Package: ldsep (via r-universe)

November 1, 2026

Title Linkage Disequilibrium Shrinkage Estimation for Polyploids

Version 2.1.7

Description Estimate haplotypic or composite pairwise linkage disequilibrium (LD) in polyploids, using either genotypes or genotype likelihoods. Support is provided to estimate the popular measures of LD: the LD coefficient D, the standardized LD coefficient D', and the Pearson correlation coefficient r. All estimates are returned with corresponding standard errors. These estimates and standard errors can then be used for shrinkage estimation. The main functions are ldfast(), ldest(), mldest(), sldest(), plot.lddf(), format_lddf(), and ldshrink(). Details of the methods are available in Gerard (2021a) <doi:10.1111/1755-0998.13349> and Gerard (2021b) <doi:10.1038/s41437-021-00462-5>.

License GPL (>= 3)

BugReports https://github.com/dcgerard/ldsep/issues

URL https://dcgerard.github.io/ldsep/

Encoding UTF-8

LazyData true

LinkingTo Rcpp, RcppArmadillo

Imports Rcpp, foreach, doParallel, ashr, corrplot, lpSolve, abind, matrixStats

Suggests testthat, covr, knitr, rmarkdown, updog (>= 2.0.2), VariantAnnotation

Depends R (>= 2.10)

VignetteBuilder knitr

Config/roxygen2/version 8.1.0

NeedsCompilation yes

Author David Gerard [aut, cre] (ORCID: <https://orcid.org/0000-0001-9450-5023>)

Maintainer David Gerard <gerard.1787@gmail.com>

Repository https://cran.r-universe.dev

Date/Publication 2026-09-09 20:40:02 UTC

RemoteUrl https://github.com/cran/ldsep

RemoteRef HEAD

RemoteSha bb8164e47fbf51ee4ff67e0d168871192b399809

Contents

Dprime . . . 2
format_lddf . . . 3
get_prob_array . . . 4
gl_to_gp . . . 5
glike . . . 6
gp . . . 7
hsm . . . 7
is.lddf . . . 8
ldest . . . 9
ldest_comp . . . 13
ldest_hap . . . 16
ldfast . . . 19
ldshrink . . . 22
mldest . . . 23
pbnorm_dist . . . 26
plot.lddf . . . 27
pvcalc . . . 28
slcor . . . 29
sldest . . . 30
uit . . . 32
zshrink . . . 33

Index **35**

`Dprime` *Get the standardized composite D'.*

Description

This function will either standardize by the maximum covariance conditional on the marginal genotype distribution, or by the maximum covariance conditional on the marginal allele frequencies.

Usage

```
Dprime(qmat, type = c("allele", "geno"), constrain = FALSE)
```

Arguments

qmat	The observed joint genotype distribution.
type	Should we condition on the marginal genotype distribution (type = "geno"), or should we condition on the allele frequency (type = "allele")?
constrain	A logical. This option is only applicable when type = "allele". Should return an value that is equal to D' under HWE (FALSE) or a value that is constrained to lie between -1 and 1 (TRUE)? Defaults to FALSE.

Details

Note that when type = "allele" and constrain = FALSE, the resulting D' is constrained to fall between -K and K, where K is the ploidy of the species. However, under HWE, this measure is equal to haplotypic D'. Using constrain = TRUE will result in a measure that is constrained to lie between -1 and 1, but it will not equal haplotypic D' under HWE.

Using type = "geno" is its own thing and will not equal D' generally under HWE. When type = "geno", then the the constrain parameter has no effect.

Value

A vector of length 2. The first element is the estimated D'. The second element is the normalization used.

Author(s)

David Gerard

Examples

```
K <- 6
qmat <- matrix(stats::runif((K+1)^2), nrow = K+1)
qmat <- qmat / sum(qmat)
Dprime(qmat, type = "geno")
Dprime(qmat, type = "allele")
```

---

format_lddf *Format an element of* mldest() *or* sldest() *into an upper-triangular matrix.*

Description

Formats the LD estimates and standard errors output from running mldest() or sldest() into a more conventional upper-triangular matrix.

Usage

```
format_lddf(obj, element = "r2")
```

Arguments

obj	An object of class lddf, usually output from running either mldest() or sldest().
element	Which element in obj should we format into an upper-triangular matrix?

Value

A matrix of the selected elements. Only the upper-triangle of the matrix is filled. The lower-triangle and the diagonal are NA's.

Author(s)

David Gerard

Examples

```
set.seed(1)

## Simulate genotypes when true correlation is 0
nloci <- 5
nind  <- 100
K <- 6
nc <- 1
genomat <- matrix(sample(0:K, nind * nloci, TRUE), nrow = nloci)

## Haplotypic LD estimates
lddf <- mldest(geno = genomat,
               K = K,
               nc = nc,
               type = "hap")

## Obtain the D estimates in matrix form
Dmat <- format_lddf(obj = lddf, element = "D")
Dmat
```

---

get_prob_array *Obtain the distribution of genotypes given haplotype frequencies under HWE*

Description

This function will calculate the (log) probabilities for all genotype combinations at two loci given just the haplotype frequencies. This is under the assumptions of HWE.

Usage

```
get_prob_array(K, prob, log_p = TRUE)
```

Arguments

K	The ploidy of the species.
prob	Haplotype frequencies in the order of (ab, Ab, aB, AB).
log_p	A logical. Should we return the log-probabilities (TRUE) or the probabilities (FALSE). Defaults to TRUE.

Value

Element (i, j) is the (log) probability of genotype i-1 at locus 1 and genotype j-1 at locus 2.

Author(s)

David Gerard

Examples

```
get_prob_array(K = 6, prob = c(0.1, 0.2, 0.3, 0.4), log_p = FALSE)
```

---

gl_to_gp *Normalize genotype likelihoods to posterior probabilities.*

Description

This will take genotype log-likelihoods and normalize them to sum to one. This corresponds to using a naive discrete uniform prior over the genotypes. It is not generally recommended that you use this function.

Usage

```
gl_to_gp(gl)
```

Arguments

gl	A three dimensional array of genotype *log*-likelihoods. Element gl[i, j, k] is the genotype log-likelihood of dosage k for individual j at SNP i.

Value

A three-dimensional array, of the same dimensions as gl, containing the posterior probabilities of each dosage.

Author(s)

David Gerard

Examples

```
data("glike")
class(glike)
dim(glike)
gl_to_gp(glike)
```

`glike` *Genotype log-likelihoods from* `uit`

Description

Contains an array of genotype log-likelihoods from the `uit` dataset. Element `gp[i, j, k]` is the log-likelihood of dosage `k-1` for individual `j` at SNP `i`.

Usage

```
glike
```

Format

A three-dimensional `array` object.

Source

doi:10.1371/journal.pone.0062355

References

- Uitdewilligen, Jan GAML, Anne-Marie A. Wolters, B. Bjorn, Theo JA Borm, Richard GF Visser, and Herman J. Van Eck. "A next-generation sequencing method for genotyping-by-sequencing of highly heterozygous autotetraploid potato." *PloS one* 8, no. 5 (2013): e62355. doi:10.1371/journal.pone.0062355

See Also

`uit` for the full `multidog()` fit.

gp — *Posterior probabilities from* `uit`

Description

Contains an array of posterior probabilities of the genotypes from the `uit` dataset. Element `gp[i, j, k]` is the posterior probability of dosage `k-1` for individual `j` at SNP `i`.

Usage

```
gp
```

Format

A three-dimensional `array` object.

Source

doi:10.1371/journal.pone.0062355

References

- Uitdewilligen, Jan GAML, Anne-Marie A. Wolters, B. Bjorn, Theo JA Borm, Richard GF Visser, and Herman J. Van Eck. "A next-generation sequencing method for genotyping-by-sequencing of highly heterozygous autotetraploid potato." *PloS one* 8, no. 5 (2013): e62355. doi:10.1371/journal.pone.0062355

See Also

`uit` for the full `multidog()` fit.

hsm — *Half-sample mode estimator*

Description

Implementation of the half-sample mode estimator of Robertson & Cryer (1974).

Usage

```
hsm(x, k = 0.5)
```

Arguments

`x`	A vector of values
`k`	Fraction to discard each iteration.

Details

Notes that in the case of ties, this version automatically chooses the smallest of the ties. So, don't use this with ties.

Value

The mode estimate.

Author(s)

David Gerard

References

- Robertson, T., & Cryer, J. D. (1974). An iterative procedure for estimating the mode. Journal of the American Statistical Association, 69(348), 1012-1016. doi:10.1080/01621459.1974.10480246

Examples

```
set.seed(1)
x <- stats::rnorm(100)
hsm(x)
```

---

is.lddf *Tests if an argument is a* lddf *object.*

Description

Tests if an argument is a `lddf` object.

Usage

```
is.lddf(x)
```

Arguments

`x`	Anything.

Value

A logical. `TRUE` if `x` is a `lddf` object, and `FALSE` otherwise.

Author(s)

David Gerard

Examples

```
is.lddf("anything")
# FALSE
```

`ldest` *Pairwise LD estimation in polyploids.*

Description

Estimates either haplotypic or composite measures of LD using either genotypes are genotype likelihoods via maximum likelihood. The usual measures of LD are estimated (D, D', and r) along with the Fisher-z transformation of r (called "z"). All estimates are returned with standard errors. See Gerard (2021) for details.

Usage

```
ldest(
  ga,
  gb,
  K,
  se = TRUE,
  type = c("hap", "comp"),
  model = c("norm", "flex"),
  pen = ifelse(type == "hap", 2, 1)
)
```

Arguments

`ga`	One of two possible inputs: 1. A vector of counts, containing the genotypes for each individual at the first locus. When `type = "comp"`, the vector of genotypes may be continuous (e.g. the posterior mean genotype). 2. A matrix of genotype log-likelihoods at the first locus. The rows index the individuals and the columns index the genotypes. That is `ga[i, j]` is the genotype likelihood of individual `i` for genotype `j-1`.
`gb`	One of two possible inputs: 1. A vector of counts, containing the genotypes for each individual at the second locus. When `type = "comp"`, the vector of genotypes may be continuous (e.g. the posterior mean genotype). 2. A matrix of genotype log-likelihoods at the second locus. The rows index the individuals and the columns index the genotypes. That is `gb[i, j]` is the genotype likelihood of individual `i` for genotype `j-1`.
`K`	The ploidy of the species. Assumed to be the same for all individuals.

se A logical. Should we calculate standard errors (TRUE) or not (FALSE). Calculating standard errors can be really slow when type = "comp", model = "flex", and when using genotype likelihoods. Otherwise, standard error calculations should be pretty fast.

type The type of LD to calculate. The available types are haplotypic LD (type = "hap") or composite LD (type = "comp"). Haplotypic LD is only appropriate for autopolyploids when the individuals are in Hardy-Weinberg equilibrium (HWE). The composite measures of LD are always applicable, and consistently estimate the usual measures of LD when HWE is fulfilled in autopolyploids. However, when HWE is not fulfilled, interpreting the composite measures of LD could be a little tricky.

model When type = "comp" and using genotype likelihoods, should we use the proportional bivariate normal model to estimate the genotype distribution (model = "norm"), or the general categorical distribution (model = "flex")? Defaults to "norm".

pen The penalty to be applied to the likelihood. You can think about this as the prior sample size. Should be greater than 1. Does not apply if model = "norm", type = "comp", and using genotype likelihoods. Also does not apply when type = "comp" and using genotypes.

Value

A vector with some or all of the following elements:

D The estimate of the LD coefficient.

D_se The standard error of the estimate of the LD coefficient.

r2 The estimate of the squared Pearson correlation.

r2_se The standard error of the estimate of the squared Pearson correlation.

r The estimate of the Pearson correlation.

r_se The standard error of the estimate of the Pearson correlation.

Dprime The estimate of the standardized LD coefficient. When type = "comp", this corresponds to the standardization where we fix allele frequencies.

Dprime_se The standard error of Dprime.

Dprimeg The estimate of the standardized LD coefficient. This corresponds to the standardization where we fix genotype frequencies.

Dprimeg_se The standard error of Dprimeg.

z The Fisher-z transformation of r.

z_se The standard error of the Fisher-z transformation of r.

p_ab The estimated haplotype frequency of ab. Only returned if estimating the haplotypic LD.

p_Ab The estimated haplotype frequency of Ab. Only returned if estimating the haplotypic LD.

p_aB The estimated haplotype frequency of aB. Only returned if estimating the haplotypic LD.

p_AB The estimated haplotype frequency of AB. Only returned if estimating the haplotypic LD.

q_ij The estimated frequency of genotype i at locus 1 and genotype j at locus 2. Only returned if estimating the composite LD.

n The number of individuals used to estimate pairwise LD.

Haplotypic LD

This section describes the methods used when `type = "hap"` is selected.

Haplotypic LD measures the association between two loci on the same haplotype. When haplotypes are known, estimating haplotypic LD is simple using just the haplotypic frequencies.

When haplotypes are not known, we can still estimate haplotypic frequencies using the genotypes or genotype likelihoods *in autopolyploids as long as Hardy-Weinberg equilibrium (HWE) is satisfied*. We do this via maximum likelihood using gradient ascent. Gradient ascent is performed over the unconstrained parameterization of the 3-simplex from Betancourt (2012). The estimated haplotype frequencies are then used to estimate haplotypic LD.

Standard errors are provided using standard maximum likelihood theory. In brief, the Hessian matrix of the log-likelihood is calculated at the MLE's of the haplotype frequencies. The negative inverse of this Hessian matrix is approximately the covariance matrix of the MLE's of the haplotype frequencies. Since all haplotypic LD measures are functions of the haplotype frequencies, we use the delta-method to obtain the standard errors for each LD estimate.

A Dirichlet(2,2,2,2) prior is placed over the frequencies of haplotypes 00, 01, 10, and 11. This corresponds to the "add two" rule of Agresti (1998). You can change this prior via the `pen` argument.

When you either do not have autopolyploids or when HWE is *not* satisfied, then the estimates using `type = "hap"` are nonsensical. However, the composite measures of LD are still applicable (see below).

Composite LD

This section describes the methods used when `type = "comp"` is selected.

When HWE is not satisfied, haplotype frequencies are not estimable. However, measures of association between two loci are still estimable. These associations may be caused by LD either on the same haplotype or between different haplotypes. Cockerham and Weir (1977) thus called such measures "composite" measures of LD.

When the genotypes are known, these composite measures have simple correspondences to well-known statistical measures of association. D is the covariance of genotypes between loci divided by the ploidy. r is the Pearson correlation of genotypes. D' is D divided by a term that involves only mean genotypes.

When genotypes are not known, we estimate the joint genotype frequencies and use these to estimate the composite measures of LD using genotype likelihoods. The distribution of genotypes is assumed to either follow a proportional bivariate normal model (by default) or a general categorical model.

These estimates of composite measures of LD estimate the haplotypic measures of LD when HWE is fulfilled, but are still applicable when HWE is not fulfilled.

When genotypes are known, standard errors are calculated using standard moment-based approaches. When genotypes are not known, standard errors are calculated using standard maximum likelihood theory, same as for the haplotypic LD estimates (see above), or using a bootstrap.

Author(s)

David Gerard

References

- Agresti, Alan, and Brent A. Coull. "Approximate is better than "exact" for interval estimation of binomial proportions." *The American Statistician* 52, no. 2 (1998): 119-126. doi:10.1080/00031305.1998.10480550
- Betancourt, Michael. "Cruising the simplex: Hamiltonian Monte Carlo and the Dirichlet distribution." In *AIP Conference Proceedings 31st*, vol. 1443, no. 1, pp. 157-164. American Institute of Physics, 2012. doi:10.1063/1.3703631
- Cockerham, C. Clark, and B. S. Weir. "Digenic descent measures for finite populations." *Genetics Research* 30, no. 2 (1977): 121-147. doi:10.1017/S0016672300017547
- Gerard, David. "Pairwise Linkage Disequilibrium Estimation for Polyploids." *Molecular Ecology Resources* 21, no. 4 (2021): 1230-1242. doi:10.1111/17550998.13349

See Also

`ldfast()` Fast, moment-based approach to LD estimation that also accounts for genotype uncertainty.

`mldest()` For calculating pairwise LD among all pairs of a collection of SNPs.

`sldest()` For calculating pairwise LD along a sliding window of SNPs.

`ldest_hap()` For the function that directly estimates haplotypic LD when HWE is fulfilled.

`ldest_comp()` For the function that directly estimates composite LD.

Examples

```
set.seed(1)
n <- 100 # sample size
K <- 6 # ploidy

## generate some fake genotypes when LD = 0.
ga <- stats::rbinom(n = n, size = K, prob = 0.5)
gb <- stats::rbinom(n = n, size = K, prob = 0.5)
head(ga)
head(gb)

## generate some fake genotype likelihoods when LD = 0.
gamat <- t(sapply(ga, stats::dnorm, x = 0:K, sd = 1, log = TRUE))
gbmat <- t(sapply(gb, stats::dnorm, x = 0:K, sd = 1, log = TRUE))
head(gamat)
head(gbmat)

## Haplotypic LD with genotypes
ldout1 <- ldest(ga = ga,
                gb = gb,
                K = K,
                type = "hap")
head(ldout1)

## Haplotypic LD with genotype likelihoods
ldout2 <- ldest(ga = gamat,
```

```
                gb = gbmat,
                K = K,
                type = "hap")
head(ldout2)

## Composite LD with genotypes
ldout3 <- ldest(ga = ga,
                gb = gb,
                K = K,
                type = "comp")
head(ldout3)

## Composite LD with genotype likelihoods and normal model
ldout4 <- ldest(ga = gamat,
                gb = gbmat,
                K = K,
                type = "comp",
                model = "norm")
head(ldout4)

## Composite LD with genotype likelihoods and general categorical model
ldout5 <- ldest(ga = gamat,
                gb = gbmat,
                K = K,
                type = "comp",
                model = "flex",
                se = FALSE)
head(ldout5)

ldout1[["D"]]
ldout2[["D"]]
ldout3[["D"]]
ldout4[["D"]]
ldout5[["D"]]
```

`ldest_comp` *Estimates of composite pairwise LD based either on genotype estimates or genotype likelihoods.*

Description

This function will estimate the composite LD between two loci, either using genotype estimates or using genotype likelihoods. The resulting measures of LD are generalizations of Burrow's "composite" LD measure.

Usage

```
ldest_comp(
  ga,
```

```
  gb,
  K,
  pen = 1,
  useboot = TRUE,
  nboot = 50,
  se = TRUE,
  model = c("norm", "flex")
)
```

Arguments

ga	One of two possible inputs: 1. A vector of counts, containing the genotypes for each individual at the first locus. When `type = "comp"`, the vector of genotypes may be continuous (e.g. the posterior mean genotype). 2. A matrix of genotype log-likelihoods at the first locus. The rows index the individuals and the columns index the genotypes. That is `ga[i, j]` is the genotype likelihood of individual `i` for genotype `j-1`.
gb	One of two possible inputs: 1. A vector of counts, containing the genotypes for each individual at the second locus. When `type = "comp"`, the vector of genotypes may be continuous (e.g. the posterior mean genotype). 2. A matrix of genotype log-likelihoods at the second locus. The rows index the individuals and the columns index the genotypes. That is `gb[i, j]` is the genotype likelihood of individual `i` for genotype `j-1`.
K	The ploidy of the species. Assumed to be the same for all individuals.
pen	The penalty to be applied to the likelihood. You can think about this as the prior sample size. Should be greater than 1. Does not apply if `model = "norm"`, `type = "comp"`, and using genotype likelihoods. Also does not apply when `type = "comp"` and using genotypes.
useboot	Should we use bootstrap standard errors `TRUE` or not `FALSE`? Only applicable if using genotype likelihoods and `model = "flex"`
nboot	The number of bootstrap iterations to use is `boot = TRUE`. Only applicable if using genotype likelihoods and `model = "flex"`.
se	A logical. Should we calculate standard errors (`TRUE`) or not (`FALSE`). Calculating standard errors can be really slow when `type = "comp"`, `model = "flex"`, and when using genotype likelihoods. Otherwise, standard error calculations should be pretty fast.
model	Should we assume the class of joint genotype distributions is from the proportional bivariate normal (`model = "norm"`) or from the general categorical distribution (`model = "flex"`). Only applicable if using genotype likelihoods.

Value

A vector with some or all of the following elements:

`D` The estimate of the LD coefficient.

`D_se` The standard error of the estimate of the LD coefficient.

`r2` The estimate of the squared Pearson correlation.

`r2_se` The standard error of the estimate of the squared Pearson correlation.

`r` The estimate of the Pearson correlation.

`r_se` The standard error of the estimate of the Pearson correlation.

`Dprime` The estimate of the standardized LD coefficient. When `type = "comp"`, this corresponds to the standardization where we fix allele frequencies.

`Dprime_se` The standard error of `Dprime`.

`Dprimeg` The estimate of the standardized LD coefficient. This corresponds to the standardization where we fix genotype frequencies.

`Dprimeg_se` The standard error of `Dprimeg`.

`z` The Fisher-z transformation of `r`.

`z_se` The standard error of the Fisher-z transformation of `r`.

`p_ab` The estimated haplotype frequency of ab. Only returned if estimating the haplotypic LD.

`p_Ab` The estimated haplotype frequency of Ab. Only returned if estimating the haplotypic LD.

`p_aB` The estimated haplotype frequency of aB. Only returned if estimating the haplotypic LD.

`p_AB` The estimated haplotype frequency of AB. Only returned if estimating the haplotypic LD.

`q_ij` The estimated frequency of genotype i at locus 1 and genotype j at locus 2. Only returned if estimating the composite LD.

`n` The number of individuals used to estimate pairwise LD.

Author(s)

David Gerard

Examples

```
set.seed(1)
n <- 100 # sample size
K <- 6 # ploidy

## generate some fake genotypes when LD = 0.
ga <- stats::rbinom(n = n, size = K, prob = 0.5)
gb <- stats::rbinom(n = n, size = K, prob = 0.5)
head(ga)
head(gb)

## generate some fake genotype likelihoods when LD = 0.
gamat <- t(sapply(ga, stats::dnorm, x = 0:K, sd = 1, log = TRUE))
gbmat <- t(sapply(gb, stats::dnorm, x = 0:K, sd = 1, log = TRUE))
head(gamat)
head(gbmat)

## Composite LD with genotypes
```

```
ldout1 <- ldest_comp(ga = ga,
                     gb = gb,
                     K = K)
head(ldout1)

## Composite LD with genotype likelihoods
ldout2 <- ldest_comp(ga = gamat,
                     gb = gbmat,
                     K = K,
                     se = FALSE,
                     model = "flex")
head(ldout2)

## Composite LD with genotype likelihoods and proportional bivariate normal
ldout3 <- ldest_comp(ga = gamat,
                     gb = gbmat,
                     K = K,
                     model = "norm")
head(ldout3)
```

`ldest_hap` *Estimate haplotypic pair-wise LD using either genotypes or genotype likelihoods.*

Description

Given genotype (allele dosage) or genotype likelihood data for each individual at a pair of loci, this function will calculate the maximum likelihood estimates and their corresponding asymptotic standard errors of some measures of linkage disequilibrium (LD): D, D', the Pearson correlation, the squared Pearson correlation, and the Fisher-z transformation of the Pearson correlation. This function can be used for both diploids and polyploids.

Usage

```
ldest_hap(
  ga,
  gb,
  K,
  reltol = 10^-8,
  nboot = 100,
  useboot = FALSE,
  pen = 2,
  grid_init = FALSE,
  se = TRUE
)
```

Arguments

ga	One of two possible inputs: 1. A vector of counts, containing the genotypes for each individual at the first locus. When type = "comp", the vector of genotypes may be continuous (e.g. the posterior mean genotype). 2. A matrix of genotype log-likelihoods at the first locus. The rows index the individuals and the columns index the genotypes. That is ga[i, j] is the genotype likelihood of individual i for genotype j-1.
gb	One of two possible inputs: 1. A vector of counts, containing the genotypes for each individual at the second locus. When type = "comp", the vector of genotypes may be continuous (e.g. the posterior mean genotype). 2. A matrix of genotype log-likelihoods at the second locus. The rows index the individuals and the columns index the genotypes. That is gb[i, j] is the genotype likelihood of individual i for genotype j-1.
K	The ploidy of the species. Assumed to be the same for all individuals.
reltol	The relative tolerance for the stopping criterion.
nboot	Sometimes, the MLE standard errors don't exist. So we use the bootstrap as a backup. nboot specifies the number of bootstrap iterations.
useboot	A logical. Optionally, you may always use the bootstrap to estimate the standard errors (TRUE). These will be more accurate but also much slower, so this defaults to FALSE. Only applicable if using genotype likelihoods.
pen	The penalty to be applied to the likelihood. You can think about this as the prior sample size. Should be greater than 1. Does not apply if model = "norm", type = "comp", and using genotype likelihoods. Also does not apply when type = "comp" and using genotypes.
grid_init	A logical. Should we initialize the gradient ascent at a grid of initial values (TRUE) or just initialize at one value corresponding to the simplex point rep(0.25, 4) (FALSE)?
se	A logical. Should we calculate standard errors (TRUE) or not (FALSE). Calculating standard errors can be really slow when type = "comp", model = "flex", and when using genotype likelihoods. Otherwise, standard error calculations should be pretty fast.

Details

Let A and a be the reference and alternative alleles, respectively, at locus 1. Let B and b be the reference and alternative alleles, respectively, at locus 2. Let paa, pAb, paB, and pAB be the frequencies of haplotypes ab, Ab, aB, and AB, respectively. Let pA = pAb + pAB and let pB = paB + pAB The ldest returns estimates of the following measures of LD.

- D: pAB - pA pB
- D': D / Dmax, where Dmax = min(pA pB, (1 - pA) (1 - pB)) if D < 0 and Dmax = min(pA (1 - pB), pA (1 - pB)) if D > 0
- r-squared: The squared Pearson correlation, r^2 = D^2 / (pA (1 - pA) pB (1 - pB))

- r: The Pearson correlation, r = D / sqrt(pA (1 - pA) pB (1 - pB))

Estimates are obtained via maximum likelihood under the assumption of Hardy-Weinberg equilibrium. The likelihood is calculated by integrating over the possible haplotypes for each pair of genotypes.

The resulting standard errors are based on the square roots of the inverse of the negative Fisher-information. This is from standard maximum likelihood theory. The Fisher-information is known to be biased low, so the actual standard errors are probably a little bigger for small n (n < 20). In some cases the Fisher-information matrix is singular, and so we in these cases we return a bootstrap estimate of the standard error.

The standard error estimate of the squared Pearson correlation is not valid when r^2 = 0.

In cases where either SNP is estimated to be monoallelic (`pA %in% c(0, 1)` or `pB %in% c(0, 1)`), this function will return LD estimates of `NA`.

Value

A vector with some or all of the following elements:

`D` The estimate of the LD coefficient.

`D_se` The standard error of the estimate of the LD coefficient.

`r2` The estimate of the squared Pearson correlation.

`r2_se` The standard error of the estimate of the squared Pearson correlation.

`r` The estimate of the Pearson correlation.

`r_se` The standard error of the estimate of the Pearson correlation.

`Dprime` The estimate of the standardized LD coefficient. When `type = "comp"`, this corresponds to the standardization where we fix allele frequencies.

`Dprime_se` The standard error of `Dprime`.

`Dprimeg` The estimate of the standardized LD coefficient. This corresponds to the standardization where we fix genotype frequencies.

`Dprimeg_se` The standard error of `Dprimeg`.

`z` The Fisher-z transformation of `r`.

`z_se` The standard error of the Fisher-z transformation of `r`.

`p_ab` The estimated haplotype frequency of ab. Only returned if estimating the haplotypic LD.

`p_Ab` The estimated haplotype frequency of Ab. Only returned if estimating the haplotypic LD.

`p_aB` The estimated haplotype frequency of aB. Only returned if estimating the haplotypic LD.

`p_AB` The estimated haplotype frequency of AB. Only returned if estimating the haplotypic LD.

`q_ij` The estimated frequency of genotype i at locus 1 and genotype j at locus 2. Only returned if estimating the composite LD.

`n` The number of individuals used to estimate pairwise LD.

Author(s)

David Gerard

Examples

```
set.seed(1)
n <- 100 # sample size
K <- 6 # ploidy

## generate some fake genotypes when LD = 0.
ga <- stats::rbinom(n = n, size = K, prob = 0.5)
gb <- stats::rbinom(n = n, size = K, prob = 0.5)
head(ga)
head(gb)

## generate some fake genotype likelihoods when LD = 0.
gamat <- t(sapply(ga, stats::dnorm, x = 0:K, sd = 1, log = TRUE))
gbmat <- t(sapply(gb, stats::dnorm, x = 0:K, sd = 1, log = TRUE))
head(gamat)
head(gbmat)

## Haplotypic LD with genotypes
ldout1 <- ldest_hap(ga = ga,
                    gb = gb,
                    K = K)
head(ldout1)

## Haplotypic LD with genotype likelihoods
ldout2 <- ldest_hap(ga = gamat,
                    gb = gbmat,
                    K = K)
head(ldout2)
```

`ldfast` *Fast bias-correction for LD Estimation*

Description

Estimates the reliability ratios from posterior marginal moments and uses these to correct the biases in linkage disequilibrium estimation caused by genotype uncertainty. These methods are described in Gerard (2021).

Usage

```
ldfast(
  gp,
  type = c("r", "r2", "z", "D", "Dprime"),
  shrinkrr = TRUE,
  se = TRUE,
  thresh = TRUE,
  upper = 10,
  mode = c("zero", "estimate"),
```

```
  win = NULL
)
```

Arguments

gp	A three-way array with dimensions SNPs by individuals by dosage. That is, gp[i, j, k] is the posterior probability of dosage k-1 for individual j at SNP i.
type	What LD measure should we estimate? "r" The Pearson correlation. "r2" The squared Pearson correlation. "z" The Fisher-z transformed Pearson correlation. "D" The LD coefficient. "Dprime" The standardized LD coefficient. Note that these are all *composite* measures of LD (see the description in ldest()).
shrinkrr	A logical. Should we use adaptive shrinkage (Stephens, 2016) to shrink the reliability ratios (TRUE) or keep the raw reliability ratios (FALSE). Defaults to TRUE.
se	Should we also return a matrix of standard errors (TRUE) or not (FALSE)? It is faster to not return standard errors. Defaults to TRUE.
thresh	A logical. Should we apply an upper bound on the reliability ratios (TRUE) or not (FALSE).
upper	The upper bound on the reliability ratios if thresh = TRUE. The default is a generous 10.
mode	A character. Only applies if shrinkrr = TRUE. When using hierarchical shrinkage on the log of the reliability ratios, should we use zero as the mode (mode = "zero") or estimate it using the procedure of Robertson and Cryer (1974) (mode = "estimate")?
win	A positive integer. The window size. This will constrain the correlations calculated to those +/- the window size. This will only improve speed if the window size is *much* less than the number of SNPs.

Value

A list with some or all of the following elements:

ldmat The bias-corrected LD matrix.

rr The estimated reliability ratio for each SNP. This is the multiplicative factor applied to the naive LD estimate for each SNP.

rr_raw The raw reliability ratios (for the covariance, not the correlation). Only returned if shrinkrr = TRUE.

rr_se The standard errors for the *log*-raw reliability ratios for each SNP. That is, we have sd(log(rr_raw)) ~ rr_se. Only returned if shrinkrr = TRUE.

semat A matrix of standard errors of the corresponding estimators of LD.

Details

Returns consistent and bias-corrected estimates of linkage disequilibrium. The usual measures of LD are implemented: D, D', r, r2, and z (Fisher-z of r). These are all *composite* measures of LD, not haplotypic measures of LD (see the description in `ldest()`). They are always appropriate measures of association between loci, but only correspond to haplotypic measures of LD when Hardy-Weinberg equilibrium is fulfilled in autopolyploids.

In order for these estimates to perform well, you need to use posterior genotype probabilities that have been calculated using adaptive priors, i.e. empirical/hierarchical Bayes approaches. There are many approaches that do this, such as `updog`, `polyRAD`, `fitPoly`, or `SuperMASSA`. Note that GATK uses a uniform prior, so would be inappropriate for use in `ldfast()`.

Calculating standard errors and performing hierarchical shrinkage of the reliability ratios are both rather slow operations compared to just raw method-of-moments based estimation for LD. If you don't need standard errors, you can double your speed by setting `se = FALSE`. It is not recommended that you disable the hierarchical shrinkage.

Due to sampling variability, the estimates sometime lie outside of the theoretical boundaries of the parameters being estimated. In such cases, we truncate the estimates at the boundary and return `NA` for the standard errors.

Mathematical formulation

Let

- r be the sample correlation of posterior mean genotypes between loci 1 and 2,
- $a1$ be the sample variance of posterior means at locus 1,
- $a2$ be the sample variance of posterior means at locus 2,
- $b1$ be the sample mean of posterior variances at locus 1, and
- $b2$ be the sample mean of posterior variances at locus 2.

Then the estimated Pearson correlation between the genotypes at loci 1 and 2 is

$$\sqrt{(a1 + b1)/a1}\sqrt{(a2 + b2)/a2}r.$$

All other LD calculations are based on this equation. In particular, the estimated genotype variances at loci 1 and 2 are $a1 + b1$ and $a2 + b2$, respectively, which can be used to calculate D and D'.

The reliability ratio for SNP i is defined by $(ai + bi)/ai$. By default, we apply `ash()` (Stephens, 2016) to the log of these reliability ratios before adjusting the Pearson correlation. Standard errors are required before using `ash()`, but these are easily obtained using the central limit theorem and the delta-method.

Author(s)

David Gerard

References

- Gerard, David. Scalable Bias-corrected Linkage Disequilibrium Estimation Under Genotype Uncertainty. *Heredity*, 127(4), 357–362, 2021. doi:10.1038/s41437021004625.
- T. Robertson and J. D. Cryer. An iterative procedure for estimating the mode. *Journal of the American Statistical Association*, 69(348):1012–1016, 1974. doi:10.1080/01621459.1974.10480246.
- M. Stephens. False discovery rates: a new deal. *Biostatistics*, 18(2):275–294, 10 2016. doi:10.1093/biostatistics/kxw041.

See Also

`ash()` Function used to perform hierarchical shrinkage on the log of the reliability ratios.

`ldest()`, `mldest()`, `sldest()` Maximum likelihood estimation of linkage disequilibrium.

Examples

```
data("gp")

ldout <- ldfast(gp, "r")
ldout$ldmat
ldout$rr
ldout$semat

ldout <- ldfast(gp, "D")
ldout$ldmat
ldout$rr
ldout$semat

ldout <- ldfast(gp, "Dprime")
ldout$ldmat
ldout$rr
ldout$semat
```

---

ldshrink *Obtain shrinkage estimates of correlation from output of* `mldest()` *or* `sldest()`.

Description

This will take the output of either `mldest()` or `sldest()`, shrink the Fisher-z transformed correlation estimates using `ash()` (Stephens, 2017; Dey and Stephens, 2018), then return the corresponding correlation estimates. You can obtain estimates of r^2 by just squaring these estimates.

Usage

```
ldshrink(obj, ...)
```

Arguments

obj	An object of class lddf, usually created using either mldest() or sldest().
...	Additional arguments to pass to ash().

Value

A correlation matrix.

Author(s)

David Gerard

References

- Stephens, Matthew. "False discovery rates: a new deal." Biostatistics 18, no. 2 (2017): 275-294.
- Dey, Kushal K., and Matthew Stephens. "CorShrink: Empirical Bayes shrinkage estimation of correlations, with applications." bioRxiv (2018): 368316.

---

mldest *Estimate all pair-wise LD's in a collection of SNPs using genotypes or genotype likelihoods.*

Description

This function is a wrapper to run ldest() for many pairs of SNPs. This takes a maximum likelihood approach to LD estimation. See ldfast() for a method-of-moments approach to LD estimation. Support is provided for parallelization through the foreach and doParallel packages. See Gerard (2021) for details.

Usage

```
mldest(
  geno,
  K,
  nc = 1,
  type = c("hap", "comp"),
  model = c("norm", "flex"),
  pen = ifelse(type == "hap", 2, 1),
  se = TRUE
)
```

Arguments

geno	One of two possible inputs: • A matrix of genotypes (allele dosages). The rows index the SNPs and the columns index the individuals. That is, `genomat[i, j]` is the allele dosage for individual `j` in SNP `i`. When `type = "comp"`, the dosages are allowed to be continuous (e.g. posterior mean genotypes). • A three-way array of genotype *log*-likelihoods. The first dimension indexes the SNPs, the second dimension indexes the individuals, and the third dimension indexes the genotypes. That is, `genolike_array[i, j, k]` is the genotype log-likelihood at SNP `i` for individual `j` and dosage `k - 1`.
K	The ploidy of the species. Assumed to be the same for all individuals.
nc	The number of computing cores to use. This should never be more than the number of cores available in your computing environment. You can determine the maximum number of available cores by running `parallel::detectCores()` in R. This is probably fine for a personal computer, but some environments are only able to use fewer. Ask your admins if you are unsure.
type	The type of LD to calculate. The available types are haplotypic LD (`type = "hap"`) or composite LD (`type = "comp"`). Haplotypic LD is only appropriate for autopolyploids when the individuals are in Hardy-Weinberg equilibrium (HWE). The composite measures of LD are always applicable, and consistently estimate the usual measures of LD when HWE is fulfilled in autopolyploids. However, when HWE is not fulfilled, interpreting the composite measures of LD could be a little tricky.
model	When `type = "comp"` and using genotype likelihoods, should we use the proportional bivariate normal model to estimate the genotype distribution (`model = "norm"`), or the general categorical distribution (`model = "flex"`)? Defaults to `"norm"`.
pen	The penalty to be applied to the likelihood. You can think about this as the prior sample size. Should be greater than 1. Does not apply if `model = "norm"`, `type = "comp"`, and using genotype likelihoods. Also does not apply when `type = "comp"` and using genotypes.
se	A logical. Should we calculate standard errors (`TRUE`) or not (`FALSE`). Calculating standard errors can be really slow when `type = "comp"`, `model = "flex"`, and when using genotype likelihoods. Otherwise, standard error calculations should be pretty fast.

Details

See `ldest()` for details on the different types of LD estimators supported.

Value

A data frame of class `c("lddf", "data.frame")` with some or all of the following elements:

`i` The index of the first SNP.

`j` The index of the second SNP.

`snpi` The row name corresponding to SNP `i`, if row names are provided.

`snpj` The row name corresponding to SNP `j`, if row names are provided.

`D` The estimate of the LD coefficient.

`D_se` The standard error of the estimate of the LD coefficient.

`r2` The estimate of the squared Pearson correlation.

`r2_se` The standard error of the estimate of the squared Pearson correlation.

`r` The estimate of the Pearson correlation.

`r_se` The standard error of the estimate of the Pearson correlation.

`Dprime` The estimate of the standardized LD coefficient. When `type = "comp"`, this corresponds to the standardization where we fix allele frequencies.

`Dprime_se` The standard error of `Dprime`.

`Dprimeg` The estimate of the standardized LD coefficient. This corresponds to the standardization where we fix genotype frequencies.

`Dprimeg_se` The standard error of `Dprimeg`.

`z` The Fisher-z transformation of `r`.

`z_se` The standard error of the Fisher-z transformation of `r`.

`p_ab` The estimated haplotype frequency of ab. Only returned if estimating the haplotypic LD.

`p_Ab` The estimated haplotype frequency of Ab. Only returned if estimating the haplotypic LD.

`p_aB` The estimated haplotype frequency of aB. Only returned if estimating the haplotypic LD.

`p_AB` The estimated haplotype frequency of AB. Only returned if estimating the haplotypic LD.

`q_ij` The estimated frequency of genotype i at locus 1 and genotype j at locus 2. Only returned if estimating the composite LD.

`n` The number of individuals used to estimate pairwise LD.

Author(s)

David Gerard

References

- Gerard, David. "Pairwise Linkage Disequilibrium Estimation for Polyploids." *Molecular Ecology Resources* 21, no. 4 (2021): 1230-1242. doi:10.1111/17550998.13349

See Also

`ldfast()` Fast, moment-based approach to LD estimation that also accounts for genotype uncertainty.

`ldest()` For the base function that estimates pairwise LD.

`sldest()` For estimating pairwise LD along a sliding window.

`format_lddf()` For formatting the output of `mldest()` as a matrix.

`plot.lddf()` For plotting the output of `mldest()`.

Examples

```
set.seed(1)

## Simulate genotypes when true correlation is 0
nloci <- 5
nind  <- 100
K <- 6
nc <- 1
genomat <- matrix(sample(0:K, nind * nloci, TRUE), nrow = nloci)

## Composite LD estimates
lddf <- mldest(geno = genomat,
               K = K,
               nc = nc,
               type = "comp")
lddf[1:6, 1:6]
```

---

pbnorm_dist *Returns distribution of proportional bivariate normal.*

Description

Returns distribution of proportional bivariate normal.

Usage

```
pbnorm_dist(mu, sigma, K, log = FALSE)
```

Arguments

mu	A vector of length 2 containing the mean.
sigma	A 2-by-2 positive definite covariance matrix
K	The ploidy of the individual.
log	A logical. If TRUE, log probabilities are returned.

Value

A matrix. Element (i,j) is the (log) probability of genotype i-1 at locus 1 and j-1 at locus 2.

Author(s)

David Gerard

plot.lddf *Plot the output of* mldest() *or* sldest() *using* corrplot()

Description

Formats the LD estimates in the form of a matrix and creates a heatmap of these estimates. This heatmap is created using the corrplot R package. I've adjusted a lot of the defaults to suit my visualization preferences.

Usage

```
## S3 method for class 'lddf'
plot(
  x,
  element = "r2",
  type = c("upper", "full", "lower"),
  method = c("color", "circle", "square", "ellipse", "number", "shade", "pie"),
  diag = FALSE,
  is.corr = NULL,
  tl.pos = "n",
  title = NULL,
  na.label = "square",
  ...
)
```

Arguments

x	An object of class lddf, usually created using either mldest() or sldest().
element	Which element of x should we plot?
type	Character, "full", "upper" (default) or "lower", display full matrix, lower triangular or upper triangular matrix.
method	See corrplot() for available options. Default value is "color".
diag	Logical, whether display the correlation coefficients on the principal diagonal.
is.corr	See corrplot(). Default behavior is TRUE if an element is constrained between -1 and 1 and FALSE otherwise.
tl.pos	See corrplot(). Default value is "n".
title	What should the title be? Defaults to the element name.
na.label	See corrplot(). Default value is "square".
...	Additional arguments to pass to corrplot(). See the documentation of that function for options.

Details

For greater plotting flexibility, see corrplot() for the parameter options.

Value

(Invisibly) returns a matrix of the selected elements.

Author(s)

David Gerard

Examples

```
set.seed(1)

## Simulate genotypes when true correlation is 0
nloci <- 5
nind  <- 100
K <- 6
nc <- 1
genomat <- matrix(sample(0:K, nind * nloci, TRUE), nrow = nloci)

## Haplotypic LD estimates
lddf <- mldest(geno = genomat,
               K = K,
               nc = nc,
               type = "hap")

## Plot estimates of z
plot(lddf, element = "z")
```

`pvcalc` *Calculate prior variances from a matrix of prior genotype probabilities.*

Description

Given a matrix of prior probabilities for the genotypes at each SNP, this function will calculate the prior variance of genotypes.

Usage

```
pvcalc(priormat)
```

Arguments

`priormat`	A matrix of prior genotype probabilities. Element `priormat[i, j]` is the prior probability of dosage `j` at SNP `i`.

Value

A vector of prior variances.

Author(s)

David Gerard

Examples

```
data("uit")
priormat <- uit$snpdf[, paste0("Pr_", 0:4)]
pvcalc(priormat)
```

`slcor` *Sliding window correlation*

Description

Calculates the pairwise Pearson correlation between all columns within a fixed window size (`win`) using the `use = "pairwise.complete.obs"` option from `cor()`. That is, the correlation between each pair of variables is computed using all complete pairs of observations on those variables.

Usage

```
slcor(x, win = 1L)
```

Arguments

`x`	A numeric matrix. The variables index the columns.
`win`	The size of the window. Defaults to 1.

Value

A correlation matrix with only the observations within a window containing calculated correlations.

Author(s)

David Gerard

Examples

```
set.seed(1)
n <- 10
p <- 100
xmat <- matrix(rnorm(n * p), ncol = n)
xmat[sample(n * p, size = 30)] <- NA_real_
slcor(xmat, win = 2)
```

sldest *Sliding window LD estimation*

Description

This function is a wrapper for `ldest()` for estimating LD along a sliding window of a fixed size. Support is provided for parallelization through the foreach and doParallel packages.

Usage

```
sldest(
  geno,
  K,
  win = 50,
  nc = 1,
  type = c("hap", "comp"),
  model = c("norm", "flex"),
  pen = ifelse(type == "hap", 2, 1),
  se = TRUE
)
```

Arguments

`geno`	One of two possible inputs: • A matrix of genotypes (allele dosages). The rows index the SNPs and the columns index the individuals. That is, `genomat[i, j]` is the allele dosage for individual `j` in SNP `i`. When `type = "comp"`, the dosages are allowed to be continuous (e.g. posterior mean genotypes). • A three-way array of genotype *log*-likelihoods. The first dimension indexes the SNPs, the second dimension indexes the individuals, and the third dimension indexes the genotypes. That is, `genolike_array[i, j, k]` is the genotype log-likelihood at SNP `i` for individual `j` and dosage `k - 1`.
`K`	The ploidy of the species. Assumed to be the same for all individuals.
`win`	The window size. Pairwise LD will be estimated plus or minus these many positions. Larger sizes significantly increase the computational load.
`nc`	The number of computing cores to use. This should never be more than the number of cores available in your computing environment. You can determine the maximum number of available cores by running `parallel::detectCores()` in R. This is probably fine for a personal computer, but some environments are only able to use fewer. Ask your admins if you are unsure.
`type`	The type of LD to calculate. The available types are haplotypic LD (`type = "hap"`) or composite LD (`type = "comp"`). Haplotypic LD is only appropriate for autopolyploids when the individuals are in Hardy-Weinberg equilibrium (HWE). The composite measures of LD are always applicable, and consistently estimate the usual measures of LD when HWE is fulfilled in autopolyploids.

However, when HWE is not fulfilled, interpreting the composite measures of LD could be a little tricky.

`model` When `type = "comp"` and using genotype likelihoods, should we use the proportional bivariate normal model to estimate the genotype distribution (`model = "norm"`), or the general categorical distribution (`model = "flex"`)? Defaults to `"norm"`.

`pen` The penalty to be applied to the likelihood. You can think about this as the prior sample size. Should be greater than 1. Does not apply if `model = "norm"`, `type = "comp"`, and using genotype likelihoods. Also does not apply when `type = "comp"` and using genotypes.

`se` A logical. Should we calculate standard errors (`TRUE`) or not (`FALSE`). Calculating standard errors can be really slow when `type = "comp"`, `model = "flex"`, and when using genotype likelihoods. Otherwise, standard error calculations should be pretty fast.

Details

See `ldest()` for details on the different types of LD estimators supported.

Value

A data frame of class `c("lddf", "data.frame")` with some or all of the following elements:

`i` The index of the first SNP.

`j` The index of the second SNP.

`snpi` The row name corresponding to SNP `i`, if row names are provided.

`snpj` The row name corresponding to SNP `j`, if row names are provided.

`D` The estimate of the LD coefficient.

`D_se` The standard error of the estimate of the LD coefficient.

`r2` The estimate of the squared Pearson correlation.

`r2_se` The standard error of the estimate of the squared Pearson correlation.

`r` The estimate of the Pearson correlation.

`r_se` The standard error of the estimate of the Pearson correlation.

`Dprime` The estimate of the standardized LD coefficient. When `type = "comp"`, this corresponds to the standardization where we fix allele frequencies.

`Dprime_se` The standard error of `Dprime`.

`Dprimeg` The estimate of the standardized LD coefficient. This corresponds to the standardization where we fix genotype frequencies.

`Dprimeg_se` The standard error of `Dprimeg`.

`z` The Fisher-z transformation of `r`.

`z_se` The standard error of the Fisher-z transformation of `r`.

`p_ab` The estimated haplotype frequency of ab. Only returned if estimating the haplotypic LD.

`p_Ab` The estimated haplotype frequency of Ab. Only returned if estimating the haplotypic LD.

p_aB The estimated haplotype frequency of aB. Only returned if estimating the haplotypic LD.

p_AB The estimated haplotype frequency of AB. Only returned if estimating the haplotypic LD.

q_ij The estimated frequency of genotype i at locus 1 and genotype j at locus 2. Only returned if estimating the composite LD.

n The number of individuals used to estimate pairwise LD.

Author(s)

David Gerard

See Also

ldest() For the base function that estimates pairwise LD.

mldest() For estimating pairwise LD between *all* provided SNPs.

ldfast() Fast, moment-based approach to LD estimation that also accounts for genotype uncertainty.

format_lddf() For formatting the output of sldest() as a matrix.

plot.lddf() For plotting the output of sldest().

Examples

```
set.seed(1)

## Simulate genotypes when true correlation is 0
nloci <- 100
nind  <- 100
win <- 5
K <- 6
nc <- 1
genomat <- matrix(sample(0:K, nind * nloci, TRUE), nrow = nloci)

## Composite LD estimates
lddf <- sldest(geno = genomat,
               K = K,
               win = win,
               nc = nc,
               type = "comp")
plot(lddf, element = "z")
```

---

uit *Updog fits on the data from Uitdewilligen et. al. (2013)*

Description

10 SNPs from the "PGSC0003DMB000000062" super scaffold were genotyped using the multidog() function from the updog R package. These data are the resulting output.

Usage

```
uit
```

Format

An object of class `multidog()`. See the documentation from the updog R package.

Source

doi:10.1371/journal.pone.0062355

References

- Uitdewilligen, Jan GAML, Anne-Marie A. Wolters, B. Bjorn, Theo JA Borm, Richard GF Visser, and Herman J. Van Eck. "A next-generation sequencing method for genotyping-by-sequencing of highly heterozygous autotetraploid potato." *PloS one* 8, no. 5 (2013): e62355. doi:10.1371/journal.pone.0062355

---

zshrink — *Shrinks Fisher-z transformed correlation estimates and returns resulting correlation estimates.*

Description

This function is a wrapper for adaptive shrinkage (Stephens, 2017) on the Fisher-z transformed estimates of the Pearson correlation. This approach was proposed in Dey and Stephens (2018) but is re-implemented here for now since the CorShrink package is not available on CRAN.

Usage

```
zshrink(zmat, smat, ...)
```

Arguments

`zmat`	The matrix of Fisher-z transformed correlation estimates.
`smat`	The matrix of standard errors of the Fisher-z transformed correlation estimates.
`...`	Additional arguments to pass to `ash()`.

Value

A matrix of correlation estimates. These are posterior means of the correlation estimates after applying the CorShrink method (Dey and Stephens, 2018).

Author(s)

David Gerard

References

- Stephens, Matthew. "False discovery rates: a new deal." Biostatistics 18, no. 2 (2017): 275-294.
- Dey, Kushal K., and Matthew Stephens. "CorShrink: Empirical Bayes shrinkage estimation of correlations, with applications." bioRxiv (2018): 368316.

Index

∗ **datasets**
glike, 6
gp, 7
uit, 32

ash, *21–23*, *33*

cor, *29*
corrplot, *27*

Dprime, 2

format_lddf, 3, *25*, *32*

get_prob_array, 4
gl_to_gp, 5
glike, 6
gp, 7

hsm, 7

is.lddf, 8

ldest, 9, *20–25*, *30–32*
ldest_comp, *12*, 13
ldest_hap, *12*, 16
ldfast, *12*, 19, *23*, *25*, *32*
ldshrink, 22

mldest, *3, 4*, *12*, *22*, 23, *23*, *27*, *32*
multidog, *32, 33*

pbnorm_dist, 26
plot.lddf, *25*, 27, *32*
pvcalc, 28

slcor, 29
sldest, *3, 4*, *12*, *22, 23*, *25*, *27*, 30

uit, *6, 7*, 32

zshrink, 33