Package 'rBahadur'

August 25, 2023

Title Assortative Mating Simulation and Multivariate Bernoulli Variates

Version 1.0.0

Description Simulation of phenotype / genotype data under assortative mating. Includes functions for generating Bahadur order-2 multivariate Bernoulli variables with general and diagonal-plus-low-rank correlation structures. Further details are provided in: Border and Malik (2022) <doi:10.1101/2022.10.13.512132>.

URL https://github.com/rborder/rBahadur

License GPL (>= 3)

Encoding UTF-8

Language en-US

Depends R (>= 3.3.0), stats

RoxygenNote 7.2.3

NeedsCompilation no

Author Richard Border [aut, cre] (<https://orcid.org/0000-0002-6293-2968>), Osman Malik [aut] (<https://orcid.org/0000-0003-4477-481X>)

Maintainer Richard Border <border.richard@gmail.com>

Repository CRAN

Date/Publication 2023-08-25 19:30:02 UTC

R topics documented:

am_covariance_structure 2
am_equilibrium_parameters 2
am_simulate 3
rb_dplr 5
rb_unstr 6

Index **8**

am_covariance_structure

Compute Diagonal plus Low Rank equilibrium covariance structure

Description

Compute Diagonal plus Low Rank equilibrium covariance structure

Usage

```
am_covariance_structure(beta, AF, r)
```

Arguments

beta	vector of standardized diploid allele-substitution effects
AF	vector of allele frequencies
r	cross-mate phenotypic correlation

Value

Vector 'U' such that $D + U U^T$ corresponds to the expected haploid LD-matrix given the specified genetic architecture (encoded by 'beta' and 'AF') and cross-mate phenotypic correlation 'r'. It is assumed that the total phenotypic variance at generation zero is one.

Examples

```
set.seed(1)
h2_0 = .5; m = 200; n = 1000; r =.5; min_MAF=.1
betas <- rnorm(m,0,sqrt(h2_0/m))
afs <- runif(m, min_MAF, 1-min_MAF)
output <- am_covariance_structure(betas, afs, r)
```

am_equilibrium_parameters

Functions to compute equilibrium parameters under assortative mating

Description

Compute heritability ('h2_eq'), genetic variance ('vg_0'), and cross-mate genetic correlation ('rg_eq') at equilibrium under univariate primary-phenotypic assortative mating. These equations can be derived from Nagylaki's results (see below) under the assumption that number of causal variants is large (i.e., taking the limit as the number of causal variants approaches infinity).

Usage

```
h2_eq(r, h2_0)

rg_eq(r, h2_0)

vg_eq(r, vg_0, h2_0)
```

Arguments

r	cross-mate phenotypic correlation
h2_0	generation zero (panmictic) heritability
vg_0	generation zero (panmictic) additive genetic variance component

Value

A single numerical quantity representing the equilibrium heritability (h2_eq), the equilibrium cross-mate genetic correlation (rg_eq), or the equilibrium genetic variance (vg_eq).

References

Nagylaki, T. Assortative mating for a quantitative character. J. Math. Biology 16, 57–74 (1982). https://doi.org/10.1007/BF00275161

Examples

```
set.seed(1)
vg_0= .6; h2_0 = .5; r =.5
h2_eq(r, h2_0)
rg_eq(r, h2_0)
vg_eq(r, vg_0, h2_0)
```

---

am_simulate *Simulate genotype/phenotype data under equilibrium univariate AM.*

Description

Simulate genotype/phenotype data under equilibrium univariate AM.

Usage

```
am_simulate(h2_0, r, m, n, afs = NULL, min_MAF = 0.1, haplotypes = FALSE)
```

Arguments

h2_0	generation zero (panmictic) heritability
r	cross-mate phenotypic correlation
m	number of biallelic causal variants
n	sample size
afs	(optional). Allele frequencies to use. If not provided, m will be drawn uniformly from the interval [min_MAF, 1-min_MAF]
min_MAF	(optional) minimum minor allele frequency for causal variants. Ignored if if afs is not NULL. Defaults to 0.1
haplotypes	logical. If TRUE, includes (phased) haploid genotypes in output. Defaults to FALSE

Value

A list including the following objects:

- y: phenotype vector
- g: heritable component of the phenotype vector
- X: matrix of diploid genotypes
- AF: vector of allele frequencies
- beta_std: standardized genetic effects
- beta_raw: unstandardized genetic effects
- H: matrix of haploid genotypes (returned only if haplotypes=TRUE)

Examples

```
set.seed(1)
h2_0 = .5; m = 200; n = 1000; r =.5

## simulate genotype/phenotype data
sim_dat <- am_simulate(h2_0, r, m, n)
str(sim_dat)

## empirical h2 vs expected equilibrium h2
(emp_h2 <- var(sim_dat$g)/var(sim_dat$y))
h2_eq(r, h2_0)
```

rb_dplr *Binary random variates with Diagonal Plus Low Rank (dplr) correlations*

Description

Generate second Bahadur order multivariate Bernoulli random variates with Diagonal Plus Low Rank (dplr) correlation structures.

Usage

```
rb_dplr(n, mu, U)
```

Arguments

n	number of observations
mu	vector of means
U	outer product component matrix

Details

This generates multivariate Bernoulli (MVB) random vectors with mean vector 'mu' and correlation matrix $C = D + UU^T$ where D is a diagonal matrix with values dictated by 'U'. 'mu' must take values in the open unit interval and 'U' must induce a valid second Bahadur order probability distribution. That is, there must exist an MVB probability distribution with first moments 'mu' and standardized central second moments C such that all higher order central moments are zero.

Value

An n-by-m matrix of binary random variates, where m is the length of 'mu'.

Examples

```
set.seed(1)
h2_0 = .5; m = 200; n = 1000; r =.5; min_MAF=.1

## draw standardized diploid allele substitution effects
beta <- scale(rnorm(m))*sqrt(h2_0 / m)

## draw allele frequencies
AF <- runif(m, min_MAF, 1 - min_MAF)

## compute unstandardized effects
beta_unscaled <- beta/sqrt(2*AF*(1-AF))

## generate corresponding haploid quantities
beta_hap <- rep(beta, each=2)
AF_hap <- rep(AF, each=2)
```

```
## compute equilibrium outer product covariance component
U <- am_covariance_structure(beta, AF, r)

## draw multivariate Bernoulli haplotypes
H <- rb_dplr(n, AF_hap, U)

## convert to diploid genotypes
G <- H[,seq(1,ncol(H),2)] + H[,seq(2,ncol(H),2)]

## empirical allele frequencies vs target frequencies
emp_afs <- colMeans(G)/2
plot(AF, emp_afs)

## construct phenotype
heritable_y <-  G%*%beta_unscaled
nonheritable_y <-  rnorm(n, 0, sqrt(1-h2_0))
y <- heritable_y + nonheritable_y

## empirical h2 vs expected equilibrium h2
(emp_h2 <- var(heritable_y)/var(y))
h2_eq(r, h2_0)
```

`rb_unstr` *Binary random variates with unstructured correlations*

Description

Generate Bahadur order-2 multivariate Bernoulli random variates with unstructured correlations.

Usage

```
rb_unstr(n, mu, C)
```

Arguments

`n`	number of observations
`mu`	vector of means
`C`	correlation matrix

Details

This generates multivariate Bernoulli (MVB) random vectors with mean vector 'mu' and correlation matrix 'C'. 'mu' must take values in the open unit interval and 'C' must induce a valid second Bahadur order probability distribution. That is, there must exist an MVB probability distribution with first moments 'mu' and standardized central second moments 'C' such that all higher order central moments are zero.

Value

An n-by-m matrix of binary random variates, where m is the length of 'mu'.

Examples

```
set.seed(1)
h2_0 = .5; m = 200; n = 500; r =.5; min_MAF=.1

## draw standardized diploid allele substitution effects
beta <- scale(rnorm(m))*sqrt(h2_0 / m)

## draw allele frequencies
AF <- runif(m, min_MAF, 1 - min_MAF)

## compute unstandardized effects
beta_unscaled <- beta/sqrt(2*AF*(1-AF))

## generate corresponding haploid quantities
beta_hap <- rep(beta, each=2)
AF_hap <- rep(AF, each=2)

## compute equilibrium outer product covariance component
U <- am_covariance_structure(beta, AF, r)

## construct Correlation matrix
S <- diag(1/sqrt(AF_hap*(1-AF_hap)))
DPLR <- U%o%U
diag(DPLR) <- 1
C <- cov2cor(S%*%DPLR%*%S)

## draw multivariate Bernoulli haplotypes
H <- rb_unstr(n, AF_hap, C)

## convert to diploid genotypes
G <- H[,seq(1,ncol(H),2)] + H[,seq(2,ncol(H),2)]

## empirical allele frequencies vs target frequencies
emp_afs <- colMeans(G)/2
plot(AF, emp_afs)

## construct phenotype
heritable_y <-  G%*%beta_unscaled
nonheritable_y <-  rnorm(n, 0, sqrt(1-h2_0))
y <- heritable_y + nonheritable_y

## empirical h2 vs expected equilibrium h2
(emp_h2 <- var(heritable_y)/var(y))
h2_eq(r, h2_0)
```

Index

am_covariance_structure, 2
am_equilibrium_parameters, 2
am_simulate, 3

h2_eq *(*am_equilibrium_parameters*)*, 2

rb_dplr, 5
rb_unstr, 6
rg_eq *(*am_equilibrium_parameters*)*, 2

vg_eq *(*am_equilibrium_parameters*)*, 2