Package 'rBahadur'

July 23, 2026

Title Assortative Mating Simulation and Multivariate Bernoulli Variates

Version 1.1.0

Description Simulation of phenotype / genotype data under assortative and disassortative mating. Includes functions for generating Bahadur order-2 multivariate Bernoulli variables with general and diagonal-plus-low-rank correlation structures, and for writing simulated genotypes to disk as binary int8 or PLINK bed files. Further details are provided in: Border and Malik (2023) <doi:10.1186/s12859-023-05442-6>.

URL https://github.com/border-lab/rBahadur

BugReports https://github.com/border-lab/rBahadur/issues

License GPL (>= 3)

Encoding UTF-8

Language en-US

Depends R (>= 3.3.0), stats

Imports utils

SystemRequirements curl, gzip (providing zcat), awk

RoxygenNote 7.3.1

Suggests testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

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

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

Repository CRAN

Date/Publication 2026-07-23 11:00:14 UTC

Contents

`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, in the open interval (-1, 1). Negative values correspond to disassortative mating.

Details

For `r > 0` the low-rank term is added and `attr(U, "sign")` is 1. For `r < 0` it is subtracted and the attribute is -1, because a positive semidefinite rank-one term can only increase genetic variance whereas disassortative mating reduces it. The returned vector in that case is the analytic continuation of the positive branch, which is purely imaginary. For `r = 0` the vector is zero, since panmixia induces no disequilibrium.

Value

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

Warning, dropping the sign attribute

The returned vector carries `attr(U, "sign")`, which records whether the rank-one term is added or subtracted. Subsetting or coercing the vector, including with `as.vector()`, `c()`, or `U[i]`, drops that attribute. Because `rb_dplr()` falls back to `sign = 1` when the attribute is absent, this silently turns a disassortative structure into an assortative one, with no error at any point. If you manipulate `U` before passing it to `rb_dplr()`, pass `sign = -1` explicitly for `r < 0` rather than relying on the attribute to survive.

Feasibility under negative assortment

Disassortative mating leaves the Bahadur order-2 feasible region sooner than assortative mating does. The returned vector can satisfy the discriminant condition checked here and still drive `rb_dplr()` outside [0, 1] during sampling, because feasibility there is a property of the realized draws rather than of the parameters alone. Infeasibility becomes more likely as `n` grows, since it only takes one individual to fall outside the region. Positive `r` was feasible in every configuration tested. Negative `r` was not: at `h2_0 = 0.5` and 1500 causal variants, `r = -0.3` sampled reliably at 4000 individuals, while `r = -0.4` sampled reliably at 2000 individuals but failed for some seeds at 4000. Raising `min_MAF` widens the envelope. If `rb_dplr()` reports infeasible probabilities, reduce the magnitude of `r`, raise `min_MAF`, or increase the number of causal variants.

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, with values in (-1, 1)
h2_0	generation zero (panmictic) heritability, with values in [0, 1)
vg_0	non-negative 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_mosaic *Simulate equilibrium assortative mating with realistic local LD*

Description

Combines the global linkage disequilibrium induced by assortative mating with the local linkage disequilibrium induced by limited recombination. Causal variants are drawn with rb_dplr(), giving the dense genome-wide structure assortative mating produces, and the remaining markers are filled in by copying contiguous haplotype blocks from a reference panel, giving realistic short-range structure. Block boundaries are sampled from the panel's genetic map, so breakpoints concentrate where recombination is high.

Usage

```
am_mosaic(
  h2_0,
  r,
  n,
  panel,
  causal_idx = NULL,
  m = NULL,
  path = NULL,
  format = c("individual", "variant", "bed"),
  batch_size = NULL
)
```

Arguments

`h2_0`	generation zero (panmictic) heritability, in `(0, 1)`
`r`	cross-mate phenotypic correlation, in the open interval (-1, 1). Negative values give disassortative mating.
`n`	positive whole-number count of individuals to simulate
`panel`	reference panel: a list with `haplotypes` (a haplotypes by markers matrix of 0 and 1, optionally stored as `raw`), `pos` (strictly increasing base pair positions), and optionally `cM` (genetic map position of each marker), `chrom`, `id`, `ref`, and `alt`. Marker metadata can have length one or one entry per marker and is carried into PLINK `.bim` output. Haplotype value 0 denotes the reference allele and value 1 the alternate allele. Without `cM`, breakpoints are drawn uniformly in physical distance. See `kg_reference()` for the bundled example.
`causal_idx`	integer indices of the markers to treat as causal. If `NULL`, `m` evenly spaced markers are used.
`m`	whole-number count of causal variants, used only when `causal_idx` is `NULL`
`path`, `format`, `batch_size`	streaming options, exactly as in `am_simulate()`. With `path = NULL` the genotype matrix is returned in memory; otherwise it is streamed to disk and omitted from the result.

Details

Genotypes at the causal loci are exactly what `rb_dplr()` drew, so the equilibrium relationships in `h2_eq()` and `vg_eq()` hold there, while surrounding markers inherit the panel's correlation structure. Combining the two rather than approximating them jointly sidesteps the feasibility limit in `am_covariance_structure()`: representing strong local LD and genome-wide assortative mating in a single Bahadur order-2 distribution is generally not possible.

The equilibrium formulas are large-locus results. Calls with fewer than 50 causal variants are allowed but warn because realized variances can differ materially from the targets. Set `options(rBahadur.warn_small_m = FALSE)` to silence this warning after deciding that the finite-locus approximation is appropriate.

Each block is copied from one panel haplotype, so within a block the simulated data reproduces panel LD exactly, while correlation across a breakpoint is broken apart from what the causal variants carry. More causal variants therefore means more, shorter blocks.

Because the panel is finite, the simulated data cannot contain haplotypes the panel does not, and a small panel will show inflated identity by descent between simulated individuals.

Value

A list with `y`, `g`, `AF` (allele frequencies at the causal loci), `beta_std`, `beta_raw`, `causal_idx`, and `pos`. With `path = NULL` it also carries `X`, an `n` by `p` integer matrix of diploid genotypes at every panel marker. With `path` supplied, `X` is omitted and `path`, `format`, `n`, `m`, `h2_0`, and `r` are added, where `m` counts all panel markers written rather than only the causal ones.

See Also

`am_simulate()` for the unlinked-loci case, and `kg_reference()` for the bundled 1000 Genomes panel.

Examples

```
panel <- kg_reference()
sim <- am_mosaic(h2_0 = 0.5, r = 0.4, n = 50, panel = panel, m = 50)
dim(sim$X)

## neighbouring markers are correlated because they are copied together,
## which is the local LD that am_simulate() cannot produce
j <- sim$causal_idx[10]
cor(sim$X[, j], sim$X[, j + 1])
```

---

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,
  path = NULL,
  format = c("individual", "variant", "bed"),
  batch_size = NULL
)
```

Arguments

h2_0	generation zero (panmictic) heritability, in (0, 1)
r	cross-mate phenotypic correlation, in the open interval (-1, 1). Negative values correspond to disassortative mating.
m	number of biallelic causal variants; a whole number of at least 2
n	positive whole-number 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
path	(optional) file prefix. If supplied, genotypes are streamed to disk in batches rather than returned in memory, and X is omitted from the result. Writes <path>.int8 for format "individual" or "variant", or <path>.bed plus <path>.bim and <path>.fam for "bed", and in every case also writes <path>.meta and <path>.rds.
format	on-disk layout when path is supplied. "individual" (the default) stores each individual's variants contiguously in one byte per genotype, "variant" stores each variant's individuals contiguously, and "bed" writes a variant-major PLINK binary file at two bits per genotype alongside .bim and .fam.
batch_size	(optional) number of individuals per batch for "individual", or variants per block for "variant" and "bed". Defaults to a value targeting a working buffer of roughly 128 MB, but actual peak memory use is roughly twice that: the "individual" branch also holds the diploid Xb and its transposed copy, and the "variant"/"bed" branches also copy the buffer passed to the callback and build a double-precision Xb from it.

Details

The "variant" and "bed" layouts stream over loci and reproduce the in-memory genotypes exactly under a given seed at any batch_size. The "individual" layout streams over people and matches only when batch_size >= n, because a batch of rows is not contiguous in R's column-major random draw. haplotypes = TRUE cannot be combined with path.

The equilibrium formulas are large-locus results. Calls with fewer than 50 causal variants are allowed but warn because realized variances can differ materially from the targets, especially when m is very small. Set options(rBahadur.warn_small_m = FALSE) to silence this warning after deciding that the finite-locus approximation is appropriate.

Value

A list. Without path it contains y, g, X, AF, beta_std, beta_raw, and H when haplotypes is TRUE. With path it is returned invisibly, omits X, and adds path, format, n, m, h2_0, r, and min_MAF; these are also saved in <path>.rds as the call's provenance record.

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)

## stream genotypes to disk instead of holding them in memory
p <- file.path(tempdir(), "am_sim")
meta <- am_simulate(h2_0, r, m, n, path = p, format = "variant")
dim(read_genotypes(p))

## disassortative mating
neg <- am_simulate(h2_0, -0.3, m, n)
var(neg$g) < var(sim_dat$g)
```

---

download_1kg_panel *Download a 1000 Genomes region and build a reference panel*

Description

Convenience wrapper that fetches a window of phased 1000 Genomes data along with a genetic map and hands both to `vcf_to_panel()`. It exists so the vignette's workflow can be reproduced at realistic scale; it requires network access and downloads a large file, so it is not run in examples or tests.

Usage

```
download_1kg_panel(
  chrom = "22",
  start = 2e+07,
  end = 2.1e+07,
  dest = tempdir(),
  ...
)
```

Arguments

`chrom`	autosome, as a string from `"1"` through `"22"`
`start`, `end`	base pair bounds of the region to keep
`dest`	directory in which to cache downloads. Defaults to a temporary directory.
`...`	further arguments passed to `vcf_to_panel()`, such as `min_maf` and `max_samples`

Details

Streams the VCF and stops reading once `end` is passed, so a small window costs far less than the whole chromosome. The data are the GRCh38 phased biallelic release, and the genetic map is the GRCh38 PLINK map distributed with BEAGLE. A new cache file is published only after the stream reaches a variant beyond `end`, preventing an interrupted download from being reused as a complete region.

Value

A panel list suitable for `am_mosaic()`.

Examples

```
## requires network access
if (interactive()) {
  panel <- download_1kg_panel("22", 20e6, 20.5e6, max_samples = 200)
  dim(panel$haplotypes)
}
```

---

kg_reference *Load the bundled 1000 Genomes reference panel*

Description

A small real reference panel, included so that examples, tests, and the vignette run offline. It covers a 1 Mb window of chromosome 22 and is intended for demonstration rather than analysis; use `vcf_to_panel()` or `download_1kg_panel()` to build a panel at realistic scale.

Usage

```
kg_reference()
```

Value

A list with `haplotypes` (520 by 2500 raw matrix of 0/1), `pos`, `cM`, `chrom`, `build`, and `source`.

Source

Phase 3 of the 1000 Genomes Project, GRCh38, chromosome 22 positions 20,000,000 to 21,000,000, restricted to biallelic single nucleotide variants with minor allele frequency at least 0.05 in the first 260 samples. Genetic map positions are interpolated from the GRCh38 PLINK maps distributed with BEAGLE.

Examples

```
panel <- kg_reference()
dim(panel$haplotypes)
range(panel$pos)

## realistic local LD is the point: correlation decays with distance
H <- matrix(as.integer(panel$haplotypes), nrow = nrow(panel$haplotypes))
cor(H[, 1], H[, 2])^2
cor(H[, 1], H[, 2000])^2
```

`rbahadur_cli_path` *Path to the* `rbahadur` *command line script*

Description

Returns the location of the executable shipped with the package, so it can be placed on the search path.

Usage

```
rbahadur_cli_path()
```

Value

A single string giving the path to the script, or `""` if the package was installed without it.

Examples

```
rbahadur_cli_path()

## to put it on your PATH:
## ln -s $(Rscript -e 'cat(rBahadur::rbahadur_cli_path())') ~/bin/rbahadur
```

`rbahadur_main` *Run the* `rbahadur` *command line interface*

Description

Dispatches a vector of command line arguments. This is what the shipped `rbahadur` script calls; it is exported so the interface can be driven from R and tested. It returns an exit status rather than terminating the session.

Usage

```
rbahadur_main(args = commandArgs(trailingOnly = TRUE))
```

Arguments

`args`	character vector of command line arguments, as returned by `commandArgs(trailingOnly = TRUE)`

Value

Invisibly, an integer exit status: `0` on success, `1` for a usage error, and `2` for a runtime failure. Usage text goes to stdout when requested and to stderr when the invocation was wrong.

Examples

```
p <- file.path(tempdir(), "cli_example")
rbahadur_main(c("simulate", "--h2", "0.5", "--r", "0.3",
                "--m", "50", "--n", "40", "--out", p, "--quiet"))
rbahadur_main(c("info", p))
```

`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, sign = NULL)
```

Arguments

`n`	positive whole-number count of observations
`mu`	non-empty vector of means strictly between 0 and 1
`U`	finite outer-product component vector, with the same length as `mu`
`sign`	either 1 or -1, selecting $C = D + UU^T$ or $C = D - UU^T$. Defaults to `attr(U, "sign")` when present and to 1 otherwise, so vectors from `am_covariance_structure()` carry the correct structure automatically.

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'.

Warning, dropping the sign attribute

`U` vectors produced by `am_covariance_structure()` carry `attr(U, "sign")`. Subsetting or coercing `U`, including with `as.vector()`, `c()`, or `U[i]`, drops that attribute. Because `sign` here falls back to `+1` when the attribute is absent, doing so silently converts a disassortative structure into an assortative one, with no error at any point. If you manipulate `U` before calling `rb_dplr()`, pass `sign = -1` explicitly rather than relying on the attribute to survive.

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	positive whole-number count of observations
mu	non-empty vector of means strictly between 0 and 1
C	finite symmetric correlation matrix matching the length of mu

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 = 30; n = 100; 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)
```

`read_genotypes` *Read genotypes from a binary file written by* `write_genotypes()`

Description

Read genotypes from a binary file written by `write_genotypes()`

Usage

```
read_genotypes(path)
```

Arguments

`path`	file prefix, the same value passed to `write_genotypes()`

Value

An integer matrix with individuals in rows and variants in columns, reconstructed to the same orientation regardless of the on-disk layout.

Examples

```
X <- matrix(sample(0:2, 20, replace = TRUE), nrow = 4)
p <- file.path(tempdir(), "example_genotypes")
write_genotypes(X, p, format = "variant")
read_genotypes(p)
```

`vcf_to_panel` *Build a reference panel from a VCF*

Description

Reads a biallelic VCF and returns a panel in the form `am_mosaic()` expects. Phased genotypes such as `0|1` are strongly recommended because the mosaic copies haplotypes rather than genotypes. Unphased genotypes such as `0/1` are accepted with a warning and their written allele order is treated as if it were phase.

Usage

```
vcf_to_panel(
  vcf,
  map = NULL,
  min_maf = 0.05,
  max_markers = NULL,
  max_samples = NULL
)
```

Arguments

`vcf`	path to a VCF, optionally gzipped
`map`	optional path to a PLINK-format genetic map with columns chromosome, marker, centimorgans, and base pair position. Without one, the panel carries no `cM` and `am_mosaic()` places breakpoints uniformly in physical distance.
`min_maf`	drop markers whose minor allele frequency in the panel falls below this. Must lie in `[0, 0.5]`; defaults to 0.05.
`max_markers`	if given, thin evenly to at most this many markers
`max_samples`	if given, keep only the first this many samples

Value

A panel list suitable for `am_mosaic()`, including chromosome, ID, reference allele, and alternate allele metadata for each retained marker.

Examples

```
## a tiny phased VCF written on the fly
vcf <- tempfile(fileext = ".vcf")
writeLines(c(
  "##fileformat=VCFv4.2",
  paste(c("#CHROM", "POS", "ID", "REF", "ALT", "QUAL", "FILTER", "INFO",
          "FORMAT", "s1", "s2", "s3", "s4"), collapse = "\t"),
  paste(c("22", "100", ".", "A", "G", ".", ".", ".", "GT",
          "0|1", "1|1", "0|0", "1|0"), collapse = "\t"),
  paste(c("22", "200", ".", "C", "T", ".", ".", ".", "GT",
          "1|0", "0|1", "1|1", "0|0"), collapse = "\t")), vcf)
panel <- vcf_to_panel(vcf, min_maf = 0)
dim(panel$haplotypes)
```

---

write_genotypes *Write genotypes to a binary file*

Description

Write genotypes to a binary file

Usage

```
write_genotypes(X, path, format = c("individual", "variant", "bed"))
```

Arguments

X	an integer matrix of genotypes with individuals in rows and variants in columns, taking values 0, 1, or 2
path	non-empty file prefix in an existing directory. Layout "individual" and "variant" write <path>.int8; "bed" writes <path>.bed plus <path>.bim and <path>.fam. All layouts write <path>.meta.
format	on-disk layout. "individual" (the default) stores each individual's variants contiguously, "variant" stores each variant's individuals contiguously, and "bed" writes a variant-major PLINK binary file at two bits per genotype.

Details

Because this function receives no marker annotations, PLINK .bim output uses placeholder chromosome, position, and allele values. PLINK files written by am_mosaic() instead preserve the reference panel's available chromosome, physical position, genetic map, ID, and allele metadata.

Value

path, invisibly.

Examples

```
X <- matrix(sample(0:2, 20, replace = TRUE), nrow = 4)
p <- file.path(tempdir(), "example_genotypes")
write_genotypes(X, p)
identical(read_genotypes(p), X)
```

Index