

# Package: corrcoverage (via r-universe)

September 8, 2024

**Type** Package

**Title** Correcting the Coverage of Credible Sets from Bayesian Genetic Fine Mapping

**Version** 1.2.0

**Maintainer** Anna Hutchinson <anna.hutchinson@mrc-bsu.cam.ac.uk>

**Description** Using a computationally efficient method, the package can be used to find the corrected coverage estimate of a credible set of putative causal variants from Bayesian genetic fine-mapping. The package can also be used to obtain a corrected credible set if required; that is, the smallest set of variants required such that the corrected coverage estimate of the resultant credible set is within some user defined accuracy of the desired coverage. Maller et al. (2012) <[doi:10.1038/ng.2435](https://doi.org/10.1038/ng.2435)>, Wakefield (2009) <[doi:10.1002/gepi.20359](https://doi.org/10.1002/gepi.20359)>, Fortune and Wallace (2018) <[doi:10.1093/bioinformatics/bty898](https://doi.org/10.1093/bioinformatics/bty898)>.

**URL** <https://annahutch.github.io/corrcoverage>

**License** MIT + file LICENSE

**BugReports** <https://github.com/annahutch/corrcoverage/issues>

**OS\_type** unix

**Encoding** UTF-8

**LazyData** true

**Suggests** covr, dplyr, knitr, mvtnorm, rmarkdown, testthat, pkgdown

**VignetteBuilder** knitr

**RoxygenNote** 6.1.1

**LinkingTo** Rcpp, RcppArmadillo

**Imports** data.table, magrittr, stats, matrixStats, Rcpp

**SystemRequirements** C++11

**Repository** <https://annahutch.r-universe.dev>

**RemoteUrl** <https://github.com/annahutch/corrcoverage>

**RemoteRef** HEAD

**RemoteSha** 3b610d1450c7936db6ab2664de37543165ac6c0e

## Contents

.zj_abf . . . . .	2
.zj_pp . . . . .	3
approx.bf.p . . . . .	4
bf_func . . . . .	5
cor2 . . . . .	6
corrcov . . . . .	7
corrcov_bhat . . . . .	8
corrcov_CI . . . . .	9
corrcov_CI_bhat . . . . .	11
corrcov_nvar . . . . .	12
corrcov_nvar_bhat . . . . .	14
corrected_cov . . . . .	15
corrected_cs . . . . .	17
corrected_cs_bhat . . . . .	18
credset . . . . .	20
credsetC . . . . .	21
credsetmat . . . . .	22
est_mu . . . . .	22
est_mu_bhat . . . . .	23
logsum . . . . .	24
logsum_matrix . . . . .	25
ppfunc . . . . .	25
ppfunc.mat . . . . .	26
prop_cov . . . . .	28
pvals_pp . . . . .	28
Var.data.cc . . . . .	29
z0_pp . . . . .	30
zj_pp . . . . .	31
zj_pp_arma . . . . .	33
z_sim . . . . .	33
<b>Index</b>	<b>35</b>

---

.zj\_abf

*Internal function: Simulate nrep ABFs from joint Z-score vector*

---

### Description

Does not include posterior probabilities for null model

**Usage**

.zj\_abf(Zj, int.Sigma, int.nrep, int.ERR, int.r)

**Arguments**

Zj	joint z vector
int.Sigma	internal sigma
int.nrep	internal nrep
int.ERR	internal error matrix
int.r	internal r

**Value**

Matrix of simulated ABFs, one simulation per row

---

.zj\_pp *Simulate posterior probabilities of causality from joint Z-score vector*

---

**Description**

Internal function: Simulate nrep posterior probabilities of causality from joint Z-score vector

**Usage**

.zj\_pp(Zj, int.Sigma, int.nrep, int.ERR, int.r)

**Arguments**

Zj	joint z vector
int.Sigma	internal sigma
int.nrep	internal nrep
int.ERR	internal error matrix
int.r	internal r

**Details**

Does not include posterior probabilities for null model

**Value**

Matrix of simulated posterior probabilities of causality, one simulation per row

---

 approx.bf.p

*Find approx. Bayes factors (ABFs)*


---

### Description

Wakefield's log asymptotic Bayes factor (LABF) with prior standard deviation of effect size as a parameter

### Usage

```
approx.bf.p(pvals, f, type, N, s, W = 0.2)
```

### Arguments

pvals	P-values
f	Minor allele frequencies
type	Type of experiment ('quant' or 'cc')
N	Total sample size
s	Proportion of cases (N1/N0+N1), ignored if type=='quant'
W	Prior for the standard deviation of the effect size parameter beta (W=0.2 default)

### Details

([Wakefield et al. 2009](<https://onlinelibrary.wiley.com/doi/abs/10.1002/gepi.20359>) This function converts p-values to log ABFs, also reporting intermediate calculations

### Value

data.frame containing LABF and intermediate calculations

### Examples

```
set.seed(1)
nsnps = 100
N0 = 5000
N1 = 5000
z_scores <- rnorm(nsnps, 0, 3)
p_values <- 2 * pnorm( - abs ( z_scores ) )

## generate example LD matrix and MAFs
library(mvtnorm)
nsamples = 1000

simx <- function(nsnps, nsamples, S, maf=0.1) {
  mu <- rep(0,nsnps)
  rawvars <- rmvnorm(n=nsamples, mean=mu, sigma=S)
  pvars <- pnorm(rawvars)
```

```

      x <- qbinom(1-pvars, 1, maf)
    }

    S <- (1 - (abs(outer(1:nsnps,1:nsnps,`-`)))/nsnps))^4
    X <- simx(nsnps,nsamples,S)
    maf <- colMeans(X)

    approx.bf.p(pvals = p_values, f = maf, type = "cc", N = N0+N1, s = N1/(N0+N1))

```

---

bf\_func

*Calculate ABFs from Z scores*


---

### Description

Calculate ABFs from Z scores

### Usage

```
bf_func(z, V, W = 0.2)
```

### Arguments

z	Vector of Z-scores
V	Variance of the estimated effect size
W	Prior for the standard deviation of the effect size parameter, beta (default 0.2)

### Details

Note, z and V should both be vectors or both be matrices

### Value

ABFs

### Examples

```

set.seed(1)
nsnps = 100
N0 = 5000
N1 = 5000
z_scores <- rnorm(nsnps, 0, 3)

## generate example LD matrix and MAFs
library(mvtnorm)
nsamples = 1000

simx <- function(nsnps, nsamples, S, maf=0.1) {
  mu <- rep(0,nsnps)

```

```
rawvars <- rmvnorm(n=nsamples, mean=mu, sigma=S)
pvars <- pnorm(rawvars)
x <- qbinom(1-pvars, 1, maf)
}

S <- (1 - (abs(outer(1:nsnps,1:nsnps,`-`))/nsnps))^4
X <- simx(nsnps,nsamples,S)
maf <- colMeans(X)

varbeta = Var.data.cc(f = maf, N = N0 + N1, s = 0.5)

bf_func(z_scores, V = varbeta)
```

---

cor2

*Correlation matrix of SNPs*

---

### **Description**

Correlation matrix of SNPs

### **Usage**

```
cor2(x)
```

### **Arguments**

x                      Phased haplotype matrix, rows as samples and columns as SNPs

### **Details**

Quick function to find a correlation matrix

### **Value**

Correlation matrix

### **Author(s)**

Chris Wallace

---

corrcov *Corrected coverage estimate using Z-scores and MAFs*

---

**Description**

Corrected coverage estimate using Z-scores and mafs

**Usage**

```
corrcov(z, f, N0, N1, Sigma, thr, W = 0.2, nrep = 1000,  
        pp0min = 0.001)
```

**Arguments**

z	Marginal Z-scores
f	Minor allele frequencies
N0	Number of controls
N1	Number of cases
Sigma	SNP correlation matrix
thr	Minimum threshold for fine-mapping experiment
W	Prior for the standard deviation of the effect size parameter, beta (default 0.2)
nrep	The number of simulated posterior probability systems to consider for the corrected coverage estimate (default 1000)
pp0min	Only average over SNPs with $pp0 > pp0min$

**Details**

This function only requires the marginal summary statistics from GWAS

**Value**

Corrected coverage estimate

**Author(s)**

Anna Hutchinson

**Examples**

```
set.seed(1)  
nsnps = 100  
N0 = 5000  
N1 = 5000  
z_scores <- rnorm(nsnps, 0, 3) # simulate a vector of Z-scores  
  
## generate example LD matrix
```

```

library(mvtnorm)
nsamples = 1000

simx <- function(nsnps, nsamples, S, maf=0.1) {
  mu <- rep(0,nsnps)
  rawvars <- rmvnorm(n=nsamples, mean=mu, sigma=S)
  pvars <- pnorm(rawvars)
  x <- qbinom(1-pvars, 1, maf)
}

S <- (1 - (abs(outer(1:nsnps,1:nsnps,`-`)))/nsnps))^4
X <- simx(nsnps,nsamples,S)
LD <- cor2(X)
maf <- colMeans(X)

corrcov(z = z_scores, f = maf, N0, N1, Sigma = LD, thr = 0.95)

```

---

corrcov_bhat	<i>Corrected coverage estimate using estimated effect sizes and their standard errors</i>
--------------	---

---

### Description

Corrected coverage estimate using estimated effect sizes and their standard errors

### Usage

```
corrcov_bhat(bhat, V, N0, N1, Sigma, thr, W = 0.2, nrep = 1000,
pp0min = 0.001)
```

### Arguments

bhat	Estimated effect sizes from single-SNP logistic regressions
V	Variance of estimated effect sizes
N0	Number of controls
N1	Number of cases
Sigma	SNP correlation matrix
thr	Minimum threshold for fine-mapping experiment
W	Prior for the standard deviation of the effect size parameter, beta (default 0.2)
nrep	The number of simulated posterior probability systems to consider for the corrected coverage estimate (default 1000)
pp0min	Only average over SNPs with $pp0 > pp0min$

### Details

This function only requires the marginal summary statistics from GWAS



**Value**

Corrected coverage estimate

**Author(s)**

Anna Hutchinson

**Examples**

```

set.seed(1)
nsnps <- 100
N0 <- 1000 # number of controls
N1 <- 1000 # number of cases

## generate example LD matrix
library(mvtnorm)
nsamples = 1000

simx <- function(nsnps, nsamples, S, maf=0.1) {
  mu <- rep(0,nsnps)
  rawvars <- rmvnorm(n=nsamples, mean=mu, sigma=S)
  pvars <- pnorm(rawvars)
  x <- qbinom(1-pvars, 1, maf)
}

S <- (1 - (abs(outer(1:nsnps,1:nsnps,`~`)))/nsnps))^4
X <- simx(nsnps,nsamples,S)
LD <- cor2(X)
maf <- colMeans(X)

varbeta <- Var.data.cc(f = maf, N = N0 + N1, s = N1/(N0+N1))

bhats = rnorm(nsnps, 0, 0.2) # log OR

corr cov_bhat(bhat = bhats, V = varbeta, N0, N1, Sigma = LD, thr = 0.95)

```

---

corr cov\_CI

*Confidence interval for corrected coverage estimate using Z-scores and MAFs*

---

**Description**

Obtain confidence interval for corrected coverage estimate using Z-scores and mafs

**Usage**

```

corr cov_CI(z, f, N0, N1, Sigma, thr, W = 0.2, nrep = 1000, CI = 0.95,
pp0min = 0.001)

```

**Arguments**

z	Marginal Z-scores
f	Minor allele frequencies
N0	Number of controls
N1	Number of cases
Sigma	SNP correlation matrix
thr	Minimum threshold for fine-mapping experiment
W	Prior for the standard deviation of the effect size parameter, beta (default 0.2)
nrep	The number of simulated posterior probability systems to consider for the corrected coverage estimate (nrep = 1000 default)
CI	The size of the confidence interval (as a decimal)
pp0min	Only average over SNPs with pp0 > pp0min

**Value**

CI for corrected coverage estimate

**Author(s)**

Anna Hutchinson

**Examples**

```
# this is a long running example
set.seed(1)
nsnps = 100
N0 = 5000
N1 = 5000
z_scores <- rnorm(nsnps, 0, 3) # simulate a vector of Z-scores

## generate example LD matrix
library(mvtnorm)
nsamples = 1000

simx <- function(nsnps, nsamples, S, maf=0.1) {
  mu <- rep(0, nsnps)
  rawvars <- rmvnorm(n=nsamples, mean=mu, sigma=S)
  pvars <- pnorm(rawvars)
  x <- qbinom(1-pvars, 1, maf)
}

S <- (1 - (abs(outer(1:nsnps, 1:nsnps, `~`)))/nsnps))^4
X <- simx(nsnps, nsamples, S)
LD <- cor2(X)
maf <- colMeans(X)
```

```
corr cov_C I(z = z_scores, f = maf, N0, N1, Sigma = LD, thr = 0.95)
```

---

corr cov_C I_b hat	<i>Confidence interval for corrected coverage estimate using estimated effect sizes and their standard errors</i>
--------------------	---

---

### Description

Obtain confidence interval for corrected coverage estimate using estimated effect sizes and their standard errors

### Usage

```
corr cov_C I_b hat(b hat, V, N0, N1, Sigma, thr, W = 0.2, nrep = 1000,
  CI = 0.95, pp0min = 0.001)
```

### Arguments

b hat	Estimated effect sizes from single-SNP logistic regressions
V	Variance of estimated effect sizes
N0	Number of controls
N1	Number of cases
Sigma	SNP correlation matrix
thr	Minimum threshold for fine-mapping experiment
W	Prior for the standard deviation of the effect size parameter beta
nrep	The number of simulated posterior probability systems to consider for the corrected coverage estimate (nrep = 1000 default)
CI	The size of the confidence interval (as a decimal)
pp0min	Only average over SNPs with pp0 > pp0min

### Value

CI for corrected coverage estimate

### Author(s)

Anna Hutchinson

## Examples

```
# this is a long running example
set.seed(1)
nsnps <- 100
N0 <- 5000 # number of controls
N1 <- 5000 # number of cases

## generate example LD matrix
library(mvtnorm)
nsamples = 1000

simx <- function(nsnps, nsamples, S, maf=0.1) {
  mu <- rep(0,nsnps)
  rawvars <- rmvnorm(n=nsamples, mean=mu, sigma=S)
  pvars <- pnorm(rawvars)
  x <- qbinom(1-pvars, 1, maf)
}

S <- (1 - (abs(outer(1:nsnps,1:nsnps,`-`)))/nsnps))^4
X <- simx(nsnps,nsamples,S)
LD <- cor2(X)
maf <- colMeans(X)

varbeta <- Var.data.cc(f = maf, N = N0 + N1, s = N1/(N0+N1))

bhats = rnorm(nsnps,0,0.2) # log OR

corrcov_CI_bhat(bhat = bhats, V = varbeta, N0, N1, Sigma = LD)
```

---

corrcov\_nvar

*Corrected coverage estimate using Z-scores and MAFs (fixing nvar)*

---

## Description

Obtain corrected coverage estimate using Z-scores and mafs (limiting simulations used for estimation to those with correct nvar)

## Usage

```
corrcov_nvar(z, f, N0, N1, Sigma, nvar, thr, W = 0.2, nrep = 10000,
  pp0min = 0.001)
```

## Arguments

z	Marginal Z-scores
f	Minor allele frequencies

N0	Number of controls
N1	Number of cases
Sigma	SNP correlation matrix
nvar	The number of variants that simulated credible sets used for estimation should contain
thr	Minimum threshold for fine-mapping experiment
W	Prior for the standard deviation of the effect size parameter, beta (default 0.2)
nrep	The number of simulated posterior probability systems to consider for the corrected coverage estimate (nrep = 10000 default due to trimming)
pp0min	Only average over SNPs with pp0 > pp0min

### Details

This function requires the marginal summary statistics from GWAS and an nvar value. It should only be used when nvar is very low (<3) and there is some evidence to suggest that only simulated credible sets with this nvar value should be used to derive the corrected coverage estimate.

### Value

Corrected coverage estimate

### Author(s)

Anna Hutchinson

### Examples

```
set.seed(1)
nsnps = 100
N0 = 5000
N1 = 5000
z_scores <- rnorm(nsnps, 0, 3) # simulate a vector of Z-scores

## generate example LD matrix
library(mvtnorm)
nsamples = 1000

simx <- function(nsnps, nsamples, S, maf=0.1) {
  mu <- rep(0, nsnps)
  rawvars <- rmvnorm(n=nsamples, mean=mu, sigma=S)
  pvars <- pnorm(rawvars)
  x <- qbinom(1-pvars, 1, maf)
}

S <- (1 - (abs(outer(1:nsnps, 1:nsnps, `~`)))/nsnps))^4
X <- simx(nsnps, nsamples, S)
LD <- cor2(X)
maf <- colMeans(X)
```

```
corr cov_nvar(z = z_scores, f = maf, N0, N1, Sigma = LD, thr = 0.95, nvar = 1, nrep = 100)

# note that nrep should be at least the default value (nrep = 10000) but is
# lower here for speed of computation
```

---

corr cov_nvar_bhat	<i>Corrected coverage estimate using estimated effect sizes and their standard errors (fixing nvar)</i>
--------------------	---

---

### Description

Obtain corrected coverage estimate using estimated effect sizes and their standard errors (limiting simulations used for estimation to those with correct nvar)

### Usage

```
corr cov_nvar_bhat(bhat, V, N0, N1, Sigma, nvar, thr, W = 0.2,
  nrep = 10000, pp0min = 0.001)
```

### Arguments

bhat	Estimated effect sizes from single-SNP logistic regressions
V	Variance of estimated effect sizes
N0	Number of controls
N1	Number of cases
Sigma	SNP correlation matrix
nvar	The number of variants that simulated credible sets used for estimation should contain
thr	Minimum threshold for fine-mapping experiment
W	Prior for the standard deviation of the effect size parameter, beta (default 0.2)
nrep	The number of simulated posterior probability systems to consider for the corrected coverage estimate (nrep = 10000 default due to trimming)
pp0min	Only average over SNPs with pp0 > pp0min

### Details

This function requires the marginal summary statistics from GWAS and an nvar value. It should only be used when nvar is very low ( $<3$ ) and there is some evidence to suggest that only simulated credible sets with this nvar value should be used to derive the corrected coverage estimate.

### Value

Corrected coverage estimate

**Author(s)**

Anna Hutchinson

**Examples**

```

set.seed(1)
nsnps <- 100
N0 <- 5000 # number of controls
N1 <- 5000 # number of cases

## generate example LD matrix
library(mvtnorm)
nsamples = 1000

simx <- function(nsnps, nsamples, S, maf=0.1) {
  mu <- rep(0,nsnps)
  rawvars <- rmvnorm(n=nsamples, mean=mu, sigma=S)
  pvars <- pnorm(rawvars)
  x <- qbinom(1-pvars, 1, maf)
}

S <- (1 - (abs(outer(1:nsnps,1:nsnps,`-`)))/nsnps))^4
X <- simx(nsnps,nsamples,S)
LD <- cor2(X)
maf <- colMeans(X)

varbeta <- Var.data.cc(f = maf, N = N0 + N1, s = N1/(N0+N1))

bhats = rnorm(nsnps,0,0.2) # log OR

corrcov_nvar_bhat(bhat = bhats, V = varbeta, N0, N1, Sigma = LD, thr = 0.95, nvar = 1, nrep = 1000)

# note that nrep should be at least the default value (nrep = 10000) but is
# lower here for speed of computation

```

---

corrected\_cov

*Corrected coverage estimate of the causal variant in the credible set*

---

**Description**

Corrected coverage estimate of the causal variant in the credible set

**Usage**

```

corrected_cov(pp0, mu, V, Sigma, thr, W = 0.2, nrep = 1000,
  pp0min = 0.001)

```

**Arguments**

pp0	Posterior probabilities of SNPs
mu	The true effect at the CV (estimate using <code>corrcoverage::est_mu</code> function)
V	Variance of the estimated effect size (can be obtained using <code>coloc::Var.beta.cc</code> function)
Sigma	SNP correlation matrix
thr	Minimum threshold for fine-mapping experiment
W	Prior for the standard deviation of the effect size parameter, beta (W=0.2 default)
nrep	Number of posterior probability systems to simulate for each variant considered causal (nrep = 1000 default)
pp0min	Only average over SNPs with $pp0 > pp0min$

**Details**

Requires an estimate of the true effect at the CV (e.g. use maximum absolute z-score or output from `corrcoverage::est_mu` function)

**Value**

Corrected coverage estimate

**Author(s)**

Anna Hutchinson

**Examples**

```
set.seed(1)
nsnps <- 100
N0 <- 5000
N1 <- 5000

## generate example LD matrix
library(mvtnorm)
nsamples = 1000

simx <- function(nsnps, nsamples, S, maf=0.1) {
  mu <- rep(0, nsnps)
  rawvars <- rmvnorm(n=nsamples, mean=mu, sigma=S)
  pvars <- pnorm(rawvars)
  x <- qbinom(1-pvars, 1, maf)
}

S <- (1 - (abs(outer(1:nsnps, 1:nsnps, `~`)))/nsnps))^4
X <- simx(nsnps, nsamples, S)
LD <- cor2(X)
maf <- colMeans(X)
```



```
## generate V (variance of estimated effect sizes)
varbeta <- Var.data.cc(f = maf, N = 5000, s = 0.5)

pp <- rnorm(nsnps, 0.2, 0.05)
pp <- pp/sum(pp)

corrected_cov(pp0 = pp, mu = 4, V = varbeta, Sigma = LD, thr = 0.95, nrep = 100)
```

---

corrected\_cs

---

*Corrected credible set using Z-scores and MAFs*


---

### Description

Corrected credible set using Z-scores and MAFs

### Usage

```
corrected_cs(z, f, N0, N1, Sigma, W = 0.2, lower = 0, upper = 1,
  desired.cov, acc = 0.005, max.iter = 20, pp0min = 0.001)
```

### Arguments

z	Z-scores
f	Minor allele frequencies
N0	Number of controls
N1	Number of cases
Sigma	Correlation matrix of SNPs
W	Prior for the standard deviation of the effect size parameter, beta (default 0.2)
lower	Lower threshold (default = 0)
upper	Upper threshold (default = 1)
desired.cov	The desired coverage of the causal variant in the credible set
acc	Accuracy of corrected coverage to desired coverage (default = 0.005)
max.iter	Maximum iterations (default = 20)
pp0min	Only average over SNPs with pp0 > pp0min

### Value

List of variants in credible set, required threshold, the corrected coverage and the size of the credible set

### Author(s)

Anna Hutchinson

**Examples**

```

# this is a long running example

# In this example, the function is used to find a corrected 95% credible set
# using Z-scores and MAFs, that is the smallest set of variants
# required such that the resultant credible set has coverage close to (/within
# some accuracy of) the "desired coverage" (here set to 0.95). Max.iter parameter
# defines the maximum number of iterations to try in the root bisection algorithm,
# this should be increased to ensure convergence to the desired coverage, but is set
# to 1 here for speed (and thus the resultant credible set will not be accurate).

set.seed(2)
nsnps = 200
N0 = 1000
N1 = 1000
z_scores <- rnorm(nsnps, 0, 1) # simulate a vector of Z-scores

## generate example LD matrix
library(mvtnorm)
nsamples = 1000

simx <- function(nsnps, nsamples, S, maf=0.1) {
  mu <- rep(0,nsnps)
  rawvars <- rmvnorm(n=nsamples, mean=mu, sigma=S)
  pvars <- pnorm(rawvars)
  x <- qbinom(1-pvars, 1, maf)
}

S <- (1 - (abs(outer(1:nsnps,1:nsnps,`-`)))/nsnps))^4
X <- simx(nsnps,nsamples,S)
LD <- cor2(X)
maf <- colMeans(X)

names(z_scores) <- seq(1,length(z_scores))

corrected_cs(z = z_scores, f = maf, N0, N1, Sigma = LD, desired.cov = 0.9, max.iter = 1)
# max.iter set low for speed, should be set to at least
# the default to ensure convergence to desired coverage

```

---

corrected\_cs\_bhat

*Corrected credible set using estimated effect sizes and their standard errors*


---

**Description**

Corrected credible set using estimated effect sizes and their standard errors

**Usage**

```
corrected_cs_bhat(bhat, V, N0, N1, Sigma, W = 0.2, lower = 0,
  upper = 1, desired.cov, acc = 0.005, max.iter = 20,
  pp0min = 0.001)
```

**Arguments**

bhat	Estimated effect sizes
V	Prior variance of estimated effect sizes
N0	Number of controls
N1	Number of cases
Sigma	Correlation matrix of SNPs
W	Prior for the standard deviation of the effect size parameter, beta (default 0.2)
lower	Lower threshold (default = 0)
upper	Upper threshold (default = 1)
desired.cov	The desired coverage of the causal variant in the credible set
acc	Accuracy of corrected coverage to desired coverage (default = 0.005)
max.iter	Maximum iterations (default = 20)
pp0min	Only average over SNPs with $pp0 > pp0min$

**Value**

List of variants in credible set, required threshold, the corrected coverage and the size of the credible set

**Author(s)**

Anna Hutchinson

**Examples**

```
# this is a long running example

# In this example, the function is used to find a corrected 95% credible set
# using bhats and their standard errors, that is the smallest set of variants
# required such that the resultant credible set has coverage close to (/within
# some accuracy of) the "desired coverage" (here set to 0.95). Max.iter parameter
# defines the maximum number of iterations to try in the root bisection algorithm,
# this should be increased to ensure convergence to the desired coverage, but is set
# to 1 here for speed (and thus the resultant credible set will not be accurate).

set.seed(18)
nsnps <- 100
N0 <- 500 # number of controls
N1 <- 500 # number of cases
```

```

# simulate fake haplotypes to obtain MAFs and LD matrix
## generate example LD matrix
library(mvtnorm)
nsamples = 1000

simx <- function(nsnp, nsamples, S, maf=0.1) {
  mu <- rep(0,nsnp)
  rawvars <- rmvnorm(n=nsamples, mean=mu, sigma=S)
  pvars <- pnorm(rawvars)
  x <- qbinom(1-pvars, 1, maf)
}

S <- (1 - (abs(outer(1:nsnp,1:nsnp,`-`)))/nsnp))^4
X <- simx(nsnp,nsamples,S)
LD <- cor2(X)
maf <- colMeans(X)

varbeta <- Var.data.cc(f = maf, N = N0 + N1, s = N1/(N0+N1))

bhats = rnorm(nsnp,0,0.2) # log OR

names(bhats) <- seq(1,length(bhats))

corrected_cs_bhat(bhat = bhats, V = varbeta, N0, N1, Sigma = LD, desired.cov = 0.9, max.iter = 1)
# max.iter set low for speed, should be set to at
# least the default to ensure convergence to desired coverage

```

---

credset

*Credible set of genetic variants*


---

## Description

Credible set of putative causal variants

## Usage

```
credset(pp, CV, thr)
```

## Arguments

pp	Vector of posterior probabilities of causality
CV	Optional parameter: Index of CV
thr	Minimum threshold for credible set size

## Details

If the CV parameter is supplied (index of causal variant) then the output includes a binary indicator of whether the CV is contained in the set

**Value**

list of the variants in the credible set, the claimed.cov (cumulative sum of the posterior probabilities of the variants forming the credible set), binary covered indicator (1 if CV is contained in the credible set) and nvar (number of variants in the set)

**Author(s)**

Anna Hutchinson

**Examples**

```
set.seed(1)
nsnps <- 100
pp <- rnorm(nsnps, 0.3, 0.05)
pp <- pp/sum(pp)

credsetC(pp, thr = 0.9)

iCV <- 71

credsetC(pp, CV = iCV, thr = 0.9)
```

---

 credsetC

*Credible set of variants from matrix of PPs*


---

**Description**

Quick credset function for matrix of posterior probabilities (using RCpp)

**Usage**

```
credsetC(pp, CV, thr)
```

**Arguments**

pp	Matrix of posterior probabilities of causality (one row per system)
CV	Vector of CV indices (one per system/row)
thr	Minimum threshold for credible set size

**Value**

Data.frame of claimed coverage (sum of posterior probabilities of variants in the set), binary covered indicator and number of variants (nvar).

**Examples**

```

set.seed(1)
nsnps <- 100

# simulate matrix of posterior probabilities
# 1 simulation per row

pp <- matrix(rnorm(nsnps*100, 0.3, 0.05), ncol = nsnps)
pp <- pp/rowSums(pp)

iCV <- rep(71, times = dim(pp)[1])

credsetC(pp, CV = iCV, thr = 0.9)

```

---

credsetmat

*Obtain credible sets from a matrix of posterior probabilities*


---

**Description**

Obtain credible sets from a matrix of posterior probabilities

**Usage**

```
credsetmat(pp, iCV, threshold)
```

**Arguments**

pp	Matrix of posterior probabilities (one row for each simulation)
iCV	A vector of the indices of the CV
threshold	The threshold to use to generate the credible set

---

est\_mu

*Estimate the true effect at the causal variant using Z-scores and MAFs*


---

**Description**

Estimate the true effect at the causal variant using Z-scores and MAFs

**Usage**

```
est_mu(z, f, N0, N1, W = 0.2)
```

**Arguments**

z	Vector of marginal Z-scores
f	Minor allele frequencies
N0	Number of controls
N1	Number of cases
W	Prior for the standard deviation of the effect size parameter, beta, default 0.2

**Value**

Estimate of the true effect at the causal variant

**Author(s)**

Anna Hutchinson

**Examples**

```
nsnps <- 100
z_scores <- rnorm(nsnps, 0, 3) # simulate a vector of Z-scores
N0 <- 5000 # number of controls
N1 <- 5000 # number of cases

maf <- runif(nsnps, 0.05, 0.5)

est_mu(z = z_scores, f = maf, N0 = N0, N1 = N1)
```

---

est_mu_bhat	<i>Estimate the true effect at the causal variant using estimated effect sizes and their standard errors</i>
-------------	--

---

**Description**

Estimate the true effect at the causal variant using estimated effect sizes and their standard errors

**Usage**

```
est_mu_bhat(bhat, V, N0, N1, p1 = 1e-04, W = 0.2)
```

**Arguments**

bhat	Vector of estimated effect sizes
V	Prior variance for estimated effect sizes
N0	Number of controls
N1	Number of cases
p1	Prior probability a SNP is associated with the trait, default 1e-4
W	Prior for the standard deviation of the effect size parameter, beta

**Value**

Estimate of the true effect at the causal variant

**Author(s)**

Anna Hutchinson

**Examples**

```
nsnps <- 100
N0 <- 5000 # number of controls
N1 <- 5000 # number of cases

maf <- runif(nsnps, 0.05, 0.3)

varbeta <- Var.data.cc(f = maf, N = N0 + N1, s = N1/(N0+N1))

bhats = rnorm(nsnps,0,0.2) # log(OR)

est_mu_bhat(bhat = bhats, V = varbeta, N0 = N0, N1 = N1)
```

---

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)**

Chris Wallace



---

logsum_matrix	<i>logsum rows of a matrix</i>
---------------	--------------------------------

---

**Description**

matrix-ified version of logsum to avoid needing apply()

**Usage**

```
logsum_matrix(x)
```

**Arguments**

x                    numeric matrix

**Value**

rowwise sums

**Author(s)**

Chris Wallace

---

ppfunc	<i>Find PPs of SNPs from Z-scores</i>
--------	---------------------------------------

---

**Description**

Posterior probabilities of causality from marginal Z-scores

**Usage**

```
ppfunc(z, V, W = 0.2)
```

**Arguments**

z                    Vector of marginal Z-scores  
V                    Variance of the estimated effect size (can be obtained using Var.beta.cc function)  
W                    Prior for the standard deviation of the effect size parameter, beta (W = 0.2 default)

**Details**

This function converts Z-scores to posterior probabilities of causality i.e. not including the null model of no genetic effects, so that the sum of the posterior probabilities for all variants is 1

**Value**

Vector of posterior probabilities

**Examples**

```
set.seed(1)
nsnps = 100
N0 = 5000
N1 = 5000
z_scores <- rnorm(nsnps, 0, 3)

## generate example LD matrix and MAFs
library(mvtnorm)
nsamples = 1000

simx <- function(nsnps, nsamples, S, maf=0.1) {
  mu <- rep(0,nsnps)
  rawvars <- rmvnorm(n=nsamples, mean=mu, sigma=S)
  pvars <- pnorm(rawvars)
  x <- qbinom(1-pvars, 1, maf)
}

S <- (1 - (abs(outer(1:nsnps,1:nsnps,`-`)))/nsnps))^4
X <- simx(nsnps,nsamples,S)
maf <- colMeans(X)

varbeta <- Var.data.cc(f = maf, N = N0+N1, s = N1/(N0+N1))

res <- ppfunc(z = z_scores, V = varbeta)
sum(res)
res
```

---

ppfunc.mat

*Find PPs of SNPs from matrix of Z-scores*


---

**Description**

Posterior probabilities of causality from matrix of marginal Z-scores (1 simulation per row)

**Usage**

```
ppfunc.mat(zstar, V, W = 0.2)
```

**Arguments**

zstar	Matrix of marginal z-scores, one replicate per row
V	Variance of the estimated effect size, one element per column of zstar
W	Prior for the standard deviation of the effect size parameter, beta

**Details**

This function converts a matrix of Z-scores (one row per simulation) to posterior probabilities of causality, not including the null model of no genetic effects, so that the sum of the posterior probabilities for each simulation (each row) is 1.

**Value**

Matrix of posterior probabilities of causality

**Author(s)**

Chris Wallace

**Examples**

```
set.seed(1)
nsnps = 100
N0 = 5000
N1 = 5000

## generate example LD matrix and MAFs
library(mvtnorm)
nsamples = 1000

simx <- function(nsnps, nsamples, S, maf=0.1) {
  mu <- rep(0,nsnps)
  rawvars <- rmvnorm(n=nsamples, mean=mu, sigma=S)
  pvars <- pnorm(rawvars)
  x <- qbinom(1-pvars, 1, maf)
}

S <- (1 - (abs(outer(1:nsnps,1:nsnps,`-`)))/nsnps))^4
X <- simx(nsnps,nsamples,S)
maf <- colMeans(X)

varbeta <- Var.data.cc(f = maf, N = N0+N1, s = N1/(N0+N1))

# simulate matrix of Z scores
# 1 simulation per row
z_scores <- matrix(rnorm(nsnps*100, 0, 3), ncol = nsnps)

# each row is a vector of simulated PPs
res <- ppfunc.mat(zstar = z_scores, V = varbeta)

rowSums(res)
```

---

prop_cov	<i>Proportion of credible sets containing the causal variant</i>
----------	--

---

**Description**

Proportion of simulated credible sets containing the causal variant

**Usage**

```
prop_cov(x)
```

**Arguments**

x                      data.frame with a binary 'covered' column

**Value**

Proportion of x with x = 1

**Author(s)**

Anna Hutchinson

---

pvals_pp	<i>Find PPs for SNPs and null model from P-values and MAFs</i>
----------	--

---

**Description**

Posterior probabilities of causality from P-values

**Usage**

```
pvals_pp(pvals, f, type, N, s, W = 0.2, p1 = 1e-04)
```

**Arguments**

pvals	P-values of SNPs
f	Minor allele frequencies
type	Type of experiment ('quant' or 'cc')
N	Total sample size
s	Proportion of cases (N1/N0+N1), ignored if type=='quant'
W	Prior for the standard deviation of the effect size parameter, beta (default 0.2)
p1	Prior probability a SNP is associated with the trait (default 1e-4)

**Details**

This function converts p-values to posterior probabilities of causality, including the null model of no genetic effect

**Value**

Posterior probabilities of null model (no genetic effect) and causality for each SNP

**Author(s)**

Anna Hutchinson

**Examples**

```
set.seed(1)
nsnps = 100
N0 = 5000
N1 = 5000
z_scores <- rnorm(nsnps, 0, 3)
p_values <- 2 * pnorm( - abs ( z_scores ) )

## generate example LD matrix and MAFs
library(mvtnorm)
nsamples = 1000

simx <- function(nsnps, nsamples, S, maf=0.1) {
  mu <- rep(0,nsnps)
  rawvars <- rmvnorm(n=nsamples, mean=mu, sigma=S)
  pvars <- pnorm(rawvars)
  x <- qbinom(1-pvars, 1, maf)
}

S <- (1 - (abs(outer(1:nsnps,1:nsnps,`-`)))/nsnps))^4
X <- simx(nsnps,nsamples,S)
maf <- colMeans(X)

res <- pvals_pp(pvals = p_values, f = maf, type = "cc", N = N0+N1, s = N1/(N0+N1))
sum(res)
res
```

**Description**

Variance of the estimated effect size for case-control data

**Usage**

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

**Arguments**

f	Minor allele frequencies
N	Total sample size (N0+N1)
s	Proportion of cases (N1/N0+N1)

**Value**

Variance of estimated effect size  $\hat{\beta}$ , V.

**Author(s)**

Chris Wallace

**Examples**

```
maf = runif(100, 0.05, 0.5)
N0 = 5000 # number of controls
N1 = 5000 # number of cases

Var.data.cc(f = maf, N = N0 + N1, s = N1/(N0+N1))
```

---

z0\_pp

*Find PPs for SNPs and null model from Z-scores and MAFs*


---

**Description**

Posterior probabilities of causality from marginal Z-scores with prior SD as a parameter

**Usage**

```
z0_pp(z, f, type, N, s, W = 0.2, p1 = 1e-04)
```

**Arguments**

z	Marginal Z-scores of SNPs
f	Minor allele frequencies
type	Type of experiment ('quant' or 'cc')
N	Total sample size
s	Proportion of cases (N1/N0+N1), ignored if type=='quant'
W	Prior for the standard deviation of the effect size parameter, beta (default 0.2)
p1	Prior probability a SNP is associated with the trait (default 1e-4)

**Details**

Converts Z-scores to posterior probabilities of causality, including the null model of no genetic effects

**Value**

Posterior probabilities of null model (no genetic effect) and causality for each SNP

**Author(s)**

Anna Hutchinson

**Examples**

```
set.seed(1)
nsnps = 100
N0 = 5000
N1 = 5000
z_scores <- rnorm(nsnps, 0, 3)

## generate example LD matrix and MAFs
library(mvtnorm)
nsamples = 1000

simx <- function(nsnps, nsamples, S, maf=0.1) {
  mu <- rep(0,nsnps)
  rawvars <- rmvnorm(n=nsamples, mean=mu, sigma=S)
  pvars <- pnorm(rawvars)
  x <- qbinom(1-pvars, 1, maf)
}

S <- (1 - (abs(outer(1:nsnps,1:nsnps,`-`)))/nsnps))^4
X <- simx(nsnps,nsamples,S)
maf <- colMeans(X)

res <- z0_pp(z = z_scores, f = maf, type = "cc", N = N0+N1, s = N1/(N0+N1))
sum(res)
res
```

---

zj\_pp

---

*Simulate posterior probabilities of causality from joint Z-score vector*


---

**Description**

Simulate nrep marginal Z-scores from joint Z-scores and convert these to posterior probabilities of causality

**Usage**

```
zj_pp(Zj, V, nrep = 1000, W = 0.2, Sigma)
```

**Arguments**

Zj	Vector of joint Z-scores (0s except at CV)
V	Variance of the estimated effect size (can be obtained using Var.beta.cc function)
nrep	Number of posterior probability systems to simulate (default 1000)
W	Prior for the standard deviation of the effect size parameter, beta (default 0.2)
Sigma	SNP correlation matrix

**Details**

Does not include posterior probabilities for null model

**Value**

Matrix of simulated posterior probabilities, one simulation per row

**Author(s)**

Anna Hutchinson

**Examples**

```
set.seed(1)
nsnps <- 100
Zj <- rep(0, nsnps)
iCV <- 4 # index of CV
mu <- 5 # true effect at CV
Zj[iCV] <- mu

## generate example LD matrix and MAFs
library(mvtnorm)
nsamples = 1000

simx <- function(nsnps, nsamples, S, maf=0.1) {
  mu <- rep(0, nsnps)
  rawvars <- rmvnorm(n=nsamples, mean=mu, sigma=S)
  pvars <- pnorm(rawvars)
  x <- qbinom(1-pvars, 1, maf)
}

S <- (1 - (abs(outer(1:nsnps, 1:nsnps, `~`)))/nsnps))^4
X <- simx(nsnps, nsamples, S)
LD <- cor2(X)
maf <- colMeans(X)

## generate V (variance of estimated effect sizes)
varbeta <- Var.data.cc(f = maf, N = 5000, s = 0.5)
```



```
res <- zj_pp(Zj, V = varbeta, nrep = 5, W = 0.2, Sigma = LD)
res[c(1:5), c(1:5)]
```

---

`zj_pp_arma`                      *Obtain pp from a matrix of Zj and ERR*

---

### Description

Obtain pp from a matrix of Zj and ERR

### Usage

```
zj_pp_arma(Zj, sigma, nrep, ERR, r)
```

### Arguments

Zj	vector of joint Z scores
sigma	SNP correlation matrix
nrep	Number of simulations
ERR	Error matrix
r	Shrinkage values

### Value

pp Matrix of posterior probabilities (one row for each simulation)

---

`z_sim`                              *Simulate marginal Z-scores from joint Z-score vector*

---

### Description

Simulate marginal z-scores ( $Z_m$ ) from the joint z-scores ( $Z_j$ ) using  $E(Z_m) = Z_j \times \Sigma$  and  $Z^* \sim MVN(E(Z_m), \Sigma)$

### Usage

```
z_sim(Zj, Sigma, nrep)
```

**Arguments**

Zj                    Vector of joint Z-scores (a vector of 0s except at the CV)  
 Sigma                SNP correlation matrix  
 nrep                 Number of Z-score systems to simulate

**Value**

Matrix of simulated posterior probabilities, one simulation per row

**Author(s)**

Anna Hutchinson

**Examples**

```
set.seed(1)
nsnps <- 100

# derive joint Z score vector
Zj <- rep(0, nsnps)
iCV <- 4 # index of CV
mu <- 5 # true effect at CV
Zj[iCV] <- mu

## generate example LD matrix
library(mvtnorm)
nsamples = 1000

simx <- function(nsnps, nsamples, S, maf=0.1) {
  mu <- rep(0, nsnps)
  rawvars <- rmvnorm(n=nsamples, mean=mu, sigma=S)
  pvars <- pnorm(rawvars)
  x <- qbinom(1-pvars, 1, maf)
}

S <- (1 - (abs(outer(1:nsnps, 1:nsnps, `~`)))/nsnps))^4
X <- simx(nsnps, nsamples, S)
LD <- cor2(X)

res <- z_sim(Zj, Sigma = LD, nrep = 100)
res[c(1:5), c(1:5)]
```

# Index

.zj\_abf, [2](#)  
.zj\_pp, [3](#)  
  
approx.bf.p, [4](#)  
  
bf\_func, [5](#)  
  
cor2, [6](#)  
corr cov, [7](#)  
corr cov\_bhat, [8](#)  
corr cov\_CI, [9](#)  
corr cov\_CI\_bhat, [11](#)  
corr cov\_nvar, [12](#)  
corr cov\_nvar\_bhat, [14](#)  
corrected\_cov, [15](#)  
corrected\_cs, [17](#)  
corrected\_cs\_bhat, [18](#)  
credset, [20](#)  
credsetC, [21](#)  
credsetmat, [22](#)  
  
est\_mu, [22](#)  
est\_mu\_bhat, [23](#)  
  
logsum, [24](#)  
logsum\_matrix, [25](#)  
  
ppfunc, [25](#)  
ppfunc.mat, [26](#)  
prop\_cov, [28](#)  
pvals\_pp, [28](#)  
  
Var.data.cc, [29](#)  
  
z0\_pp, [30](#)  
z\_sim, [33](#)  
zj\_pp, [31](#)  
zj\_pp\_arma, [33](#)