Package ‘coloc’

July 5, 2022

Type Package

Imports data.table, ggplot2, methods, viridis, stats, grDevices, susieR (>= 0.11.33), utils

Suggests knitr, testthat, snpStats, mvtnorm, magrittr, rmarkdown

Title Colocalisation Tests of Two Genetic Traits

Version 5.1.0.1

Date 2021-06-08

Maintainer Chris Wallace `<cew54@cam.ac.uk>`

Description Performs the colocalisation tests described in
Giambartolomei et al (2013) <doi:10.1371/journal.pgen.1004383>,
Wallace (2020) <doi:10.1371/journal.pgen.1008720>,
Wallace (2021) <doi:10.1101/2021.02.23.432421>.

License GPL

LazyLoad yes

VignetteBuilder knitr

RoxygenNote 7.1.1

Encoding UTF-8

URL `https://github.com/chr1swallace/coloc`

BugReports `https://github.com/chr1swallace/coloc/issues`

Collate 'check.R' 'claudia.R' 'coloc-package.R' 'plot.R' 'private.R' 'sensitivity.R' 'split.R' 'susie.R' 'testdata.R' 'zzz.R'

Depends R (>= 2.10)

NeedsCompilation no

Author Chris Wallace [aut, cre],
Claudia Giambartolomei [aut],
Vincent Plagnol [ctb]

Repository CRAN

Date/Publication 2022-07-05 11:36:25 UTC

R topics documented:

`coloc-package` *Colocalisation tests of two genetic traits*

Description

Performs the colocalisation tests described in Plagnol et al (2009) and Wallace et al (2020) and draws some plots.

Author(s)

Chris Wallace cew54@cam.ac.uk

`alpha_to_logbf` *alpha_to_logbf*

Description

convert alpha matrix to log BF matrix

Usage

```
alpha_to_logbf(alpha, pi)
```

Arguments

`alpha`	an L by p+1 matrix of posterior inclusion probabilites
`pi`	per snp prior probability of causality

Value

L by p+1 matrix of log BF

Author(s)

Chris Wallace

`approx.bf.estimates` *Internal function, approx.bf.estimates*

Description

Internal function, approx.bf.estimates

Usage

```
approx.bf.estimates(z, V, type, suffix = NULL, sdY = 1)
```

Arguments

`z`	normal deviate associated with regression coefficient and its variance
`V`	its variance
`type`	"quant" or "cc"
`suffix`	suffix to append to column names of returned data.frame
`sdY`	standard deviation of the trait. If not supplied, will be estimated.

Details

Calculate approximate Bayes Factors using supplied variance of the regression coefficients

Value

data.frame containing lABF and intermediate calculations

Author(s)

Vincent Plagnol, Chris Wallace

`approx.bf.p` *Internal function, approx.bf.p*

Description

Internal function, approx.bf.p

Usage

```
approx.bf.p(p, f, type, N, s, suffix = NULL)
```

Arguments

`p`	p value
`f`	MAF
`type`	"quant" or "cc"
`N`	sample size
`s`	proportion of samples that are cases, ignored if type=="quant"
`suffix`	suffix to append to column names of returned data.frame

Details

Calculate approximate Bayes Factors

Value

data.frame containing lABF and intermediate calculations

Author(s)

Claudia Giambartolomei, Chris Wallace

`bin2lin` *binomial to linear regression conversion*

Description

Convert binomial to linear regression

Usage

```
bin2lin(D, doplot = FALSE)
```

Arguments

`D`	standard format coloc dataset
`doplot`	plot results if TRUE - useful for debugging

Details

Estimate beta and varbeta if a linear regression had been run on a binary outcome, given log OR and their variance + MAF in controls

sets beta = cov(x,y)/var(x) varbeta = (var(y)/var(x) - cov(x,y)^2/var(x)^2)/N

Value

D, with original beta and varbeta in beta.bin, varbeta.bin, and beta and varbeta updated to linear estimates

Author(s)

Chris Wallace

`check_alignment` *check alignment*

Description

check alignment between beta and LD

Usage

```
check_alignment(D, thr = 0.2, do_plot = TRUE)

check.alignment(...)
```

Arguments

`D`	a coloc dataset
`thr`	plot SNP pairs in absolute LD > thr
`do_plot`	if TRUE (default) plot the diagnostic
`...`	arguments passed to check_alignment()

Value

proportion of pairs that are positive

Author(s)

Chris Wallace

`check_dataset` *check_dataset*

Description

Check coloc dataset inputs for errors

Usage

```
check_dataset(d, suffix = "", req = c("snp"), warn.minp = 1e-06)

check.dataset(...)
```

Arguments

`d`	dataset to check
`suffix`	string to identify which dataset (1 or 2)
`req`	names of elements that must be present
`warn.minp`	print warning if no p value < warn.minp
`...`	arguments passed to check_dataset()

Details

Coloc is flexible, requiring perhaps only p values, or z scores, or effect estimates and standard errors, but with this flexibility, also comes difficulties describing exactly the combinations of items required.

pvalues P-values for each SNP in dataset 1

N Number of samples in dataset 1

MAF minor allele frequency of the variants

beta regression coefficient for each SNP from dataset 1

varbeta variance of beta

type the type of data in dataset 1 - either "quant" or "cc" to denote quantitative or case-control

s for a case control dataset, the proportion of samples in dataset 1 that are cases

sdY for a quantitative trait, the population standard deviation of the trait. if not given, it can be estimated from the vectors of varbeta and MAF

snp a character vector of snp ids, optional. If present, it will be used to merge dataset1 and dataset2. Otherwise, the function assumes dataset1 and dataset2 contain results for the same SNPs in the same order.

Some of these items may be missing, but you must always give `type`.

Then scalars describing the samples used:

`N` If `type == "cc"`, `s`. If `type == "quant"` and `sdY` known, `sdY`. If `sdY` is unknown, it will be approximated, and this will require `beta`, `varbeta`, `N`, `MAF`.

Then, if not already covered above, the summary statistics describing the results

preferably `beta`, `varbeta`

alternatively `pvalues`, `MAF`

`check_dataset` call stop() unless a series of expectations on dataset input format are met

This is a helper function for use by other coloc functions, but you can use it directly to check the format of a dataset to be supplied to coloc.abf(), coloc.signals(), finemap.abf(), or finemap.signals().

Value

NULL if no errors found

Author(s)

Chris Wallace

`coloc.abf` *Fully Bayesian colocalisation analysis using Bayes Factors*

Description

Bayesian colocalisation analysis

Usage

```
coloc.abf(dataset1, dataset2, MAF = NULL, p1 = 1e-04, p2 = 1e-04, p12 = 1e-05)
```

Arguments

`dataset1` a list with specifically named elements defining the dataset to be analysed. See `check_dataset` for details.

`dataset2` as above, for dataset 2

`MAF` Common minor allele frequency vector to be used for both dataset1 and dataset2, a shorthand for supplying the same vector as parts of both datasets

`p1` prior probability a SNP is associated with trait 1, default 1e-4

`p2` prior probability a SNP is associated with trait 2, default 1e-4

`p12` prior probability a SNP is associated with both traits, default 1e-5

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.

Value

a list of two `data.frames`:

- summary is a vector giving the number of SNPs analysed, and the posterior probabilities of H0 (no causal variant), H1 (causal variant for trait 1 only), H2 (causal variant for trait 2 only), H3 (two distinct causal variants) and H4 (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.H4 of the SNP being causal for the shared signal *if* H4 is true. This is only relevant if the posterior support for H4 in summary is convincing.

Author(s)

Claudia Giambartolomei, Chris Wallace

`coloc.bf_bf` *Coloc data through Bayes factors*

Description

Colocalise two datasets represented by Bayes factors

Usage

```
coloc.bf_bf(
  bf1,
  bf2,
  p1 = 1e-04,
  p2 = 1e-04,
  p12 = 5e-06,
  overlap.min = 0.5,
  trim_by_posterior = TRUE
)
```

Arguments

`bf1` named vector of BF, or matrix of BF with colnames (cols=snps, rows=signals)

`bf2` named vector of BF, or matrix of BF with colnames (cols=snps, rows=signals)

`p1` prior probability a SNP is associated with trait 1, default 1e-4

`p2` prior probability a SNP is associated with trait 2, default 1e-4

`p12` prior probability a SNP is associated with both traits, default 1e-5

`overlap.min` see trim_by_posterior

`trim_by_posterior` it is important that the signals to be colocalised are covered by adequate numbers of snps in both datasets. If TRUE, signals for which snps in common do not capture least overlap.min proportion of their posteriors support are dropped and colocalisation not attempted.

Details

This is the workhorse behind many coloc functions

Value

coloc.signals style result

Author(s)

Chris Wallace

`coloc.detail` *Bayesian colocalisation analysis with detailed output*

Description

Bayesian colocalisation analysis, detailed output

Usage

```
coloc.detail(
  dataset1,
  dataset2,
  MAF = NULL,
  p1 = 1e-04,
  p2 = 1e-04,
  p12 = 1e-05
)
```

Arguments

`dataset1` a list with specifically named elements defining the dataset to be analysed. See `check_dataset` for details.

`dataset2` as above, for dataset 2

`MAF` Common minor allele frequency vector to be used for both dataset1 and dataset2, a shorthand for supplying the same vector as parts of both datasets

`p1` prior probability a SNP is associated with trait 1, default 1e-4

`p2` prior probability a SNP is associated with trait 2, default 1e-4

`p12` prior probability a SNP is associated with both traits, default 1e-5

Details

This function replicates coloc.abf, but outputs more detail for further processing using coloc.process

Intended to be called internally by coloc.signals

Value

a list of three `data.tables`s:

- summary is a vector giving the number of SNPs analysed, and the posterior probabilities of H0 (no causal variant), H1 (causal variant for trait 1 only), H2 (causal variant for trait 2 only), H3 (two distinct causal variants) and H4 (one common causal variant)
- df is an annotated version of the input data containing log Approximate Bayes Factors and intermediate calculations, and the posterior probability SNP.PP.H4 of the SNP being causal for the shared signal
- df3 is the same for all 2 SNP H3 models

Author(s)

Chris Wallace

See Also

`coloc.process`, `coloc.abf`

`coloc.process` *Post process a coloc.details result using masking*

Description

Internal helper function

Usage

```
coloc.process(
  obj,
  hits1 = NULL,
  hits2 = NULL,
  LD = NULL,
  r2thr = 0.01,
  p1 = 1e-04,
  p2 = 1e-04,
  p12 = 1e-06,
  LD1 = LD,
  LD2 = LD,
  mode = c("iterative", "allbutone")
)
```

Arguments

`obj`	object returned by coloc.detail()
`hits1`	lead snps for trait 1. If length > 1, will use masking
`hits2`	lead snps for trait 2. If length > 1, will use masking
`LD`	named LD matrix (for masking)
`r2thr`	r2 threshold at which to mask
`p1`	prior probability a SNP is associated with trait 1, default 1e-4
`p2`	prior probability a SNP is associated with trait 2, default 1e-4
`p12`	prior probability a SNP is associated with both traits, default 1e-5
`LD1`	named LD matrix (for masking) for trait 1 only
`LD2`	named LD matrix (for masking) for trait 2 only
`mode`	either "iterative" (default) - successively condition on signals or "allbutone" - find all putative signals and condition on all but one of them in each analysis

Value

data.table of coloc results

Author(s)

Chris Wallace

coloc.signals *Coloc with multiple signals per trait*

Description

New coloc function, builds on coloc.abf() by allowing for multiple independent causal variants per trait through conditioning or masking.

Usage

```
coloc.signals(
  dataset1,
  dataset2,
  MAF = NULL,
  LD = NULL,
  method = c("single", "cond", "mask"),
  mode = c("iterative", "allbutone"),
  p1 = 1e-04,
  p2 = 1e-04,
  p12 = NULL,
  maxhits = 3,
  r2thr = 0.01,
  pthr = 1e-06
)
```

Arguments

`dataset1` a list with specifically named elements defining the dataset to be analysed. See `check_dataset` for details.

`dataset2` as above, for dataset 2

`MAF` Common minor allele frequency vector to be used for both dataset1 and dataset2, a shorthand for supplying the same vector as parts of both datasets

`LD` required if method="cond". matrix of genotype *correlation* (ie r, not r^2) between SNPs. If dataset1 and dataset2 may have different LD, you can instead add LD=LD1 to the list of dataset1 and a different LD matrix for dataset2

`method` default "" means do no conditioning, should return similar to coloc.abf. if method="cond", then use conditioning to coloc multiple signals. if method="mask", use masking to coloc multiple signals. if different datasets need different methods (eg LD is only available for one of them) you can set method on a per-dataset basis by adding method="..." to the list for that dataset.

`mode` "iterative" or "allbutone". Easiest understood with an example. Suppose there are 3 signal SNPs detected for trait 1, A, B, C and only one for trait 2, D.

```
Under "iterative" mode, 3 coloc will be performed:
* trait 1 - trait 2
* trait 1 conditioned on A - trait 2
* trait 1 conditioned on A+B - trait 2

Under "allbutone" mode, they would be
* trait 1 conditioned on B+C - trait 2
* trait 1 conditioned on A+C - trait 2
* trait 1 conditioned on A+B - trait 2

Only iterative mode is supported for method="mask".

The allbutone mode is optimal if the signals are known with
certainty (which they never are), because it allows each
signal to be tested without influence of the others.  When
there is uncertainty, it may make sense to use iterative mode,
because the strongest signals aren't affected by conditioning
incorrectly on weaker secondary and less certain signals.
```

`p1` prior probability a SNP is associated with trait 1, default 1e-4

`p2` prior probability a SNP is associated with trait 2, default 1e-4

`p12` prior probability a SNP is associated with both traits, default 1e-5

`maxhits` maximum number of levels to condition/mask

`r2thr` if masking, the threshold on r2 should be used to call two signals independent. our experience is that this needs to be set low to avoid double calling the same strong signal.

`pthr` if masking or conditioning, what p value threshold to call a secondary hit "significant"

Value

data.table of coloc results, one row per pair of lead snps detected in each dataset

Author(s)

Chris Wallace

`coloc.susie` *run coloc using susie to detect separate signals*

Description

colocalisation with multiple causal variants via SuSiE

Usage

```
coloc.susie(
  dataset1,
  dataset2,
  back_calculate_lbf = FALSE,
  susie.args = list(),
  ...
)
```

Arguments

`dataset1` *either* a coloc-style input dataset (see check_dataset), or the result of running runsusie on such a dataset

`dataset2` *either* a coloc-style input dataset (see check_dataset), or the result of running runsusie on such a dataset

`back_calculate_lbf`
by default, use the log Bayes factors returned by susie_rss. It is also possible to back-calculate these from the posterior probabilities. It is not advised to set this to TRUE, the option exists really for testing purposes only.

`susie.args` a named list of additional arguments to be passed to runsusie

`...` other arguments passed to coloc.bf_bf, in particular prior values for causal association with one trait (p1, p2) or both (p12)

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 H4 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)

Chris Wallace

`coloc.susie_bf` *run coloc using susie to detect separate signals*

Description

coloc for susie output + a separate BF matrix

Usage

```
coloc.susie_bf(dataset1, bf2, p1 = 1e-04, p2 = 1e-04, p12 = 5e-06, ...)
```

Arguments

`dataset1`	a list with specifically named elements defining the dataset to be analysed. See `check_dataset` for details.
`bf2`	named vector of BF, names are snp ids and will be matched to column names of susie object's alpha
`p1`	prior probability a SNP is associated with trait 1, default 1e-4
`p2`	prior probability a SNP is associated with trait 2, default 1e-4
`p12`	prior probability a SNP is associated with both traits, default 1e-5
`...`	other arguments passed to coloc.bf_bf, in particular prior values for causal association with one trait (p1, p2) or both (p12)

Value

coloc.signals style result

Author(s)

Chris Wallace

`coloc_test_data` *Simulated data to use in testing and vignettes in the coloc package*

Description

Simulated data to use in testing and vignettes in the coloc package

Usage

```
data(coloc_test_data)
```

Format

A four of two coloc-style datasets. Elements D1 and D2 have a single shared causal variant, and 50 SNPs. Elements D3 and D4 have 100 SNPs, one shared causal variant, and one variant unique to D3. Use these as examples of what a coloc-style dataset for a quantitative trait should look like.

Examples

```
data(coloc_test_data)
names(coloc_test_data)
str(coloc_test_data$D1)
check_dataset(coloc_test_data$D1) # should return NULL if data structure is ok
```

`combine.abf` *combine.abf*

Description

Internal function, calculate posterior probabilities for configurations, given logABFs for each SNP and prior probs

Usage

```
combine.abf(l1, l2, p1, p2, p12)
```

Arguments

`l1`	merged.df$lABF.df1
`l2`	merged.df$lABF.df2
`p1`	prior probability a SNP is associated with trait 1, default 1e-4
`p2`	prior probability a SNP is associated with trait 2, default 1e-4
`p12`	prior probability a SNP is associated with both traits, default 1e-5

Value

named numeric vector of posterior probabilities

Author(s)

Claudia Giambartolomei, Chris Wallace

`estgeno.1.ctl` *estgeno1*

Description

Estimate single snp frequency distibutions

Usage

```
estgeno.1.ctl(f)

estgeno.1.cse(G0, b)
```

Arguments

`f`	MAF
`G0`	single snp frequency in controls (vector of length 3) - obtained from estgeno.1.ctl
`b`	log odds ratio

Value

relative frequency of genotypes 0, 1, 2

Author(s)

Chris Wallace

See Also

estgeno2

`est_cond` *generate conditional summary stats*

Description

Internal helper function for est_all_cond

Usage

```
est_cond(x, LD, YY, sigsnps, xtx = NULL)
```

Arguments

`x`	coloc dataset
`LD`	named matrix of r
`YY`	sum((Y-Ybar)^2)
`sigsnps`	names of snps to jointly condition on
`xtx`	optional, matrix X'X where X is the genotype matrix. If not available, will be estimated from LD, MAF, beta and sample size (the last three should be part of the coloc dataset)

Value

data.table giving snp, beta and varbeta on remaining snps after conditioning

Author(s)

Chris Wallace

`find.best.signal` *Pick out snp with most extreme Z score*

Description

Internal helper function

Usage

```
find.best.signal(D)
```

Arguments

`D`	standard format coloc dataset

Value

z at most significant snp, named by that snp id

Author(s)

Chris Wallace

`finemap.abf` *Bayesian finemapping analysis*

Description

Bayesian finemapping analysis

Usage

```
finemap.abf(dataset, p1 = 1e-04)
```

Arguments

`dataset`	a list with specifically named elements defining the dataset to be analysed. See `check_dataset` for details.
`p1`	prior probability a SNP is associated with the trait 1, default 1e-4

Details

This function calculates posterior probabilities of different causal variant for a single 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.

Value

a `data.frame`:

- an annotated version of the input data containing log Approximate Bayes Factors and intermediate calculations, and the posterior probability of the SNP being causal

Author(s)

Chris Wallace

`finemap.signals` *Finemap multiple signals in a single dataset*

Description

This is an analogue to finemap.abf, adapted to find multiple signals where they exist, via conditioning or masking - ie a stepwise procedure

Usage

```
finemap.signals(
  D,
  LD = D$LD,
  method = c("single", "mask", "cond"),
  r2thr = 0.01,
  sigsnps = NULL,
  pthr = 1e-06,
  maxhits = 3
)
```

Arguments

`D` list of summary stats for a single disease, see check_dataset

`LD` matrix of signed r values (not rsq!) giving correlation between SNPs

`method` if method="cond", then use conditioning to coloc multiple signals. The default is mask - this is less powerful, but safer because it does not assume that the LD matrix is properly allelically aligned to estimated effect

`r2thr` if mask==TRUE, all snps will be masked with r2 > r2thr with any sigsnps. Otherwise ignored

`sigsnps` SNPs already deemed significant, to condition on or mask, expressed as a numeric vector, whose *names* are the snp names

`pthr` when p > pthr, stop successive searching

`maxhits` maximum depth of conditioning. procedure will stop if p > pthr OR abs(z)<zthr OR maxhits hits have been found.

`mask` use masking if TRUE, otherwise conditioning. defaults to TRUE

Value

list of successively significant fine mapped SNPs, named by the SNPs

Author(s)

Chris Wallace

`logbf_to_pp` *logbf 2 pp*

Description

convert logbf matrix to PP matrix

Usage

```
logbf_to_pp(bf, pi, last_is_null)
```

Arguments

`bf` an L by p or p+1 matrix of log Bayes factors

`pi` *either* a scalar representing the prior probability for any snp to be causal, *or* a full vector of per snp / null prior probabilities

`last_is_null` TRUE if last value of the bf vector or last column of a bf matrix relates to the null hypothesis of no association. This is standard for SuSiE results, but may not be for BF constructed in other ways.

Value

matrix of posterior probabilies, same dimensions as bf

Author(s)

Chris Wallace

`logdiff` *logdiff*

Description

Internal function, logdiff

Usage

```
logdiff(x, y)
```

Arguments

`x`	numeric
`y`	numeric

Details

This function calculates the log of the difference of the exponentiated logs taking out the max, i.e. insuring that the difference is not negative

Value

max(x) + log(exp(x - max(x,y)) - exp(y-max(x,y)))

Author(s)

Chris Wallace

`logsum` *logsum*

Description

Internal function, logsum

Usage

```
logsum(x)
```

Arguments

`x`	numeric vector

Details

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

Value

max(x) + log(sum(exp(x - max(x))))

Author(s)

Claudia Giambartolomei

---

map_cond *find the next most significant SNP, conditioning on a list of sigsnps*

Description

Internal helper function for finemap.signals

Usage

```
map_cond(D, LD, YY, sigsnps = NULL)
```

Arguments

`D`	dataset in standard coloc format
`LD`	named matrix of r
`YY`	sum(y^2)
`sigsnps`	names of snps to mask

Value

named numeric - Z score named by snp

Author(s)

Chris Wallace

`map_mask` *find the next most significant SNP, masking a list of sigsnps*

Description

Internal helper function for finemap.signals

Usage

```
map_mask(D, LD, r2thr = 0.01, sigsnps = NULL)
```

Arguments

`D`	dataset in standard coloc format
`LD`	named matrix of r
`r2thr`	mask all snps with r2 > r2thr with any in sigsnps
`sigsnps`	names of snps to mask

Value

named numeric - Z score named by snp

Author(s)

Chris Wallace

`plot.coloc_abf` *plot a coloc_abf object*

Description

plot a coloc_abf object

Usage

```
## S3 method for class 'coloc_abf'
plot(x, ...)
```

Arguments

`x`	coloc_abf object to be plotted
`...`	other arguments

Value

ggplot object

Author(s)

Chris Wallace

---

plot_dataset *plot a coloc dataset*

Description

Plot a coloc structured dataset

Usage

```
plot_dataset(
  d,
  susie_obj = NULL,
  alty = NULL,
  ylab = "-log10(p)",
  show_legend = TRUE,
  color = c("dodgerblue2", "green4", "#6A3D9A", "#FF7F00", "gold1", "skyblue2",
    "#FB9A99", "palegreen2", "#CAB2D6", "#FDBF6F", "gray70", "khaki2", "maroon",
   "orchid1", "deeppink1", "blue1", "steelblue4", "darkturquoise", "green1", "yellow4",
    "yellow3", "darkorange4", "brown"),
  ...
)
```

Arguments

`d`	a coloc dataset
`susie_obj`	optional, the output of a call to runsusie()
`alty`	default is to plot a standard manhattan. If you wish to plot a different y value, pass it here. You may also want to change ylab to describe what you are plotting.
`ylab`	label for y axis, default is -log10(p) and assumes you are plotting a manhattan
`show_legend`	optional, show the legend or not. default is TRUE
`color`	optional, specify the colours to use for each credible set when susie_obj is supplied. Default is shamelessly copied from susieR::susie_plot() so that colours will match
`...`	other arguments passed to the base graphics plot() function

Author(s)

Chris Wallace

`print.coloc_abf` *print.coloc_abf*

Description

Print summary of a coloc.abf run

Usage

```
## S3 method for class 'coloc_abf'
print(x, ...)
```

Arguments

`x`	object of class `coloc_abf` returned by coloc.abf() or coloc.signals()
`...`	optional arguments: "trait1" name of trait 1, "trait2" name of trait 2

Value

x, invisibly

Author(s)

Chris Wallace

`process.dataset` *process.dataset*

Description

Internal function, process each dataset list for coloc.abf

Usage

```
process.dataset(d, suffix)
```

Arguments

`d`	list
`suffix`	"df1" or "df2"

Value

data.frame with log(abf) or log(bf)

Author(s)

Chris Wallace

runsusie *Run susie on a single coloc-structured dataset*

Description

run susie_rss storing some additional information for coloc

Usage

```
runsusie(
  d,
  suffix = 1,
  p = NULL,
  trimz = NULL,
  r2.prune = NULL,
  maxit = 100,
  repeat_until_convergence = TRUE,
  s_init = NULL,
  ...
)
```

Arguments

`d` coloc dataset, must include LD (signed correlation matrix)

`suffix` suffix label that will be printed with any error messages

`p` prior probability a snp is causal (equivalent to p1 or p2 in coloc.abf). By default, this is set to NULL, upon which we will set a small null_weight to pass to susie_rss() (see vignette a06 for details why). You can override this by setting p as you would p1 or p2 in a coloc function, but note that you may miss some true signals that way. Also note that neither of these options correspond to the susie_rss() defaults, because our goal here is not fine mapping alone.

`trimz` used to trim datasets for development purposes

`r2.prune` sometimes SuSiE can return multiple signals in high LD. if you set r2.prune to a value between 0 and 1, sets with index SNPs with LD greater than r2.prune

`maxit` maximum number of iterations for the first run of susie_rss(). If susie_rss() does not report convergence, runs will be extended assuming repeat_until_convergence=TRUE. Most users will not need to change this default.

`repeat_until_convergence`
keep running until susie_rss() indicates convergence. Default TRUE. If FALSE, susie_rss() will run with maxit iterations, and if not converged, runsusie() will error. Most users will not need to change this default.

`s_init` used internally to extend runs that haven't converged. don't use.

`...` arguments passed to susie_rss. In particular, if you want to match some coloc defaults, set

- prior_variance=0.2^2 (if a case-control trait) or (0.15/sd(Y))^2 if a quantitative trait
- estimate_prior_variance=FALSE

otherwise susie_rss will estimate the prior variance itself

Value

results of a susie_rss run, with some added dimnames

Author(s)

Chris Wallace

Examples

```
library(coloc)
data(coloc_test_data)
result=runsusie(coloc_test_data$D1)
summary(result)
```

---

sdY.est *Estimate trait variance, internal function*

Description

Estimate trait standard deviation given vectors of variance of coefficients, MAF and sample size

Usage

```
sdY.est(vbeta, maf, n)
```

Arguments

`vbeta`	vector of variance of coefficients
`maf`	vector of MAF (same length as vbeta)
`n`	sample size

Details

Estimate is based on var(beta-hat) = var(Y) / (n * var(X)) var(X) = $2maf$(1-maf) so we can estimate var(Y) by regressing n*var(X) against 1/var(beta)

Value

estimated standard deviation of Y

Author(s)

Chris Wallace

`sensitivity` *Prior sensitivity for coloc*

Description

Shows how prior and posterior per-hypothesis probabilities change as a function of p12

Usage

```
sensitivity(
  obj,
  rule = "",
  dataset1 = NULL,
  dataset2 = NULL,
  npoints = 100,
  doplot = TRUE,
  plot.manhattans = TRUE,
  preserve.par = FALSE,
  row = 1
)
```

Arguments

`obj` output of coloc.detail or coloc.process

`rule` a decision rule. This states what values of posterior probabilities "pass" some threshold. This is a string which will be parsed and evaluated, better explained by examples. "H4 > 0.5" says post prob of H4 > 0.5 is a pass. "H4 > 0.9 & H4/H3 > 3" says post prob of H4 must be > 0.9 AND it must be at least 3 times the post prob of H3."

`dataset1` optional the dataset1 used to run SuSiE. This will be used to make a Manhattan plot if plot.manhattans=TRUE.

`dataset2` optional the dataset2 used to run SuSiE. This will be used to make a Manhattan plot if plot.manhattans=TRUE.

`npoints` the number of points over which to evaluate the prior values for p12, equally spaced on a log scale between p1*p2 and min(p1,p2) - these are logical limits on p12, but not scientifically sensible values.

`doplot` draw the plot. set to FALSE if you want to just evaluate the prior and posterior matrices and work with them yourself

`plot.manhattans` if TRUE, show Manhattans of input data

`preserve.par` if TRUE, do not change par() of current graphics device - this is to allow sensitivity plots to be incoporated into a larger set of plots, or to be plot one per page on a pdf, forexample

`row` when coloc.signals() has been used and multiple rows are returned in the coloc summary, which row to plot

Details

Function is called mainly for plotting side effect. It draws two plots, showing how prior and posterior probabilities of each coloc hypothesis change with changing p12. A decision rule sets the values of the posterior probabilities considered acceptable, and is used to shade in green the region of the plot for which the p12 prior would give and acceptable result. The user is encouraged to consider carefully whether some prior values shown within the green shaded region are sensible before accepting the hypothesis. If no shading is shown, then no priors give rise to an accepted result.

Value

list of 3: prior matrix, posterior matrix, and a pass/fail indicator (returned invisibly)

Author(s)

Chris Wallace

`Var.data` *Var.data*

Description

variance of MLE of beta for quantitative trait, assuming var(y)=1

Usage

```
Var.data(f, N)
```

Arguments

`f`	minor allele freq
`N`	sample number

Details

Internal function

Value

variance of MLE beta

Author(s)

Claudia Giambartolomei

`Var.data.cc` *Var.data*

Description

variance of MLE of beta for case-control

Usage

```
Var.data.cc(f, N, s)
```

Arguments

`f`	minor allele freq
`N`	sample number
`s`	???

Details

Internal function

Value

variance of MLE beta

Author(s)

Claudia Giambartolomei

Index