS.

See Also

Other colocboost_utilities: get_cormat(), get_cos_purity(), get_cos_summary(), get_hierarchical_clusters(), get_ucos_summary()

Examples

```
# colocboost example
set.seed(1)
N <- 1000
P <- 100
# Generate X with LD structure
sigma <- 0.9^abs(outer(1:P, 1:P, "-"))
X <- MASS::mvrnorm(N, rep(0, P), sigma)
colnames(X) <- paste0("SNP", 1:P)
L <- 3
true_beta <- matrix(0, P, L)
true_beta[10, 1] <- 0.5 # SNP10 affects trait 1
true_beta[10, 2] <- 0.4 # SNP10 also affects trait 2 (colocalized)
true_beta[50, 2] <- 0.3 # SNP50 only affects trait 2
true_beta[80, 3] <- 0.6 # SNP80 only affects trait 3
Y <- matrix(0, N, L)
for (l in 1:L) {
  Y[, l] <- X %*% true_beta[, l] + rnorm(N, 0, 1)
}
res <- colocboost(X = X, Y = Y)
get_cos(res, coverage = 0.99, X = X)
get_cos(res, coverage = 0.99, X = X, min_abs_corr = 0.95)
```

---

get_cos_purity *Calculate purity within and in-between CoS*

Description

Calculate purity statistics between all pairs of colocalization confidence sets (CoS)

Usage

```
get_cos_purity(cos, X = NULL, Xcorr = NULL, n_purity = 100)
```

Arguments

`cos` List of variables in CoS

`X` Genotype matrix of values of the p variables. Used to compute correlations if Xcorr is not provided.

`Xcorr` Correlation matrix of correlations between variables. Alternative to X.

`n_purity` The maximum number of CoS variables used in calculating the correlation (“purity”) statistics. When the number of variables included in the CoS is greater than this number, the CoS variables are randomly subsampled.

Value

A list containing three matrices (min_abs_cor, max_abs_cor, median_abs_cor) with purity statistics for all pairs of CoS. Diagonal elements represent within-CoS purity.

See Also

Other colocboost_utilities: `get_cormat()`, `get_cos()`, `get_cos_summary()`, `get_hierarchical_clusters()`, `get_ucos_summary()`

Examples

```
# colocboost example
set.seed(1)
N <- 1000
P <- 100
# Generate X with LD structure
sigma <- 0.9^abs(outer(1:P, 1:P, "-"))
X <- MASS::mvrnorm(N, rep(0, P), sigma)
colnames(X) <- paste0("SNP", 1:P)
L <- 3
true_beta <- matrix(0, P, L)
true_beta[10, 1] <- 0.5
true_beta[10, 2] <- 0.4
true_beta[50, 2] <- 0.3
true_beta[80, 3] <- 0.6
Y <- matrix(0, N, L)
for (l in 1:L) {
  Y[, l] <- X %*% true_beta[, l] + rnorm(N, 0, 1)
}
res <- colocboost(X = X, Y = Y)
cos_res <- get_cos(res, coverage = 0.8)
get_cos_purity(cos_res$cos, X = X)
```

`get_cos_summary` *Get colocalization summary table from a ColocBoost output.*

Description

`get_cos_summary` get the colocalization summary table with or without the outcomes of interest.

Usage

```
get_cos_summary(
  cb_output,
  outcome_names = NULL,
  interest_outcome = NULL,
  region_name = NULL
)
```

Arguments

`cb_output` Output object from `colocboost` analysis

`outcome_names` Optional vector of names of outcomes, which has the same order as Y in the original analysis.

`interest_outcome`
Optional vector specifying a subset of outcomes from `outcome_names` to focus on. When provided, only colocalization events that include at least one of these outcomes will be returned.

`region_name` Optional character string. When provided, adds a column with this gene name to the output table for easier filtering in downstream analyses.

Value

A summary table for colocalization events with the following columns:

`focal_outcome` The focal outcome being analyzed if exists. Otherwise, it is `FALSE`.

`colocalized_outcomes`
Colocalized outcomes for colocalization confidence set (CoS)

`cos_id` Unique identifier for colocalization confidence set (CoS)

`purity` Minimum absolute correlation of variables with in colocalization confidence set (CoS)

`top_variable` The variable with highest variant colocalization probability (VCP)

`top_variable_vcp`
Variant colocalization probability for the top variable

`cos_npc` Normalized probability of colocalization

`min_npc_outcome`
Minimum normalized probability of colocalized traits

`n_variables` Number of variables in colocalization confidence set (CoS)

`colocalized_index`
Indices of colocalized variables

`colocalized_variables`
List of colocalized variables

`colocalized_variables_vcp`
Variant colocalization probabilities for all colocalized variables

Source

See detailed instructions in our tutorial portal: https://statfungen.github.io/colocboost/articles/Interpret_ColocBoost_Output.html

See Also

Other colocboost_utilities: `get_cormat()`, `get_cos()`, `get_cos_purity()`, `get_hierarchical_clusters()`, `get_ucos_summary()`

Examples

```
# colocboost example
set.seed(1)
N <- 1000
P <- 100
# Generate X with LD structure
sigma <- 0.9^abs(outer(1:P, 1:P, "-"))
X <- MASS::mvrnorm(N, rep(0, P), sigma)
colnames(X) <- paste0("SNP", 1:P)
L <- 3
true_beta <- matrix(0, P, L)
true_beta[10, 1] <- 0.5 # SNP10 affects trait 1
true_beta[10, 2] <- 0.4 # SNP10 also affects trait 2 (colocalized)
true_beta[50, 2] <- 0.3 # SNP50 only affects trait 2
true_beta[80, 3] <- 0.6 # SNP80 only affects trait 3
Y <- matrix(0, N, L)
for (l in 1:L) {
  Y[, l] <- X %*% true_beta[, l] + rnorm(N, 0, 1)
}
res <- colocboost(X = X, Y = Y)
get_cos_summary(res)
```

`get_hierarchical_clusters` *Perform modularity-based hierarchical clustering for a correlation matrix*

Description

This function performs a modularity-based hierarchical clustering approach to identify clusters from a correlation matrix.

Usage

```
get_hierarchical_clusters(cormat, min_cluster_corr = 0.8)
```

Arguments

`cormat` A correlation matrix.

`min_cluster_corr`
The small correlation for the weights distributions across different iterations to be decided having only one cluster. Default is 0.8.

Value

A list containing:

`cluster` A binary matrix indicating the cluster membership of each variable.

`Q_modularity` The modularity values for the identified clusters.

See Also

Other colocboost_utilities: `get_cormat()`, `get_cos()`, `get_cos_purity()`, `get_cos_summary()`, `get_ucos_summary()`

Examples

```
# Example usage
set.seed(1)
N <- 100
P <- 4
sigma <- matrix(0.2, nrow = P, ncol = P)
diag(sigma) <- 1
sigma[1:2, 1:2] <- 0.9
sigma[3:4, 3:4] <- 0.9
X <- MASS::mvrnorm(N, rep(0, P), sigma)
cormat <- get_cormat(X)
clusters <- get_hierarchical_clusters(cormat)
clusters$cluster
clusters$Q_modularity
```

get_robust_colocalization

Recalibrate and summarize robust colocalization events.

Description

`get_robust_colocalization` get the colocalization by discarding the weaker colocalization events or colocalized outcomes

Usage

```
get_robust_colocalization(
  cb_output,
  cos_npc_cutoff = 0.5,
  npc_outcome_cutoff = 0.2,
  pvalue_cutoff = NULL,
  weight_fudge_factor = 1.5,
  coverage = 0.95
)
```

Arguments

`cb_output`	Output object from `colocboost` analysis
`cos_npc_cutoff`	Minimum threshold of normalized probability of colocalization (NPC) for CoS.
`npc_outcome_cutoff`	Minimum threshold of normalized probability of colocalized traits in each CoS.
`pvalue_cutoff`	Maximum threshold of marginal p-values of colocalized variants on colocalized traits in each CoS.
`weight_fudge_factor`	The strength to integrate weight from different outcomes, default is 1.5
`coverage`	A number between 0 and 1 specifying the “coverage” of the estimated colocalization confidence sets (CoS) (default is 0.95).

Value

A `"colocboost"` object with some or all of the following elements:

`cos_summary`	A summary table for colocalization events.
`vcp`	The variable colocalized probability for each variable.
`cos_details`	A object with all information for colocalization results.
`data_info`	A object with detailed information from input data
`model_info`	A object with detailed information for colocboost model
`ucos_from_cos`	A object with information for trait-specific effects if exists after removing weaker signals.

Source

See detailed instructions in our tutorial portal: https://statfungen.github.io/colocboost/articles/Interpret_ColocBoost_Output.html

See Also

Other colocboost_inference: `get_ambiguous_colocalization()`, `get_colocboost_summary()`

Examples

```
# colocboost example
set.seed(1)
N <- 1000
P <- 100
# Generate X with LD structure
sigma <- 0.9^abs(outer(1:P, 1:P, "-"))
X <- MASS::mvrnorm(N, rep(0, P), sigma)
colnames(X) <- paste0("SNP", 1:P)
L <- 3
true_beta <- matrix(0, P, L)
true_beta[10, 1] <- 0.5 # SNP10 affects trait 1
true_beta[10, 2] <- 0.4 # SNP10 also affects trait 2 (colocalized)
true_beta[50, 2] <- 0.3 # SNP50 only affects trait 2
true_beta[80, 3] <- 0.6 # SNP80 only affects trait 3
Y <- matrix(0, N, L)
for (l in 1:L) {
  Y[, l] <- X %*% true_beta[, l] + rnorm(N, 0, 1)
}
res <- colocboost(X = X, Y = Y)
res$cos_details$cos$cos_index
filter_res <- get_robust_colocalization(res, cos_npc_cutoff = 0.5, npc_outcome_cutoff = 0.2)
filter_res$cos_details$cos$cos_index
```

`get_ucos_summary` *Get trait-specific summary table from a ColocBoost output.*

Description

`get_ucos_summary` produces a trait-specific summary table for uncolocalized (single-trait) associations from ColocBoost results. This is particularly useful for examining trait-specific signals or for summarizing results from single-trait FineBoost analyses.

Usage

```
get_ucos_summary(
  cb_output,
  outcome_names = NULL,
  region_name = NULL,
  ambiguous_cos = FALSE,
  min_abs_corr_between_ucos = 0.5,
  median_abs_corr_between_ucos = 0.8
)
```

Arguments

`cb_output` Output object from `colocboost` analysis

`outcome_names` Optional vector of names of outcomes, which has the same order as Y in the original analysis.

`region_name` Optional character string. When provided, adds a column with this gene name to the output table for easier filtering in downstream analyses.

`ambiguous_cos` Logical indicating whether to include ambiguous colocalization events. The default is FALSE.

`min_abs_corr_between_ucos`
Minimum absolute correlation for variants across two trait-specific (uncolocalized) effects to be considered colocalized. The default is 0.5.

`median_abs_corr_between_ucos`
Median absolute correlation for variants across two trait-specific (uncolocalized) effects to be considered colocalized. The default is 0.8.

Value

A list containing:

- `ucos_summary`: A summary table for trait-specific, uncolocalized associations with the following columns:
  - `outcomes`: Outcome being analyzed
  - `ucos_id`: Unique identifier for trait-specific confidence sets
  - `purity`: Minimum absolute correlation of variables within trait-specific confidence sets
  - `top_variable`: The variable with highest variant-level probability of association (VPA)
  - `top_variable_vpa`: Variant-level probability of association (VPA) for the top variable
  - `ucos_npc`: Normalized probability of causal association for the trait-specific confidence set
  - `n_variables`: Number of variables in trait-specific confidence set
  - `ucos_index`: Indices of variables in the trait-specific confidence set
  - `ucos_variables`: List of variables in the trait-specific confidence set
  - `ucos_variables_vpa`: Variant-level probability of association (VPA) for all variables in the confidence set
  - `region_name`: Region name if provided through the region_name parameter
- `ambiguous_cos_summary`: A summary table for ambiguous colocalization events with the following columns:
  - `outcomes`: Outcome in the ambiguous colocalization event
  - `ucos_id`: Unique identifiers for the ambiguous event
  - `min_between_purity`: Minimum absolute correlation between variables across trait-specific sets in the ambiguous event

- median_between_purity: Median absolute correlation between variables across trait-specific sets in the ambiguous event
- overlap_idx: Indices of variables that overlap between ambiguous trait-specific sets
- overlap_variables: Names of variables that overlap between ambiguous trait-specific sets
- n_recalibrated_variables: Number of variables in the recalibrated colocalization set from an ambiguous event
- recalibrated_index: Indices of variables in the recalibrated colocalization set from an ambiguous event
- recalibrated_variables: Names of variables in the recalibrated colocalization set from an ambiguous event
- recalibrated_variables_vcp: Variant colocalization probabilities for recalibrated variables from an ambiguous event
- region_name: Region name if provided through the region_name parameter

Source

See detailed instructions in our tutorial portal: https://statfungen.github.io/colocboost/articles/Interpret_ColocBoost_Output.html

See Also

Other colocboost_utilities: get_cormat(), get_cos(), get_cos_purity(), get_cos_summary(), get_hierarchical_clusters()

Examples

```
# colocboost example with single trait analysis
set.seed(1)
N <- 1000
P <- 100
# Generate X with LD structure
sigma <- 0.9^abs(outer(1:P, 1:P, "-"))
X <- MASS::mvrnorm(N, rep(0, P), sigma)
colnames(X) <- paste0("SNP", 1:P)
L <- 1  # Only one trait for single-trait analysis
true_beta <- matrix(0, P, L)
true_beta[10, 1] <- 0.5 # SNP10 affects the trait
true_beta[80, 1] <- 0.2 # SNP11 also affects the trait but with lower effect
Y <- X %*% true_beta + rnorm(N, 0, 1)
res <- colocboost(X = X, Y = Y, output_level = 2)
# Get the trait-specifc effect summary
get_ucos_summary(res)
```

`Heterogeneous_Effect` *Individual level data for 2 traits and 2 causal variants with heterogeneous effects*

Description

An example dataset with simulated genotypes and traits for 2 traits and 2 common causal variants with heterogeneous effects

Usage

```
Heterogeneous_Effect
```

Format

`Heterogeneous_Effect`**:**

A list with 3 elements

X List of genotype matrices

Y List of traits

variant indices of two causal variants

Source

The Heterogeneous_Effect dataset contains 2 simulated phenotypes alongside corresponding genotype matrices. There are two causal variants, both of which have heterogeneous effects on two traits. Due to the file size limitation of CRAN release, this is a subset of simulated data to generate Figure 2b in Cao etc. 2025. See full dataset in colocboost paper repo `https://github.com/StatFunGen/colocboost-paper`.

See Also

Other colocboost_data: `Ambiguous_Colocalization`, `Ind_5traits`, `Non_Causal_Strongest_Marginal`, `Sumstat_5traits`, `Weaker_GWAS_Effect`

`Ind_5traits` *Individual level data for 5 traits*

Description

An example dataset with simulated genotypes and traits for 5 traits

Usage

```
Ind_5traits
```

Format

`Ind_5traits`**:**

A list with 3 elements

X List of genotype matrices

Y List of traits

true_effect_variants List of causal variants

Source

The Ind_5traits dataset contains 5 simulated phenotypes alongside corresponding genotype matrices. The dataset is specifically designed for evaluating and demonstrating the capabilities of ColocBoost in multi-trait colocalization analysis with individual-level data. See Cao etc. 2025 for details. Due to the file size limitation of CRAN release, this is a subset of simulated data. See full dataset in colocboost paper repo `https://github.com/StatFunGen/colocboost-paper`.

See Also

Other colocboost_data: `Ambiguous_Colocalization`, `Heterogeneous_Effect`, `Non_Causal_Strongest_Marginal`, `Sumstat_5traits`, `Weaker_GWAS_Effect`

`Non_Causal_Strongest_Marginal` *Individual level data for 2 traits and 2 causal variants, but the strongest marginal association is not causal*

Description

An example dataset with simulated genotypes and traits for 2 traits and 2 common causal variants, but the strongest marginal association is not causal variant.

Usage

```
Non_Causal_Strongest_Marginal
```

Format

`Non_Causal_Strongest_Marginal`**:**

A list with 3 elements

X List of genotype matrices

Y List of traits

variant indices of two causal variants

Source

The Non_Causal_Strongest_Marginal dataset contains 2 simulated phenotypes alongside corresponding genotype matrices. There are two causal variants, but the strongest marginal association is not a causal variant. Due to the file size limitation of CRAN release, this is a subset of simulated data to generate Figure 2b in Cao etc. 2025. See full dataset in colocboost paper repo `https://github.com/StatFunGen/colocboost-paper`.

See Also

Other colocboost_data: `Ambiguous_Colocalization`, `Heterogeneous_Effect`, `Ind_5traits`, `Sumstat_5traits`, `Weaker_GWAS_Effect`

`Sumstat_5traits` *Summary level data for 5 traits*

Description

An example dataset with simulated statistics for 5 traits

Usage

```
Sumstat_5traits
```

Format

`Sumstat_5traits`**:**

A list with 2 elements

sumstat Summary statistics for 5 traits

true_effect_variants List of causal variants

Source

The Sumstat_5traits dataset contains 5 simulated summary statistics, where it is directly derived from the Ind_5traits dataset using marginal association. The dataset is specifically designed for evaluating and demonstrating the capabilities of ColocBoost in multi-trait colocalization analysis with summary association data. See Cao etc. 2025 for details. Due to the file size limitation of CRAN release, this is a subset of simulated data. See full dataset in colocboost paper repo `https://github.com/StatFunGen/colocboost-paper`.

See Also

Other colocboost_data: `Ambiguous_Colocalization`, `Heterogeneous_Effect`, `Ind_5traits`, `Non_Causal_Strongest_Ma`
`Weaker_GWAS_Effect`

`Weaker_GWAS_Effect` *Individual level data for 2 traits and 2 causal variants with weaker effects for focal trait*

Description

An example dataset with simulated genotypes and traits for 2 traits and 2 common causal variants with heterogeneous effects

Usage

```
Weaker_GWAS_Effect
```

Format

`Weaker_GWAS_Effect`**:**

A list with 3 elements

- **X** List of genotype matrices
- **Y** List of traits
- **variant** indices of two causal variants

Source

The Weaker_GWAS_Effect dataset contains 2 simulated phenotypes alongside corresponding genotype matrices. There are two causal variants, one of which has a weaker effect on the focal trait compared to the other trait. Due to the file size limitation of CRAN release, this is a subset of simulated data to generate Figure 2b in Cao etc. 2025. See full dataset in colocboost paper repo `https://github.com/StatFunGen/colocboost-paper`.

See Also

Other colocboost_data: `Ambiguous_Colocalization`, `Heterogeneous_Effect`, `Ind_5traits`, `Non_Causal_Strongest_Ma` `Sumstat_5traits`

Index