

# Package ‘cophescan’

May 8, 2026

**Title** Adaptation of the Coloc Method for PheWAS

**Version** 1.4.3

**Maintainer** Ichcha Manipur <im504@cam.ac.uk>

**Description** A Bayesian method for Phenome-wide association studies (PheWAS) that identifies causal associations between genetic variants and traits, while simultaneously addressing confounding due to linkage disequilibrium. For details see Manipur et al (2024, Nature Communications) <doi:10.1038/s41467-024-49990-8>.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**VignetteBuilder** knitr

**RoxygenNote** 7.3.3

**Depends** R (>= 3.5.0)

**URL** <https://github.com/ichcha-m/cophescan>,  
<https://ichcha-m.github.io/cophescan/>

**BugReports** <https://github.com/ichcha-m/cophescan/issues>

**Imports** Rcpp (>= 1.0.7), coloc, data.table, ggplot2, ggrepel, pheatmap, methods, viridis, stats, grDevices, magrittr, utils, matrixStats, dplyr

**Suggests** knitr, testthat (>= 3.0.0), rmarkdown, RColorBrewer, ggpubr

**Collate** 'cophescan-package.R' 'singlevar.R' 'multivarsusie.R' 'multitrait.R' 'cophe\_hyp\_predict.R' 'copheplots.R' 'testdata.R' 'RcppExports.R' 'metrop\_hier\_priors.R' 'zzz.R'

**LinkingTo** Rcpp, RcppArmadillo

**Config/testthat/edition** 3

**ByteCompile** true

**NeedsCompilation** yes

**Author** Ichcha Manipur [aut, cre],  
Chris Wallace [aut]

**Repository** CRAN

**Date/Publication** 2026-03-18 15:00:03 UTC

## Contents

cophescan-package	3
adjust_priors	3
average_piks	4
average_piks_list	5
average_posterior_prob	5
average_posterior_prob_list	6
combine.bf	7
cophe.hyp.predict	7
cophe.multitrait	8
cophe.single	10
cophe.single.lbf	11
cophe.susie	12
cophe.susie.lbf	13
cophe_heatmap	15
cophe_multi_trait_data	15
cophe_plot	16
get_beta	17
get_posterior_prob	17
Hc.cutoff.fdr	18
hypothesis.priors	18
logd_alpha	19
logd_beta	19
logd_gamma	20
loglik	20
logpost	21
logpriors	21
logsum	22
logsumexp	23
metrop_run	23
multitrait.simplify	24
pars2pik	25
pars_init	25
per.snp.priors	26
piks	27
plot_trait_manhat	27
posterior_prob	28
prepare_plot_data	28
propose	29
run_metrop_priors	30
sample_alpha	31
sample_beta	32
sample_gamma	32
summary.cophe	33
target	33

---

cophescan-package      *The 'cophescan' package.*

---

## Description

Coloc adapted Phenome-wide Scans

## Author(s)

**Maintainer:** Ichcha Manipur <im504@cam.ac.uk>

Authors:

- Chris Wallace

## See Also

Useful links:

- <https://github.com/ichcha-m/cophescan>
- <https://ichcha-m.github.io/cophescan/>
- Report bugs at <https://github.com/ichcha-m/cophescan/issues>

---

adjust\_priors      *adjust\_priors*

---

## Description

adjust fixed priors when nsnp in region is high

## Usage

```
adjust_priors(  
  nsnp,  
  pa = 3.82e-05,  
  pc = 0.00182,  
  p1 = NULL,  
  p2 = NULL,  
  p12 = NULL  
)
```

**Arguments**

nsnps	number of SNPs
pa	prior probability that a non-query variant is causally associated with the query trait (cophescan prior), default 3.82e-5
pc	prior probability that the query variant is causally associated with the query trait (cophescan prior), default 1.82e-3 (cophescan prior)
p1	prior probability a SNP is associated with trait 1, (coloc prior), pc derived by using $pc = p12/p1 + p12$ ; use p1, p2, p12 only when pa and pc are unavailable (See vignettes)
p2	prior probability a SNP is associated with trait 2, (coloc prior), pa derived by using $pa = p2$
p12	prior probability a SNP is associated with both traits, (coloc prior), pc derived by using $pc = p12/p1 + p12$

**Value**

vector of pn, pa and pc adjusted prior probabilities

---

average_piks	<i>Average of priors: pnk, pak and pck</i>
--------------	--------------------------------------------

---

**Description**

Average of priors: pnk, pak and pck

**Usage**

```
average_piks(params, nsnps, covar_vec, nits, thin, covar = FALSE)
```

**Arguments**

params	Vector of parameters: $\alpha$ , $\beta$ and $\gamma$
nsnps	number of snps
covar_vec	Vector of the covariate
nits	Number of iterations run in mcmc
thin	thinning
covar	logical: was the covariate information used? default: False

**Value**

average pik matrix of priors: pnk, pak and pck

---

average\_piks\_list      *Average of priors: pnk, pak and pck from list (memory intensive)*

---

**Description**

Average of priors: pnk, pak and pck from list (memory intensive)

**Usage**

```
average_piks_list(params, nsnps, covar_vec, nits, thin, covar = FALSE)
```

**Arguments**

params	Vector of parameters: $\alpha$ , $\beta$ and $\gamma$
nsnps	number of snps
covar_vec	Vector of the covariate
nits	Number of iterations run in mcmc
thin	thinning
covar	logical: was the covariate information used? default: False

**Value**

average pik matrix of priors: pnk, pak and pck

---

average\_posterior\_prob  
*Average of posterior probabilities: Hn, Ha and Hc*

---

**Description**

Average of posterior probabilities: Hn, Ha and Hc

**Usage**

```
average_posterior_prob(
  params,
  lbf_mat,
  nsnps,
  covar_vec,
  nits,
  thin,
  covar = FALSE
)
```

**Arguments**

params	Vector of parameters: $\alpha$ , $\beta$ and $\gamma$
lbf_mat	matrix of log bayes factors: IBF.Ha and IBF.Hc
nsnps	number of snps
covar_vec	Vector of the covariate
nits	Number of iterations run in mcmc
thin	thinning
covar	logical: was the covariate information used? default: False

**Value**

matrix with average of all the posterior probabilities: Hn, Ha and Hc

---

average\_posterior\_prob\_list

*Average of posterior probabilities: Hn, Ha and Hc from list (memory intensive)*

---

**Description**

Average of posterior probabilities: Hn, Ha and Hc from list (memory intensive)

**Usage**

```
average_posterior_prob_list(
  params,
  lbf_mat,
  nsnps,
  covar_vec,
  nits,
  thin,
  covar = FALSE
)
```

**Arguments**

params	Vector of parameters: $\alpha$ , $\beta$ and $\gamma$
lbf_mat	matrix of log bayes factors: IBF.Ha and IBF.Hc
nsnps	number of snps
covar_vec	Vector of the covariate
nits	Number of iterations run in mcmc
thin	thinning
covar	logical: was the covariate information used? default: False

**Value**

matrix with average of all the posterior probabilities: Hn, Ha and Hc

---

combine.bf	<i>combine.bf</i>
------------	-------------------

---

**Description**

Calculate posterior probabilities for all the configurations

**Usage**

```
combine.bf(lBF_df, pn, pa, pc)
```

**Arguments**

lBF_df	dataframe with log bayes factors of hypothesis Ha and Hn: column names should be lBF.Ha and lBF.Hc
pn	prior probability that none of the SNPs/variants in the region are associated with the query trait
pa	prior probability that a non-query variant is causally associated with the query trait
pc	prior probability that the query variant is causally associated with the query trait

**Value**

named numeric vector of posterior probabilities and bayes factors

**Author(s)**

Ichcha Manipur

---

cophe.hyp.predict	<i>Predict cophescan hypothesis for tested associations</i>
-------------------	-------------------------------------------------------------

---

**Description**

Predict cophescan hypothesis for tested associations

**Usage**

```
cophe.hyp.predict(
  cophe.res,
  grouping.vars = c("querysnp", "querytrait"),
  Hc.cutoff = 0.6,
  Hn.cutoff = 0.2
)
```

**Arguments**

<code>cophe.res</code>	results obtained from <code>cophe.single</code> , <code>cophe.susie</code> or <code>cophe.multitrait</code> or <code>data.frame</code> with the following columns: <code>PP.Hn</code> , <code>PP.Hc</code> , <code>PP.Ha</code> , <code>querysnp</code> , <code>querytrait</code>
<code>grouping.vars</code>	This is important for results from <code>cophe.susie</code> where there are multiple signals. These will be collapsed into one call. If you want to return all signals set this to a single variable eg: <code>grouping.vars = c('querysnp')</code>
<code>Hc.cutoff</code>	threshold for <code>PP.Hc</code> above which the associations are called <code>Hc</code>
<code>Hn.cutoff</code>	threshold for <code>PP.Hn</code> above which the associations are called <code>Hn</code>

**Value**

returns dataframe with posterior probabilities of `Hn`, `Hc` and `Ha` with the predicted hypothesis based on the provided cut.off.

**See Also**

[cophe.single](#), [cophe.susie](#), [cophe.multitrait](#), [multitrait.simplify](#)

---

<code>cophe.multitrait</code>	<i>Run cophescan on multiple traits at once</i>
-------------------------------	-------------------------------------------------

---

**Description**

Run cophescan on multiple traits at once

**Usage**

```
cophe.multitrait(
  trait.dat,
  querysnpid,
  querytrait.names,
  LDmat = NULL,
  method = "single",
  simplify = FALSE,
  predict.hyp = TRUE,
  Hn.cutoff = 0.2,
  Hc.cutoff = 0.6,
  est.fdr.based.cutoff = FALSE,
  fdr = 0.05,
  ...
)
```

**Arguments**

<code>trait.dat</code>	Named(traits) list of coloc structured data for k traits (Total number of traits)
<code>querysnpid</code>	vector of query variant ids = length(trait.dat), if the same variant
<code>querytrait.names</code>	vector of names for the query traits, if the names of the multi.dat list contain the trait names please pass querytrait.names=names(multi.dat)
<code>LDmat</code>	LD matrix
<code>method</code>	either 'single' for <code>cophe.single</code> or 'susie' for <code>cophe.susie</code>
<code>simplify</code>	if TRUE removes intermediate results from output using 'multitrait.simplify'
<code>predict.hyp</code>	if TRUE predicts the hypothesis based on the provided thresholds for pp.Hc and pp.Hn (overrides simplify) using <code>cophe.hyp.predict</code>
<code>Hn.cutoff</code>	threshold for PP.Hn above which the associations are called Hn
<code>Hc.cutoff</code>	threshold for PP.Hc above which the associations are called Hc
<code>est.fdr.based.cutoff</code>	if True calculates the Hc.cutoff using $1 - \text{mean}(\text{PP.Hc})   \text{PP.Hc} > \text{cutoff}$
<code>fdr</code>	fdr threshold to estimate Hc.cutoff
<code>...</code>	additional arguments of priors for <code>cophe.susie</code> or <code>cophe.single</code>

**Value**

if `simplify` is False returns multi-trait list of lists, each with:

- a summary data.frame of the cophescan results
- priors used
- querysnp
- querytrait

if `simplify` is TRUE only returns dataframe with posterior probabilities of Hn, Hc and Ha with no intermediate results if `predict.hyp` is TRUE returns a dataframe with output of `simplify` and the predicted hypotheses for all associations

**Author(s)**

Ichcha Manipur

---

 cophe.single

*Bayesian cophescan analysis using Approximate Bayes Factors*


---

## Description

Bayesian cophescan analysis under single causal variant assumption

## Usage

```
cophe.single(
  dataset,
  querysnpid,
  querytrait,
  MAF = NULL,
  pa = 3.82e-05,
  pc = 0.00182,
  p1 = NULL,
  p2 = NULL,
  p12 = NULL
)
```

## Arguments

dataset	a list with specifically named elements defining the query trait dataset to be analysed.
querysnpid	Id of the query variant, (id in dataset\$snp)
querytrait	Query trait name
MAF	Minor allele frequency vector
pa	prior probability that a non-query variant is causally associated with the query trait (cophescan prior), default 3.82e-5
pc	prior probability that the query variant is causally associated with the query trait (cophescan prior), default 1.82e-3 (cophescan prior)
p1	prior probability a SNP is associated with trait 1, (coloc prior), pc derived by using $pc = p12/p1 + p12$ ; use p1, p2, p12 only when pa and pc are unavailable (See vignettes)
p2	prior probability a SNP is associated with trait 2, (coloc prior), pa derived by using $pa = p2$
p12	prior probability a SNP is associated with both traits, (coloc prior), pc derived by using $pc = p12/p1 + p12$

## Details

This function calculates posterior probabilities of different causal variant configurations under the assumption of a single causal variant for each trait.

If regression coefficients and variances are available, it calculates Bayes factors for association at each SNP. If only p values are available, it uses an approximation that depends on the SNP's MAF and ignores any uncertainty in imputation. Regression coefficients should be used if available. Find more input data structure details in the coloc package

**Value**

a list of two data.frames:

- summary is a vector giving the number of SNPs analysed, and the posterior probabilities of Hn (no shared causal variant), Ha (two distinct causal variants) and Hc (one common causal variant)
- results is an annotated version of the input data containing log Approximate Bayes Factors and intermediate calculations, and the posterior probability SNP.PP.Hc of the SNP being causal for the shared signal *if* Hc is true. This is only relevant if the posterior support for Hc in summary is convincing.

**Author(s)**

Ichcha Manipur

**Examples**

```
library(cophescan)
data(cophe_multi_trait_data)
query_trait_1 <- cophe_multi_trait_data$summ_stat[['Trait_1']]
querysnpid <- cophe_multi_trait_data$querysnpid
res.single <- cophe.single(query_trait_1, querysnpid = querysnpid, querytrait='Trait_1')
summary(res.single)
```

---

`cophe.single.lbf`      *cophe.single.lbf*

---

**Description**

Calculate log bayes factors for each hypothesis (Single causal variant assumption)

**Usage**

```
cophe.single.lbf(dataset, querysnpid, querytrait, MAF = NULL)
```

**Arguments**

dataset	a list with specifically named elements defining the query trait dataset to be analysed.
querysnpid	Id of the query variant, (id in dataset\$snp)
querytrait	Query trait name
MAF	Minor allele frequency vector

**Value**

data frame with log bayes factors for H<sub>n</sub> and H<sub>a</sub> hypotheses

**Author(s)**

Ichcha Manipur

**See Also**

[cophe.single](#)

**Examples**

```
library(cophescan)
data(cophe_multi_trait_data)
query_trait_1 <- cophe_multi_trait_data$summ_stat[['Trait_1']]
querysnpid <- cophe_multi_trait_data$querysnpid
res.single.lbf <- cophe.single.lbf(query_trait_1, querysnpid = querysnpid, querytrait='Trait_1')
res.single.lbf
```

---

cophe.susie

*run cophe.susie using susie to detect separate signals*

---

**Description**

Check if a variant causally associated in one trait might be causal in another trait

**Usage**

```
cophe.susie(
  dataset,
  querysnpid,
  querytrait,
  pa = 3.82e-05,
  pc = 0.00182,
  p1 = NULL,
  p2 = NULL,
  p12 = NULL,
  susie.args = list()
)
```

**Arguments**

dataset	<i>either</i> a list with specifically named elements defining the dataset to be analysed. (see <a href="#">check_dataset</a> )
querysnpid	Id of the query variant
querytrait	Query trait name

pa	prior probability that a non-query variant is causally associated with the query trait (cophescan prior), default 3.82e-5
pc	prior probability that the query variant is causally associated with the query trait (cophescan prior), default 1.82e-3
p1	prior probability a SNP is associated with trait 1, (coloc prior), pc derived by using $pc = p12/p1 + p12$ ; use p1, p2, p12 only when pa and pc are unavailable (See vignettes)
p2	prior probability a SNP is associated with trait 2, (coloc prior), pa derived by using $pa = p2$
p12	prior probability a SNP is associated with both traits, (coloc prior), pc derived by using $pc = p12/p1 + p12$
susie.args	a named list of additional arguments to be passed to <a href="#">runsusie</a>

**Value**

a list, containing elements

- summary a data.table of posterior probabilities of each global hypothesis, one row per pairwise comparison of signals from the two traits
- results a data.table of detailed results giving the posterior probability for each snp to be jointly causal for both traits *assuming Hc is true*. Please ignore this column if the corresponding posterior support for H4 is not high.
- priors a vector of the priors used for the analysis

**Author(s)**

Ichcha Manipur

**Examples**

```
library(cophescan)
data(cophe_multi_trait_data)
query_trait_1 <- cophe_multi_trait_data$summ_stat[['Trait_1']]
querysnpid <- cophe_multi_trait_data$querysnpid
query_trait_1$LD <- cophe_multi_trait_data$LD
res.susie <- cophe.susie(query_trait_1, querysnpid = querysnpid, querytrait='Trait_1')
summary(res.susie)
```

---

cophe.susie.lbf

*cophe.susie.lbf*

---

**Description**

Calculate log bayes factors for each hypothesis (SuSIE - multiple causal variant assumption)

**Usage**

```
cophe.susie.lbf(  
  dataset,  
  queriesnpid,  
  querytrait,  
  switch = TRUE,  
  susie.args = list(),  
  MAF = NULL  
)
```

**Arguments**

dataset	a list with specifically named elements defining the query trait dataset to be analysed.
queriesnpid	Id of the query variant, (id in dataset\$snp)
querytrait	Query trait name
switch	Set switch=TRUE to obtain single BF when credible sets not found with SuSIE
susie.args	a named list of additional arguments to be passed to <a href="#">runsusie</a>
MAF	Minor allele frequency vector

**Value**

data frame with log bayes factors for Hn and Ha hypotheses

**Author(s)**

Ichcha Manipur

**See Also**

[cophe.susie](#)

**Examples**

```
library(cophescan)  
data(cophe_multi_trait_data)  
query_trait_1 <- cophe_multi_trait_data$summ_stat[['Trait_1']]  
query_trait_1$LD <- cophe_multi_trait_data$LD  
queriesnpid <- cophe_multi_trait_data$queriesnpid  
res.susie.lbf <- cophe.susie.lbf(query_trait_1, queriesnpid = queriesnpid,  
                                querytrait='Trait_1', switch=T)  
  
res.susie.lbf
```

---

cophe_heatmap	<i>Heatmap of multi-trait cophescan results</i>
---------------	-------------------------------------------------

---

**Description**

Heatmap of multi-trait cophescan results

**Usage**

```
cophe_heatmap(
  multi.dat,
  querysnpid,
  query_trait_names,
  thresh_Hc = 0.5,
  thresh_Ha = 0.5,
  ...
)
```

**Arguments**

multi.dat	multi trait cophescan results returned from <code>cophe.multitrait</code> or formatted in the same way with <code>multitrait.simplify</code>
querysnpid	query variant
query_trait_names	names of phenotypes corresponding to the multi.dat results
thresh_Hc	Hc threshold to be displayed
thresh_Ha	Ha threshold to be displayed
...	additional arguments to be passed to <a href="#">pheatmap</a>

**Value**

heatmap of posterior probabilities of the phenotypes above the set threshold

---

cophe_multi_trait_data	<i>Simulated multi-trait data</i>
------------------------	-----------------------------------

---

**Description**

Simulated multi-trait data

**Usage**

```
data(cophe_multi_trait_data)
```

**Format**

list of coloc structured datasets for 24 traits (cophe\_multi\_trait\_data\$summ\_stat), LD matrix (cophe\_multi\_trait\_data\$LD) and the id of the query snp (cophe\_multi\_trait\_data\$querysnpid). #' The trait dataset are simulated summary statistics (1000 SNPs) for 10 Hn, 10 Ha and 10 Hc.

---

cophe_plot	<i>cophe_plots showing the Ha and Hc of all traits and labelled above the specified threshold</i>
------------	---------------------------------------------------------------------------------------------------

---

**Description**

cophe\_plots showing the Ha and Hc of all traits and labelled above the specified threshold

**Usage**

```
cophe_plot(
  multi.dat,
  querysnpid,
  query_trait_names,
  thresh_Hc = 0.5,
  thresh_Ha = 0.5,
  beta_p = NULL,
  traits.dat = NULL
)
```

**Arguments**

multi.dat	multi trait cophescan results returned from cophe.multitrait or multitrait.simplify
querysnpid	query variant (only a single variant for PheWAS plots)
query_trait_names	list of phenotype names
thresh_Hc	Hc threshold to be displayed
thresh_Ha	Ha threshold to be displayed
beta_p	data.frame (from the get.beta function) with four columns : 1. "beta_plot": indicating beta direction (p or n) 2. "pval_plot": -log10(pval) of the queried variant 3. "querysnp" 4. "querytrait".
traits.dat	list of multi-trait coloc structured datasets

**Value**

cophescan plots of Ha and Hc

**See Also**

[cophe.single](#), [cophe.susie](#), [cophe.multitrait](#), [multitrait.simplify](#)

---

get_beta	<i>Extract beta and p-values of queried variant</i>
----------	-----------------------------------------------------

---

**Description**

Extract beta and p-values of queried variant

**Usage**

```
get_beta(traits.dat, querysnpid, querytrait)
```

**Arguments**

traits.dat	list of coloc structured dataset
querysnpid	vector of querysnpid
querytrait	vector of querytrait names

**Value**

data.frame with one column named beta\_plot: indicating beta direction (n/p) and another column named pval\_plot with  $-\log_{10}(\text{pval})$  of the queried variant

---

get_posterior_prob	<i>Calculation of the posterior prob of Hn, Ha and Hc</i>
--------------------	-----------------------------------------------------------

---

**Description**

Calculation of the posterior prob of Hn, Ha and Hc

**Usage**

```
get_posterior_prob(params, lbf_mat, nsnp, covar_vec, covar = FALSE)
```

**Arguments**

params	Vector of parameters: $\alpha$ , $\beta$ and $\gamma$
lbf_mat	matrix of log bayes factors: IBF.Ha and IBF.Hc
nsnp	number of snps
covar_vec	Vector of the covariate
covar	logical: should the covariate information be used? default: False

**Value**

posterior prob of Hn, Ha and Hc

---

Hc.cutoff.fdr	<i>Estimate the Hc.cutoff for the required FDR</i>
---------------	----------------------------------------------------

---

**Description**

Estimate the Hc.cutoff for the required FDR

**Usage**

```
Hc.cutoff.fdr(ppHc, fdr = 0.05, return_plot = TRUE)
```

**Arguments**

ppHc	a vector containing the PP.Hc (the posterior probability of causal association) of all tests
fdr	FDR default: 0.05
return_plot	default: TRUE, plot the fdr estimated at the different Hc.cutoff

**Value**

the Hc.cutoff value for the specified FDR, if return\_plot is True returns a plot showing the FDR calculated at different Hc thresholds

---

hypothesis.priors	<i>hypothesis.priors</i>
-------------------	--------------------------

---

**Description**

Estimate priors for each hypothesis

**Usage**

```
hypothesis.priors(nsnps, pn, pa, pc)
```

**Arguments**

nsnps	number of SNPs
pn	prior probability that none of the SNPs/variants in the region are associated with the query trait
pa	prior probability that a non-query variant is causally associated with the query trait
pc	prior probability that the query variant is causally associated with the query trait

**Value**

hypotheses priors

**Author(s)**

Ichcha Manipur

---

logd_alpha	<i>dnorm for alpha</i>
------------	------------------------

---

**Description**

dnorm for alpha

**Usage**

```
logd_alpha(a, alpha_mean = -10, alpha_sd = 0.5)
```

**Arguments**

a	current alpha
alpha_mean	prior for the mean of alpha
alpha_sd	prior for the standard deviation of alpha

**Value**

log dnorm

---

logd_beta	<i>dgamma for beta</i>
-----------	------------------------

---

**Description**

dgamma for beta

**Usage**

```
logd_beta(b, beta_shape = 2, beta_scale = 2)
```

**Arguments**

b	current beta
beta_shape	prior for the shape (gamma distribution) of beta
beta_scale	prior for the scale of beta

**Value**

log dgamma

---

logd_gamma	<i>dgamma for gamma</i>
------------	-------------------------

---

**Description**

dgamma for gamma

**Usage**

```
logd_gamma(g, gamma_shape = 2, gamma_scale = 2)
```

**Arguments**

g	current gamma
gamma_shape	prior for the shape (gamma distribution) of gamma
gamma_scale	prior for the scale of gamma

**Value**

log dgamma

---

loglik	<i>Log likelihood calculation</i>
--------	-----------------------------------

---

**Description**

Log likelihood calculation

**Usage**

```
loglik(params, lbf_mat, nsnps, covar_vec, covar = FALSE)
```

**Arguments**

params	Vector of parameters: $\alpha$ , $\beta$ and $\gamma$
lbf_mat	matrix of log bayes factors: IBF.Ha and IBF.Hc
nsnps	number of snps
covar_vec	Vector of the covariate
covar	logical: should the covariate information be used? default: False

**Value**

logpost log of the posteriors

---

logpost	<i>Log posterior calculation</i>
---------	----------------------------------

---

**Description**

Log posterior calculation

**Usage**

```
logpost(params, lbf_mat, nsnps, covar_vec, covar = FALSE)
```

**Arguments**

params	Vector of parameters: $\alpha$ , $\beta$ and $\gamma$
lbf_mat	matrix of log bayes factors: IBF.Ha and IBF.Hc
nsnps	number of snps
covar_vec	Vector of the covariate
covar	logical: should the covariate information be used? default: False

**Value**

logpost log of the posteriors

---

logpriors	<i>Calculate log priors</i>
-----------	-----------------------------

---

**Description**

Calculate log priors

**Usage**

```
logpriors(  
  params,  
  covar = FALSE,  
  alpha_mean = -10,  
  alpha_sd = 0.5,  
  beta_shape = 2,  
  beta_scale = 2,  
  gamma_shape = 2,  
  gamma_scale = 2  
)
```

**Arguments**

params	Vector of parameters: $\alpha$ , $\beta$ and $\gamma$
covar	logical: Should the covariate information be used? default: False
alpha_mean	prior for the mean of alpha
alpha_sd	prior for the standard deviation of alpha
beta_shape	prior for the shape (gamma distribution) of beta
beta_scale	prior for the scale of beta
gamma_shape	prior for the shape (gamma distribution) of gamma
gamma_scale	prior for the scale of gamma

**Value**

log priors

---

logsum

*logsum*

---

**Description**

Internal function, logsum Function directly taken from coloc This function calculates the log of the sum of the exponentiated logs taking out the max, i.e. insuring that the sum is not Inf

**Usage**

logsum(x)

**Arguments**

x	numeric vector
---	----------------

**Value**

$\max(x) + \log(\text{sum}(\exp(x - \max(x))))$

---

logsumexp	<i>Log sum</i>
-----------	----------------

---

**Description**

Log sum

**Usage**

```
logsumexp(x)
```

**Arguments**

x                    vector of log scale values to be added

**Value**

log sum of input

---

metrop_run	<i>Run the hierarchical mcmc model to infer priors</i>
------------	--------------------------------------------------------

---

**Description**

Run the hierarchical mcmc model to infer priors

**Usage**

```
metrop_run(  
  lbf_mat,  
  nsnps,  
  covar_vec,  
  covar = FALSE,  
  nits = 10000L,  
  thin = 1L,  
  alpha_mean = -10,  
  alpha_sd = 0.5,  
  beta_shape = 2,  
  beta_scale = 2,  
  gamma_shape = 2,  
  gamma_scale = 2  
)
```

**Arguments**

lbf_mat	matrix of log bayes factors: IBF.Ha and IBF.Hc
nsnps	number of snps
covar_vec	Vector of the covariate
covar	logical: Should the covariate information be used? default: False
nits	Number of iterations run in mcmc
thin	thinning
alpha_mean	prior for the mean of alpha
alpha_sd	prior for the standard deviation of alpha
beta_shape	prior for the shape (gamma distribution) of beta
beta_scale	prior for the scale of beta
gamma_shape	prior for the shape (gamma distribution) of gamma
gamma_scale	prior for the scale of gamma

**Value**

named list of log likelihood (ll) and parameters: alpha, beta and gamma

---

multitrait.simplify *Simplifying the output obtained from* cophe.multitrait,  
cophe.single *or* cophe.susie

---

**Description**

Simplifying the output obtained from cophe.multitrait, cophe.single or cophe.susie

**Usage**

```
multitrait.simplify(multi.dat, only_BF = FALSE)
```

**Arguments**

multi.dat	output obtained from cophe.multitrait, cophe.single or cophe.susie
only_BF	return only bayes factors and not posterior probabilities (default=FALSE)

**Value**

dataframe with posterior probabilities of Hn, Hc and Ha

---

pars2pik	<i>Conversion of parameters alpha, beta and gamma to p<sub>nk</sub>, p<sub>ak</sub> and p<sub>ck</sub></i>
----------	------------------------------------------------------------------------------------------------------------

---

**Description**

Conversion of parameters alpha, beta and gamma to p<sub>nk</sub>, p<sub>ak</sub> and p<sub>ck</sub>

**Usage**

```
pars2pik(params, nsnps, covar_vec, covar = FALSE)
```

**Arguments**

params	Vector of parameters: $\alpha$ , $\beta$ and $\gamma$
nsnps	number of snps
covar_vec	Vector of the covariate
covar	logical: should the covariate information be used? default: False

**Value**

pik matrix of priors: p<sub>nk</sub>, p<sub>ak</sub> and p<sub>ck</sub>

---

pars_init	<i>Initiate parameters alpha, beta and gamma</i>
-----------	--------------------------------------------------

---

**Description**

Initiate parameters alpha, beta and gamma

**Usage**

```
pars_init(  
  covar = FALSE,  
  alpha_mean = -10,  
  alpha_sd = 0.5,  
  beta_shape = 2,  
  beta_scale = 2,  
  gamma_shape = 2,  
  gamma_scale = 2  
)
```

**Arguments**

covar	logical: Should the covariate information be used? default: False
alpha_mean	prior for the mean of alpha
alpha_sd	prior for the standard deviation of alpha
beta_shape	prior for the shape (gamma distribution) of beta
beta_scale	prior for the scale of beta
gamma_shape	prior for the shape (gamma distribution) of gamma
gamma_scale	prior for the scale of gamma

**Value**

params  $\alpha$ ,  $\beta$  and  $\gamma$

---

per.snp.priors      *per.snp.priors*

---

**Description**

Estimate per snp priors

**Usage**

```
per.snp.priors(
  nsnp,
  pa = 3.82e-05,
  pc = 0.00182,
  p1 = NULL,
  p2 = NULL,
  p12 = NULL
)
```

**Arguments**

nsnp	number of SNPs
pa	prior probability that a non-query variant is causally associated with the query trait (cophescan prior), default 3.82e-5
pc	prior probability that the query variant is causally associated with the query trait (cophescan prior), default 1.82e-3 (cophescan prior)
p1	prior probability a SNP is associated with trait 1, (coloc prior), pc derived by using $pc = p12/p1 + p12$ ; use p1, p2, p12 only when pa and pc are unavailable (See vignettes)
p2	prior probability a SNP is associated with trait 2, (coloc prior), pa derived by using $pa = p2$
p12	prior probability a SNP is associated with both traits, (coloc prior), pc derived by using $pc = p12/p1 + p12$

**Value**

priors at the query variant

**Author(s)**

Ichcha Manipur

---

piks *List of priors: pn, pa and pc over all iterations*

---

**Description**

List of priors: pn, pa and pc over all iterations

**Usage**

piks(params, nsnp, covar\_vec, covar = FALSE)

**Arguments**

params	Vector of parameters: $\alpha$ , $\beta$ and $\gamma$
nsnp	number of snps
covar_vec	Vector of the covariate
covar	logical: was the covariate information used? default: False

**Value**

List of priors (len: iterations): pnk, pak and pck

---

plot\_trait\_manhat *Plot region Manhattan for a trait highlighting the queried variant*

---

**Description**

Plot region Manhattan for a trait highlighting the queried variant

**Usage**

plot\_trait\_manhat(trait.dat, querysnpid, alt.snpid = NULL)

**Arguments**

trait.dat	dataset used as input for running cophescan
querysnpid	the id of the causal variant as present in trait.dat\$snp, plotted in red
alt.snpid	the id of the other variants as a vector to be plotted, plotted in blue

**Value**

regional manhattan plot

---

posterior_prob	<i>List of posterior probabilities: Hn, Ha and Hc over all iterations</i>
----------------	---------------------------------------------------------------------------

---

**Description**

List of posterior probabilities: Hn, Ha and Hc over all iterations

**Usage**

```
posterior_prob(params, lbf_mat, nsnps, covar_vec, covar = FALSE)
```

**Arguments**

params	Vector of parameters: $\alpha$ , $\beta$ and $\gamma$
lbf_mat	matrix of log bayes factors: IBF.Ha and IBF.Hc
nsnps	number of snps
covar_vec	Vector of the covariate
covar	logical: was the covariate information used? default: False

**Value**

List of posterior probabilities (len: iterations): Hn, Ha and Hc

---

prepare_plot_data	<i>Prepare data for plotting</i>
-------------------	----------------------------------

---

**Description**

Prepare data for plotting

**Usage**

```
prepare_plot_data(
  multi.dat,
  querysnpid,
  query_trait_names,
  thresh_Ha = 0.5,
  thresh_Hc = 0.5,
  hmp = FALSE,
  cophe.plot = TRUE
)
```

**Arguments**

<code>multi.dat</code>	multi trait cophescan results returned from <code>cophe.multitrait</code> or <code>multitrait.simplify</code>
<code>querysnpid</code>	query variant
<code>query_trait_names</code>	vector of names of the query traits
<code>thresh_Ha</code>	Ha threshold to be displayed
<code>thresh_Hc</code>	Hc threshold to be displayed
<code>hmp</code>	return for heatmap
<code>cophe.plot</code>	default: TRUE, return for <code>cophe.plot</code>

**Value**

plot list

**See Also**

[cophe.plot](#), [cophe.susie](#), [cophe.multitrait](#), [multitrait.simplify](#) default NULL

---

propose

*Proposal distribution*

---

**Description**

Proposal distribution

**Usage**

```
propose(params, propsd = 0.5)
```

**Arguments**

<code>params</code>	Vector of parameters: $\alpha$ , $\beta$ and $\gamma$
<code>propsd</code>	Standard deviation for the proposal

**Value**

vector : proposal

---

run\_metrop\_priors      *Run the hierarchical Metropolis Hastings model to infer priors*

---

### Description

Run the hierarchical Metropolis Hastings model to infer priors

### Usage

```
run_metrop_priors(
  multi.dat,
  covar = FALSE,
  covar_vec = NULL,
  is_covar_categorical = FALSE,
  nits = 10000,
  thin = 1,
  posterior = FALSE,
  avg_pik = TRUE,
  avg_posterior = TRUE,
  pik = FALSE,
  alpha_mean = -10,
  alpha_sd = 0.5,
  beta_shape = 2,
  beta_scale = 2,
  gamma_shape = 2,
  gamma_scale = 2
)
```

### Arguments

multi.dat	matrix of bf values, rows=traits, named columns=("IBF.Ha", "IBF.Hc", "nsnps")
covar	whether to include covariates
covar_vec	vector of covariates
is_covar_categorical	only two categories supported (default=FALSE) - Experimental
nits	number of iterations
thin	burnin
posterior	default: FALSE, estimate posterior probabilities of the hypotheses
avg_pik	default: FALSE, estimate the average of the pik
avg_posterior	default: FALSE, estimate the average of the posterior probabilities of the hypotheses
pik	default: FALSE, inferred prior probabilities
alpha_mean	prior for the mean of alpha
alpha_sd	prior for the standard deviation of alpha

beta_shape	prior for the shape (gamma distribution) of beta
beta_scale	prior for the scale of beta
gamma_shape	prior for the shape (gamma distribution) of gamma
gamma_scale	prior for the scale of gamma

**Value**

List containing the posterior distribution of the parameters alpha, beta, gamma (if covariate included) and the loglikelihood

if avg\_posterior=TRUE matrix with average of all the posterior probabilities of Hn, Ha and Hc

if avg\_pik=TRUE matrix with average of all the priors: pn, pa and pc

data, nits and thin contain the input data, number of iterations and burnin respectively specified for the hierarchical model

---

sample_alpha	<i>sample alpha</i>
--------------	---------------------

---

**Description**

sample alpha

**Usage**

```
sample_alpha(alpha_mean = -10, alpha_sd = 0.5)
```

**Arguments**

alpha_mean	prior for the mean of alpha
alpha_sd	prior for the standard deviation of alpha

**Value**

sample from mnorm for  $\alpha$

sample\_beta

*sample beta*

---

**Description**

sample beta

**Usage**

```
sample_beta(beta_shape = 2, beta_scale = 2)
```

**Arguments**

beta\_shape      prior for the shape (gamma distribution) of beta  
beta\_scale      prior for the scale of beta

**Value**sample from rgamma for  $\beta$ 

---

sample\_gamma

*sample gamma*

---

**Description**

sample gamma

**Usage**

```
sample_gamma(gamma_shape = 2, gamma_scale = 2)
```

**Arguments**

gamma\_shape      prior for the shape (gamma distribution) of gamma  
gamma\_scale      prior for the scale of gamma

**Value**sample from rgamma for  $\gamma$

---

summary.cophe	<i>print the summary of results from cophescan single or susie</i>
---------------	--------------------------------------------------------------------

---

**Description**

print the summary of results from cophescan single or susie

**Usage**

```
## S3 method for class 'cophe'
summary(object, ...)
```

**Arguments**

object	Result from either cophe.susie or cophe.single
...	additional arguments affecting the summary produced.

**Value**

log bayes and posterior probabilities

**See Also**

[cophe.single](#), [cophe.susie](#)

---

target	<i>Target distribution</i>
--------	----------------------------

---

**Description**

Target distribution

**Usage**

```
target(params, lbf_mat, nsnp, covar_vec, covar = FALSE)
```

**Arguments**

params	Vector of parameters: $\alpha$ , $\beta$ and $\gamma$
lbf_mat	matrix of log bayes factors: IBF.Ha and IBF.Hc
nsnp	number of snps
covar_vec	Vector of the covariate
covar	logical: Should the covariate information be used? default: False

**Value**

target

# Index

## \* datasets

    cophe\_multi\_trait\_data, 15

adjust\_priors, 3  
average\_piks, 4  
average\_piks\_list, 5  
average\_posterior\_prob, 5  
average\_posterior\_prob\_list, 6

check\_dataset, 12  
combine.bf, 7  
cophe.hyp.predict, 7  
cophe.multitrait, 8, 8, 16, 29  
cophe.single, 8, 10, 12, 16, 33  
cophe.single.lbf, 11  
cophe.susie, 8, 12, 14, 16, 29, 33  
cophe.susie.lbf, 13  
cophe\_heatmap, 15  
cophe\_multi\_trait\_data, 15  
cophe\_plot, 16, 29  
cophescan (cophescan-package), 3  
cophescan-package, 3

get\_beta, 17  
get\_posterior\_prob, 17

Hc.cutoff.fdr, 18  
hypothesis.priors, 18

logd\_alpha, 19  
logd\_beta, 19  
logd\_gamma, 20  
loglik, 20  
logpost, 21  
logpriors, 21  
logsum, 22  
logsumexp, 23

metrop\_run, 23  
multitrait.simplify, 8, 16, 24, 29

pars2pik, 25  
pars\_init, 25  
per.snp.priors, 26  
pheatmap, 15  
piks, 27  
plot\_trait\_manhat, 27  
posterior\_prob, 28  
prepare\_plot\_data, 28  
propose, 29

run\_metrop\_priors, 30  
runsusie, 13, 14

sample\_alpha, 31  
sample\_beta, 32  
sample\_gamma, 32  
summary.cophe, 33

target, 33